technische universität
dortmund

fakultät für informatik

Andreas Timm

Grundlagen einer industrietauglichen AOP Lösung für C

Diplomarbeit im Studiengang Informatik

22. September 2010

interne berichte
internal reports

Gutachter:
Prof. Dr. Spinczyk
Dr. Beuche

LS12 - Lehrstuhl Eingebettete Systeme

fakultät für
informatik

GERMANY | D-44221 DORTMUND

Abstract

Nowadays, safety critical systems that provide state of the art technologies for automobiles, are developed in order to remain competitive. For efficiency reasons these systems are based on the programming language C which meets efficiency criteria in the area of embedded systems. These projects are usually very complex and are developed in collaborative teams consisting of several developers. The usage of IDEs reduces the complexity and provides functionality to support the developers. Moreover, the concept of modularization splits a complex system into several manageable parts according to their specific functionality. For instance, the OOP paradigm follows this approach. The lack of the possibility to address functionality which is spread over several modules as a single entity in OOP is compensated by means of AOP. A number of advantages result from it. For instance AOP enables static configuration of software systems, suppressing the requirement of realtime configuration checks, minimizing configuration overhead. The use of AOP has the disadvantage that a developer loses control over all aspects of his implementation, resulting in unwanted changes through AOP which is hard to debug. Therefore this thesis develops fundamentals for an industry-capable AOP solution for C.

Zusammenfassung

Sicherheitskritische Systeme, die aktuelle Technologien für Automobile zur Verfügung stellen, werden heutzutage mit dem Ziel entwickelt, konkurrenzfähig zu bleiben. Aus Gründen der Effizienz werden diese System in der Programmiersprache C entwickelt, welche Effizienzkriterien aus dem Bereich der eingebetteten Systeme erfüllt. Diese üblicherweise sehr komplexen Systeme werden von ProgrammiererInnen in Arbeitsgruppen entwickelt. Der Einsatz integrierter Entwicklungsplattformen reduziert die Komplexität und unterstützt die ProgrammiererInnen durch enthaltene Funktionalität. Durch Modularisierung wird ein System in überschaubare Elemente, welche je spezifische Funktionalität enthalten, aufgeteilt. Die Objekt-orientierte Programmierung beispielsweise verfolgt diesen Ansatz. Das Fehlen der Möglichkeit, querschneidende Belange in OOP als Entitäten zu implementieren, wird durch Aspekt-orientierte Programmierung aufgefangen. Daraus folgen zum Beispiel die Vorteile, dass ein Software System mittels AOP statisch konfiguriert werden kann, wodurch maßgeschneiderte Software erstellt werden kann. Der Nachteil bei dem Einsatz von AOP ist, dass ein Entwickler die Übersicht über die implementierten Aspekte verlieren kann, was zu ungewollten Änderungen am Quellcode durch AOP führt. Die daraus entstehenden Fehlfunktionen zu beheben kann eine schwierige Aufgabe darstellen. Um dieses Problem zu beheben, werden in dieser Diplomarbeit die Grundlagen für eine industrietaugliche AOP Lösung für C entwickelt.

Contents

Chapter 1

Analysis

This chapter summarizes the main goals of this thesis. Starting with an introduction of the realm of the problem, important background knowledge is presented. Here a description of development platforms is given and basic Integrated Development Environments (IDE) purposes are described. There then follows a short introduction to Aspect-Oriented Programming (AOP) and the role of the programming language C in embedded systems in the industry. Existing related solutions for AOP development are presented. Finally the motivation for this thesis will be presented and its main challenges are indicated.

1.1 Introduction

Nowadays, safety critical systems that provide state of the art technologies for automobiles, are developed in order to remain competitive. For efficiency reasons these systems are based on the programming language C which meets efficiency criteria in the area of embedded systems. These projects are usually very complex and are developed in collaborative teams consisting of several developers. The usage of IDEs reduces the complexity and provides functionality to support the developers. Moreover, the concept of modularization splits a complex system into several manageable parts according to their specific functionality. For instance, the Object Oriented Programming (OOP) paradigm follows this approach. The lack of the possibility to address functionality which is spread over several modules as a single entity in OOP is compensated by means of aspect oriented programming. In particular the development of automotive solutions is supported by this approach. A number of advantages result from it. For instance AOP enables static configuration of software systems, suppressing the requirement of realtime configuration checks, minimizing configuration overhead. The use of AOP has the disadvantage that a developer loses control over all aspects of his implementation, resulting in unwanted changes through AOP which is hard to debug. Therefore this thesis develops fundamentals for an industry-capable AOP solution for C.

1.2 Basics

The following sections introduce the main theoretical aspects of the present thesis. First, the most important reasons for the use of IDEs will be specified. Afterwards the paradigm of AOP and its main advantages will be pointed out. Furthermore the importance of the programming language C and its usage in the automotive industry is outlined. Finally related work will be presented.

1.2.1 Integrated Development Environment

The usage of tools that support a programmer's work can be seen as mandatory, as it optimizes the programmer's efficiency. IDEs are commonly used to simplify and support the necessary steps from implementation to building. Java development for instance requires the developer to implement classes and afterwards run a compiler that finally creates binary files which can be executed via a Java interpreter. The implementation of classes is simply a special kind of text. Therefore it can be created with a text editor of the programmer's choice. It is possible to use the most simple editor to create these classfiles. However, the use of an IDE offers several advantages. The code is continuously checked for errors, keywords are highlighted for a better usability and drop down menus give a context-specific choice of available functions of an object. These help the developers by increasing the efficience of their works in progress and can help to improve the quality of the produced code.

1.2.2 Aspect Oriented Programming

AOP is a programming paradigm that allows the modular implementation of crosscutting concerns. A crosscutting concern requires extensions in different modules of a software system. Regarding OOP such a crosscutting concern can not be implemented as a single entity [CE00]. For instance tracing all function calls of a software system is a very work intensive issue when it is implemented in OOP. As an AOP advice it is relatively simple and can be performed quickly because it is possible to express the idea 'after every function call print out the function name' directly as an AOP advice. A specialized syntax is required to express aspects. This syntax is language dependent.

Several advices can be grouped together in an aspect, representing a single entity that implements a crosscutting concern. Every advice consists of a pointcut and advicecode. The pointcut is an expression which determines where the adviceblock has to be inserted. This insertion point is called joinpoint, thus a pointcut defines a set of joinpoints. Pointcut expressions can be created through a grammar that is defined within the AOP language. Common pointcut expressions are unions or intersections of *call(function specifier)* and *execution(function specifier)*. With *call* those parts of the base system match where a

call to the specified function should be executed. An *execution* matches those functions the function specifier matches. Such a function specifier matches a function in a regular-expression like manner.

These pointcut expressions are usually prefixed with a further instruction which is one of *before*, *around* or *after*. The interpretation is thoroughly intuitive, simply defining how an adviceblock should be inserted into the control flow. Before and after advices insert additional functionality. Around advices can be used to replace existing functionality through suppressing its execution.

The adviceblock itself may contain labels that have to be substituted with context-sensitive information like functionnames, parameters to the function or return values and types. This offers the possibility of developing advices which react based on the execution context, an important flexibility.

However, to be able to use AOP there is need for a new tool which connects these aspects with the software system. This tool is called *weaver*, its action is called *weaving* and the result is called *woven code*. During the weaving process the software system is extended by advices that match the code. Weaving can be done at compiletime or at runtime.

An example of a compiletime weaver for C is shown in figure 1.1. It is executed once and creates a new version of the C source code that includes the additions from the aspects that reside in the aspect files. This new version, or woven code, will then be handed over to further preprocessors and finally to the compiler which creates a binary file from it.

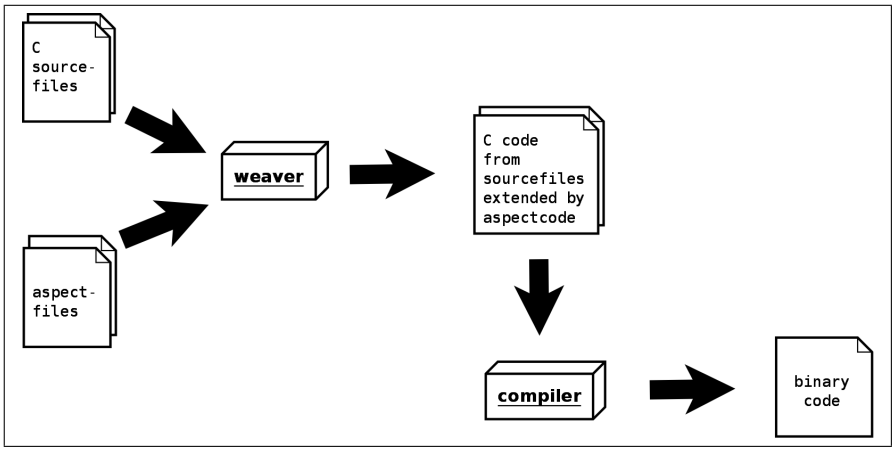

Figure 1.1: The place of the compiletime weaver in the toolchain

As a major drawback AOP changes the control flow. The woven code is extended base code, that means that these extensions will be executed at runtime as well, even though they were invisible in the base code. This behaviour can possibly be considered as dangerous, because the executed code is not the same as the code the programmer looks at. By keeping all influences from aspects in mind, the programmer will know what the

executed code looks like, but especially in larger teams this is almost impossible for every single programmer and so it creates risks for them which they would surely like to avoid.

1.2.3 C and the Industry

The programming language C was initially invented by Dennis Ritchie in the early 1970s and is widespread within the industry. In the automotive sector the CAN-Bus (Controller Area Network, 1987 by Bosch and Intel) is used to connect several embedded systems within a car [Gmb91]. These embedded systems run operating systems with specialized software written in C. Before such an embedded system is certified to be connected to the CAN-Bus, the source code of the software that runs on the embedded system will be statically analyzed to maximize its reliability and safety. The code is checked to fulfill the MISRA (Motor Industry Software Reliability Association) rules. The latest set of rules is called *MISRA-C:2004* ("Guidelines for the use of the C language in critical systems") and contains 121 required rules and additional 20 advisories.

The configuration of C programs is usually done through #ifdef instructions. Nevertheless, a lot of #ifdef instructions are necessary in order to be able to configure a system at a high level of granularity. This unfortunately increases the complexity of configuration maintenance.

A possible solution for this issue is to realize the configuration tasks by using optional aspects. Doing so improves still further readability and maintainability of the resulting software.

To efficiently use AOP for the use cases shown above, it will be necessary to extend existing IDEs to fulfill several requirements due to adequate user support functionalities.

1.2.4 Related Work

The existing projects that relate to this work are plugins for IDEs and standalone commandline applications, which enable and support Aspect Oriented Software Development (AOSD). Besides C related AOP solutions AspectJ Development Tools (AJDT), as *the* AOP solution for Java, is also a subject of the analysis.

ACDT

The Eclipse plugin AspectC/C++ Development Tools (ACDT) supports AOP for C++ and is actually available in version 3.1.0 for Eclipse 3.2.x [acd]. This plugin is based on the weaver of AspectC++ and on the C/C++ Development Tooling (CDT) plugin [aspb, cdt]. It features syntax highlighting and a special outline view for aspect files. This view shows the aspect, the advices within that aspect and the joinpoints of those advices, which also have navigation functionality. The joinpoints are marked in the editor,

visualizing the existence of an advice that matches a specific position. Debugging of the software supports setting of breakpoints in the advice code and stepping through the code, accessing aspect files naturally. The *comparison mode* can be used to track changes to the aspects from the project's repository.

The weaver from AspectC++ is used by the plugin. As it is a compiletime weaver, it is inserted into the buildchain. The AspectC++ language is a state of the art aspect language for C++. The weaver produces C++ compatible woven code in a new directory directly from the aspect files and the C++ source code.

AXDT

The XWeaver Development Environment (AXDT) is a Eclipse plugin that offers a language independent AOP solution, as long as the weaver gets an XML version of the source code as its input. As the name suggests this plugin is based on the XWeaver project, a commandline weaver, which uses the XML-based AOP language AspectX [axd]. The latest plugin is available for Eclipse 3.2.2, made available in may 2007. The plugins features are guided settings of XWeaver project file attributes like language, project, aspects and destination, aspect editing based on a visualized tree to create a XML representation with properties that contain modifier code. A cross reference view shows which aspect's advice alters a target file and at what point. Markers show where an advice joins the source code in the editor. An interesting part of this project is the meta overview of classes that visualizes order and source of advices that change a source file.

The weaver works on XML data. The aspects are saved as XML files, any sourcecode that should be a target for the aspects has to be transformed into a corresponding XML model. Afterwards the weaver weaves the base code XML model and the aspects into a XML model of modified code, which will finally be transformed back into the original language. As of today the project ("beta prototype" [bpx]) claims to support C, C++ and Java code.

AJDT

The AspectJ Development Tools are another plugin for Eclipse. They rely on the AspectJ language, an AOP language for Java, which is deployed as a separate plugin and accessed from the plugin, offering the possibility to select the version of the AspectJ language independently from the AJDT version. AJDT is currently available in version 2.1.0 [ajd].

The plugins features provide information about changes of matching advices to a updated part of the sourcecode. Changes to the aspects of a system can be traced with the crosscutting comparison view. Next to the aspect outline its effect is presented in a cross reference view, showing the joinpoints of the advices. AspectJ also supports refactoring, syntax highlighting and syntax error marking.

C4

The CrossCutting C Code Toolkit C4 reads aspects which are written inside of the C source files. They will be removed and afterwards be used to be woven into the system. The C files are initially preprocessed by the ASTEC [MB05]. The aspects inside the C files are surrounded by blocks that include information about being an aspect and the point at which they are to be woven. These blocks are removed, parsed and afterwards they will be woven into the function that they came from. Then the woven version can be compiled.

Even though there is no graphical user interface available, this approach differs from other existing approaches as it integrates the aspect code with the C code that is meant to be extended [c4m06, RFG].

Aspicere

This aspicere project [aspa] offers a linktime weaver that is based on a low-level virtual machine, "a collection of modular and reusable compiler and toolchain technologies" [llv] and also a compiletime weaver that only implements a subset of the AOP language. A graphical frontend is not available. The AOP language Aspicere has a pointcut language based on prologue.

1.3 Motivation

The amount of existing solutions for AOP for C as well as several related papers show that this rather new area is a topical science subject. Although the functional parts of these solutions work properly, they lack acceptance within industry development projects. The main problem is, that the control flow becomes invisible to the programmer, as the software will be altered just before it is compiled. The C code visible to the programmer does not exactly correspond to the code that is executed at runtime.

In projects with small numbers of programmers this problem cannot be considered as a serious issue. But in companies with over a dozen programmers working on one piece of software, this represents a serious problem. In addition to the need to know about the main parts of the software, together with detailed knowledge of all actively developed parts, it would be necessary to know about every implemented aspect, as the effect of changes to the base software could change the result the weaver produces. For instance, renaming a function could suppress the matching of a pointcut on that function. On the other hand it could create a match for that function, without noticing the developer during implementation. At testing time this effect would finally become visible and create the need for debugging. The occurance of such problems would seriously reduce the benefit of AOP for big software systems.

State of the art C development is usually undertaken with the support of IDEs like Eclipse bundled with CDT. These tools generate syntax highlighting, outline view, error markers, type hierarchy, call graph and support refactoring, team development and step by step debugging.

These days available AOP solutions are mostly stand-alone weavers, which are inserted into the buildchain. Some of the tools even have an editor that supports syntax highlighting and debugging. Though these tools can be used for AOP, they unfortunately support AOP development just as well as simple texteditors support programming. To be able to use AOP techniques effectively, new, supporting tools are required. The main problem to be tackled is the control flow invisibility. Therefore the goal of this thesis is to create the foundations for a development supporting tool that visualizes the control flow, continuously reflects effects of the latest code changes and offers support for aspect creation.

1.4 Challenges

In order to create a tool that shows the control flow, several steps have to be accomplished in this thesis. First an AOP language has to be specified, as it is the basis for all further steps. A choice between existing languages and the option to create a new AOP language for C will be considered.

As Eclipse has been used in various previous LS12 projects it will be selected as the IDE platform within this thesis. Therefore an Eclipse plugin has to be developed that provides several functionalities to give the necessary control back to the programmer.

Besides syntax highlighting the software should visualize the control flow. The desired approach is development time weaving. While the programmer implements new advices or extends the C base system, the software should update the C base system, permanently producing dynamically woven C source code. This requires a special data storage, the aspect is saved in an aspect file and its advices adviceblocks are woven into the C files, duplicating the advicecode. A parser for the aspect files, as well as a system to find joinpoints from a pointcut expression, will be required. The visible code shows the programmer exactly what an external weaver's result would be. This dynamically updated information lets the developer concentrate on the implementation as it already contains the advice code. Otherwise the developer would permanently have to worry about the impacts of non-optional aspects on the existing code.

Because the update process of the C code should be done automatically, special actions that should trigger updates are to be determined within the IDE.

As the developer can edit source files in his favourite texteditor, these changes can create inconsistent sourcecode. Several possible problems need to be analysed and the developer should receive resolution proposals which bring the software system back into a consistent state.

Chapter 2

Concept

Before focussing on the implementation, detailed discussions are required to create concepts for their solutions. The project's challenges are the analysis of the requirements for an AOP language and the selection of an existing language or alternatively the creation of a new language to match the needs. The datastorage, which is different from existing AOP solutions, has to be analyzed to gain knowledge about implementation specific information for the duplicate datastorage. This directly motivates the definition of AOP consistency and a discussion about possible resolutions for inconsistency. Consistency is, of course, the desired result of the development time weaving process, which also has the task of keeping the system consistent while updating it. The new concept of a development time weaver has to be inspected in detail to possibly fulfill this requirement. The weaver, as one part of the plugin for Eclipse, will need information from the development platform. Therefore information about the interfaces of the platform that provide the required information has to be collected.

2.1 Extending Eclipse

An Integrated Development Environment supports the developer in his daily work. Eclipse is a widely used framework for the creation of developer tools and a lot of documentation is available. It has already been used in recent LS12 projects and is therefore the IDE of choice.

The graphical user interface of Eclipse is called the workbench. It has different perspectives, views and editors that provide access to the functionalities.

Each Eclipse instance works with a single workspace. The workspace holds Eclipse configuration and projects. Projects, directories and files are called *Resources* in Eclipse. For reasons of consistency, instead of the filesystem, the *Resource* should be altered.

Extension points are used in order to create an Eclipse plugin. A plugin itself can offer new extension points, so other plugins can extend existing functionality [eclc, CR08].

2.1.1 Workbench

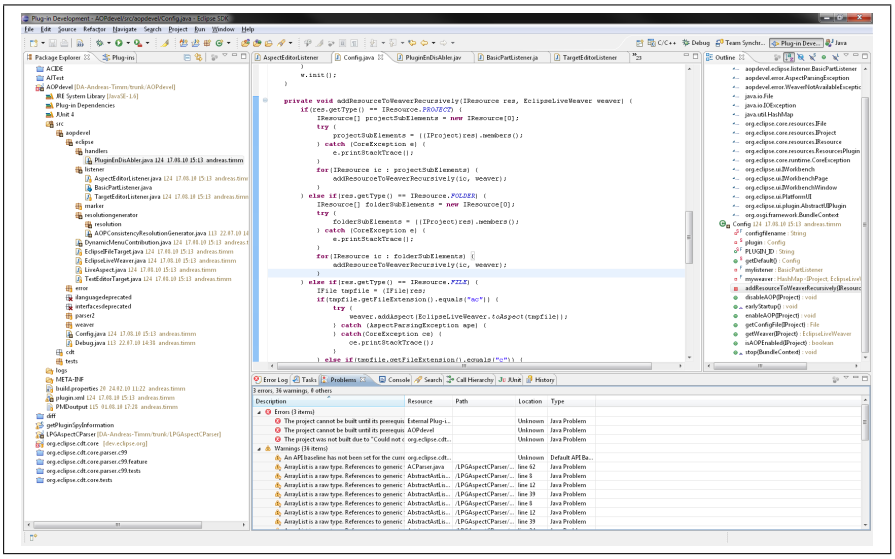

Figure 2.1: Example perspective of the Eclipse Workbench with different views

Figure 2.1 shows the GUI (Graphical User Interface) of Eclipse. Its content is specified by a selectable perspective. The usual Java perspective shows a Project Explorer view on the left. It visualizes the available projects of this workspace in an explorable tree. This view provides the functionality to open a file in an editor, which is generated in the center of the GUI. An editor is usually text-based and supports syntax highlighting, syntax error marking and proposals for error solution. Based on the content of the editor, an Outline of the file is shown on the right. Blocks like functions, nested classes or classvariables are listed and can be used for navigation and re-ordering. The bottom of the window contains a view providing several tabs. Those display the commandline output generated by applications and a list of the errors occuring in the project.

By selecting a different perspective, these views may change. A different perspective is ,for example, the Debug Perspective. The Project Explorer is removed and functionality is added to be able to step through the sourcecode while the application is being executed. Editors are shared among the different perspectives, so they are still visible even after the current perspective has changed. An additional view shows values of variables from the application.

2.1.2 Workspace

An Eclipse instance runs on a single workspace, which is a directory containing one configuration directory and one directory for each project. It is possible to switch between different workspaces within Eclipse. After a workspace switch Eclipse will be restarted and reads the workspace's data.

The configuration directory called *.metadata* holds a *.log* file for logging and a *.lock* file to ensure exclusive usage of this workspace. The *.plugin* directory is the path where Eclipse plugins stores its configuration information.

2.1.3 Resources

All files, directories and projects of one workspace are accessible in Eclipse through the required pre-installed resources plugin. The *Resources* are organized like they are in the filesystem. A project resource contains directory and file resources, just like a directory resource. The difference is that all directory and file resources from a project resource belong to that project resource. The content of a file resource equals the content of the corresponding file.

A project can be closed. Contents of a closed project are not accessible. The contents of an open project are available through the Project Explorer view. This view enables navigation through the tree hierarchy.

2.1.4 Plugins

The development of an Eclipse plugin can be performed using the Eclipse PDE (Plugin Development Environment) [ecla].

Eclipse provides extension points which can be used to integrate new functionality into the core. These extension points offer possibilies to, for example, connect an editor to a specific filetype, add entries to menus, insert new views etc. There are 25 extension points for the Eclipse core and 78 extension points for the Eclipse user interface. Further extension points exist for many parts of the IDE. Nevertheless several other extension points exist because each plugin may offer its own extension point set.

Every extension point is defined by an XML (eXtended Markup Language) schema file. The schema defines optional and required elements and attributes an extension has to contribute in order to use the extension point.

An extension point can be accessed by an extension from a plugin. This extension is also defined in a XML file and has to match the requirements of the extension point. Required attributes have to be set and contributed classes have to implement the defined interfaces.

Within Eclipse special editors support the definition of extension points and extensions, while raw XML editing is replaced by guided editing [Vog].

After installation of a new Eclipse plugin it is available inside the specific Eclipse installation. This is independent of the active workspace. Plugins are stored in a distributed manner, usually every plugin is available from another location.

The Eclipse platform makes extensive use of plugins. To reduce startup time and still offer a lot of functionality through the plugins, the Eclipse team created a lazy startup mechanism. A plugin is loaded as soon as functionality it offers gets used. Compared to the loading all installed plugins at startup this method is much faster and the initial memory footprint is smaller.

By providing an extension to the extension point *org.eclipse.ui.startup*, the plugin can request to be loaded immediately.

A great number of plugins are available for Eclipse. For instance, they include support for SVN (subversion) or LaTeX and development environments for Ruby, PHP, Java, C and many more.

2.1.5 C/C++ Development Tooling

The C/C++ Development Tooling (CDT) plugin [cdt] extends Eclipse with comprehensive C and C++ development support. It is actively developed by several global enterprises including IBM, Nokia, Intel and freescale.

Features of CDT include project creation support with a choice of build toolchains, syntax highlighting and outline view, navigatable source code and type hierarchy. Furthermore debugging is supported with a special view. The analogy of these features to those of the Java Development Tools (JDT) is based on the fact that the CDT is developed on the basis of an early JDT version that was modified to support C and C++ instead of Java.

Parts of the CDT had to be developed from scratch, like for instance the C/C++ parser. A lot of the plugin's functionality is based on CDTs own parser that creates an abstract syntax tree representation of the source code that is cached to provide quick response times.

At its base, CDT is developed in several different projects. The single projects can be developed independently, providing their functionality to the plugin which integrates the functionality of all the parts. Therefore some projects provide extension points which are used by other projects to extend the functionality. The single modules are then developed by different teams based on a plan for the next major version release, with the actual version being 7.0.

2.2 Development of SimpleAspectC

A variety of AOP languages for C already exists. As experience has shown, these are not solutions applicable for use in the new approach of development time weaving. It requires the woven code be changed minimal, keeping the global view of a target file similar to the

original version to prevent confusion of the developer. Furthermore the woven code needs to be easy to read by providing information about the source of the woven code at the same time.

The identification of every aspect and every advice is necessary in order to be able to trace the source of woven code. As the woven state of the source code is now always visible to the developer, this information is crucial for debugging the code. Moreover, the identification is fundamental, enabling the development of further enhanced IDE functionality for AOP.

A basic demand for a new AOP language is the orientation of its design towards state of the art AOP languages. Especially AspectC++, being developed and improved for over five years now, and AspectJ, as the major AOP solution for Java, should be considered as important influences to the design. Taking these languages as standards shows that it is necessary to include before, after and around advices in an AOP language.

The design of a new AOP language should furthermore be intuitive to understand, which helps developers to quickly adopt it. This includes reuse of known C patterns for the grammar of aspects, advices and pointcuts, as well as the usage of common AOP identifiers to introduce the syntax.

Research has shown that aspects are not always placed inside exclusive aspect files, but this is required in order to be able to weave the aspects directly into the target files.

SimpleAspectC

Existing AOP languages for C, like aspectc [aspb], C4 [c4m06] or WeaveC [Nvvd07], do not meet all of the usability demands made by a development time weaver. A new AOP language for C that meets all the requirements must be created as it is the base for the further work. Its design orients on AspectC++ [aspb] to result in a state of the art AOP language.

Aspect file

SimpleAspectC uses aspect files with the suffix .ac to store the aspect's information. Each file consists of the following three parts:

1. the aspect file header
2. the aspect
3. the aspect file footer

Every aspect file requires an aspect, which may contain advice and pointcut declarations. There are no further demands on the implementation of an aspect, header and footer are optional parts and can therefore be empty.

Header and footer

These optional parts of the aspect file will be directly written into target files in the weaving process. Therefore they may only contain conventional C source code. They are included once in every file the aspect extends. The position within the aspect file defines the position in each target file. C code in the aspect file's header will be added to the target file's top, code at the aspect file's footer being added to the target file's bottom.

The aspect

Every aspect file requires exactly one aspect. It is the basic construct that enables creation of advices. Each aspect is identified by a name that is unique within the project. The aspect body contains advice and pointcut definitions.

Example

Listing 2.1: Aspect file example

```
1 #include <xyz_lib.h>;
2 void aspect_log(char[]);
3
4 aspect Foo {
5   [...]
6 }
7
8 void aspect_log(char[] info) {
9   puts(info);
10 }
```

This sample shows the usage of the aspect header to set an include and a forward declaration. The aspect itself is not of interest yet. In the aspect footer an implementation for the forward declaration is given in plain C.

The advice

The different types of advices enable additions to almost every part of C source code. An advice is identified by a name that is unique in the aspect scope. The advice type is identified by a position, which is one of before, around and after, followed by a pointcut method. The available methods are call, exec, decl, def and file.

The position defines how the advice will be woven at a joinpoint, declaring when the advice code has to be executed relatively to the joinpoint. A joinpoint is the match of the given pointcut in a target C file. The different available pointcut methods are defined like this:

call Match every call to a function that is specified in the pointcut, thus inserting advice code around the call inside another function instead of altering the called function itself.

exec Extend the execution of a function body given in the pointcut. Can be used to redefine a function using the around position.

decl Targets the declarations in C header files and adds code to them.

def Adds advice code around a target function definition. The function itself will not be changed by this advice, but can be target for a exec pointcut to access a function defined in this advice.

file Expects a filematch as an argument to specify files that should also be extended with the aspects header and footer code and the adviceblock. Usable to insert a new function into a specific set of files.

All methods except file expect a pointcut as an argument. Every advice requires the implementation of an adviceblock. It may contain special symbols that are used or substituted in the weaving process, hence providing further possibilities for the developer.

Special symbols

Depending on the position and method of the advice, there are different special symbols available for usage in every adviceblock. These are:

returntype() positions: all; methods: call, exec, decl, def; Insert the string representation of the function's return type here.

returnvalue() positions: around, after; methods: call, exec, decl, def; Will be substituted with the variable that holds the return value.

argset(x) positions: all; methods: call, exec; Set x gives access to the passed arguments to the function, accessible through x.size: the amount of elements x.arg[i]: element at position i x.type[i]: mangled char of the type of the element at position i

functionname() positions: all; methods: call, exec; Substituted by a string with the name of the current function.

proceed() position: around; methods: call, exec; Executes the original function.

yield() position: around; methods: decl, def, file; Splits the adviceblock into two parts, head and foot, the head will be inserted before the declaration or definition of the matched function or, if the method was file, the head will be inserted at the top of that file. The foot will be inserted after the declaration or definition or if the method was file at the bottom of the file.

Pointcut

To choose functions that should be extended by an advice, a pointcut needs to be defined. The following atomic constructs are available:

⟨ **c data type** ⟩ ⟨ **function name** ⟩**()** To use it on a specific function the function declaration (without arguments) can be inserted. This minimal set can be extended through the use of the wildcard "%" in the function name and return type.

within(filematch) It is also possible to weave the aspect only into a specific set of files through the within(filematch) method.

args(arglist) To restrict advices on functions with a specific argument signature the args(arglist) method can be used. The arglist allows restrictions on the argument types and argument amount. It can also be used to access the function's arguments.

⟨ **pointcut** ⟩ A previously defined pointcut may be inserted.

!⟨ **pointcut** ⟩ Invert the pointcut to match all joinpoints except those the pointcut matches.

These constructs can be combined using the && and || operators. "A && B" restricts the pointcut to match both pointcuts A and B. "C || D" matches if at least one of the pointcuts C or D matches.

Arglist

As an argument for the args pointcut an arglist has to be provided. It is useable as a filter for the matched functions by call or exec pointcut methods. An arglist consists of several arguments, with each required argument being separated by a comma. A required argument may be empty, which means exactly one argument is demanded and its type is irrelevant. If a type is given it may be followed by a label. The argument will then be available through the label in the following advice block. The wildcard for the arglist is "...", meaning there may be zero or more arguments.

The arglist's production rule P is

$P ::= P, P | \epsilon | ... | < cdatatype >< label(optional) >$

Examples

args(,,) matches, if the function expects three arguments

args(,,...) matches functions with at least two arguments

args(int x, int y, char[, ...)] matches only functions with the first three argument types being (int, int, char[]). The matched function may expect further arguments. The first two int arguments will be available in the adviceblock through the variables x and y.

Filematch

To restrict the weaving onto specific files a filematch has to be created. It is a comma-separated list of strings with "*" as wildcard operator. The given path has to be relative to the project's root path.

Examples

file("lib/sec/caesar.c", "sec/lib_*.c") matches the file 'lib/sec/caesar.c' and all C files beginning with 'lib_' in the 'sec/' directory

file("sec/*", "sec/*/*") will match all C files in the 'sec/' directory and in each of 'sec/'s subdirectories

Aspect example

Example 1

A network attached, battery powered embedded system, intended to be used in a closed environment, transmits plain data for improved battery lifetime.

Listing 2.2: Examplary device functions

```
1 #send characters
2 void send(char * chrptr)
3 {
4   ...
5 }
6
7 #receive new data
8 char * receive()
9 {
10  ...
11 }
```

In a new project some of these devices will be attached to an open network, requiring the transmitted data to be encrypted. The following aspect satisfies this requirement.

Listing 2.3: Encrypt data that will be sent over a network

```
1 #include <crypt.h>;
2 aspect Datacrypting {
3   advice befsend : before exec("void send()" && args(char * c, ...)) {
4     c = encrypt(c);
5   }
```

```
6   advice aftreceive : after exec("char * receive()") {
7     return decrypt(returnvalue());
8   }
9 }
```

Example 2

Listing 2.4: Trace every function call and print the functionname and all parameters of the type int and char.

```
1 #include <mangling.h>
2 aspect Trace {
3   advice trace : before execution("% %()    ) {
4     argset(args);
5     printf( TRACE  function functionname()    );
6     int i;
7     for(i = 0; i < args.size; i++) {
8       if(strcmp(mangle(    int    ), args.type[i]) == 0) {
9         printf( param  #%d: % d  , i, args.arg[i]);
10      } else if(strcmp(mangle(    char    ), args.type[i]) == 0) {
11        printf( param  #%d: % s  , i, args.arg[i]);
12      }
13    }
14  }
15 }
```

2.3 Principle of permanent woven code

In contrast to compiletime and runtime weaving methods, this project focuses on permanent weaving techniques. The C source files will always contain additional code from matching advices. They will be extended in exactly the same way as a compiletime weaver extends them. The difference is that the woven code will be updated as soon as an aspect is changed.

This approach requires specification of the management of the duplicate data. The implementation of aspects only takes place in aspect files, while the changes in these aspect files will afterwards result in updates of the target files. Before an update can take place, old aspect code has to be removed in order to compute correct pointcut matches. This method, in contrast to weaving, will be called *unweaving*. As this project is supposed to be an initial foundation, different techniques for updating may be implemented for testing purposes in addition to this work in the future.

This duplicate datastorage bears the risk, that either woven code, aspectcode or both are changed without usage of this projects plugin, that means that these changes do not lead to an update or error information within the woven code. These possibilities and related consistency criteria have to be evaluated and will be discussed in chapter 2.4.

Since every advice can have an arbitrary set of joinpoints, and every joinpoint can be a target for an arbitrary set of advices, information about the source of the woven code is necessary. Only with knowledge of the source is a developer able to change the corresponding advice. Furthermore, the identification of advice code offers the possibility of extracting aspect files from woven code or add more developer supporting functionality to the plugin. These ideas may be a future addition to this thesis and are not part of it.

To identify the woven code, every line of it has to be extended by a special comment. There are three sources for woven code: the aspect's header and footer code, the advicecode and administrative code. From every advice that is woven into a target file its aspect's header and footer code is inserted into that target file once. These header and footer codes will be identified by the name of the aspects from which they originate with the additional information about being header or footer code. The identification of aspectcode can simply be done through the information about the advice's aspectname and the advicename, as they both have to be unique and thus identify the code unambiguously. Again, every single line of code is extended by this identification. At last there are further code pieces that originate from the context and are supplied by the weaver. The possible cases are

- function hijacking and renaming
- adviceblock begin / end
- renamed function call
- return of original functionvalue
- forward declaration
- aop block begin / end

Their exact use will be described in the weaver section (2.5).

2.4 Consistency

The already mentioned possibility of changing aspectcode, woven code or both with an external tool that does not support the development time weaving feature of this thesis' Eclipse plugin bears the risk of damaging the AOP consistency. This consistency is crucial for AOP development, because every update of the woven aspectcode will start with the removal of old aspectcode. If the obsolete aspectcode can not be identified, it can not

be removed. This state can not be distinguished from the initial state where a target file contains no aspectcode at all, thus the weaver will compute joinpoints and insert aspectcode based on the corrupt code. The correct consistent state of a system will be defined as follows:

Definition

An AOP software system with permanent woven code is consistent if the software system's sourcecode contains exactly those code changes that a weaver would compute from the software system's included aspects.

Any change to a file of an AOP software system with duplicate datastorage can possibly compromise the consistency of the software system. Changes to an inconsistent system can lead to dysfunction of the whole software. Therefore its consistency has to be checked after editing any file.

An aspect file's advices can potentially have pointcuts that provide joinpoints for every single sourcefile. Therefore changes to an aspect file can result in an update for every sourcefile. Each sourcefile will then be unwoven, the aspects' advices' joinpoints will be computed and the weaving process will be executed for every target sourcefile. This will guarantee the consistency of the software system.

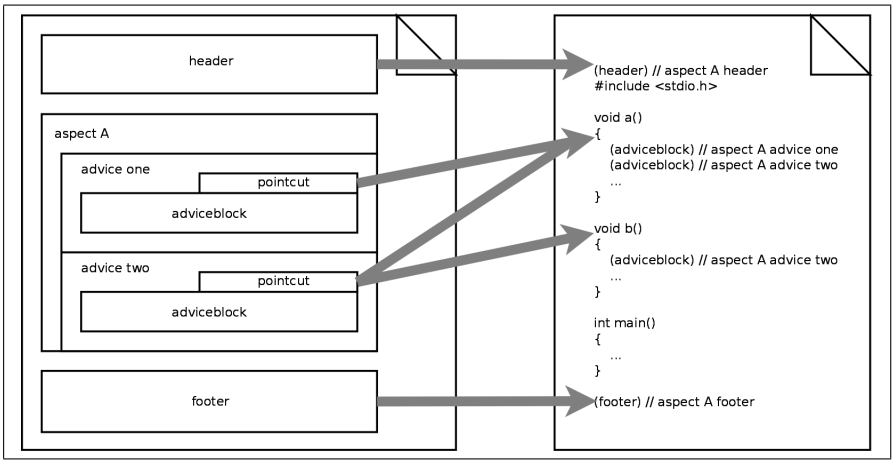

Figure 2.2: Example of an aspect file and how it extends the source file

Source files that were extended by advices afterwards contain aspectcode. Figure 2.2 shows a simple example where one aspect file is woven into one C source file. The aspectcode exists in different parts of the source file, as an aspect has a header section, several advices and a footer section. The header and footer sections will be inserted once into every targetfile that is matched by at least one of the aspects' advices. Furthermore, an advice can extend a function, for example, resulting in the renaming of the matched function, the hijacking of the original function to place the adviceblock and a call to the

renamed function. Also a forward declaration to register the function name in the compiler is added. All the inserted codepieces will simply be called *aspectcode* for now.

A source file can potentially contain matches for every aspect. It is possible to create consistency through unweaving of the active aspectcode, computing joinpoints for every advice from every aspect in the unwoven version of the targetfile and weaving of the matching advices afterwards. Again, this will guarantee the consistency of the software system. If woven aspectcode is changed, it can potentially be changed in such a way that it is impossible to check whether a line of code results from an aspect or not. The changes to aspectcode can happen in the header, a forward declaration, a woven adviceblock or the footer.

All these corruptions can be automatically solved if the identifying comments are unchanged. If this is the case, each of those parts can be completely removed and an active version of the part can be inserted.

2.5 Development time weaver outline

The principles of duplicate datastorage and consistency require a slightly different approach to the weaver than the most related type of weaving, compiletime weaving. Compiletime weaving is executed once, directly before the resulting code gets compiled, and is usually created as a standalone program. The weaver that is necessary for this project works at development time.

The main task of the development time weaver is to assure the system's consistency. This can only be achieved through direct integration into the development process. Changes to source files and aspect files need to be tracked for the creation of immediate updates.

It has to be taken into consideration that the developer can interrupt the work and close the IDE at any time. Before closing, all open files will be saved. That triggers the development time weaver to update accordingly and keeps the system in a consistent state.

If the files stay untouched until next startup, consistency is guaranteed. If files are changed in the meantime, the system may be inconsistent, so it has to be checked at IDE startup time.

Comparing the figures 1.1 and 2.3 the difference of the weaving mechanism appears to be very small. The C source code always includes the systems aspects, so the weaver is not included in the toolchain at compiletime. Also it uses its output as input instead of the C sourcefiles.

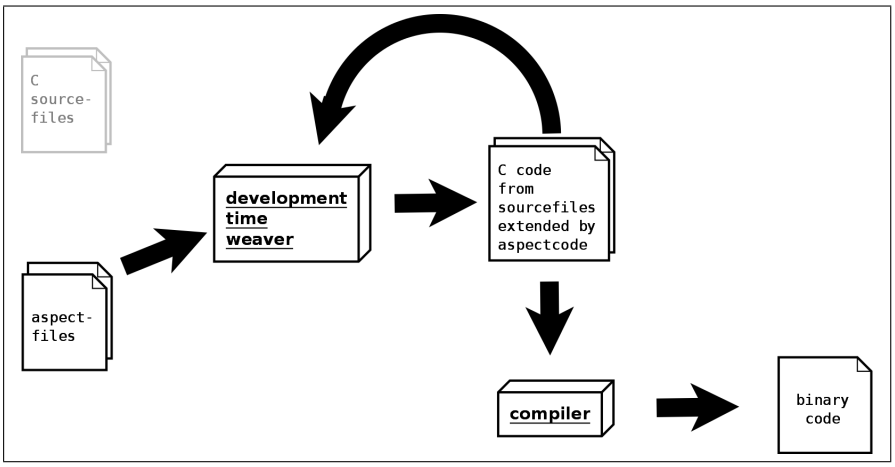

Figure 2.3: The weaver included in the development

Chapter 3

Design & Implementation

With a concept for the implementation at hand, first the environment has to be explored in order to find those interfaces that enable the final implementation. As Eclipse is the platform for the plugin, implementation-specific details about the IDE are required to accomplish the goal of creating an Eclipse plugin based on the concept. Besides analyzing plugin development and available extension points from Eclipse, the Eclipse plugin CDT needs to be reviewed for possibilities of use as a source for C specific information. Furthermore, a solution to the question of how CDT may be extended to accomplish the task of creating an Aspect Editor which should be able to provide feedback to the developer about the correctness of the AOP grammar as well as about C codepieces inside an aspect file has to be found.

The second part will show the design of the plugin, the tasks of the different classes and their connections. The implementation will be discussed, and its details are presented in the final section 3.3, enabling rapid further development based on this work.

The development of the plugin takes place inside the Eclipse Plugin Development Environment with the programming language Java. The plugin is being tested by running it inside another Eclipse instance, a directly executable option from the project's popup menu. This feature supports the immediate testing of the latest implementation. Direct feedback is shown in the development Eclipse instance's console.

Software Development

The management of software development is important to assure the high quality of the product. For the development of this thesis' Eclipse plugin, milestones are created, each consisting of tasks that need to be solved within a specified time. The later milestones' tasks may require the previous tasks to be finished successfully. Three milestones, starting with basic functionalities up to the complete system, show the tasks for a specified period of time. With this plan at hand, the single steps to fulfill each milestone's requirements are

developed in an iterative way. For every new part of the software a design is chosen based on the existing model of the software, the task itself and further background information. The following implementation needs to be tested until it fulfills the tasks requirements. The implementation step may also detect implementation problems which require modifications of the design to create a working solution. If the implementation is successful, it needs to be tested for robustness against false input and correctness for allowed input. If the further development requires changes to existing parts of the software to support the solution for the active task, these changes need to be verified in terms of their suitability for the intended purpose. After finishing a task for a milestone, the next task to reach the active milestone needs to be solved until all milestones tasks, and therefore the milestone itself, are completed.

The usage of the programming language Java is mandatory to create a plugin for Eclipse. The support through the PDE plugin to develop an Eclipse plugin is unchallenged, hence the programming took place in Eclipse. Being a highly developed and widespread platform, this was not a bad choice, as regularly required working steps are supported through inbuilt functionality like refactoring, code completion and navigatable source code. Furthermore, positive experience from different practical courses encouraged the onward usage of Eclipse.

3.1 Prerequisites

The development of an Eclipse plugin that supports development time weaving and therefore shows its effects directly to the user of course requires knowledge about the creation of a plugin for Eclipse. The creation of a plugin begins with the creation of a “Plug-in Project“ in Eclipse from which implementations for a wide range of plugins are possible. The plugin development is directly supported by the Eclipse plugin called Plugin Development Environment (PDE), hence the development of a Eclipse plugin can be done within Eclipse. After a survey of the basics of plugin development, detailed information about available and useful extension points will be given. The plugin should offer a possibility to enable or disable AOP support per project. An intuitive and extendable position has to be identified where this functionality should be added. Furthermore, the plugin has to provide functionality to react to the file updates from an AOP enabled project. Suitable techniques to retrieve useful information in this area are also required.

CDT has a long development history managed by the project leader Doug Schaefer from the Intel Corporation. The core of CDT is the handwritten parser, which has been extensively tested and improved. The reuse of this freely available[1] plugin gives access to robust information and places this project on a solid base. The extension of CDT requires an overview of the included and therefore accessible information from CDT about C files

[1] Licensed under the Eclipse Public License (EPL)

at first. The extension points that are offered by CDT have to be checked for usability. Furthermore, the direct Java based access of CDT within Eclipse has to be tested if it fits this projects needs and is therefore a usable base.

3.1.1 Eclipse plugin development

As mentioned earlier, it is possible to develop Eclipse plugins using the Eclipse plugin Plugin Development Environment (PDE) [ecla]. It is shipped as a bundle together with Eclipse, so that it is ready for immediate use. It offers a wizard for creating new plugin projects, including several examples that can be used to extend the given functionality. Furthermore, it includes a specialized editor which simplifies editing the plugin settings that are located within plugin.xml files. Some of the most important settings will now be introduced.

Core class The class that is the plugin's core class and will be instanciated if the plugin is used. The use of a plugin will be triggered if a contributed extension is activated. This class has to extend AbstractUIPlugin.

Dependencies The dependencies for this plugin that are necessary in order to be executable, which are usually the result of specified extensions. It also includes a minimum version for the required plugin.

Extensions which external extension points will be accessed and what will be sent to them, nicely guided, much better than raw xml writing, guidance is directly based on the extension point definition, so it is only possible to create approvable extensions

Extension Points Specify the plugin's extension points so that other plugins can contribute to it. This function is also often used to split large plugins into several smaller plugins which can then be developed independently.

Another feature of PDE is the plugin spy. It is an important support tool for plugin developers as it shows information about active parts of Eclipse. The extracted class names and plugin sources can be used directly for development. To invoke the plugin spy for user interface parts press Alt+Shift+F1 anywhere inside of Eclipse. Another plugin spy for menus is available by pressing Alt+Shift+F2. The menu spy presents information about the entry of a menu, like contribution item identifier, the URI (Unified Resource Identifier) of the active contribution, the active class and again its source plugin [CR08].

3.1.2 Eclipse core and extension points

The plugin has to be started with Eclipse because it has to compute status information before the user can interact with the development platform. The used extension points are described based on the implementation of this thesis' plugin [eclb].

One task of this plugin is to check all projects to determine whether they have AOP enabled. The enabled projects automatically get a weaver connected to them, and the project's aspect files and source files will be added to the weaver. Eclipse offers the possibility of loading a plugin immediately through an early startup technique.

Early startup

The standard way for Eclipse to load a plugin is called "lazy loading" because an installed plugin will only be loaded if its functions are required. That is the case if a contribution to an extension point exists, and the extension points function was activated. Loading of the plugin causes a short interruption, but it boosts the startup time of Eclipse as it relies on the intense use of plugins. If a plugin always and immediately needs to provide its functionality, it can contribute an extension to the extension point org.eclipse.ui.startup. A class has to be set as startup attribute, and that class needs to implement the interface IStartup. After instantiation the function *earlyStartup()* will be called, so all the startup action should be implemented there.

Enabling or disabling AOP support for a project will be done through an entry to the context menu of a project in the Project Explorer and in the Package Explorer since both of them handle projects as their entries and are standard tools for navigation through the project's resource tree. The menu entry should reflect whether or not a project is AOP enabled, so a dynamic menu contribution is necessary.

Menu contribution

Extending menus can be achieved by offering an extension for the extension point org.eclipse.ui.menus. A *menuContribution* entry at this extension is used to add entries at a place that will be defined by the attribute *locationURI*. This URI combines the global place for the entry with an identifier. The place is one of *menu*, *toolbar* and *popup*. Using the plugin spy it is simple to get the identifier which is called "the active menu contribution identifier" in the plugin spy popup. To extend the popup menu of the Package Explorer the locationURI would be 'popup:org.eclipse.jdt.ui.PackageExplorer'. This first step was just the pointer for the menu entry that will be added. It is defined as a subelement of the *menuContribution* entry and can, for example, be *command* or *dynamic*. Using the *command* subelement, it is possible to add a static function to a menu. This only requires usage of the extension points for *commands* that will be shown later. Adding a *dynamic* subelement offers the possibility of choosing a class that will be used to contribute to the menu. This class has to extend the CompoundContributionItem and implement the function *getContributionItems()* which returns an array of *IContributionItems*. These IContributionItems can then be chosen by the user and will trigger an action. This action

has to be registered as a *command* entry in the extension for org.eclipse.ui.commands. Such a *command* requires the declaration of its special attributes *name* and *id*. Finally a connection is built to execute the desired functionality at the menu entry using the org.eclipse.ui.handlers extension point. A *handler* entry for this extension point's extension has the required attribute *commandId* which is the *id* of the *command* that is connected to the menu entry. The *handler* also has a *class* attribute where a class may be registered that implements IHandler or extends AbstractHandler. These classes' *execute()* functions will be called if the *command* is triggered.

Eclipse has a mapping of file extensions to editors that claim to be the best choice for editing a specific filetype. This results in the automatic opening of a file in that registered editor on double click. It is also possible to choose an editor from a list that should open the target file, even if that editor did not register for the target's filetype.

Filetype

Registering a file extension for an editor can be done via the extension point org.eclipse.core.contenttype.contentTypes. By providing a *content-type* definition a filetype can receive a name that will be presented in the file properties.

Editor

Besides textual editing of a file, an editor could also offer a graphical representation of the file's content and offer special editing methods, for instance a UML diagram could be edited graphically and be saved as a XML file.

Every editor class has to implement the IEditorPart interface which is part of the contribution to the extension point org.eclipse.ui.editors, where all editors need to register. This extension point for example also offers possibilities to set a name for the editor or an icon that is shown in the editor tab.

It is a useful feature to show errors within an editor. To create such an error marker it has to be specified as a new marker type. The quickfix feature (see section 3.2.6) of Eclipse can then be used to suggest resolutions to solve the errors.

Marker

If a resource can contain wrong parts, and they are to be indicated to the user, a marker can be assigned to the resource. By registering a new marker type at the extension point org.eclipse.core.resources.markers individualised problem solvers can get access to the error. The extension has to provide the marker's name and an identifier for an attribute. Attributes are generic and are saved as name/value pairs. Names can be defined through

the extension or through constants from the IMarker interface to be able to access standard information. A marker is submitted to an IMarkerResolution if the user chooses the corresponding quickfix.

The creation of a marker is done directly on the resource, calling the *createMarker(String type)* function. The type can be one of the constants from IMarker or is self-defined by the extension.

Resolution generator

The previously added markers can get resolution proposals shown by Eclipse in the editor through the extension point org.eclipse.ui.ide.markerResolution. The entry *markerResolutionGenerator* refers to the *markerType* that it is designed for. The *class* which is set in the extension has to implement the interface *IMarkerResolutionGenerator*. This class is asked if it can return a resolution for a marker. It will then additionally be asked which resolutions, that are instances of IMarkerResolution, may be presented to the user as choice options. The resolutions have a label, an image and a method that will be executed if the user chooses the corresponding quickfix.

Knowledge about relevant extension points that can be used is important, but other techniques are also important. Information about active selections or a project's resources may be required for a plugin to fulfill its task. This information is available through classes that can be accessed through Java directly.

The *HandlerUtil* provides basic functions to support the usage of handlers. An IHandler implementation, that is registered as a command, receives an ExecutionEvent when it is called. This can be sent to different static methods of the HandlerUtil to get the active part from the Eclipse workbench. This part can be, for example, an editor or any other visible part of the workbench.

Accessing the Eclipse workbench requires use of the static class *PlatformUI*. The workbench is an instance of *IWorkbench*. As shown in figure 3.1, a strict hierarchy exists that handles editors inside the workbench and furthermore the editor's source, an IFile.

The unique workspace instance can be retrieved from the *ResourcesPlugin*. Figure 3.2 shows a diagram of the IResource Tree. An *IWorkspace* contains several *IProject*s. Each of those contains any number of *IFolder*s and *IFile*. Every IFolder and IFile of an IProject is associated with that specific IProject. Finally an IFolder can contain further IFolders and IFiles. The *IResource* that is extended by the other interfaces contains definitions for manipulation of an implementation like deletion or moving. IResource instances that are available at a lot of locations within Eclipse may easily be compared using the type of the IResource instance and the constants from IResource like IResource.FOLDER,

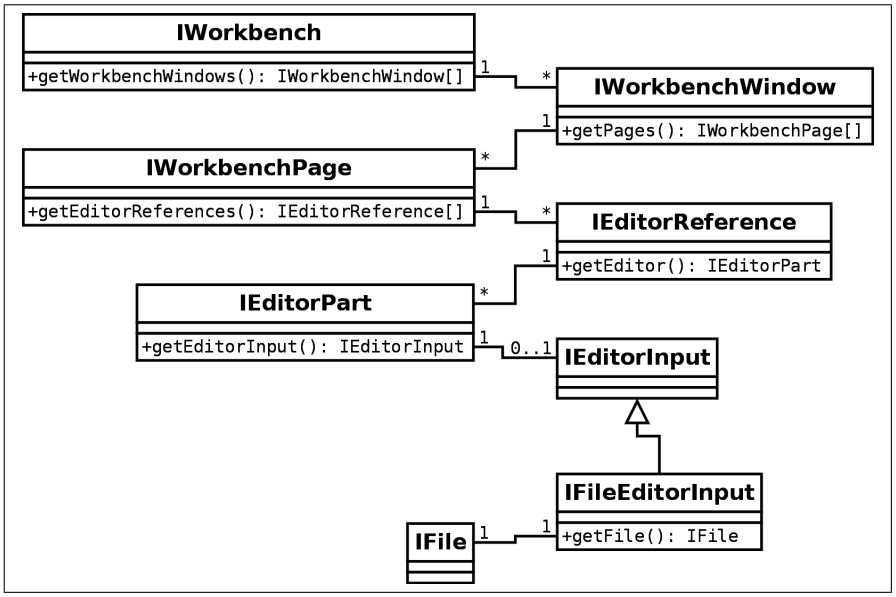

Figure 3.1: How to access an IFile from the IWorkbench

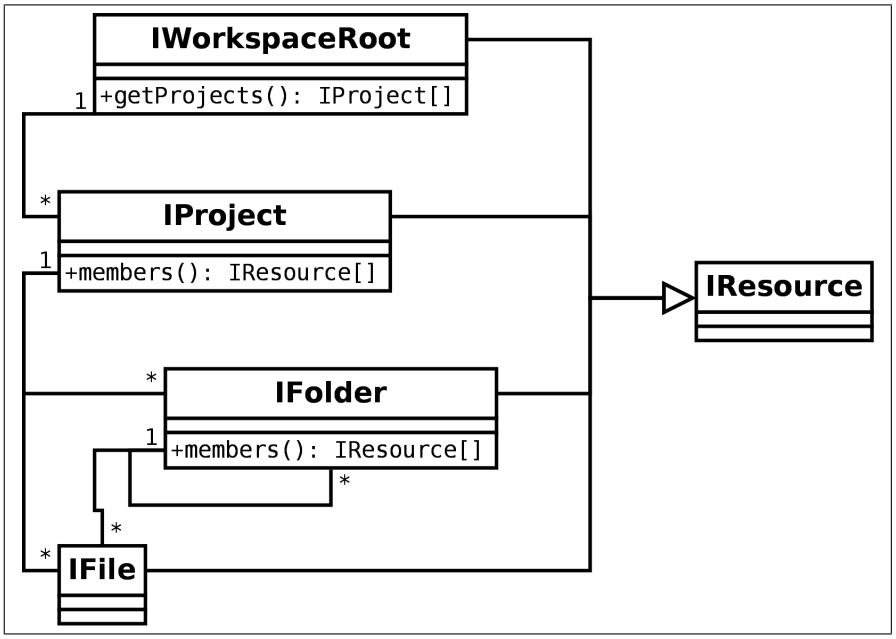

Figure 3.2: Hierarchical tree of IResource elements

IResource.FILE or IResource.PROJECT. Furthermore, the IResource defines support for creation, search and deletion of *IMarker*s, which are used to mark errors. [CR08]

3.1.3 CDT overview

CDT, being *the* C/C++ development solution for Eclipse, offers free access to its source code via Concurrent Version System (CVS). The latest version can be fetched from the CVS repository *:pserver:anonymous@dev.eclipse.org:2401/cvsroot/tools* by checking out *org.eclipse.cdt/all*.

To alter syntax checking and highlighting in the editor, the CDT parser needs to be modified. The project called *org.eclipse.cdt.core* contains all the classes that have to be altered.

CDT's C parser is entirely written in Java. It can be found in the package *org.eclipse.cdt.internal.core.dome.parser*. It parses a token list based on many functions, each specific for a state of the parsing and returning a node for the Abstract Syntax Tree (AST) that will be the parsers result. Helper functions provide access to the tokens list:

LA() returns the next token without advancing, also known as look ahead

consume() returns the next token and removes it from the list

LT(int i) returns type of the i'th next token

consume(int type) consumes next token if it is of the specific type or throws an exception

The mentioned type of a token is specified within the interface *IToken*, where every token type receives an unique integer. These integers are mapped to keywords within the *Keywords* class in the package *org.eclipse.cdt.core.parser*. This connection between keywords and identificators is, of course, necessary to afterwards parse the tokens. The *Keywords* class also contains the keywords that will be used for highlighting within the editor. These keywords are added to sets of keywords in the *Keywordset* class. By directly adding keywords to C syntax keywordsets, the editor will highlight those added keywords in the C editor.

The *IASTNodes* are created through a *CNodeFactory*. Additional functions need to be registered that create the node which is the return value for the parser functions. The *CNodeFactory* can be found in the package *org.eclipse.cdt.internal.core.dom.parser.c*. To insert new nodes into the AST, these need to be specified as interfaces in the package *org.eclipse.cdt.core.dom.ast*, with implementing classes inside the package *org.eclipse.cdt.internal.core.dom.parser.c*.

With the presented changes to the *GNUCSourceParser*, the existing C syntax parsing can be used and modified to support the syntax of SimpleAspectC.

A programmatical access to the basic information about projects and resources is provided through the *CoreModel* class. It contains static functions to check if a project is C project, if an *IFile* is a translationunit and if given source code is valid or contains syntax errors. This information can be used to delimit the possibility to enable AOP for a project to C projects only.

3.2 Design

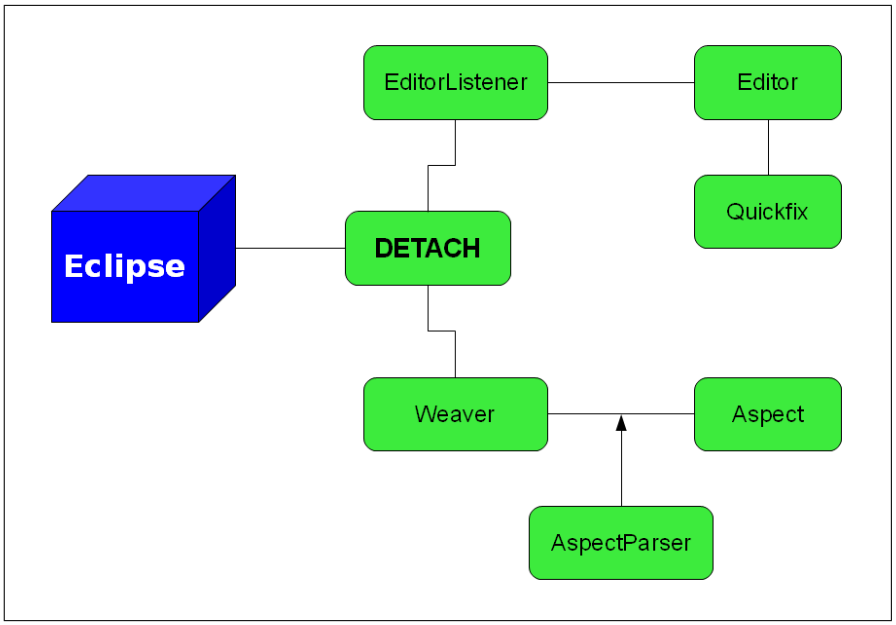

Figure 3.3: Abstract model of the plugin's components

In chapter 2 all tasks and the triggers for their activation are discussed. Additional environmental information is gathered in section 3.1. With this collected data the main components of the plugin can be identified. Figure 3.3 shows the basic setup of the components. The plugin's core that is accessed from Eclipse is called *DETACH* (DEvelopment Time Aop for C Hatchery). It handles the *Weavers* for the projects. Aspect source files need to be parsed into *Aspects*, using the *AspectParser*. The Aspects will be used by the Weaver to update the C source files of a project after changes to an Aspect. Changes are effected editing the source file, so *EditorListeners* are used to track changes to those files. The modification of a file may corrupt the consistency, so the Weaver has to check it after changes to any file. An inconsistent state will be marked and *Quickfixes* will be made available to return to a consistent state inside an *Editor*.

The AspectEditor is developed on the base of CDT and exported as a separate plugin. The markers, the Quickfixes and the EditorListeners will of course affect the AspectEditor, which supports syntax highlighting and grammar check for aspect files. The development required direct extension of the CDT parser.

3.2.1 Plugin core

The core of every plugin is a class that is registered as the starting point in the plugin.xml. To grant Eclipse access to that class, it has to extend the class *AbstractUIPlugin*. For this thesis' Eclipse plugin, this core class is *DETACHPlugin*, an abbreviation for "DEvelopment Time Aop for C Hatchery".

The main tasks of the core are to set up the plugin and integrate it into Eclipse. Every available project has to be checked, if AOP support is enabled for it. Then a weaver will be created for those projects. Furthermore the weaver will get all the project's files, including all aspect files. Besides weaver setup it is necessary to track changes in the workbench, like opening editors, to be able to update after saving and to update C files directly in the editor. Those changes can be monitored by an *IPartListener* which is added to the existing workbench pages, as they all may contain editors. The implementing class *BasicPartListener* will be described in section 3.2.2. It is added to the workbench from the "enable AOP" method, also requesting it to initialize itself, which triggers the check of all open editors and to add listeners to editors that may sometime require the weaver to update the system.

Since Eclipse introduces the "lazy loading" mechanism, a plugin is only loaded if a user action triggers the necessity to execute a part of the plugin that is registered through an extension. Lazy loading is not appropriate for this plugin, as its functionality is required from the beginning. Therefore the "early startup" extension has to be registered, requiring this plugin to implement the *IStartup* interface. This interface obligates the implementation of the *public void earlyStartup()* method that is called as soon as the workspace is available. This starting point will then be used to check the existing projects and intialize weavers.

Managing the weaver that belongs to a specific project is the main function of this class at runtime. After the initialisation, a map of existing projects that are AOP enabled exists, and this map can be altered through the public methods *enableAOP(IProject)* and *disableAOP(IProject)*, checked through the function *isAOPEnabled(IProject)* and read through the function *getWeaver(IProject)*. A check should be done before trying to get the weaver for a project. Enabling AOP support executes the previously described proce-

dure of creating a weaver and adding all the project's files to that weaver, of course also putting it into the map and saving this information inside the project folder to recognize it for the next startup. Disabling the AOP feature will drop this recognition, reset the weaver and finally remove it.

3.2.2 The listeners

Two listeners build the base that retrieve relevant changes in the workbench. Opening an editor, for example, has to trigger the addition of a listener to the editor to be able to inform the weaver of the necessity to check consistency or update the system. Only those editors that access aspect files or source files from an AOP enabled project are of interest.

Besides the knowledge about the existence of those editors, their changes have to be tracked. The aspect files contain aspects. Changes to one aspect requires an update of all C source files. To track those changes an *AspectEditorListener* will be added to every editor with access to an aspect file. Changes to a C source file should lead to an update of that file to keep the consistency, therefore a *TargetEditorListener* will be added to editors that access C sourcefiles. Details about the need for updates can be found in section 2.4.

The **BasicPartListener** has the task of registering AspectEditorListeners or TargetEditorListeners on opening editors in the workbench. Activation of the BasicPartListener is requested at any time that a project is set to be AOP enabled. This can happen at startup, to return to the last state the user was working in, and at runtime by user interaction. At startup Eclipse may return to the last working setup, so the editors that were opened the last time will be opened again. The editing of a C project before enabling AOP may also result in editors that are already open and have to be tracked. Therefore the method *initWorkbenchPage(IWorkbenchPage)* has to be executed to check all editors that are already open and register listeners on the interesting ones.

Editors with access to aspect files of AOP enabled projects will have an **AspectEditorListener** attached. The listener informs the weaver of the project about the need to update as soon as the editor's input is saved.

The **TargetEditorListener** has the task of updating error markers after saving. This triggers first the removal of existing markers and afterwards the consistency check which will automatically create markers in the editor for missing aspectcode.

3.2.3 Weaver

The *Weaver* class includes the most important procedures. Lists for aspects and C source files, called *targets*, are maintained to react immediately to the demand to update and keep the system consistent. Furthermore, the Weaver relies on a *ILanguageSpecifics* implementation which serves helper functions as a source for language specific constructs. This approach was chosen to possibly enable integration of other languages into the plugin. More information about the possibilities and challenges associtated witht the use of this plugin for different programming or AOP languages can be found in chapter 6.

Weaving is of course one of the main tasks of this class. For this development time weaver the weaving procedure consists of the unweaving of old aspectcode and weaving of active aspectcode for every target. Assuming the target contains no aspectcode, the first action will be the compution of joinpoints from every advice. Iterating over the lexicographically ordered aspects the advices will be checked in order of appearance. The joinpoint data is being stored within the *Target* (section 3.2.5). Possibly several advices target the same joinpoint, so the newly acquired information about those sets of *execution advices* extending the same joinpoint can be combined. The joinpoints will be listed in descending order before iterating over them, so that the inserted aspectcode will not affect the position of other joinpoints, which might require updates of all the joinpoints.

The combination of *execution advices* reduces the amount of new functions that include the aspectcode. In comparison to creating renamed functions for every advice that matches a joinpoint, this technique improves readability. The code from every advice will be capsulated in its own block to ensure that the adviceblocks will not overwrite existing variables. This also has the positive side-effect that variable declarations, which have to be in the first position of a block, do not need to be combined in order to produce fully ANSI-C compatible code. This of course requires correct code within the adviceblock.

Advices are ordered for every joinpoint. This order has direct consequences for the combination of the advices. A single *before* or *after execution advice* creates one forward declaration, one hijacked function and one renamed function.

Listing 3.1: Example of one woven *before exec advice*

```
1  void foobar_AspectOne_BeforeAdvice(); //forward declaration
2  void foobar() //hijacked function
3  { //hijacked function block begin
4    do { //secure block begin
5          advice_code; //adviceblock from aspect AspectOne advice BeforeAdvice
6    } while(0); //secure block end
7    foobar_AspectOne_BeforeAdvice(); //call to renamed function
8  } //hijacked function block end
9  void foobar_AspectOne_BeforeAdvice() //renamed function
10 {
11   original_foobar_code;
12 }
```

This example shows how a *before execution advice* extends the C code. An *after execution advice* would create similar code, except that the adviceblock and its container block would be executed after the call of the renamed function. If the function had a non-void return type, the return value of the call to the renamed function would be stored in a temporary variable which would be returned in the end of the hijacked function. This example shows directly how advices can be combined. Every adviceblock is encapsulated in a secure block and the renamed function has to be called after the *before advices* and before the *after advices*.

Any number of following *before advices* can be kept in its order. *After advices* that exist directly after those *before advices* can be grouped together with the preceding *before advices*. Another advice after the last after advice in row will create a new group of combined advices. Figure 3.4 shows how several advices could be attached to a joinpoint in this specific order, where A denotes *after* and B denotes *before*. Following *before* and *after advices* could, for example, have the same aspect as their source. If each level in figure 3.6 presents an aspect, these could, for example, be that the first aspect creates a connection to a remote server for some function, disconnecting from that server at the end. The next aspect could be an authentication and the last might be a data transfer. This example shows that of course the *after advice* from the first aspect has to be reached as the last advice. In the meantime the authentication aspect has to execute its *before advice* to be able to transfer data. Afterwards the application could be logged out before finally cutting the connection.

Figure 3.4: Possible [B]efore and [A]fter advice array with the same joinpoint

Figure 3.5: Separation of groups of combinable [B]efore and [A]fter advices from figure 3.4

Each group will be inserted into one hijacked function that calls the original renamed function after its *before advices*. In figure 3.6 each group would get its own function calling in the top down manner from the pointers. The lowest group of course calls the original function.

The existing knowledge about these woven parts can be used for updating. The existing aspectcode parts simply need to be removed to start from the beginning with a sourcefile

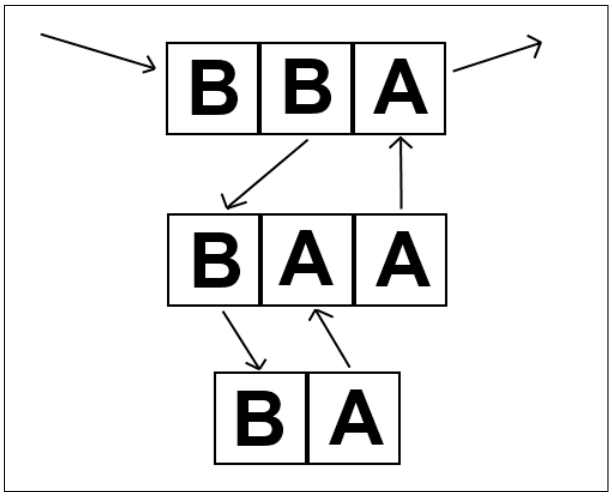

Figure 3.6: Call order of the grouped advices

that contains no aspectcode. This procedure guarantees consistency. The direct update of the woven aspectcode is far more complex and goes beyond the scope of this thesis.

Similar to updating the consistency can be checked. This check tests whether the woven code contains the aspectcode that could be generated from the systems aspects for it. Therefore the test will unweave the woven code, generate matches from the aspects' advices and check if the code that would be woven is contained within the target. If some pieces are missing, these will be marked. The different pieces so far are *header*, *forward declaration*, *adviceblock* and *footer*. Section 3.2.6 describes available solutions for those errors.

Within Eclipse the weaver class is called *EclipseLiveWeaver* and it extends the *Weaver* class. The Weaver implements those functions that are independent of Eclipse. The plugin is not available as a standalone version at the moment, because the AST from the CDT is retrieved from the installed CDT plugin in Eclipse. If the CDT parser can be extracted, it could be used to create the AST instead, but this approach goes beyond the scope of this thesis.

The explanation of the weaver's features already suggests that only a subset of the AOP language has been implemented for testing purposes. The plugin basis can now easily be extended to support the full language or a variation of it. Right now only the *exec* method is available to create a pointcut. It can be combined with before or after and will be successfully woven. Furthermore, the pointcut supports the wildcard symbol "%", thus making it more expressible. With union "||" and intersection "&&" as available pointcut extensions, a moderate strength of pointcut expressions is being achieved. Information about the pointcut system can be found in section 3.2.4.

3.2.4 Aspect parser

The information that is contained in an aspect file is needed in specific data structures to be able to compute joinpoints from the pointcuts and weave adviceblocks. Therefore a handwritten aspect parser was created. Three steps are required here: first tokenizing, then improving and finally parsing. The tokenizer will create tokens for known keywords from the aspect language. Active keywords are “aspect”, “advice”, “pointcut”, “before”, “after” and “exec”. Other strings will be handled as string tokens. Symbols that will be translated into tokens are: “:”, “{”, “}”, “(”, “)” and the newline. Finally whitespaces will also be inserted as stringtokens, because they will be needed later.

Improving these tokens is necessary to create maintainable syntax check and parsing. The improved parts of the aspect file are simply the combination of all tokens before the aspect begins into a C code header, the tokens from an adviceblock are put into an adviceblock and the final lines after the aspect block become a C code footer. Due to the fact that a tokenizer is not aware of the context these steps were separated.

The subsequent check of the existing tokens is implemented similar to the BNF from the aspect grammar. A set of expected or correct further tokens is always available, and the next token in line is checked if it was expected. Based on every possible token, the following operation that updates the expected tokens set is determined.

The Aspect class is used in the weaving process and contains advices that have the same order as in the aspect file. This order is crucial for the weaver to create proper output. Figure 3.7 shows the contents of an Aspect and its Advices, as well as the model for the pointcut.

A *Pointcut* consists of a list of *IPCExpression*s. For example the advice declaration *advice example : before exec(“char get%()” || “int get%()”)* consists of the three IPCExpressions *PCEFunction* (“char get%()”), *PCEUnion* (||) and another *PCEFunction* (“int get%()”). The algorithm that calculates the matches will first look for function definitions that match the first PCEFunction and store them temporarily. The next IPCExpression determines what will happen to the temporary store. The PCEUnion directly transfers it into the result set. Alternatively, a PCEIntersection would store the intersection of the actual temporary joinpoint set and the next IPCMatchableExpression in the temporary store. This mechanism, that is at the moment limited to the classes in figure 3.7, can be easily extended to support further pointcut expressions.

The PCEFunction uses the AST from the CDT. All functions from the C source file are available as IASTFunctionDefinition instances which themself are declarations. The declarations are available from the AST via *getDeclarations()*. The IASTFunctionDefinition contains a specifier, which is the return type of the function, and a declarator, the function name. This information can be used to match against the PCEFunction. A positive match can be considered as a correct joinpoint.

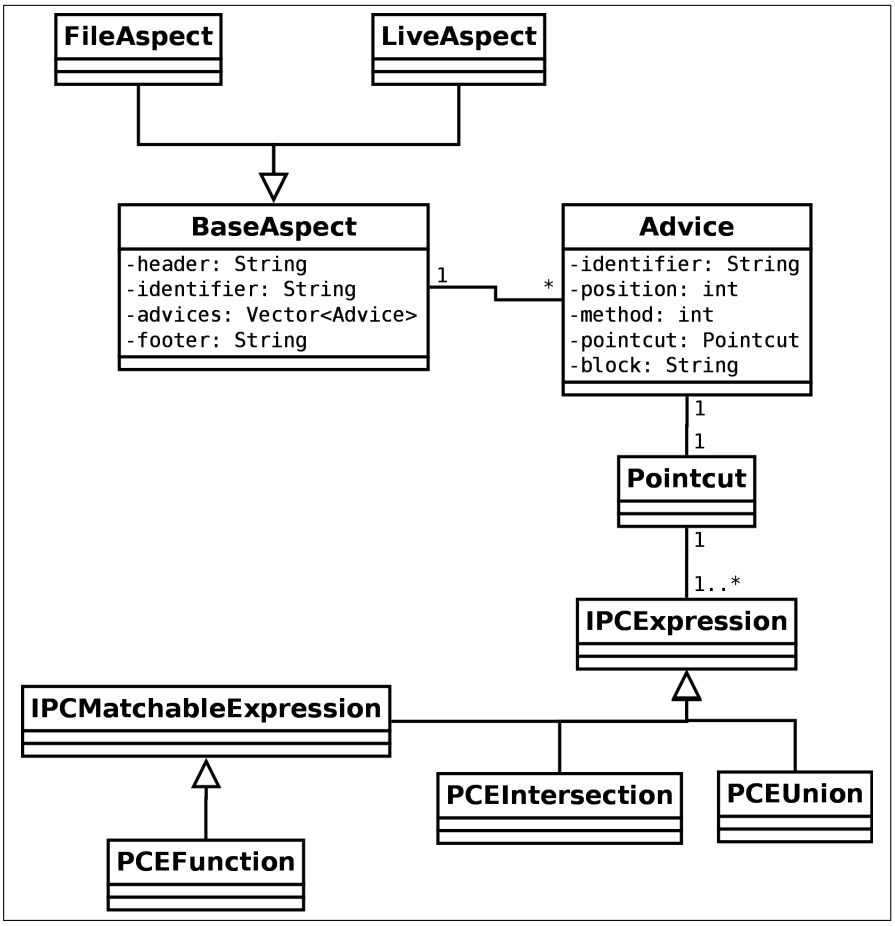

Figure 3.7: Class diagram and connection to the pointcut expression hierarchy

3.2.5 Target file handler

The abstract class *ATarget* is used as a handle for one C sourcefile. This file needs to be updated by the weaver from time to time. The C sourcefile is the standard case for a Target. The methods *get()* and *set()* are available for accessing the content of a file. Eclipse requires another setup of the Target, as every change to a file is traced and can be undone. A file with many joinpoints that are woven one after another would create a long list of unnecessary undo steps. To suppress this behaviour, a simple buffer was created within Target that would be asked to set its content in a final step after the update. After an initial intialization the buffer will be edited and a final *setBuffer()* method sets the content of the buffer to the Target's output channel. Several subclasses exist to operate on an IFile or an ITextEditor within Eclipse.

Besides being a handle for a file, the Target also knows about active joinpoints and joinpoints for the next update. These lists can manage several advices per joinpoint. The active joinpoints list supports direct checking if the consistency from the last weaving has been damaged by the user. Therefore a simple check for the existence of the aspectcode from the advices is executed. Damaged parts can then be reset by quickfix resolutions.

The creation of error markers within Eclipse provides the opportunity to notify the target itself about those errors and give it the task to inform the user about those error markers. Within Eclipse they will be visualized and provide quickfix resolutions, a possible commandline tool also already has this accessible method implemented and can therefore inform the developer, even without an IDE.

3.2.6 Quickfix resolution generator

Quickfixes are the Eclipse way of providing solutions for errors in the sourcecode to the developer. These solutions are connected to the errors through markers for the errors. Markers need to be registered within Eclipse and may be created on an IResource. A registered ResolutionGenerator will be asked about its ability to provide a solution for an error.

The markers are generic and only need to be created. They receive attributes about the line in which they are to be placed, a range is also possible, and about the error itself. The *AOPConsistencyResolutionGenerator* provides solutions for errors that the Weaver finds in its *checkConsistency()* method.

Identified errors can be one of four different types. Header, footer, forward declarations and the adviceblock with hijacked and renamed functions are the code pieces the weaver creates and each of them can be tested if the target contains the specific piece. This information is saved in the marker attribute “aop_error_type“, the possible types are defined constants within *IAOPMarker*. A missing or changed piece will create a marker

at the line where the piece should exist. This information is available from the list of joinpoints every Target manages.

Available solutions for every marked error are put together by the AOPConsistencyResolutionGenerator method *getResolutions(IMarker marker)* which decides which resolutions should be available for the user to choose from.

So far implemented resolutions provide possibilities for creating consistency within the target. A simple case is the removal of all active aspect code and afterwards weaving, which is a generic possibility to gain consistency. Being aware of special weaving parts is crucial, for instance a hijacked function line may not be deleted. The removal would lead to a C block without a function definition, which of course would be a C error. That means the hijacked function comment may be removed, but the function declaration has to stay. Any further development must take this into account.

A specialized quickfix solves errors within forward declarations that were damaged by a developer, removes the damaged code and inserts forward declarations for the active advices at the specific joinpoint. For development purposes a quickfix that checks the consistency again was implemented.

3.2.7 Aspect editor

The CDT *CEditor* will be used as a base for the aspect editor. As described in section 3.1.3, it is necessary to directly modify the C parser to check the syntax. Keywords that should be highlighted have to be set in the same package that contains the C parser. The keywords “aspect”, “advice” and “pointcut” are registered as additional keywords for C syntax, and will be highlighted in the *CEditor*.

More work is required to insert the AOP syntax from the SimpleAspectC language into the C parser. For the keywords “aspect”, “pointcut”, “advice”, “before”, “after”, “around”, “exec”, “file”, “decl”, “def” and “call”, tokens will be registered. Furthermore, object handles are required for the AST for AOP treenodes. Therefore, interfaces for the nodes *IASTAOPAdviceStatement*, *IASTAOPAspectDeclaration*, *IASTAOPCompoundStatement*, *IASTAOPProblem*, *IASTAOPProblemStatement* and *IASTAOPStatement* are created in the package *org.eclipse.cdt.core.dom.ast*. Implementing classes are required for each of these interfaces in the package *org.eclipse.cdt.internal.core.dom.parser.c*. All of them will be created in the *CNodeFactory*, where functions need to be added to create these objects.

Based on this basic setup, the parser can then be extended. The aspect statement parsing function needs to be called from the declaration function. From there the function calls follow the SimpleAspectC grammar. The previous described objects will be created if the specific tokens are found, else an error token has to be created.

3.2.8 Main components interaction summary

Every Eclipse plugin has one core class that is registered in the plugin setup to be instantiated as soon as the plugin is required. The core of DETACH is called *DETACH-Plugin*. Due to the lazy loading that Eclipse uses, this plugin provides an extension to org.eclipse.ui.startup to be started early, just after the startup of the workbench. At startup the core checks the existing projects for their AOP state and adds one Weaver instance per project that has AOP support enabled to an internal mapping of IProjects and Weavers. If at startup or at runtime, a project that receives the status of being AOP enabled will be initialized instantly. Therefore listeners are added to notify the Weaver of updates within aspect files or C source files, and update the system accordingly to keep it consistent. Besides the insertion of listeners all the projects aspect files and C source files will be added to the Weaver. With this complete set of resources it is now possible for the Weaver to check the system's consistency, as it is a required precondition for further updates to be correct.

After this startup the plugin will respond to user actions at runtime. AOP support can be enabled or disabled within the popup menu of a project in the Project Explorer and in the Package Explorer. By enabling AOP support for a project, this information will be saved within the project's workspace, which is necessary for the detection of AOP enabled projects at startup. Furthermore, the project will be mapped to a new Weaver and its resources will be added to the Weaver and checked for consistency.

Other possible modes of interaction between user and plugin are the editing of files and the selection of a quickfix for an error marker. Whenever the user saves an aspect file in an editor from an AOP enabled project, this saving process triggers an update of the Aspect this file holds, which will then result in an update of all target C source files of the project. The aspect file's content will be parsed and creates an Aspect instance that has several Advices with the same order as inside of the aspect file. The Aspect itself will replace a previous version of the Aspect or it will be inserted into a list that is lexicographically ordered. Every Advice's pointcut will be handled as a Pointcut with a list of IPCExpressions. The Pointcut supports compution of joinpoints based on its IPCExpressions.

The update starts with the removal of all aspectcode within a target file, the compution of joinpoints per advice and finally the weaving. By saving a target file in an editor this file's consistency will be checked based on the saved joinpoints from the ATarget that handles this target file. The consistency check may return errors like missing or changed aspectcode, and these errors will be marked, hence become visible to the developer. The AOPConsistencyResolutionGenerator provides quickfix proposals the developer can choose from to solve the error.

3.3 Implementation

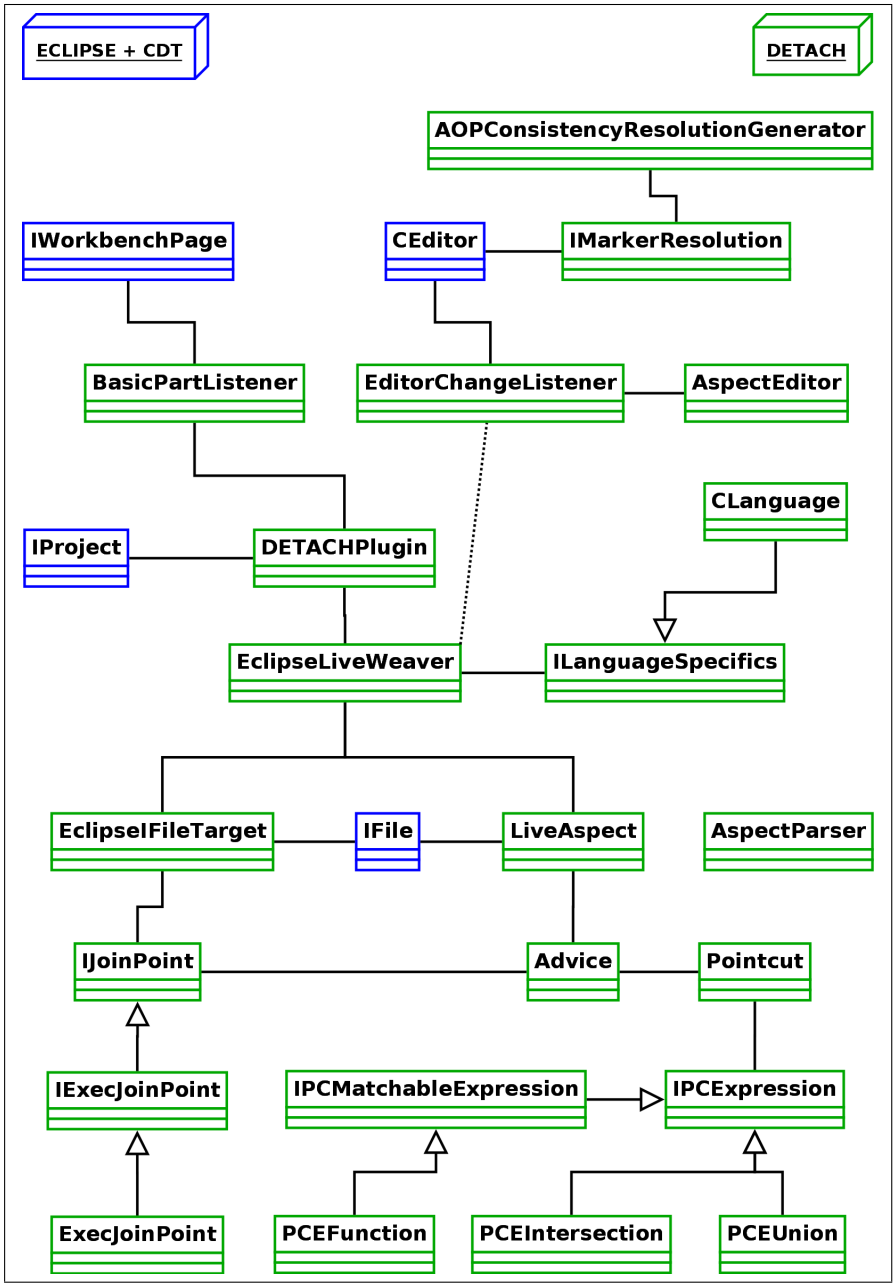

Figure 3.8: Overview of the classes and their connections.

The class diagram visible in figure 3.8 shows the actual implementation, specified by the design from section 3.2. For readability reasons, some parts of the plugin are left out, like interfaces and superclasses that would overload the diagram without providing necessary information for the interaction and access of the classes. Furthermore, existing Eclipse and CDT classes, to which parts of the system are connected, are outlined blue, classes from the DETACH plugin a colored in green.

The core of the plugin is the *DETACHPlugin* class. This class is loaded by Eclipse at startup, since it registered at the org.eclipse.ui.startup extension point, to be able to provide its functionality from the beginning. This is necessary, because the plugin will automatically enable AOP for those projects, which were set to be AOP enabled in an earlier Eclipse lifecycle. All projects will be scanned by the core, the ones which contain the information of being AOP enabled receive an internally mapped connection to a *EclipseLiveWeaver* instance. This instance then receives information about all the project's files. C source files are handled in *EclipseIFileTargets*, which know about their source that is an *IFile*, the Eclipse file handle. Aspect files will be parsed by the *AspectParser*, and are handled within *LiveAspect* instances, each connected to an *IFile*. Every *LiveAspect* consists of several *Advices*, ordered exactly as within the aspect file, as this order is crucial for the logical execution order of the *EclipseLiveWeaver*.

Another important information to be able to weave an advice is the *Pointcut* that is implemented for the advice. It is internally represented as a list of *IPCExpressions*. This technology is easy to be extended. At the moment support for simple constructs is available, a pointcut can contain function identifiers which are connected by *PCEUnion* and *PCEIntersection* instances. Their effect is similar to basic logic "and" and "or" set modifiers. To support all the different types of pointcut expressions from the SimpleAspectC language described in section 2.2. The interface *IPCMatchableExpression* is used to distinguish between atomic operations and matchable expression that require the calling of their own matching functions. A *Pointcut* can be asked to return matches for a *Target*, the *EclipseIFileTargets* superclass. It will then compute the *IJoinPoints* for the target's value based on the list of *IPCExpressions*. The returned array of *IJoinPoints* is used in two cases. First, it is used to be simply added to the target file as an information for the *Weaver* for the upcoming update. Second, these *IJoinPoints* are used by the consistency check, to see if all the aspect code that should exist is included in a target, and that the actually woven advices are not different from the expected advices, so, like the name says, it checks the consistency. Other constructs from the SimpleAspectC language for pointcut methods will result in further sub-classes of the *IJoinPoint*. The *Weaver*, as the superclass of the *EclipseLiveWeaver*, will have to distinguish between the different *IJoinPoint* subclasses, foundations and comments for this work are included in the source code.

The next visible connection of the *EclipseLiveWeaver* is that to *ILanguageSpecifics*. As a first step towards support for different languages, basic information that is required by the *Weaver* is handled in the implementation *CLanguage* of that interface. These are common used information like the block begin, statement end or comment begin. At the moment, this information is used by the *Weaver* to generate the source code that is visible in chapter 4.

The *BasicPartListener* is added to existing *IWorkbenchPages* to see opening editors and check existing editors. They might be interesting for the development time weaver,

as they can edit files of an AOP enabled project. If this is the case, the editor will receive an *EditorChangeListener*, which triggers updates of the system or consistency checks for a target file if the editor was saved. This listener will listen on an *CEditor*, the C editor from CDT, and any other text editor that contains a C file from an observed project. If the target files contains AOP consistency errors, markers are generated that visualize the error. The *AOPConsistencyResolutionGenerator* handles quickfix proposals for Eclipse, based on the type of the error. The developer can choose from several existing quickfixes to resolve the error and return the current file into a consistent state. An *EditorChangeListener* is also added to *AspectEditors*, where saving triggers the update of the whole system, because an aspect can potentially change every single source file of the software.

The single classes implement functionality based on the description from the design. Information for the further development of the plugin can be found in chapter 6. The listed possibilities to extend the existing functionality can be implemented in further work, based on the present foundation.

Chapter 4

Test

The plugin has to be tested in three disciplines. The usage will be explored first to identify features and optimization potentials. This also includes AOP development with the plugin, and leads directly to the next discipline which is a check of the output's correctness. This test will show the active capabilities and limits of the implementation for recognition of C source code and insertion of aspect code. At last the performance of different operations and tasks will be tested and discussed to discover the plugin's potential.

The tests will be based on C source code from open source projects. Furthermore, these tests will provide information about weaving into common C source code, the performance of weaving and consistency checks and the usability of the plugin.

Before the tests can begin, the environment has to be set up. This consists of installing Eclipse and the plugins and installing a C compiler toolchain. The performance tests required some extensions to the Eclipse plugin, capturing exact execution times per function.

4.1 Test environment installation

Software that is required for the use and testing of this plugin will be introduced. The derivate of Eclipse is operating system dependent. The plugins for Eclipse only depend on the version of the IDE and can be installed through the internal software installer.

After installing the software, the creation of a C project will be shown. Usage of the AOP plugin is project independent and therefore covered in its own section. Finally, these basics enable the tests with foreign source code.

4.1.1 Eclipse

Eclipse[1] serves as a development platform for several programming and scripting languages, therefore bundles including plugins to support the language of choice can be

[1] An archive containing Eclipse is available for download from http://www.eclipse.org/downloads.

downloaded, enabling the development to start immediate. The bundles are available for Linux, Mac OS X and Windows, each in 32 Bit and 64 Bit. This wide range of Eclipse versions makes it a usable tool for the majority of developers.

The test environment for this thesis' plugin requires the CDT plugin for Eclipse. A bundled version including CDT is available and has to be downloaded. The zip file contains the Eclipse folder with the executable, the plugins and further settings files. Afterwards all that needs to be done is start the Eclipse executable.

The execution of Eclipse does not require the execution of a previous installation routine. This provides the possibility of extracting the archive into different folders, each presenting an independent Eclipse instance. Plugins that are going to be installed reside inside the specific Eclipse instance, they are not globally available. Different setups for testing and development purposes can therefore exist in parallel.

4.1.2 CDT

An existing Eclipse instance can be extended to support C/C++ development by installing the CDT plugin. Eclipse manages plugins through an internal installer. It is accessible through the main menu "Help" under the entry "Install New Software...". A so called *site*, the location of an online repository from which to download the plugin, needs to be added to install CDT. The location of CDT depends on the Eclipse version, the latest bearing the name "Helios"[2]. Afterwards the installer presents trees that show main features and optional features. Simply installing the whole set will work for the plugin test.

4.1.3 DETACH installation

Just like CDT, the DEveleopment Time Aop for C Hatchery (DETACH) plugin needs to be installed using the "Install New Software..." dialog from the "Help" main menu. By activating the "Add..." button in the top right corner, another dialog pops up, providing the possibility to choose an archive through the according "Archive..." button. By executing this operation for the archive *detach_base.jar* and final selection to install the plugin in the first dialog, the DETACH plugin will be successfully installed and usable.

The aspect editor exists in a development version. The source code is set up in a workspace directory that can be opened with Eclipse Helios. By executing the *org.eclipse.cdt.ui* project, an instance of Eclipse with the modified C editor is opened and can be used.

[2]The location for the installation of CDT for Eclipse Helios is http://download.eclipse.org/tools/cdt/releases/helios. All available CDT versions can be found at http://www.eclipse.org/cdt/downloads.php.

4.2 Test tasks

The tests for usage, correctness and performance of the plugin need to be specified. The interaction possibilities and automatic triggering of functions will be checked in section 4.3.1 by comparing expected behaviour with the observed behaviour.

The check for correctness will be based on the weaving result. It needs to create correct C syntax and change the control flow in a correct way. Finally, the compiled executable that includes aspects has to show its effect.

As a last test, the performance of highly frequented functions will be tested for different inputs. This required modification of the plugin's source code is intended to measure the elapsed time of important functions.

4.2.1 Usage

The plugin's functions have to be tested for correct execution. This includes enabling AOP for a project, the editing of project files and aspect files and the effect of saving them. Creation and other changes to the files and folders of a project should be tested for their effect. Disabling AOP support for a project should return into a correct state, as well as re-enabling afterwards.

Quickfixes for error markers for missing or changed aspect code have to be tested to see if they work correctly. Therefore the different areas where these quickfixes are available need to be tested as well as the quickfix solutions.

The system's reaction, including detailed information about the steps to execute the test, will be presented for every testcase. Afterwards a brief conclusion for the specific testcase will be illustrated.

4.2.2 Correctness

The weaving of aspects into the source code of the base system has to generate correct C source code. This presupposes, of course, that the project in each test consists of correct C source code. Therefore different aspects have to be created and tested, to test the possible outcome of the weaving process.

States that lead to false output need to be identified as limits on the applicability of this plugin, as they clearly corrupt the result's correctness.

4.2.3 Performance

Testing is required on those parts of the software, which contain important functionality and build the core of the plugin. Three functions were identified to implement crucial mechanisms, so each of them will be tested with different setups in the two following tests.

The first test will consider the *initialization* of the plugin, as it delays Eclipse's startup. Initialization consists of creating weaver instances for every project that is AOP enabled, the registration of all the project's files for the weaver, the compution of joinpoints which are expected to be woven and a final consistency check for all target files. The initialization time will be the average duration from 100 initialization executions.

Next, the performance of the consistency check will be tested. This check is executed for every file at initialization time and after saving a target file from an editor in Eclipse. It returns information about inconsistency that will be shown in the editor. The performance will be inspected in the tests by measuring the execution time in relation to the file size and amount of potential joinpoints based on an static amount of advices that need to be checked. Due to the limitation of pointcut expressions in order to be able to match function executions, the amount of potential joinpoints will be the quantity of function declarations. To minimize influence from concurrent processes within Eclipse or from the operating system on the results, the test will be set up to automatically execute 100 times and return an average in the end.

The weaving performance will be the last test. It will be executed by changing an advice, an action that triggers the update for the whole project. So every target file will be checked to determine whether it needs an update. The removal of old aspect code, compution of updated joinpoints and the final weaving are involved here. The time will be shown in reference to the amount of files that need an update and the amount of potential joinpoints. This test will also be set up to execute 100 times automatically to get usable data.

The time that the different functions require is not an absolute dimension. An important factor for it is the hardware that is used to run the tests. The test hardware is an AMD Athlon 64 X2 Dual Core Processor 3800+ running at 2GHz with 6GB of RAM[3]. As it turns out, the plugin requires a big amount of RAM to be able to handle large projects. Consequently the Java Virtual Machine in which Eclipse runs was set to 3GB of RAM with the parameter *-Xmx3072m* being used for the Eclipse launch.

4.2.4 Description of the projects used within this test

To perform the described tests, C projects are required as targets for exemplary AOP implementations. To test the performance for different sized projects, one small project and one large project are chosen. The plugin does not rely on the specific source code from the chosen projects. They are exchangeable and can be replaced by other projects. For this test the gzip project and the linux kernel project will act as the test subjects.

[3]Due to hardware problems with the harddrive, a ramdisk is set up that is used as the target for temporary files. This has the side-effect, that the harddrive speed does not influence the measured times.

The gzip project

The gzip project provides an open source commandline compression utility. It is written entirely in C and available in the latest version 1.2.4 [gzi], developed by Jean-loup Gailly and Mark Adler [jlg, mad]. They were rewarded with the Software Tools User Group (STUG) award 2009 for the freely available results of their work on compression algorithms [stu]. Besides compression it also supports decompression of archives and is delivered as a standard tool of several popular linux distributions.

The linux kernel

To test a large software system, the linux kernel is chosen. Initiated by Linus Torvalds in 1991, the linux kernel is used in a lot of distributions and operating systems [lin]. Version 2.2.26 was chosen because its sources require less than 100MB. It could therefore be handled by the test's hardware setup. In comparison, the kernel 2.4.37 requires 185MB of harddisk space and the plugin was not able to handle this amount. Optimization possibilities need to be determined.

4.3 Execution of the tests

The project independent discipline usage will be checked before the correctness and performance for the projects gzip and the linux kernel 2.2.26[4] are tested. These two projects are chosen because they show the differences between using the plugin for small and large C projects in terms of the pure number of files, the range of file sizes and an expected wide range of C syntax usage.

4.3.1 Usage

The plugin's usage test requires a C project in order to be performed. In this test the source code from the gzip project will be used, although any other C source code could be used equally well. The test aspect 4.1 is saved in a new file called test.ac that will be used to perform several tasks throughout the test.

Listing 4.1: The aspect for the usage test

```
1 /* This file is modified by test.ac */
2
3 aspect Test {
4   advice greetings : before exec("% main()") {
5     puts("Greetings from test.ac");
6   }
```

[4]This specific version is available from ftp://ftp.kernel.org/pub/linux/kernel/v2.2/.

```
7   advice goodbye : after exec("% main()") {
8     puts("test.ac says goodbye!");
9   }
10 }
11
12 /* This file was modified by test.ac */
```

Testcase: Enable AOP support for a C project through the project's popup menu entry "enable AOP support".

The project becomes AOP enabled through activation of the project's popup menu. The triggered intitialization of the weaver becomes visible as an active operation in the workbench which finishes quickly. Its effect is that most of the project's C files receive an marker that is visible in the Project Explorer, informing the developer about problems inside the file. All errors can be viewed in the Error Log below the editor area, providing the possibility of jumping directly to the specific location.

Conclusion: Enabling AOP for a C project works correctly.

Testcase: Check the possibility of enabling AOP support for a non-C project.

This thesis' plugin is designed to support AOP for C only. Therefore it may only be possible to enable it for C projects. With the given Eclipse setup, it is possible to develop in the programming languages C, C++ and Java. New projects, one for C++ and one for Java, will be created. The popup menu for either of them does not show an AOP entry, so the AOP support can only be activated for C projects.

Conclusion: AOP support can not be enabled for non-C projects. This behaviour is correct.

Testcase: Execution of quickfix proposals for missing aspectcode in target files after first time initialization of the C project.

A file with markers will be opened in an editor, showing markers in every line that contains an AOP error. The errors occur in lines with function declarations that match an advice's pointcut and at the first and last line of the file. Each marker transports information about the problem. The marker in the first line shows that aspect header code is missing. A quickfix provides the solutions to either weave the file again or to check the consistency again. The first choice fixes not only this problem but also other problems in the file. It inserts all the aspect code that should be inside the file after cleaning it to remove obsolete aspect code. Thus it creates a consistent output, as the final consistency check for the file shows through the absence of AOP error markers. The alternative solution, to check the consistency again, will remove active AOP error markers from the file and check its consistency again. This quickfix needs to be provided as Eclipse requires the registration of jobs to run updates on project resources and the addition of error

markers to target files. A manager schedules the jobs based on their information of being a long running or quick procedure and the required resources. If an active job has access to a specific resource, another job that needs access to that resource will not be started until the first job finishes. This behaviour suppresses deadlocks but also alters the order in which the jobs will be executed. That directly influences the plugin's behaviour, because the update for a target and its consistency check are both jobs which can be executed in reverse order, creating markers for all those parts where an update for the aspect code is required, even though that update was executed afterwards. The Java development within Eclipse shows similar effects. When it is used in a quick way, warnings and error markers are not removed from an editor. Hence this problem can be seen as an Eclipse issue.
Conclusion: The available quickfixes offer applicable generic solutions.

Testcase: Update an aspect of a system that was in an inconsistent state.

At the moment the system is in a inconsistent state. One quickfix was executed for the file trees.c, hence this file is in a consistent state. The other files have not been woven and contain no aspectcode. The aspect test.ac and the target file tree.c will now be opened in an editor to test the effect of an update to an aspect. The advice *almostforgot* from listing 4.2 will be added to the aspect in listing 4.1 as the last advice. Now the aspect file contains the advices *greetings*, *goodbye* and *almostforgot*. The update should insert aspectcode into the target files, including trees.c, at the correct joinpoints. The correctness of the joinpoints and aspectcode will be checked in the correctness section 4.3.2. After an aspect update the system should be consistent and therefore not contain any AOP error markers. The consistency will be checked after every weaving operation. Saving the aspect now triggers the update of every target file, which results in a consistent system.
Conclusion: Works except for some consistency checks which were executed out of order and created wrong error markers.

Listing 4.2: An advice that will be added to the Test aspect.

```
1 advice almostforgot : before exec("% m%()") {
2   puts("A function starting with M was called");
3 }
```

Testcase: Update an aspect of a project that is in a consistent state.

The advice *almostforgot* from listing 4.2 will receive a modified pointcut and additional code in the advice block (see listing 4.3). Saving the aspect results in an update of all target files. Afterwards, as in the previous testcase, the system is in a consistent state but shows some wrong AOP error markers which can be removed through the quickfix "check consistency again".

Conclusion: As in the case of a system in an inconsistent state, the aspect update creates correct results except for some consistency checks which are executed out of order and therefore create wrong markers.

Listing 4.3: The modified Test aspect.

```
1 /* This file is modified by test.ac */
2
3 aspect Test {
4   advice greetings : before exec("% main()") {
5         puts("Greetings from test.ac");
6   }
7   advice goodbye : after exec("% main()") {
8         puts("test.ac says goodbye!");
9   }
10  advice almostforgot : before exec("% m%()" || "% n%()") {
11        puts("A function starting with M or N was called");
12  }
13 }
14
15 /* This file was modified by test.ac */
```

Testcase: Create a syntax error in an aspect and save the aspect to check the effect.

An erroneous aspect can be present in a software system, but it may not result in changes to the system that create inconsistencies. The aspect's modification will first be a change to the syntax of an advice declaration and afterwards to the aspect declaration. After saving the aspect with corrupt advice syntax, the system stays in its active state. The modification of the aspect does not result in a change of the system, the last correct version of the aspect is used instead. Changing parts of the aspect syntax shows the same effect. The intact consistency is positive, but based on another anticipation the wrong aspect could also lead to the removal of the aspect.

Conclusion: An aspect with syntax errors is not used by the weaver, therefore it shall not harm the system's consistency.

Testcase: Look at the effect of creating and deleting aspect and target files of an AOP enabled project.

Within an AOP enabled C project, a new aspect file is created. The creation of the aspect in the editor and the final saving of it trigger the weaver to insert the information from the aspect into the source code of the project. The creation of a C file, as well as the removal of an aspect file, are not yet supported operations. The weaver is not informed about created files or deleted files. A ResourceListener is needed, which tracks changes of the resources in the workspace and informs the weaver about them.

Conclusion: Resource changes, like creation and deletion of files of a project, are not observed. However, a newly created aspect affects the system and is woven.

Testcase: Modify the aspectcode of a target file in an editor and save it.

There are several different places in which a target file can include aspectcode. Besides header and footer code, hijacked functions and forward declarations for the renamed functions exist. First the code itself will be modified, and the AOP comments will be left unchanged. All the previously listed locations are modified slightly by removing one character each. After saving this target file, error markers pop up at the header, the forward declaration, the hijacked function and at the footer. In the first line, at the changed header, a marker stating that "the header was changed" appears. Its possible quickfixes are "weave again" and "check consistency again". At the hijacked function, two markers exist, one for the changed forward declarations and one for the changed advice block beneath. The quickfixes available to solve this problem are "weave again", "reset forward declaration", "reset this adviceblock" and "check consistency again". In the last line of the file a marker shows that the footer was changed. Quickfix resolution proposals are "weave again" and "check consistency again". The error markers appear correctly and give an idea about what the problem is.

Limits to code modifications exist within the AOP comments. The algorithm that removes obsolete code from a corrupt file, before it is a target for the weaving process to put it back into a consistent state, has, of course, specific conditions to be able to identify aspectcode as such. Only lines that include correct AOP comments from a specific set of patterns will be regarded as such and therefore removed. There are exceptions to the removal of a line, for example the hijacked function will not be removed. Instead the following comment and the lines up to the renamed function will be removed. Therefore changing the AOP comment to the point where it can not be identified by a regular expression anymore will lead to keeping that line which will exist "somewhere" in the target file, presumably breaking C syntax. AOP comments are not supposed to be altered at all. A violation of this rule can have unforeseeable consequences.

Conclusion: A modification of woven aspectcode can be detected, as long as the identifyable AOP comments exist. Markers are inserted showing the unexpected change to the aspectcode and providing quickfix possibilities.

Testcase: Create a new function that is a match to a pointcut from the test advice in a target file and save it.

Beginning with a consistent system, one target file will be changed in this test. A function that matches a test advice's pointcut will be added to the file. After saving it, an AOP error marker appears at the new function, informing about the need to weave because a new joinpoint was found. The generic quickfix to reset the woven code is available and can resolve the issue.

Conclusion: Target files are checked for changes after saving, the consistency check successfully detects new joinpoints.

Testcase: Disable AOP support for the project and check previous testcases if the AOP support is truely disabled.

Disabling AOP support by activating the menu entry from the popup menu of a project will remove all the AOP error markers from the project. In the case of a large project, this action takes some time since the resources can only be accessed using Eclipse's job manager. Afterwards the editing of aspect files or other project files does not result in any plugin activity. The entry in the project's popup menu returned to the initial state and shows "enable AOP support".

Conclusion: Disabling AOP support does remove the AOP features the plugin provides for a specific project.

Testcase: Re-enable AOP support for the project and check if the previous testcases behave correctly.

After the disabling of AOP support for a project, that project will now receive AOP support again. Two editors, one showing the test.ac file and the other presenting the contents of the trees.c file are open as the popup's menu entry "enable AOP support" for the specific project is selected. A notification about the initialization quickly appears and is removed a short time later. The saving of changes to the aspect file in the open editor correctly result in an update of the aspectcode of the C files. Now a new function will be inserted into the trees.c file, which is another joinpoint for the advice *almostforgot*. After saving the target file's editor, the newly created function receives an error marker that complains about the need to weave because a new joinpoint has appeared. The execution of the quickfix "weave again" correctly inserts the aspectcode computed from the *almostforgot* advice. The plugin returned to a fully functional state and successfully completed this testcase.

Conclusion: After being disabled for a project, the subsequent re-enabling results in correct behaviour of the plugin, exactly as occured after the first activation.

Testcase: Check if the aspect editor correctly highlights the syntax and also shows errors for syntax errors.

The editor is used to modify an aspect file. Keywords are highlighted correctly. The syntax check works for an aspect with a single advice, the support for further advice syntax highlighting needs to be implemented. An aspect with a single advice in correct syntax is therefore successfully checked. If, for example, the closing right brace is removed, the whole aspect is marked to contain a syntax error. Further code after the first advice is not checked.

Conclusion: Keywords are highlighted correctly, the syntax check works only for an aspect with a single advice.

4.3.2 Correctness

The impact of advices on the source code through the development time weaver need to be checked for their correctness. The result has to match the theory, create correct ANSI-C source code and show the advices supposed effect at runtime.

The advices need to be woven in a specific order. Section 3.2.3 explains the impact of different orders on the grouping of advices. The anticipated result will be based on this theoretical background. Advices have to be woven in the order of their appearance per joinpoint. The appearance is based on the aspects names, which are checked in lexicographic order, and the advice's order inside the aspect. Multiple before advices from the joinpoint will be grouped together with directly following after advices. If there are further advices, beginning with a before advice, these will be combined into a different group. Each group will be woven into one hijacked function. In this example, the advices for a joinpoint will be {before(1), after(2)} and {before(3)}. The groups will be {before(1), after(2)} and {before(3)}. The function *void example()* will be hijacked by the last group, the AOP comment will include the information about the source of the change to be advice before(3). The new function body will begin with the advice block of advice before(1), afterwards call the renamed function and at last include the advice block of [advice after(2)]. The second group will get its own function, which will be the renamed function of the first group. This function gets hijacked, with the AOP comment information that the change was created based on advice before(3). The body of this function will begin with the advice block of advice before(3) and finally the call to a renamed function which contains the original function body. An additional *do { .. } while(0)*; will surround the advice blocks. This block secures the functions namespace and therefore supports creation of generic advices.

gzip

The first aspect *ShowInfo* will consist of one before and one after advice, shown in listing 4.4. The advice *saygoodbye* inserts a message into the *do_exit()* function which will be run before gzip finishes. Another message is inserted through the advice *extendHelp* to the *help()* function which shows possible commandline options for gzip.

Listing 4.4: Example aspect ShowInfo

```
1 aspect HelpInfo {
2   advice saygoodbye : before exec("void do_exit()") {
3     fprintf(stdout, "Thank you for using gzip!");
4   }
5
```

```
6   advice extendHelp : after exec("void help()") {
7     fprintf(stdout, "FTI: gzip -d <file-to-decompress>");
8   }
9 }
```

The aspect ShowInfo is saved in the file "aspects/showinfo.ac" within the gzip project inside of Eclipse. After enabling AOP support for the gzip project, two AOP error markers pop up in the file "gzip.c" at the specified functions. After choosing the quickfix "weave again" for one of the errors, the advices are woven, resulting in the code shown in listing 4.5. As supposed, each advice creates a forward declaration for the renamed function it creates and calls from the hijacked function. The hijacked function and the advice blocks are marked through AOP comments. The added source code is correct ANSI-C. One problem that is visible in this example is the makro expansion that was not checked correctly and therefore it was not added to the renamed function. The forward declaration was even inserted after it, so it became "local" itself. The parameter *exitcode* for the original function is passed on to the renamed function. Furthermore the hijacked function does not have a return statement as it is defined as void. Therefore the call to the renamed function, which also has the return type void, does not get cached to be returned later.

Listing 4.5: Effect of the ShowInfo aspect on the file gzip.c.

```
1 [...]
2 local void help_HelpInfo_extendHelp();/*
    AC:HelpInfo:extendHelp.forward_declaration */
3 void help()
4 /* AC:HelpInfo:extendHelp.hijacked_function */
5 {/* AC:HelpInfo:extendHelp.block_begin */
6   help_HelpInfo_extendHelp();/* AC:HelpInfo:extendHelp.renamed_function_call */
7   do {/* AC:HelpInfo:extendHelp.adviceblock_begin */
8     fprintf(stdout, "FTI: gzip -d <file-to-decompress>");/* AC:HelpInfo:extendHelp
        */
9   } while(0);/* AC:HelpInfo:extendHelp.adviceblock_end */
10 }/* AC:HelpInfo:extendHelp.block_end */
11 void help_HelpInfo_extendHelp()/* AC:HelpInfo:extendHelp.renamed_function */
12 {
13     static char  *help_msg[] = {
14 #if O_BINARY
15  " -a --ascii       ascii text; convert end-of-lines using local conventions",
16 #endif
17  " -c --stdout      write on standard output, keep original files unchanged",
18
19 [...]
20
21  " file...          files to (de)compress. If none given, use standard input.",
22   0};
23     char **p = help_msg;
24
25     fprintf(stderr,"%s %s (%s)\n", progname, VERSION, REVDATE);
26     usage();
27     while (*p) fprintf(stderr, "%s\n", *p++);
28 }
```

```
29
30 [...]
31
32 local void do_exit_HelpInfo_saygoodbye(exitcode);/*
       AC: HelpInfo: saygoodbye. forward_declaration */
33 void do_exit(exitcode)
34     int exitcode;
35 /* AC: HelpInfo: saygoodbye. hijacked_function */
36 {/* AC: HelpInfo: saygoodbye. block_begin */
37   do {/* AC: HelpInfo: saygoodbye. adviceblock_begin */
38     fprintf(stdout, "Thank you for using gzip!");/* AC: HelpInfo: saygoodbye */
39   } while(0);/* AC: HelpInfo: saygoodbye. adviceblock_end */
40   do_exit_HelpInfo_saygoodbye(exitcode);/*
         AC: HelpInfo: saygoodbye. renamed_function_call */
41 }/* AC: HelpInfo: saygoodbye. block_end */
42 void do_exit_HelpInfo_saygoodbye(exitcode)/*
       AC: HelpInfo: saygoodbye. renamed_function */
43     int exitcode;/* AC: HelpInfo: saygoodbye. renamed_function */
44 {
45     static int in_exit = 0;
46
47     if (in_exit) exit(exitcode);
48     in_exit = 1;
49     if (env != NULL)  free(env),  env  = NULL;
50     if (args != NULL) free((char*)args), args = NULL;
51     FREE(inbuf);
52     FREE(outbuf);
53     FREE(d_buf);
54     FREE(window);
55 #ifndef MAXSEG_64K
56     FREE(tab_prefix);
57 #else
58     FREE(tab_prefix0);
59     FREE(tab_prefix1);
60 #endif
61     exit(exitcode);
62 }
63 [...]
```

The next example shows that C source code without makro expansions can be woven without problems and that return statements are generated and return values are cached based on the defined return type of the matched function. The aspects *ConnectToServer* and *Authenticate* from the listings 4.6 and 4.7 will be woven. Their expected impact will be discussed before the result is presented.

Listing 4.6: Aspect 1ConnectToServer for the example of grouping advices.

```
1 #include "serverconnection.h"
2
3 aspect 1ConnectToServerAspect {
4   advice connect : before exec("int in_prev_args()") {
5         do_connect();
6   }
7
8   advice disconnect : after exec("int in_prev_args()") {
9         do_disconnect();
10  }
11 }
```

Listing 4.7: Aspect 2Authenticate for the example of grouping advices.

```
1 #include "authentication.h"
2
3 aspect 2AuthenticateAspect {
4   advice login : before exec("int in_prev_args()") {
5         do_login();
6   }
7   advice sendinfo : before exec("int in_prev_args()") {
8         send_info("something happens");
9   }
10  advice logout : after exec("int in_prev_args()") {
11        do_logout();
12  }
13 }
```

All advices have the same pointcut, hence they will be listed at the joinpoint in their actual order: connect, disconnect, login, sendinfo, logout. They will be combined into the two groups connect, disconnect and login, sendinfo, logout. This preserves the correct execution order, which is: connect, login, sendinfo, execute original function, logout, disconnect. The function's returntype is int, so the returned values from calls to renamed function need to be returned at the end of the function's execution. Listing 4.8 shows the result of the weaving process.

Listing 4.8: More complex example with two aspects and five advices being woven.

```
1 #include "serverconnection.h" /* AC:1ConnectToServerAspect.header */
2 #include "authentication.h" /* AC:2AuthenticateAspect.header */
3 [...]
4 static int in_prev_args_1ConnectToServerAspect_connect(arg, argv, argc);/*
      AC:1ConnectToServerAspect:connect.forward_declaration */
5 static int in_prev_args_2AuthenticateAspect_login(arg, argv, argc);/*
      AC:2AuthenticateAspect:login.forward_declaration */
6 static int in_prev_args (arg, argv, argc)
7       char *arg, **argv;
8       int argc;
9 /* AC:2AuthenticateAspect:login.hijacked_function */
10 {/* AC:1ConnectToServerAspect:connect.block_begin */
11   do {/* AC:1ConnectToServerAspect:connect.adviceblock_begin */
12         do_connect();/* AC:1ConnectToServerAspect:connect */
```

```
13   } while(0);/* AC:1 ConnectToServerAspect:connect.adviceblock_end */
14   int tmp = in_prev_args_1ConnectToServerAspect_connect(arg, argv, argc);/*
         AC:1 ConnectToServerAspect:connect.renamed_function_call */
15   do {/* AC:1 ConnectToServerAspect:disconnect.adviceblock_begin */
16         do_disconnect();/* AC:1 ConnectToServerAspect:disconnect */
17   } while(0);/* AC:1 ConnectToServerAspect:disconnect.adviceblock_end */
18   return tmp;/* AC:1 ConnectToServerAspect:connect.return_original */
19 }/* AC:1 ConnectToServerAspect:connect.block_end */
20 static int in_prev_args_1ConnectToServerAspect_connect(arg, argv, argc)/*
       AC:1 ConnectToServerAspect:connect.renamed_function */
21       char *arg, **argv;/* AC:1 ConnectToServerAspect:connect.renamed_function */
22       int argc;/* AC:1 ConnectToServerAspect:connect.renamed_function */
23 {/* AC:2 AuthenticateAspect:login.block_begin */
24   do {/* AC:2 AuthenticateAspect:login.adviceblock_begin */
25         do_login();/* AC:2 AuthenticateAspect:login */
26   } while(0);/* AC:2 AuthenticateAspect:login.adviceblock_end */
27   do {/* AC:2 AuthenticateAspect:sendinfo.adviceblock_begin */
28         send_info("something happens");/* AC:2 AuthenticateAspect:sendinfo */
29   } while(0);/* AC:2 AuthenticateAspect:sendinfo.adviceblock_end */
30   int tmp = in_prev_args_2AuthenticateAspect_login(arg, argv, argc);/*
         AC:2 AuthenticateAspect:login.renamed_function_call */
31   do {/* AC:2 AuthenticateAspect:logout.adviceblock_begin */
32         do_logout();/* AC:2 AuthenticateAspect:logout */
33   } while(0);/* AC:2 AuthenticateAspect:logout.adviceblock_end */
34   return tmp;/* AC:2 AuthenticateAspect:login.return_original */
35 }/* AC:2 AuthenticateAspect:login.block_end */
36 static int in_prev_args_2AuthenticateAspect_login(arg, argv, argc)/*
       AC:2 AuthenticateAspect:login.renamed_function */
37       char *arg, **argv;/* AC:2 AuthenticateAspect:login.renamed_function */
38       int argc;/* AC:2 AuthenticateAspect:login.renamed_function */
39 {
40   int i, is_in_args;
41
42   is_in_args = 0;
43   for (i = 1; i < argc - 1; i++)
44     if (stricmp (arg, argv[i]) == 0)
45       is_in_args = 1;
46   return is_in_args;
47 }
```

As discussed, the groups of advices were woven into two separate functions. The hijacked original function contains the advices from the *1ConnectToServer* aspect, the renamed function *foo_2Authenticate_login()* includes the advices from the *2Authenticate* aspect. Besides forward declarations and renamed or hijacked functions, the AOP comments are new parts a developer needs to get familiar with. The AOP comments inform about the source of the changes and help to understand the changes to the source code.

The "static" declaration was identified and added to all created function definitions. The int return type was also identified, therefore the return value was cached and carried on as the hijacked function's own return value.

4.3.3 Performance

The init time measuring is performed using a modified implementation to execute the weaver initialization method 100 times. This method will usually be executed once in an Eclipse lifecycle, either at startup for projects which obtained AOP support in an earlier Eclipse instance or at runtime to enable the AOP support for a project. Before intialization one aspect with ten advices is created for testing purposes, consisting of five before and five after advices. Potential joinpoints within the target files are counted. The high amount of iteration is chosen so that the influence of concurrent operations on the execution time is minimized. A single execution will therefore probably not exactly match the result of this testrun.

In the same Eclipse instance which was used for this initialization test, the duration for checking the consistency and weaving an update were tested. A consistency check is triggered after every update of a target file, either through a weaver that inserts updated aspectcode or through the developer that changed the file's contents. A single target file receives an update if the developer chooses the corresponding quickfix to solve an AOP inconsistency or if an aspect file is saved. Again, those methods were executed 100 times for every target file, so the result is an average of those 100 runs. Several diagrams visualize the outcome.

The last performance test measures the average time the system requires to update all target files. This function is executed every time an aspect file is saved. The difference to the previous test is that this time the execution time of the main function, which iterates over all the target files, is measured. Like the other tests, this one is also executed 100 times and the result is the average execution time of those runs.

The data that is visualized in the resulting diagrams can be found in the appendix, section 6.

gzip

The following data is the result of the initialization test:

- Target files: 22
- Potential joinpoints: 300
- Advices: 10
- Average duration for the initialization: 158.051ms

For this small project an average init time of about .16 seconds is good, showing that the plugin will start almost unnoticed. If the next tests are as good, the plugin is ready to be used on small projects.

The relation between the amount of joinpoints in a file and the duration of a consistency check execution is visible in figure 4.1. The values for the duration were calculated as the average of 100 consistency checks. The variance is influenced by the size of the files, which is another factor in the performance for a single file. The biggest variance at point (7, 11.335) results from the checked file also being the second largest file. In comparison, the largest file has also got the biggest amount of potential joinpoints but a smaller variance, as its point (22, 21.259) shows. As a result of this diagram, the relation between the amount of joinpoints and the consistency check execution time is linear.

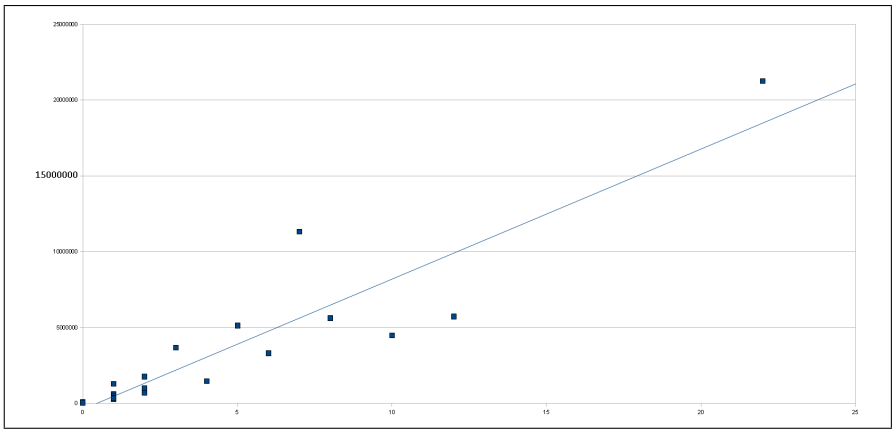

Figure 4.1: This diagram shows the relation between the absolute amount of potential joinpoints in a file on the X axis and the average duration for the consistency check of the file on the Y axis in nanoseconds.

Figure 4.2 shows the duration of a consistency check for different filesizes. The duration is the average of 100 executions. The set of data is also the base for figure 4.1. The variance is influenced by the amount of joinpoints that a source file provides. The duration for a single file from this project is definitely below 25 milliseconds, most files are even checked within less than 7 milliseconds, which is surely quick enough to be executed several times during development.

Besides the duration to execute consistency checks on the project's files, the time required to update the system through the changing of an aspect is measured. Figure 4.3 shows the relation between the amount of potential joinpoints in a target file to the duration of the execution of the update. The linearity is reminiscent of the diagram in figure 4.1, the main differences being the higher absolute execution time and the different variance. The update of the file with the biggest amount of joinpoints takes 33.961 milliseconds, which is much longer compared to the duration of a consistency check. Most of the other files can be updated within 10 milliseconds, which is a good value for a function which is executed after every saving of an aspect file. Overall, the execution of this

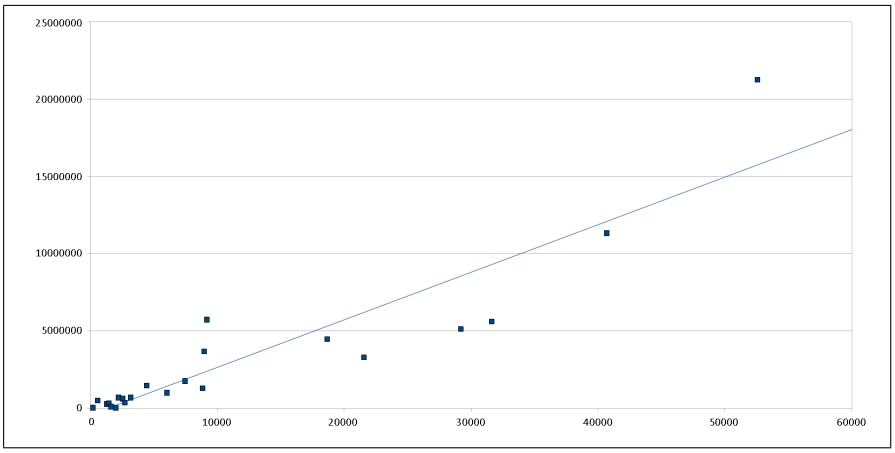

Figure 4.2: The average consistency check duration on the Y axis in nanoseconds is visualized in relation to the filesize on the X axis.

method shows a linear relation between the amount of potential joinpoints in a file and the duration of an execution.

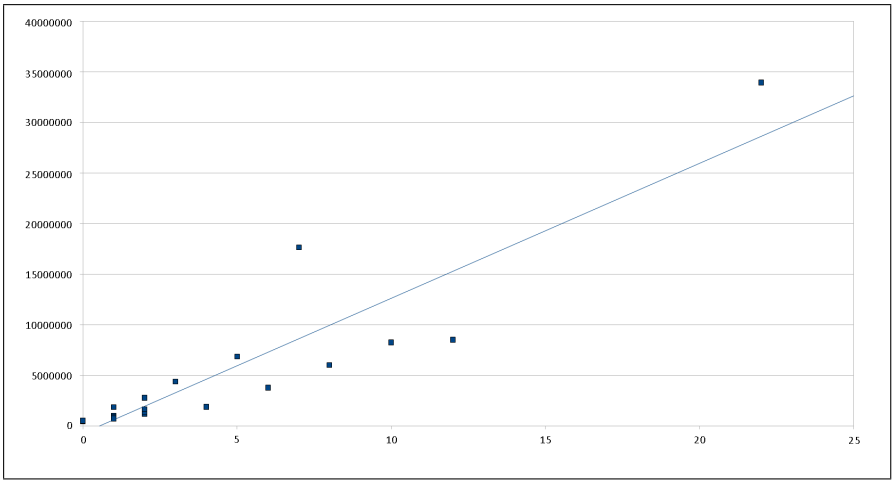

Figure 4.3: An aspect was changed, resulting in the update of all target files. This diagram shows the relation between the amount of potential joinpoints in a target file on the X axis and the average update duration on the Y axis in nanoseconds.

As in the consistency check duration test, the time to update a target file is related to the filesize as shown in figure 4.4. The variance of several points is comparable to the variance of figure 4.2, as a further factor for the duration is the amount of potential joinpoints of a target file. The durations are exactly those shown in figure 4.3, this permitting the conclusion that the operation is fast enough for the relatively small files that were tested.

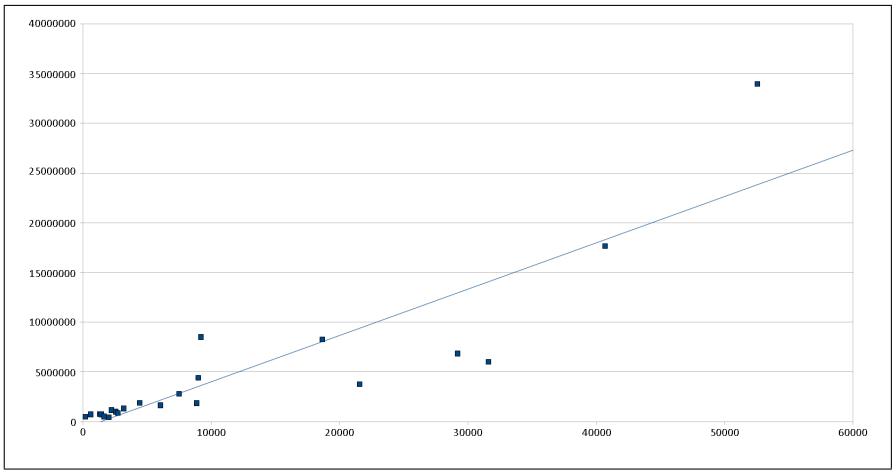

Figure 4.4: The update for all target files is executed, this diagram shows the relation between the filesize on the X axis and the average update duration on the Y axis in nanoseconds.

The following data shows the average duration of an aspect update, which includes unweaving, computation of joinpoints from the potential joinpoints and a final weaving for all target files:

- Target files: 22
- Potential joinpoints: 300
- Advices: 10
- Average duration for a complete update: 105.947ms

The complete update of the system requires only about .1 seconds. With an init time of .16 seconds and durations for updates and consistency checks for single files of mostly beneath 10 milliseconds, it shows that the performance of the plugin is good enough to be used for small projects. Therefore the performance of the plugin in bigger projects will now be tested in the next section using the Linux kernel project.

Linux kernel

The init time for this thesis plugin, performed on the linux kernel sources in version 2.2.26 was not as quick as for the gzip project. The kernel sources in this version require 90.3MB of space on the harddrive, including a total of 5913 files. The test aspect has five before and five after advices, it is exactly the same aspect as in the previous test.

- Target files: 2409
- Potential joinpoints: 125450

- Advices: 10
- Average duration for initialization: 33.587 s

Due to the amount of target files and potential joinpoints, half a minute is an acceptable duration. In comparison, the CDT initialization, which creates the abstract syntax tree and checks includes by parsing all project files, takes about half a minute on the test machine too[5]. As this method only needs to be called once per Eclipse lifecycle, it is fast enough to be used.

The duration of a consistency check for a target file in relation to the amount of potential joinpoints is shown in figure 4.5. It shows an almost linear relation, with a slightly higher increase and variance for files with over 50 potential joinpoints. Like for the high durations for files with very few joinpoints, next to the Y axis, the variance results from very large files. Figure 4.6 shows a zoom in on the cloud. The 20 largest files of the project are not shown in this diagram. This shows some files with a lot longer and shorter execution times. Regarding the fact that this diagram visualizes 2389 points, those few exceptions have no immense impact on the result that the consistency check duration is almost linear relative to the amount of potential joinpoints of a file. This finding supports the result of the performance tests of the gzip project. As most files are checked within 30 milliseconds. This per file performance confirms the possibility of using this plugin at runtime as the consistency check happens almost unnoticed.

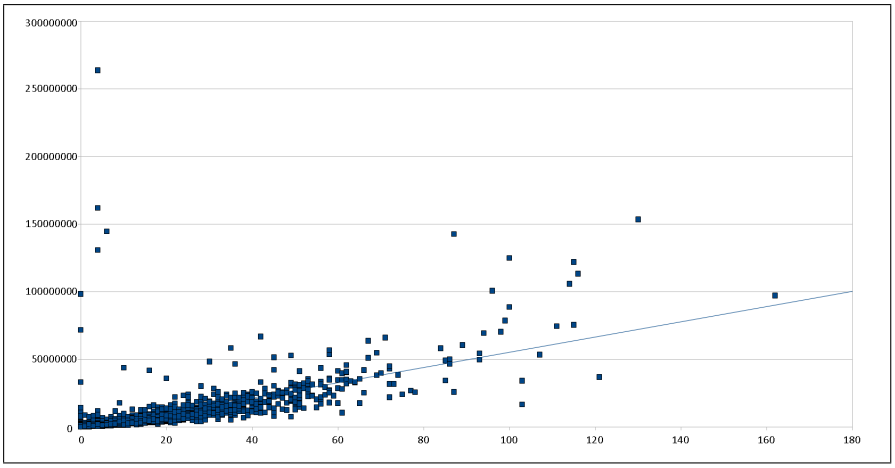

Figure 4.5: Relation of the amount of potential joinpoints in a target file on the X axis to the corresponding consistency check duration on the Y axis in nanoseconds.

It can be clearly seen from figure 4.7 that large files require longer consistency checks. The linear relation between filesize and duration of consistency checks is also visible in the zoomed version in figure 4.8, where the 20 largest files are removed to get a better view

[5]Estimated CDT startup time.

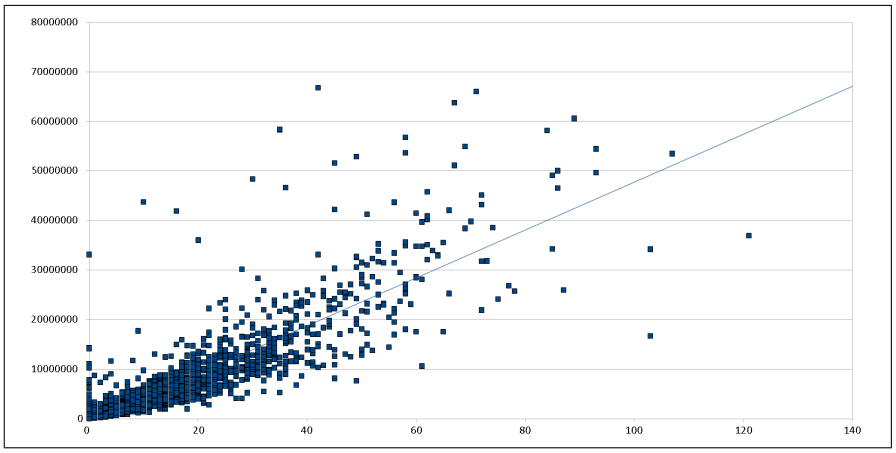

Figure 4.6: Relation of potential joinpoints on the X axis to consistency check duration on the Y axis in nanoseconds without the 20 largest files.

of the cloud. The existing variance results from different amounts of potential joinpoints of a target file. A few files are checked in very little time, right next to the X axis, because they have very few or no potential joinpoints at all. The small files which require longer consistency checks than other files of their size have bigger amounts of potential joinpoints. Most of the 2409 points visible in figure 4.7 are located next to the linear regression, showing that the consistency check time is linear related to the size.

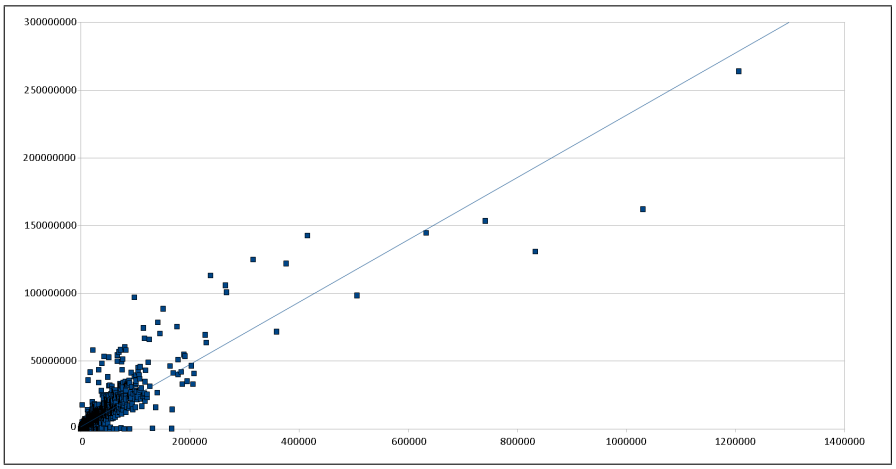

Figure 4.7: Average duration of a consistency check on the Y axis in nanoseconds in relation to the filesize on the X axis.

Besides the duration of consistency checks, the time to update a target file in relation to its filesize and the amount of potential joinpoints is measured. The results in figure 4.9 show that the update time is linear in relation to the amount of joinpoints, as the gzip project's results also indicated.

The performance of the duration of an update in relation to the filesize is visible in figure 4.10. In figure 4.11 the 20 largest files are excluded to zoom in on the cloud. It

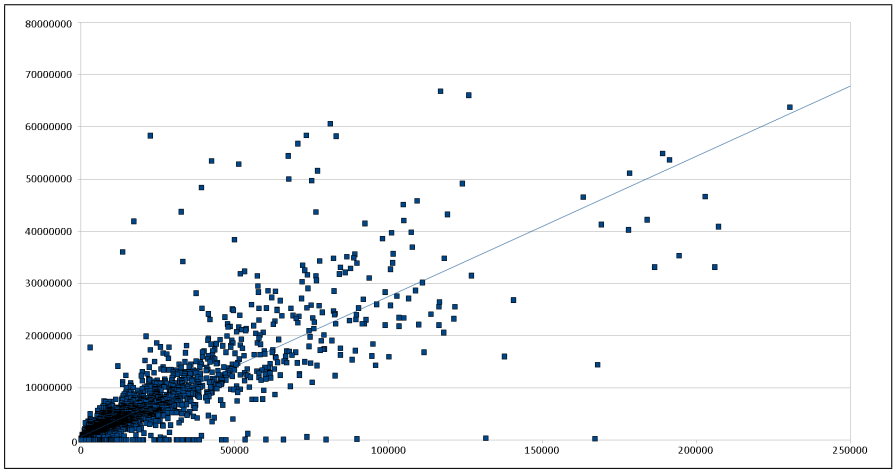

Figure 4.8: Average consistency check duration on the Y axis in nanoseconds in relation to the filesize on the X axis excluding the 20 largest files.

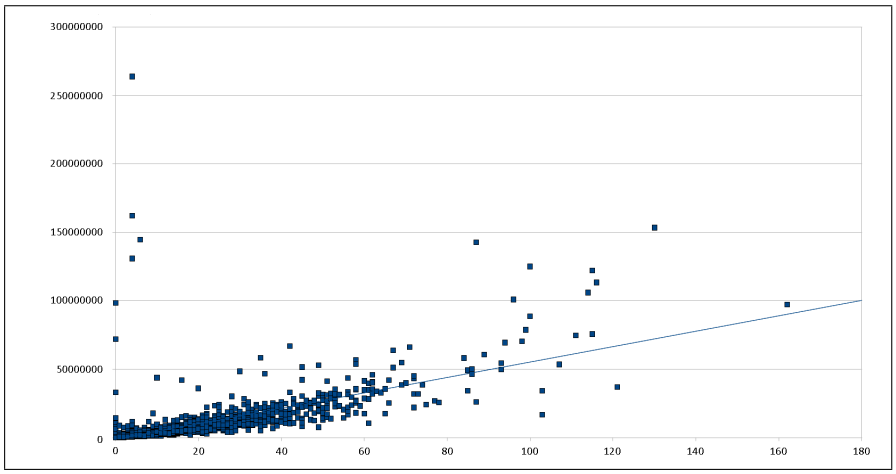

Figure 4.9: Relation of amount of potential joinpoints on the X axis to the update duration for the weaving on the Y axis in nanoseconds.

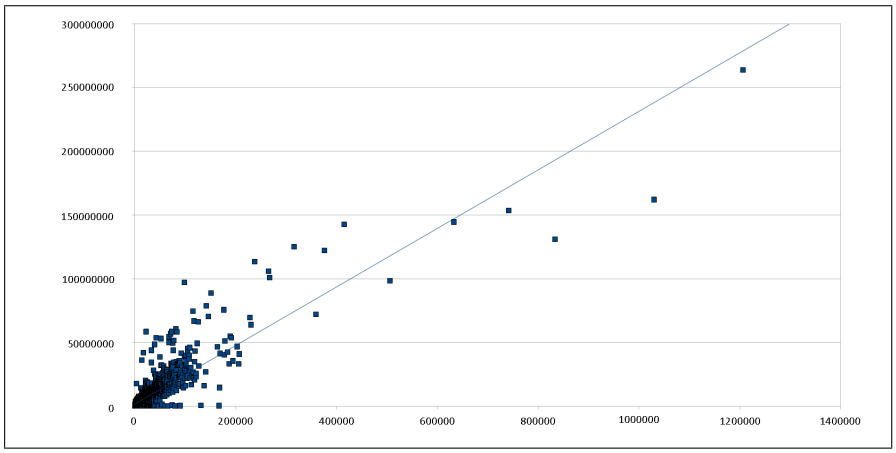

Figure 4.10: Duration of an update for every single target file on the Y axis in nanoseconds in relation to the filesize.

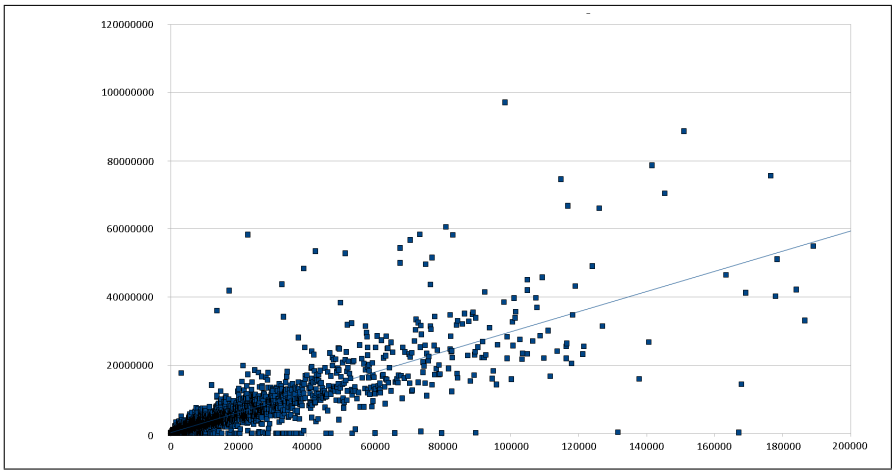

Figure 4.11: Update of every target file, duration on the Y axis in nanoseconds in relation to the filesize on the X axis, excluding the 20 largest target files.

shows the linear relation of the two dimensions, and the fact that most of the files are updated within only 20 milliseconds.

The duration of an update for all target files returns the following results:

- Target files: 2409
- Potential joinpoints: 125450
- Advices: 10
- Average duration to update all target files: 46.368s

Since it takes 46 seconds to finish, an update will definitely pause the development. This shows that a performance optimization is necessary to use the plugin for such immense projects. The execution of the consistency check and the updating of a single file finishes very quickly, within 30 milliseconds as seen in figure 4.11. For applications using AOP it may not be necessary to insert generic advices that have joinpoints in a lot of different files. Implementing the *filefilter* pointcut expression and using it for the aspects could resolve this issue.

4.4 Result

Overall, the three different tests demonstrate that the implemented basic functionality performs well. The tests cover the usage of the plugin, the correctness of the woven code and measuring the performance of crucial functions.

The testcases from the usage section 4.3.1 range from activating AOP support for a specific project over the effect of modifying an aspect file and a target file, to the availability and execution of quickfixes. They show that required front-end functionality works correctly, only minor drawbacks can be found. These drawbacks can be solved quite easy, requiring only some further development without the need to change the design. Solution approaches for the identified deficits will be presented in section 6. Testcases in which the weaver is activated have positive conclusions. Hence, further development of the plugin core is possible without the need to access and modify the implemented functionality that extends Eclipse itself.

The second test, presented in section 4.3.2, checks the weaving result. The tasks are correct weaving of single *after* and *before advices* with different pointcut expressions, and a complex situation with two aspects and five advices, all with the same pointcut. The single advices were inserted correctly besides the issue of a makro expansion that was not discovered and therefore ignored. The complex testcase results in correct ANSI-C source code. The return type of the function at the joinpoint is identified as well as the *static* modifier for the function. The modifier is added to the generated functions, and a variable corresponding to the return type is passed on through the hijacked and renamed functions. Those functions are generated exactly as theory requires, preserving the logical execution order. Furthermore, function parameters are also passed on to the renamed function containing the original function's code block. These testcases show that ANSI-C source code without makro expansions can be woven already, but a modification to support makro expansions will surely be required. A solution is to investigate the CDT to get the required information on this issue as described in section 6.

The third and final test analyses the performance of the plugin, introduced in section 4.3.3. Important functions which are executed often are chosen to be checked for their runtime relative to specific context information. This test is executed for two different projects to gain information about runtimes for small and large projects. In conclusion, the runtimes of the consistency check and of the update for a single file are linear related to a combination of the file's size and its number of potential joinpoints. All the measured runtimes of the functions for the *gzip* project are very low, initialization and update of the whole system needed about .1 seconds, concluding that the plugin can be used for small projects already. The *linux kernel* source code is used to test the plugin on a large software system. For both projects the performance is tested. The absolute runtime for the consistency check of a single file is below 30 milliseconds in the majority of cases, the

update of a single file even finishes within 20 milliseconds for most files. The update of the whole *linux kernel* takes about 46 seconds. This is hardly acceptable to be executed at development time, as it definitively pauses the work. The implementation of further caching methods, a specialized update method and the implementation of further pointcut expression are solutions to reduce this number immensely. Detailed information is available in section 6. The initialisation of the system requires about 33 seconds. For this crucial process, which has to be executed once per Eclipse lifecycle, the performance is acceptable. With some performance optimizations for the plugin, and furthermore the use of up-to-date hardware[6], it will be possible to use it for large projects. As intended, the foundations to reach this target are implemented.

The tests show that the plugin does fulfill the initial requirements. The goal of visualising the control flow of AOP in a C project is accomplished. Moreover, the performance results show that with minimal further effort it will be possible to use the plugin for large projects, which can be found in the industry. The identified issues can be solved within the existing design, as those are not resulting from design-flaws. The intention of this thesis, to create the foundations of an industry-capable AOP solution for C, is therefore achieved.

[6]The hardware used in the test was an AMD Athlon 64 X2 Dual Core Processor 3800+ with 6GB RAM, up-to-date in 2006.

Chapter 5

Summary & Conclusion

AOP is a programming paradigm in which concerns from the design of a software system can be implemented as single entities. This focuses the view on specific functionality, as it is encapsulated in a single object. The need to change many different files to implement this crosscutting concern, as it would be necessary in OOP, is dramatically reduced. Another advantage is the possibility to configure software statically. With the possibility to add or remove aspects from a software system, these can be woven and so extend the functionality as it is needed. Alternatively, for example, the software could receive configuration settings based on startup parameters, along with further code that checks the parameters subsequently. A statically configured system has therefore the advantage of fitting exactly the requirements, minimizing its size and creating a better predictable runtime. The weaving of existing AOP system happens at compiletime or at runtime, so the aspects are inserted after the development has finished. Their effect shows up initially at execution time. The problem of this procedure is that the control flow becomes invisible, as the additional aspectcode, that gets executed, is inserted at a time at which a developer starts debugging. If the developer's implementation contains a small mistake, it can result in system dysfunction instead of a compiletime error. The developer has to know about the complete set of aspects and the whole base system all the time to be able to know the result of his ongoing work. For Example, an advice's pointcut expression could match a newly created function or the renaming of a function results in the absence of the match of a desired advice's pointcut expression. These problems will show effect at runtime, being a challenge to debug. As a large number of developers are working on a software system, each of them would have to know about all the existing parts to be able to modify existing behaviour or to create new functionality correctly. This error-prone development suppresses the use of AOP in the industry. To be able to use AOP, the control flow has to become visible, always informing the developer about the advices which extend the actually regarded code.

IDEs are a standard tool to improve development efficiency as well as the quality of the produced code. With a large number of supporting features like syntax highlighting, error discovery, navigateable source code, outline view and templates which insert often used code pieces, the development of large software projects without an IDE is almost unimaginable. IDEs exist for a vast amount of programming languages, of course including C, C++ and Java.

By integrating the visualisation of advices from AOP in an IDE, the control flow can be visualised exactly where the developer is working, inside the IDE. Existing related solutions show, for example, meta information about the existance of advices which are added to a specific location. This is insufficient, because it requires further interaction from the developer to get all the information about the code that will be woven. The solution is, that weaving itself has to be executed at that time, when developing happens, at development time. Aspects will be saved in special aspect files. This preserves the advantage of a single entity that contains the implementation. Together with development time weaving, this directly implies the need for duplicate datastorage, and the need to be able to identify the advices which are source of the woven code. The consistency of the duplicate datastorage needs to be defined for this new concept. Changes to C source files can happen outside of the IDE, suppressing triggers of the development time weaver. This results in missing updates and the system is left in an inconsistent state. Therefore it is necessary to check the consistency at the IDE's startup time. An AOP language has to be developed to support the possibility to identify the aspectcode to track it back to its source. This feature was not found in any other AOP language, and along with other requirements a new AOP language for C is developed, described in section 2.2. A parser to get a model from the aspects in the aspect files is created. Furthermore, additions to the IDE to trigger automatic updates at specific times are discussed, based on the IDE's possibilities. The IDE of choice is Eclipse, used in previous LS12 projects.

With the gathered knowledge about the development platform's possibilities, the design for a plugin implementation is created. A core is required by Eclipse, which is registered to be started at Eclipse's startup time instead of using the lazy loading mechanism, because its functionality is required from the start. Every project needs to be checked if it has AOP enabled, as this fact is saved for every project, and the project will then be initialised, loading all aspects and target files into internal data structures. Afterwards the consistency of that project needs to be checked, a function that is implemented in the weaver. Every AOP enabled project receives a single instance of the weaver, which of course also features updating the system after the modification of an aspect file. The changes to an aspect file can possibly have effect on every file of the software system. Hence, the whole system has to be checked for the need of updates after saving an aspect file. The editor in which an aspect file is modified receives a listener that informs the weaver about any change, and so it triggers the update. C source files can also be changed and, for example, receive new

joinpoints. This needs to be tested by the consistency check. The result of a consistency check are markers which indicate AOP problems to the developer. Those markers receive resolution proposals, called quickfix in Eclipse, which can be used to programmatically solve the AOP error.

All of the implemented parts need to be tested if they work as supposed. The usage was tested to see if the possible interactions from the developer result in correct behaviour or if problems arise. The correctness of the woven code is also tested, showing that it works for complex use cases, as long as makro expansions are avoided. The plugin's performance was tested in a large test with two projects, which resulted in a linear relation between consistency check duration to the file size and the amount of potential joinpoints, with the result of less than 30 milliseconds per file for the majority of files. Similar, the relation of the runtime of an update execution to the file size and the amount of potential joinpoints was discovered to be less than 20 milliseconds per file for most files. The init time, the duration of the first time the plugin is started for a project, an execution that is required once per Eclipse lifecycle, was also measured and shows acceptable results. For the Linux kernel, which is about 100MB large and contains 2410 C source files, the initialisation for ten advices took 33 seconds. The update for the whole system took 46 seconds. This is too slow for a method that needs to be executed relatively often, as it halts development for a moment. The performance can be optimized with further work like implementing more pointcut expression functionality to filter the files which should be target to the advice, or by caching a lot of subsequent function calls or unchanged aspect pointcuts. For small projects the plugin is already usable, as shown for the gzip project, except where makro expansions are used to alter function modifiers.

Conclusion

As the test results already show, some more development is required to have a applicable solution that can be used for AOP development of large software systems. The foundations are so far performing well, with linear relations between frequently executed functions to the file size and the amount of potential joinpoints. The output visualizes the control flow and it also shows the source for the visible sourcecode. The changes to the sourcecode can be minimized, for example by inserting before advices directly into the target function, or by bundling more advices from one joinpoint to generate the minimal set of necessary new functions. This would also improve the readability even more. The new approach of development time weaving is successfully accomplished in this thesis. Besides theoretical background information, the DETACH plugin shows basic functionality, supporting this result. Several interfaces, to be extended in later work, are described and implemented. The checking of external changes to the system's source code is created. The plugin core implements necessary functionality to track changes at development time and react accordingly. All the additional functionality, mentioned earlier, is being discussed in section 6, and can be inserted into this foundation.

The present thesis researched techniques and implemented a plugin for Eclipse that represents the foundations of an industry-capable AOP solution for C.

Chapter 6

Perspective & Further information

The development of this plugin is not finished with the actual version. The following section will introduce several possibilities to extend it.

The complete SimpleAspectC language should be supported, therefore the aspect parser needs to be updated, as well as the parser from the AspectEditor. The pointcut expressions need more classes and interfaces to accomplish the goal, the matching algorithm will need modifications as well. The weaver will need to be able to react to the different new IJoinPoint subclasses. The context information has to be made available through replacing special strings from every adviceblock. Special variables to get handles for the function's parameters also need to be implemented with sufficient access from C.

A listener is required to react to other Eclipse functions like renaming files, adding files and deleting files to keep the internal structural representation correct. These changes can be tracked through a ResourceListener.

To improve the performance, a map with potential joinpoints could be created. This map would only need to be updated after changes to a target file, so the parsing of target files is minimized. The downside is that probably the amount of RAM required to handle such a map for a large project may be very high, but this is surely the best possible optimization. Furthermore, the weaver has to check exactly if an update is an required by comparing the active joinpoint's advices with the computed joinpoint's advices.

The woven code, more exactly the amount of newly created functions and their forward declarations, could be minimized. Based on the theory about the correct execution order, before aspects could be added in order, and after aspects would need to be added in reversed order to secure the correct execution order. Around advices will still need new functions.

A meta view of aspects that match target files and therefore alter them could be created using the AJDT visualiser (from http://www.eclipse.org/ajdt/visualiser). This was used for the AspectJ Development Tools to create such a view, and it does provide a good overview. All the required information to be visualized in such a meta view is already available from the target handle, which has a list of active joinpoints and each them have a list of advices.

If the C parser can be extracted from the CDT, it would be possible to create a commandline tool based on the Weaver. This class contains all the weaving functions without the connections to Eclipse, which are necessary at the moment. Therefore the EclipseLiveWeaver was implemented, to use the weaver within Eclipse. The CDT AST usage is implemented within the EclipseIFileTarget's method getAST(), accessing the building of the AST directly, which is only possible through the plugin's dependency on CDT at the moment. With a separate AST creation unit, a standalone commandline version could be created quickly, as all other basic classes implement Eclipse independent functionality, and special classes for the use with Eclipse extend those basic classes.

The possibility to configure the aspects that should be woven into the system might be useful. Furthermore, by extending the pure::variants Eclipse plugin to control the weaving of the aspects, it would be possible to create support for product lines.

Further information

The following sections include information for different possible extension ideas as well as ideas with solutions. It is directed at developers to provide information to start with development right away.

The ILanguageSpecifics interface defines all the information the Weaver needs to weave. All the necessary C syntax to generate the woven code is returned from the defined functions, implemented in the class CLanguage.

The removal of all aspect code, identified by AOP comments, is implemented within the class EclipseIFileTarget. It is called remOrHideAspectcode, it can be used to remove the parts completely or fill them with whitespaces to preserve the index of the original source code, extended by aspects.

The development and tests were done using a ramdisk. The used software is called RAMDisk by Dataram (http://www.dataram.com) for Windows 7. This ramdisk soft-

ware is free to use and provides the possibility to create one ramdisk with up to 4GB space. Furthermore, a disk image to initialize the ramdisk can be chosen, and this can also be set as the target to save the contents at shutdown.

The issue of makro expansions in function definitions can be solved by further investigation of CDT's AST and CDT's indexing functions as these should provide access to the information about makro expansions, because they are discovered correctly in the C editor provided by CDT. A C source file with makro expansions receives several markers, showing that the C syntax is not correct where makro expansion is used, but these markers are removed quickly, and the makro then contain information about being a makro expansion.

List of Figures

Bibliography

[acd] Eclipse AspectC/C++ Development Tools. http://acdt.aspectc.org. Checked at 2010-09-19.

[ajd] http://www.eclipse.org/ajdt. http://www.eclipse.org/ajdt. Checked at 2010-08-2.

[aspa] Aspicere, more AOSD for C. http://sailhome.cs.queensu.ca/ bram/aspicere/. Checked at 2010-07-18.

[aspb] The Home of AspectC++. http://www.aspectc.org/. Checked at 2010-08-12.

[axd] The XWeaver Project. http://www.xweaver.org. Checked at 2010-07-10.

[bpx] XWeaver Overview. http://www.xweaver.org/XWeaver/Overview.html. Checked at 2010-07-10.

[c4m06] *Making Extensibility of System Software Practical with the C4 Toolkit*, 2006.

[cdt] Eclipse CDT (C/C++ Development Tooling). http://www.eclipse.org/cdt/. Checked at 2010-08-6.

[CE00] Krysztof Czarnecki and Ulrich Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley Professional, June 2000.

[CR08] Eric Clayberg and Dan Rubel. *eclipse Plug-ins (3rd Edition)*. Addison-Wesley, December 2008.

[ecla] [Checked at 2010-09-2.

[eclb] Eclipse documentation. http://help.eclipse.org/helios/index.jsp?topic=/org.eclipse.platform.doc.isv/reference/extension-points/index.html. Checked at 2010-08-12.

[eclc] Eclipse.org. http://eclipse.org. Checked at 2010-08-2.

[Gmb91] R. BOSCH GmbH. *CAN Specification Version 2.0*. ROBERT BOSCH GmbH, September 1991.

[gzi] The gzip homepage. http://www.gzip.org. Checked at 2010-09-13.

[jlg] Projects by Jean-loup Gailly. http://gailly.net/. Checked at 2010-09-13.

[lin] Linux Kernel. http://de.wikipedia.org/wiki/Linux_(Kernel). Checked at 2010-09-14.

[llv] The LLVM Compiler Infrastructure. http://llvm.org. Checked at 2010-07-18.

[mad] Mark Adler. http://en.wikipedia.org/wiki/Mark_Adler. Checked at 2010-09-13.

[MB05] Bill McCloskey and Eric Brewer. Astec: a new approach to refactoring c. *SIGSOFT Softw. Eng. Notes*, 30(5):21–30, 2005.

[Nvvd07] István Nagy, Remco Engelen van, and Durk Ploeg van der. An overview of mirjam and weavec. In Remco Engelen van and Jeroen Voeten, editors, *Ideals: evolvability of software-intensive high-tech systems*, pages 69–86. Embedded Systems Institute, Eindhoven, 2007.

[RFG] Alison Reynolds, Marc E. Fiuczynski, and Robert Grimm. On the feasibility of an AOSD approach to Linux kernel extensions. Princeton University & New York University.

[stu] Software Tools User Group. http://www.usenix.org/about/stug.html. Checked at 2010-09-13.

[Vog] Lars Vogel. Eclipse Plugin Development tutorial. http://www.vogella.de/articles/EclipsePlugIn/article.html. Checked at 2010-08-23.

Appendix A

gzip diagram data

file	length	amount of function declarations	amount of joinpoints	average consistency check time [ns]	average update duration [ns]
"/gzip/unzip.c"	6024	2	0	1009515	1641928
"/gzip/unlzh.c"	9203	12	7	5728716	8513300
"/gzip/bits.c"	7480	2	1	1761644	2789718
"/gzip/amiga/tailor.c"	4439	4	0	1463120	1895648
"/gzip/deflate.c"	29179	5	1	5131781	6849966
"/gzip/sample/add.c"	1444	1	0	334418	733462
"/gzip/sample/sub.c"	2721	1	0	356173	860771
"/gzip/unpack.c"	8987	3	2	3670183	4396331
"/gzip/vms/vms.c"	2233	2	0	696161	1169275
"/gzip/crypt.c"	189	0	0	43090	505187
"/gzip/lzw.c"	588	1	0	502516	727918
"/gzip/unlzw.c"	8840	1	0	1300741	1863496
"/gzip/trees.c"	40691	7	4	11334997	17651799
"/gzip/util.c"	18664	10	5	4474913	8254542
"/gzip/inflate.c"	31613	8	0	5620339	6030748
"/gzip/zip.c"	3170	2	0	699299	1331074
"/gzip/sample/makecrc.c"	2569	1	0	619618	995446
"/gzip/sample/zread.c"	1314	1	0	282398	758726
"/gzip/getopt.c"	21538	6	0	3302054	3772836
"/gzip/primos/primos.c"	2027	0	0	41994	425131
"/gzip/gzip.c"	52560	22	10	21259245	33960807
"/gzip/msdos/tailor.c"	1635	0	0	99778	527339

Linux kernel diagram data

file	length	amount of function declarations	amount of joinpoints	average consistency check time [ns]	average update duration [ns]
"/kernel/arch/alpha/boot/bootp.c"	5523	5	4	1956884	6429523
"/kernel/arch/alpha/boot/main.c"	4358	7	3	1780605	3257185
"/kernel/arch/alpha/boot/tools/mkbb.c"	3536	1	0	567968	1105531
"/kernel/arch/alpha/boot/tools/objstrip.c"	6097	2	0	1048841	1699930
"/kernel/arch/alpha/kernel/alpha_ksyms.c"	5495	0	0	4722530	8017116
"/kernel/arch/alpha/kernel/bios32.c"	39191	4	1	801641	3676743
"/kernel/arch/alpha/kernel/check_asm.c"	1140	1	0	258204	748044
"/kernel/arch/alpha/kernel/core_apecs.c"	14451	12	1	3541656	5287566
"/kernel/arch/alpha/kernel/core_cia.c"	17523	11	1	4189994	5934263
"/kernel/arch/alpha/kernel/core_irongate.c"	9163	9	0	1574617	2359054
"/kernel/arch/alpha/kernel/core_lca.c"	15769	16	3	4186300	6923464
"/kernel/arch/alpha/kernel/core_mcpcia.c"	21867	15	3	5886267	12097730
"/kernel/arch/alpha/kernel/core_polaris.c"	7416	10	0	2097168	2894382
"/kernel/arch/alpha/kernel/core_pyxis.c"	15489	15	1	4259022	6163676
"/kernel/arch/alpha/kernel/core_t2.c"	16916	12	1	3866690	5707754
"/kernel/arch/alpha/kernel/core_tsunami.c"	11823	12	1	3405185	4667556
"/kernel/arch/alpha/kernel/es1888.c"	1325	1	0	358079	844542
"/kernel/arch/alpha/kernel/fpreg.c"	6030	4	0	1213050	1831990
"/kernel/arch/alpha/kernel/irq.c"	23280	19	13	8207261	25384581
"/kernel/arch/alpha/kernel/osf_sys.c"	34352	47	1	13063862	15349800
"/kernel/arch/alpha/kernel/process.c"	13133	19	12	6139471	13664414
"/kernel/arch/alpha/kernel/ptrace.c"	18226	14	1	4738263	6112640
"/kernel/arch/alpha/kernel/setup.c"	22801	9	0	8062158	9246310
"/kernel/arch/alpha/kernel/signal.c"	19951	17	0	5755875	6649177
"/kernel/arch/alpha/kernel/smc37c669.c"	63143	1	0	8718983	12962135
"/kernel/arch/alpha/kernel/smc37c93x.c"	6399	0	0	691266	2347438
"/kernel/arch/alpha/kernel/smp.c"	26068	33	19	13622901	34738249
"/kernel/arch/alpha/kernel/sys_alcor.c"	6705	7	4	2076471	4133940
"/kernel/arch/alpha/kernel/sys_cabriolet.c"	11121	11	4	3011272	5697962
"/kernel/arch/alpha/kernel/sys_dp264.c"	13960	16	5	5283345	9215831
"/kernel/arch/alpha/kernel/sys_eb64p.c"	5400	5	2	1489463	2766094
"/kernel/arch/alpha/kernel/sys_eiger.c"	5608	7	3	2386642	4076236
"/kernel/arch/alpha/kernel/sys_jensen.c"	3560	6	4	1816254	3467582
"/kernel/arch/alpha/kernel/sys_miata.c"	9084	8	3	3083253	4836745
"/kernel/arch/alpha/kernel/sys_mikasa.c"	7964	7	2	2363309	6542381
"/kernel/arch/alpha/kernel/sys_nautilus.c"	16245	10	3	4535631	7045241
"/kernel/arch/alpha/kernel/sys_noritake.c"	10940	9	3	3195212	8045932
"/kernel/arch/alpha/kernel/sys_rawhide.c"	5095	5	2	1768002	3045574
"/kernel/arch/alpha/kernel/sys_ruffian.c"	9248	10	3	3111750	5229165
"/kernel/arch/alpha/kernel/sys_rx164.c"	6645	7	5	2641921	5215817
"/kernel/arch/alpha/kernel/sys_sable.c"	7331	6	3	2128304	3942929
"/kernel/arch/alpha/kernel/sys_sio.c"	13300	14	1	4119046	5475533
"/kernel/arch/alpha/kernel/sys_sx164.c"	5212	6	4	2104538	6653739
"/kernel/arch/alpha/kernel/sys_takara.c"	7601	8	3	3415418	4776232
"/kernel/arch/alpha/kernel/time.c"	14880	8	4	3410228	6607334
"/kernel/arch/alpha/kernel/traps.c"	29810	19	7	8943486	19996338
"/kernel/arch/alpha/lib/callback_init.c"	2205	0	0	327866	817250
"/kernel/arch/alpha/lib/checksum.c"	4051	7	0	1167853	1686782
"/kernel/arch/alpha/lib/csum_partial_copy.c"	8842	8	0	1920071	2665176
"/kernel/arch/alpha/lib/io.c"	9941	23	11	5790815	11071031
"/kernel/arch/alpha/lib/memcpy.c"	3181	3	2	931783	2023346

"/kernel/arch/alpha/lib/srm_printk.c"	906	1	0	223255	706930
"/kernel/arch/alpha/lib/srm_puts.c"	468	1	0	159862	776624
"/kernel/arch/alpha/lib/stacktrace.c"	2655	4	1	899729	1583929
"/kernel/arch/alpha/lib/strcasecmp.c"	491	1	0	147997	623192
"/kernel/arch/alpha/math-emu/math.c"	9400	2	0	1690406	2087424
"/kernel/arch/alpha/mm/extable.c"	2414	3	0	785164	1296371
"/kernel/arch/alpha/mm/fault.c"	4617	2	1	1122595	2017285
"/kernel/arch/alpha/mm/init.c"	8962	14	6	3838526	6811094
"/kernel/arch/arm/boot/compressed/misc.c"	6485	10	5	3944518	5912211
"/kernel/arch/arm/kernel/armksyms.c"	6182	0	0	6129404	6647803
"/kernel/arch/arm/kernel/arthur.c"	1947	2	1	1013319	1440803
"/kernel/arch/arm/kernel/dec21285.c"	6526	14	5	3284784	5690435
"/kernel/arch/arm/kernel/dma-a5k.c"	2329	6	3	1443304	2685001
"/kernel/arch/arm/kernel/dma-arc.c"	3842	6	3	1625112	3005591
"/kernel/arch/arm/kernel/dma-dummy.c"	451	4	2	713833	1557910
"/kernel/arch/arm/kernel/dma-footbridge.c"	1990	6	3	1476047	5108642
"/kernel/arch/arm/kernel/dma-isa.c"	4038	6	3	1552596	2959830
"/kernel/arch/arm/kernel/dma-rpc.c"	7714	10	7	3332755	6672882
"/kernel/arch/arm/kernel/dma.c"	5542	11	2	2485735	3756444
"/kernel/arch/arm/kernel/ecard.c"	21642	29	19	12656944	27231757
"/kernel/arch/arm/kernel/fiq.c"	4788	7	4	1915272	3697690
"/kernel/arch/arm/kernel/hw-footbridge.c"	18931	3	3	2036131	4673870
"/kernel/arch/arm/kernel/iic.c"	3593	10	6	2256953	4444464
"/kernel/arch/arm/kernel/init_task.c"	755	0	0	310305	794900
"/kernel/arch/arm/kernel/ioport.c"	647	1	0	148028	527969
"/kernel/arch/arm/kernel/irq.c"	11710	11	7	4114254	8232050
"/kernel/arch/arm/kernel/isa.c"	1293	1	0	553812	1092328
"/kernel/arch/arm/kernel/leds-ebsa110.c"	617	1	1	426341	1092670
"/kernel/arch/arm/kernel/leds-footbridge.c"	4920	2	1	819175	1689908
"/kernel/arch/arm/kernel/oldlatches.c"	1169	0	0	82897	570860
"/kernel/arch/arm/kernel/process.c"	7263	17	12	4791352	12140567
"/kernel/arch/arm/kernel/ptrace.c"	19877	18	1	5477468	6883912
"/kernel/arch/arm/kernel/setup.c"	12187	8	0	3919538	4922234
"/kernel/arch/arm/kernel/signal.c"	15737	12	0	4647096	5101797
"/kernel/arch/arm/kernel/sys_arm.c"	8833	14	0	2948589	3750004
"/kernel/arch/arm/kernel/time.c"	3860	5	3	1367043	2753050
"/kernel/arch/arm/kernel/traps.c"	10396	16	12	5824858	12179790
"/kernel/arch/arm/lib/getconsdata.c"	2706	0	0	665198	1183278
"/kernel/arch/arm/lib/io.c"	861	3	3	1006429	2061397
"/kernel/arch/arm/mm/extable.c"	1366	2	0	492084	1199582
"/kernel/arch/arm/mm/fault-armo.c"	2510	6	2	1228249	2332717
"/kernel/arch/arm/mm/fault-armv.c"	11189	5	1	1363797	2637126
"/kernel/arch/arm/mm/fault-common.c"	4806	5	4	1903787	3756402
"/kernel/arch/arm/mm/init.c"	6838	9	4	2757868	4781166
"/kernel/arch/arm/mm/ioremap.c"	4256	5	1	1112963	2001879
"/kernel/arch/arm/mm/mm-armv.c"	2320	2	1	701049	1477067
"/kernel/arch/arm/mm/mm-ebsa110.c"	464	0	0	17062	459752
"/kernel/arch/arm/mm/mm-footbridge.c"	2851	2	0	289783	626570
"/kernel/arch/arm/mm/mm-nexuspci.c"	720	0	0	17505	410842
"/kernel/arch/arm/mm/mm-rpc.c"	2462	3	0	562048	888494
"/kernel/arch/arm/mm/mm-tbox.c"	1991	0	0	18050	473768
"/kernel/arch/arm/mm/small_page.c"	5190	3	3	1665480	3082314
"/kernel/arch/arm/nwfpe/double_cpdo.c"	6565	1	0	1401627	196440648
"/kernel/arch/arm/nwfpe/extended_cpdo.c"	6366	1	0	1324433	1970779
"/kernel/arch/arm/nwfpe/fpa11_cpdo.c"	3677	1	0	660442	1194302
"/kernel/arch/arm/nwfpe/fpa11_cpdt.c"	8404	13	8	4606114	8123548
"/kernel/arch/arm/nwfpe/fpa11_cprt.c"	8691	5	0	2108385	5293525
"/kernel/arch/arm/nwfpe/fpa11.c"	5084	4	3	1530867	3034228
"/kernel/arch/arm/nwfpe/fpmodule.c"	5326	1	1	1070967	1999176
"/kernel/arch/arm/nwfpe/fpopcode.c"	4515	8	0	1424918	1982884
"/kernel/arch/arm/nwfpe/single_cpdo.c"	5684	1	0	1221938	1833700
"/kernel/arch/arm/nwfpe/softfloat.c"	167861	55	2	14419524	32841172
"/kernel/arch/i386/boot/compressed/misc.c"	8807	17	12	6444268	12228747
"/kernel/arch/i386/boot/tools/build.c"	5296	4	3	2283936	3331196
"/kernel/arch/i386/kernel/apm.c"	42816	31	10	11763299	29911633
"/kernel/arch/i386/kernel/bios32.c"	36610	12	3	4500042	9714068
"/kernel/arch/i386/kernel/bluesmoke.c"	5054	5	5	2566003	4645521
"/kernel/arch/i386/kernel/cpuid.c"	3778	5	2	1365403	2486217
"/kernel/arch/i386/kernel/dmi_scan.c"	3967	0	0	990516	1252771
"/kernel/arch/i386/kernel/i386_ksyms.c"	3376	0	0	1943859	3791206
"/kernel/arch/i386/kernel/init_task.c"	669	0	0	309303	756655
"/kernel/arch/i386/kernel/io_apic.c"	33521	22	14	11798348	33621804
"/kernel/arch/i386/kernel/ioport.c"	2700	3	1	861708	1668402
"/kernel/arch/i386/kernel/irq.c"	28785	23	16	9796181	27137791
"/kernel/arch/i386/kernel/ldt.c"	3653	3	0	836115	1429197
"/kernel/arch/i386/kernel/mca.c"	23444	17	8	5521044	13272330
"/kernel/arch/i386/kernel/microcode.c"	9069	6	1	2503284	4140999
"/kernel/arch/i386/kernel/msr.c"	6028	9	1	1828862	2841866
"/kernel/arch/i386/kernel/mtrr.c"	59190	22	1	7825534	12164408
"/kernel/arch/i386/kernel/process.c"	21608	28	18	10308869	25887241
"/kernel/arch/i386/kernel/ptrace.c"	18304	11	1	3948450	8526656
"/kernel/arch/i386/kernel/setup.c"	41648	3	2	8353051	17526876
"/kernel/arch/i386/kernel/signal.c"	18167	17	0	5390834	5984332
"/kernel/arch/i386/kernel/smp.c"	57869	28	18	19506460	56894412
"/kernel/arch/i386/kernel/sys_i386.c"	5674	7	0	1715017	5020804
"/kernel/arch/i386/kernel/time.c"	22350	11	5	5378507	10514819
"/kernel/arch/i386/kernel/traps.c"	20053	34	20	13112921	30423808
"/kernel/arch/i386/kernel/visws_apic.c"	9405	12	12	4974093	11183693
"/kernel/arch/i386/kernel/vm86.c"	19811	23	14	8468461	22578198
"/kernel/arch/i386/lib/delay.c"	1760	5	5	1409087	3033692
"/kernel/arch/i386/lib/old-checksum.c"	395	1	0	150840	624343

”/kernel/arch/i386/lib/usercopy.c”	3184	7	0	1130810	1333528
”/kernel/arch/i386/math-emu/errors.c”	18698	18	10	7171847	15298316
”/kernel/arch/i386/math-emu/fpu_arith.c”	3033	18	18	4977989	10269049
”/kernel/arch/i386/math-emu/fpu_aux.c”	4295	13	13	3597026	8119235
”/kernel/arch/i386/math-emu/fpu_entry.c”	22262	5	2	3788122	9220183
”/kernel/arch/i386/math-emu/fpu_etc.c”	3317	5	5	1665993	3489625
”/kernel/arch/i386/math-emu/fpu_tags.c”	2799	14	7	2850320	5046432
”/kernel/arch/i386/math-emu/fpu_trig.c”	41467	25	21	16132557	55039664
”/kernel/arch/i386/math-emu/get_address.c”	11737	5	0	2301340	3175016
”/kernel/arch/i386/math-emu/load_store.c”	9028	1	0	1364852	4674664
”/kernel/arch/i386/math-emu/poly_2xm1.c”	4757	1	0	921148	1485067
”/kernel/arch/i386/math-emu/poly_atan.c”	6749	1	0	1174620	1806152
”/kernel/arch/i386/math-emu/poly_l2.c”	7750	3	1	1822621	2708444
”/kernel/arch/i386/math-emu/poly_sin.c”	11543	2	2	2449351	4080865
”/kernel/arch/i386/math-emu/poly_tan.c”	7208	1	1	1443428	2514346
”/kernel/arch/i386/math-emu/reg_add_sub.c”	9344	3	0	1527876	2281749
”/kernel/arch/i386/math-emu/reg_compare.c”	8626	10	6	2877020	5877491
”/kernel/arch/i386/math-emu/reg_constant.c”	3833	9	9	3094999	6248880
”/kernel/arch/i386/math-emu/reg_convert.c”	1678	1	0	214139	733801
”/kernel/arch/i386/math-emu/reg_divide.c”	5273	1	0	743883	1221628
”/kernel/arch/i386/math-emu/reg_ld_str.c”	33744	21	2	8434212	11398194
”/kernel/arch/i386/math-emu/reg_mul.c”	4170	1	0	529599	1081139
”/kernel/arch/i386/mm/extable.c”	1367	2	0	492704	996358
”/kernel/arch/i386/mm/fault.c”	7289	2	1	1634018	2755064
”/kernel/arch/i386/mm/init.c”	12743	15	7	4729082	9126978
”/kernel/arch/i386/mm/ioremap.c”	3700	5	1	1558454	2100342
”/kernel/arch/m68k/amiga/amiga_ksyms.c”	881	0	0	552038	1041040
”/kernel/arch/m68k/amiga/amiints.c”	13830	16	15	8501932	16851460
”/kernel/arch/m68k/amiga/amisound.c”	2542	3	3	1358432	2602162
”/kernel/arch/m68k/amiga/chipram.c”	3717	4	0	1065024	1616072
”/kernel/arch/m68k/amiga/cia.c”	4995	7	4	2136015	4001374
”/kernel/arch/m68k/amiga/config.c”	26084	19	8	9545266	18038771
”/kernel/arch/m68k/amiga/pcmcia.c”	2300	6	5	1595541	3300866
”/kernel/arch/m68k/apollo/config.c”	6146	12	5	3863529	8882124
”/kernel/arch/m68k/apollo/dn_debug.c”	429	1	0	196291	692278
”/kernel/arch/m68k/apollo/dn_ints.c”	3305	11	8	3335735	5724878
”/kernel/arch/m68k/atari/ataints.c”	19365	33	0	8865024	9847446
”/kernel/arch/m68k/atari/atakeyb.c”	26983	29	26	13436766	40630555
”/kernel/arch/m68k/atari/atari_ksyms.c”	1264	0	0	1353659	1832280
”/kernel/arch/m68k/atari/atasound.c”	2645	2	0	833233	977190
”/kernel/arch/m68k/atari/config.c”	24680	7	3	6437752	11859769
”/kernel/arch/m68k/atari/debug.c”	10204	12	4	3563192	6178300
”/kernel/arch/m68k/atari/joystick.c”	3609	7	1	1452617	2349281
”/kernel/arch/m68k/atari/stdma.c”	4934	6	4	1653402	3428202
”/kernel/arch/m68k/atari/stram.c”	42677	7	2	3499119	8000204
”/kernel/arch/m68k/atari/time.c”	11401	10	3	3178876	5466001
”/kernel/arch/m68k/bvme6000/bvmeints.c”	3582	8	2	1751086	2908172
”/kernel/arch/m68k/bvme6000/config.c”	13086	22	5	5993166	9501534
”/kernel/arch/m68k/bvme6000/rtc.c”	4138	4	0	1047265	1633856
”/kernel/arch/m68k/hp300/config.c”	2046	5	3	1210489	2427652
”/kernel/arch/m68k/hp300/hil.c”	12342	6	5	3391139	6715342
”/kernel/arch/m68k/hp300/ints.c”	5572	5	4	1980967	3965427
”/kernel/arch/m68k/hp300/ksyms.c”	202	0	0	17501	458452
”/kernel/arch/m68k/hp300/time.c”	1939	3	2	824261	1630823
”/kernel/arch/m68k/kernel/bios32.c”	13891	0	0	29627	720794
”/kernel/arch/m68k/kernel/ints.c”	7365	13	9	4489491	11573400
”/kernel/arch/m68k/kernel/kgdb.c”	37255	12	4	6715384	14041913
”/kernel/arch/m68k/kernel/m68k_defs.c”	3927	1	0	609924	1179204
”/kernel/arch/m68k/kernel/m68k_ksyms.c”	2475	0	0	2244316	4471175
”/kernel/arch/m68k/kernel/process.c”	8415	14	6	3883280	7002946
”/kernel/arch/m68k/kernel/ptrace.c”	16003	9	1	3619100	5214883
”/kernel/arch/m68k/kernel/setup.c”	13196	7	3	5608460	7714380
”/kernel/arch/m68k/kernel/signal.c”	30059	21	2	8326279	11603026
”/kernel/arch/m68k/kernel/sys_m68k.c”	13572	9	0	2980875	3908918
”/kernel/arch/m68k/kernel/time.c”	5610	7	4	2215276	3959764
”/kernel/arch/m68k/kernel/traps.c”	28290	10	7	5659368	13641724
”/kernel/arch/m68k/lib/ashldi3.c”	1635	1	0	454042	987690
”/kernel/arch/m68k/lib/ashrdi3.c”	1690	1	0	460737	921692
”/kernel/arch/m68k/lib/checksum.c”	10503	3	0	1663583	5078058
”/kernel/arch/m68k/lib/lshrdi3.c”	1623	1	0	452517	948067
”/kernel/arch/m68k/lib/memcmp.c”	257	1	0	146173	569172
”/kernel/arch/m68k/lib/memcpy.c”	1479	1	0	329520	688122
”/kernel/arch/m68k/lib/memset.c”	1204	1	0	283263	734240
”/kernel/arch/m68k/mac/adb-bus.c”	74866	32	22	25856693	87658891
”/kernel/arch/m68k/mac/adb-misc.c”	3420	4	3	1187554	2531462
”/kernel/arch/m68k/mac/bootparse.c”	2988	2	2	1224842	2041674
”/kernel/arch/m68k/mac/config.c”	28890	15	9	9246859	19730816
”/kernel/arch/m68k/mac/debug.c”	10967	13	6	5365182	7731520
”/kernel/arch/m68k/mac/iop.c”	20283	22	16	8412720	25022423
”/kernel/arch/m68k/mac/mac_ksyms.c”	339	0	0	198720	696050
”/kernel/arch/m68k/mac/macboing.c”	8410	6	6	3813339	6602476
”/kernel/arch/m68k/mac/macints.c”	15718	15	11	6542257	17479649
”/kernel/arch/m68k/mac/mackeyb.c”	21472	9	6	5556929	14196392
”/kernel/arch/m68k/mac/oss.c”	7444	10	9	4584294	7688509
”/kernel/arch/m68k/mac/psc.c”	4618	9	8	2769009	8296034
”/kernel/arch/m68k/mac/via.c”	20081	22	17	9789536	23551396
”/kernel/arch/m68k/math-emu/fp_arith.c”	14767	17	1	4508878	5760985
”/kernel/arch/m68k/math-emu/fp_log.c”	2194	11	0	1306901	1840212
”/kernel/arch/m68k/math-emu/fp_trig.c”	2779	18	0	2017521	2645973
”/kernel/arch/m68k/mm/extable.c”	1259	2	0	474970	1521519
”/kernel/arch/m68k/mm/fault.c”	4664	2	0	901753	1467540

"/kernel/arch/m68k/mm/hwtest.c"	2429	2	0	370706	878240
"/kernel/arch/m68k/mm/init.c"	12595	11	4	4073483	6981416
"/kernel/arch/m68k/mm/kmap.c"	8253	6	4	2336969	4578969
"/kernel/arch/m68k/mm/memory.c"	15031	14	3	3760574	6473978
"/kernel/arch/m68k/mvme147/147ints.c"	3197	8	2	1661327	5512938
"/kernel/arch/m68k/mvme147/config.c"	6025	18	4	4053956	8847317
"/kernel/arch/m68k/mvme16x/16xints.c"	3357	8	2	1666896	2814735
"/kernel/arch/m68k/mvme16x/config.c"	10446	17	5	5130690	8201702
"/kernel/arch/m68k/mvme16x/mvme16x_ksyms.c"	135	0	0	63866	531738
"/kernel/arch/m68k/mvme16x/rtc.c"	3848	4	0	1005986	1408714
"/kernel/arch/m68k/q40/config.c"	10214	17	5	5532310	8330372
"/kernel/arch/m68k/q40/q40ints.c"	8374	12	3	3348883	5371280
"/kernel/arch/m68k/sun3x/config.c"	2703	11	7	2777229	4815002
"/kernel/arch/m68k/sun3x/dvma.c"	4592	3	0	930228	1553064
"/kernel/arch/m68k/sun3x/sbus.c"	769	6	2	915122	1814606
"/kernel/arch/m68k/sun3x/time.c"	1918	4	2	1289306	4515296
"/kernel/arch/m68k/tools/amiga/dmesg.c"	1656	1	0	271132	757754
"/kernel/arch/mips/arc/cmdline.c"	1261	2	1	608276	1286080
"/kernel/arch/mips/arc/console.c"	1134	2	1	484846	1185705
"/kernel/arch/mips/arc/env.c"	449	2	0	221390	979156
"/kernel/arch/mips/arc/file.c"	1449	10	0	1444790	1570097
"/kernel/arch/mips/arc/identify.c"	1643	0	0	345044	809084
"/kernel/arch/mips/arc/init.c"	1319	1	0	526237	1157579
"/kernel/arch/mips/arc/memory.c"	4940	6	3	1749579	3310572
"/kernel/arch/mips/arc/misc.c"	1382	8	6	1870131	3638481
"/kernel/arch/mips/arc/printf.c"	693	1	1	368374	1062500
"/kernel/arch/mips/arc/salone.c"	728	3	0	373570	829751
"/kernel/arch/mips/arc/time.c"	395	2	0	215700	635422
"/kernel/arch/mips/arc/tree.c"	2881	10	3	2112867	3123802
"/kernel/arch/mips/baget/baget.c"	2996	3	0	762530	1290304
"/kernel/arch/mips/baget/balo.c"	4507	10	7	2997910	5650414
"/kernel/arch/mips/baget/irq.c"	10166	20	14	6548777	15543360
"/kernel/arch/mips/baget/print.c"	2540	9	8	2379570	4835903
"/kernel/arch/mips/baget/prom/init.c"	390	1	0	194416	680358
"/kernel/arch/mips/baget/reset.c"	526	4	4	1051471	2153329
"/kernel/arch/mips/baget/setup.c"	16126	8	8	5367638	10857714
"/kernel/arch/mips/baget/time.c"	2146	6	4	1404843	3017224
"/kernel/arch/mips/baget/vacserial.c"	76335	56	30	31455553	123136468
"/kernel/arch/mips/baget/wbflush.c"	385	2	2	587385	1438862
"/kernel/arch/mips/boot/addinitrd.c"	3213	2	0	557364	1062811
"/kernel/arch/mips/boot/elf2ecoff.c"	18301	11	3	4764688	7885249
"/kernel/arch/mips/boot/mkboot.c"	18133	13	9	6050216	12855430
"/kernel/arch/mips/dec/boot/decstation.c"	2210	2	0	619549	1144868
"/kernel/arch/mips/dec/irq.c"	6156	12	8	3662589	7095914
"/kernel/arch/mips/dec/prom/cmdline.c"	790	1	1	410592	1109088
"/kernel/arch/mips/dec/prom/identify.c"	2404	1	0	662128	1142335
"/kernel/arch/mips/dec/prom/init.c"	2777	2	1	2167667	2357960
"/kernel/arch/mips/dec/prom/memory.c"	2515	5	2	1286654	2280190
"/kernel/arch/mips/dec/promcon.c"	1277	5	0	801156	1312082
"/kernel/arch/mips/dec/reset.c"	384	3	3	779752	1776791
"/kernel/arch/mips/dec/rtc-dec.c"	785	3	1	641574	1356761
"/kernel/arch/mips/dec/serial.c"	2073	0	0	19093	481700
"/kernel/arch/mips/dec/setup.c"	13395	9	9	5950942	10985326
"/kernel/arch/mips/dec/time.c"	12991	10	5	3872579	7084127
"/kernel/arch/mips/dec/wbflush.c"	2341	5	5	1660628	3359427
"/kernel/arch/mips/jazz/floppy-jazz.c"	3289	18	10	3872365	7335574
"/kernel/arch/mips/jazz/io.c"	3194	10	7	2669760	5216342
"/kernel/arch/mips/jazz/jazzdma.c"	14512	15	7	4766746	9551546
"/kernel/arch/mips/jazz/kbd-jazz.c"	2186	8	6	2068733	4161054
"/kernel/arch/mips/jazz/reset.c"	730	4	4	1011277	2315049
"/kernel/arch/mips/jazz/rtc-jazz.c"	820	3	1	802822	1294152
"/kernel/arch/mips/jazz/setup.c"	3156	4	4	2340070	6056974
"/kernel/arch/mips/kernel/branch.c"	4014	1	0	636094	1202635
"/kernel/arch/mips/kernel/fpe.c"	1404	3	2	884787	1775942
"/kernel/arch/mips/kernel/gdb-stub.c"	20258	13	7	6214649	11975707
"/kernel/arch/mips/kernel/init_task.c"	750	0	0	311492	789644
"/kernel/arch/mips/kernel/ioport.c"	976	2	0	339160	735446
"/kernel/arch/mips/kernel/ipc.c"	2544	1	0	465478	880441
"/kernel/arch/mips/kernel/irixelf.c"	35639	19	2	7859947	11970194
"/kernel/arch/mips/kernel/irixinv.c"	2145	3	0	666470	1184064
"/kernel/arch/mips/kernel/irixioctl.c"	5820	3	0	1288322	1922470
"/kernel/arch/mips/kernel/irixsig.c"	22787	18	1	6881928	8557771
"/kernel/arch/mips/kernel/irq.c"	8846	16	10	4882064	11390168
"/kernel/arch/mips/kernel/mips_ksyms.c"	3084	0	0	2188330	2444765
"/kernel/arch/mips/kernel/pci.c"	2418	0	0	18717	484082
"/kernel/arch/mips/kernel/proc.c"	3037	1	0	602864	954902
"/kernel/arch/mips/kernel/process.c"	5005	8	5	2320300	4402658
"/kernel/arch/mips/kernel/ptrace.c"	13191	7	1	3241239	6683000
"/kernel/arch/mips/kernel/reset.c"	493	3	3	710685	1706059
"/kernel/arch/mips/kernel/setup.c"	6050	2	1	1816072	2804749
"/kernel/arch/mips/kernel/signal.c"	14553	15	2	4593198	6801750
"/kernel/arch/mips/kernel/syscall.c"	6796	10	1	2756627	3524208
"/kernel/arch/mips/kernel/sysirix.c"	56210	72	3	21909348	28449560
"/kernel/arch/mips/kernel/sysmips.c"	2475	3	0	626482	1138898
"/kernel/arch/mips/kernel/time.c"	15458	10	5	4039715	10574380
"/kernel/arch/mips/kernel/traps.c"	16270	22	21	10879056	26512370
"/kernel/arch/mips/kernel/unaligned.c"	9786	2	1	1588259	2818306
"/kernel/arch/mips/kernel/vm86.c"	206	1	0	143223	607250
"/kernel/arch/mips/lib/csum_partial_copy.c"	2088	3	0	503424	862798
"/kernel/arch/mips/lib/dump_tlb.c"	3469	8	7	2318578	4726442
"/kernel/arch/mips/lib/floppy-no.c"	1261	3	1	617255	1324228

”/kernel/arch/mips/lib/floppy-std.c”	3101	18	11	3980433	7771084
”/kernel/arch/mips/lib/ide-no.c”	1843	8	3	1658746	2928694
”/kernel/arch/mips/lib/ide-std.c”	2228	8	3	1757032	3098112
”/kernel/arch/mips/lib/kbd-no.c”	1202	8	6	1896693	3692876
”/kernel/arch/mips/lib/kbd-std.c”	1670	8	6	2005561	3890570
”/kernel/arch/mips/lib/rtc-no.c”	861	3	1	601779	1309562
”/kernel/arch/mips/lib/rtc-std.c”	780	3	1	598667	1259674
”/kernel/arch/mips/lib/tinycon.c”	2533	8	8	2950367	5027736
”/kernel/arch/mips/mm/andes.c”	2436	15	10	3625184	6906685
”/kernel/arch/mips/mm/extable.c”	1266	2	0	499116	969152
”/kernel/arch/mips/mm/fault.c”	3629	1	0	659431	3905558
”/kernel/arch/mips/mm/init.c”	10362	16	6	4186550	7599965
”/kernel/arch/mips/mm/loadmmu.c”	3080	1	1	1822523	2671491
”/kernel/arch/mips/mm/r2300.c”	17090	26	17	10059008	22285052
”/kernel/arch/mips/mm/r4xx0.c”	67539	86	62	50029772	222565048
”/kernel/arch/mips/mm/r6000.c”	5368	17	11	4601662	9243952
”/kernel/arch/mips/mm/tfp.c”	2369	15	10	3518896	6853941
”/kernel/arch/mips/mm/umap.c”	4816	9	2	1854739	3108343
”/kernel/arch/mips/sgi/kernel/indy_hpc.c”	2270	1	1	1148852	1655471
”/kernel/arch/mips/sgi/kernel/indy_int.c”	14794	23	18	10059212	27110797
”/kernel/arch/mips/sgi/kernel/indy_mc.c”	4930	2	1	1586965	2199422
”/kernel/arch/mips/sgi/kernel/indy_rtc.c”	848	3	1	614392	1313782
”/kernel/arch/mips/sgi/kernel/indy_sc.c”	4927	4	4	1771844	3560812
”/kernel/arch/mips/sgi/kernel/indy_timer.c”	8093	10	6	3311521	8869162
”/kernel/arch/mips/sgi/kernel/promcon.c”	1299	5	0	705131	1319043
”/kernel/arch/mips/sgi/kernel/reset.c”	5594	14	11	4919544	8855628
”/kernel/arch/mips/sgi/kernel/setup.c”	4907	10	8	3135199	6750370
”/kernel/arch/mips/sgi/kernel/system.c”	2512	2	1	810892	1612761
”/kernel/arch/mips/sgi/kernel/time.c”	399	1	1	330936	1069336
”/kernel/arch/mips/sni/io.c”	4359	10	7	2758438	5542270
”/kernel/arch/mips/sni/pci.c”	5052	0	0	21168	552161
”/kernel/arch/mips/sni/pcimt_scache.c”	987	1	1	342795	936185
”/kernel/arch/mips/sni/reset.c”	959	4	4	1020877	2316612
”/kernel/arch/mips/sni/setup.c”	4678	5	5	2356504	4512852
”/kernel/arch/mips/tools/offset.c”	6512	5	5	2275140	6057840
”/kernel/arch/ppc/8xx_io/commproc.c”	6601	7	5	2916504	5142287
”/kernel/arch/ppc/8xx_io/enet.c”	29486	7	2	5267055	8652556
”/kernel/arch/ppc/8xx_io/fec.c”	25305	14	8	7339519	18657752
”/kernel/arch/ppc/8xx_io/uart.c”	68953	39	23	23864156	87117390
”/kernel/arch/ppc/amiga/amiga_ksyms.c”	42	0	0	17472	478060
”/kernel/arch/ppc/amiga/amiints.c”	14699	16	15	9095463	17268111
”/kernel/arch/ppc/amiga/amisound.c”	39	0	0	16811	466618
”/kernel/arch/ppc/amiga/bootinfo.c”	1795	1	1	825533	1399104
”/kernel/arch/ppc/amiga/chipram.c”	38	0	0	16667	447100
”/kernel/arch/ppc/amiga/cia.c”	34	0	0	15621	268180
”/kernel/arch/ppc/amiga/config.c”	24287	19	8	9746652	20183293
”/kernel/arch/ppc/amiga/ints.c”	3774	6	4	1962926	3680380
”/kernel/arch/ppc/amiga/time.c”	2742	4	0	739552	1244624
”/kernel/arch/ppc/boot/kbd.c”	4393	4	1	1317381	2183433
”/kernel/arch/ppc/boot/misc.c”	17306	27	13	8817448	18454129
”/kernel/arch/ppc/boot/mkprep.c”	6982	6	3	2601905	4123097
”/kernel/arch/ppc/boot/ns16550.c”	1202	4	0	660649	1154346
”/kernel/arch/ppc/boot/of1275.c”	7265	27	1	4097401	4974244
”/kernel/arch/ppc/boot/vreset.c”	18577	13	3	5408494	8002804
”/kernel/arch/ppc/chrpboot/main.c”	3883	4	2	1637417	4426996
”/kernel/arch/ppc/chrpboot/mknote.c”	876	1	0	158635	644592
”/kernel/arch/ppc/chrpboot/no_initrd.c”	41	0	0	107970	467340
”/kernel/arch/ppc/chrpboot/piggyback.c”	1404	1	0	359542	667508
”/kernel/arch/ppc/chrpboot/start.c”	4890	15	4	3254766	5042398
”/kernel/arch/ppc/coffboot/chrpmain.c”	6044	4	2	2095823	3364310
”/kernel/arch/ppc/coffboot/coffmain.c”	3755	4	2	1702398	2786552
”/kernel/arch/ppc/coffboot/dummy.c”	30	1	0	105400	559971
”/kernel/arch/ppc/coffboot/hack-coff.c”	2314	1	0	318742	647227
”/kernel/arch/ppc/coffboot/main.c”	4444	4	2	1717321	2705348
”/kernel/arch/ppc/coffboot/mknote.c”	876	1	0	158914	644662
”/kernel/arch/ppc/coffboot/no_initrd.c”	41	0	0	106745	470990
”/kernel/arch/ppc/coffboot/piggyback.c”	1404	1	0	355938	679304
”/kernel/arch/ppc/coffboot/start.c”	5152	16	4	3414426	5247046
”/kernel/arch/ppc/coffboot/zlib.c”	66427	26	3	15099618	26171668
”/kernel/arch/ppc/kernel/align.c”	7719	1	0	1080077	1849346
”/kernel/arch/ppc/kernel/apus_setup.c”	16087	17	7	7208272	11753276
”/kernel/arch/ppc/kernel/bitops.c”	4813	6	3	2881886	2946978
”/kernel/arch/ppc/kernel/checks.c”	1302	1	0	212649	716722
”/kernel/arch/ppc/kernel/chrp_pci.c”	11573	21	0	4425290	5020890
”/kernel/arch/ppc/kernel/chrp_setup.c”	15504	12	6	5416159	9850474
”/kernel/arch/ppc/kernel/chrp_time.c”	5142	6	2	1733626	3014156
”/kernel/arch/ppc/kernel/feature.c”	27229	12	7	6649780	20404441
”/kernel/arch/ppc/kernel/find_name.c”	843	1	0	192162	704734
”/kernel/arch/ppc/kernel/gemini_pci.c”	2983	8	2	1621934	2664776
”/kernel/arch/ppc/kernel/gemini_setup.c”	13233	10	4	4596000	9441874
”/kernel/arch/ppc/kernel/i8259.c”	3772	6	5	2042320	3963616
”/kernel/arch/ppc/kernel/idle.c”	8525	6	2	2879103	4084770
”/kernel/arch/ppc/kernel/indirect_pci.c”	2936	6	0	1084161	1640554
”/kernel/arch/ppc/kernel/irq.c”	13463	12	8	4286756	9246620
”/kernel/arch/ppc/kernel/mbx_pci.c”	7561	10	0	1642830	2324814
”/kernel/arch/ppc/kernel/mbx_setup.c”	11990	13	7	5324230	9017768
”/kernel/arch/ppc/kernel/mk_defs.c”	5525	1	0	834650	1454367
”/kernel/arch/ppc/kernel/open_pic.c”	3682	1	1	894571	1709093
”/kernel/arch/ppc/kernel/openpic.c”	16295	31	22	10469617	29943208
”/kernel/arch/ppc/kernel/pci.c”	6372	13	0	2389092	3109966
”/kernel/arch/ppc/kernel/pmac_pci.c”	23941	33	4	9841288	14230382

"/kernel/arch/ppc/kernel/pmac_pic.c"	15663	3	1	3155778	4846148
"/kernel/arch/ppc/kernel/pmac_setup.c"	17917	12	8	6553042	12533360
"/kernel/arch/ppc/kernel/pmac_support.c"	12203	19	1	5882512	7429475
"/kernel/arch/ppc/kernel/pmac_time.c"	6946	5	1	1584515	5395332
"/kernel/arch/ppc/kernel/ppc_htab.c"	15843	6	0	3353092	4396844
"/kernel/arch/ppc/kernel/ppc_ksyms.c"	7148	0	0	6451580	6733238
"/kernel/arch/ppc/kernel/ppc-stub.c"	17680	21	9	6061388	13075258
"/kernel/arch/ppc/kernel/ppc8xx_pic.c"	1222	3	3	964657	2069388
"/kernel/arch/ppc/kernel/prep_nvram.c"	3363	8	1	1744048	2647588
"/kernel/arch/ppc/kernel/prep_pci.c"	37305	14	2	8779581	13436340
"/kernel/arch/ppc/kernel/prep_setup.c"	21855	15	8	8118089	15066139
"/kernel/arch/ppc/kernel/prep_time.c"	6740	4	0	1665594	2233812
"/kernel/arch/ppc/kernel/process.c"	14334	20	9	6150769	14859334
"/kernel/arch/ppc/kernel/prom.c"	84355	33	7	16310326	42875386
"/kernel/arch/ppc/kernel/ptrace.c"	14211	11	3	3635233	6243896
"/kernel/arch/ppc/kernel/residual.c"	25746	5	0	3881476	4595932
"/kernel/arch/ppc/kernel/setup.c"	19918	13	7	6545377	12826827
"/kernel/arch/ppc/kernel/signal.c"	19117	10	0	4008932	5040760
"/kernel/arch/ppc/kernel/smp.c"	13642	6	6	4166577	8244974
"/kernel/arch/ppc/kernel/softemu8xx.c"	3470	1	0	448224	980582
"/kernel/arch/ppc/kernel/syscalls.c"	7313	15	1	3256448	3974746
"/kernel/arch/ppc/kernel/time.c"	9263	7	6	3150067	6193052
"/kernel/arch/ppc/kernel/totalmp.c"	3318	1	1	804456	1641141
"/kernel/arch/ppc/kernel/traps.c"	7571	13	12	4340511	12397601
"/kernel/arch/ppc/lib/locks.c"	5506	10	9	3170167	9529338
"/kernel/arch/ppc/lib/strcase.c"	380	2	0	284576	736941
"/kernel/arch/ppc/mbxboot/mbxtty.c"	4648	4	2	1720116	2417406
"/kernel/arch/ppc/mbxboot/misc.c"	14013	23	11	7945724	14796879
"/kernel/arch/ppc/mm/extable.c"	2228	4	1	995345	1768818
"/kernel/arch/ppc/mm/fault.c"	9669	2	1	2728775	2724819
"/kernel/arch/ppc/mm/init.c"	45355	37	19	19215352	49555497
"/kernel/arch/ppc/xmon/adb.c"	3661	9	6	2274363	4452781
"/kernel/arch/ppc/xmon/ppc-dis.c"	5495	2	0	957410	4274154
"/kernel/arch/ppc/xmon/ppc-opc.c"	117931	38	0	20569978	30312664
"/kernel/arch/ppc/xmon/setjmp.c"	525	2	1	435281	1114411
"/kernel/arch/ppc/xmon/start.c"	8709	17	6	5143920	7594360
"/kernel/arch/ppc/xmon/subr_prf.c"	969	3	3	862218	1917184
"/kernel/arch/ppc/xmon/xmon.c"	24702	44	28	18606817	52400539
"/kernel/arch/s390/kernel/cpcmd.c"	1489	1	1	491146	1217958
"/kernel/arch/s390/kernel/debug.c"	31416	38	8	12589464	22338532
"/kernel/arch/s390/kernel/ebcdic.c"	17916	0	0	2521530	3635328
"/kernel/arch/s390/kernel/floatlib.c"	19640	32	0	5495681	6472834
"/kernel/arch/s390/kernel/gdb-stub.c"	12329	13	9	4825772	12959787
"/kernel/arch/s390/kernel/init_task.c"	855	0	0	306007	784145
"/kernel/arch/s390/kernel/irq.c"	9925	5	2	1325442	2831149
"/kernel/arch/s390/kernel/irqextras390.c"	758	1	1	296806	967738
"/kernel/arch/s390/kernel/mathemu.c"	34102	103	5	16710902	23683464
"/kernel/arch/s390/kernel/process.c"	11763	16	5	4794896	9375107
"/kernel/arch/s390/kernel/ptrace.c"	15595	12	4	4156325	10311702
"/kernel/arch/s390/kernel/s390_ext.c"	2445	2	0	681575	1197388
"/kernel/arch/s390/kernel/s390_ksyms.c"	1099	0	0	1082513	1502242
"/kernel/arch/s390/kernel/s390dyn.c"	4689	3	0	1464424	1689750
"/kernel/arch/s390/kernel/s390fpu.c"	3447	4	2	1161720	2199352
"/kernel/arch/s390/kernel/s390io.c"	127010	50	7	31504558	66898392
"/kernel/arch/s390/kernel/s390mach.c"	10698	8	4	3598488	5866128
"/kernel/arch/s390/kernel/setup.c"	10327	11	8	4397673	8700097
"/kernel/arch/s390/kernel/signal.c"	14178	16	0	3991649	4967846
"/kernel/arch/s390/kernel/smp.c"	25335	20	16	10056884	28872927
"/kernel/arch/s390/kernel/sys_s390.c"	5749	8	0	1921620	2480866
"/kernel/arch/s390/kernel/time.c"	7123	7	6	2950624	5556916
"/kernel/arch/s390/kernel/traps.c"	10043	17	6	4887298	8117106
"/kernel/arch/s390/lib/checksum.c"	1818	2	0	373543	859429
"/kernel/arch/s390/lib/delay.c"	1366	2	2	621540	1477386
"/kernel/arch/s390/mm/extable.c"	1698	2	0	518530	998327
"/kernel/arch/s390/mm/fault.c"	11631	2	2	2861391	4313175
"/kernel/arch/s390/mm/init.c"	14543	14	7	5725793	10036012
"/kernel/arch/s390/mm/ioremap.c"	3141	5	1	1120565	1938272
"/kernel/arch/s390/tools/dasdfmt/dasdfmt.c"	22691	9	4	5081707	11928978
"/kernel/arch/s390/tools/hwc/hwc_cntl_key.c"	2546	1	0	428078	935993
"/kernel/arch/s390/tools/hwc/hwc_measure.c"	1704	1	0	515245	776368
"/kernel/arch/s390/tools/silo/cfg.c"	8039	19	9	4715080	8944353
"/kernel/arch/s390/tools/silo/silo.c"	14490	11	0	3280140	4042396
"/kernel/arch/sparc/ap1000/aplib.c"	10973	11	2	2969280	4719420
"/kernel/arch/sparc/ap1000/apmmu.c"	33151	103	56	34194990	119572384
"/kernel/arch/sparc/ap1000/approm.c"	2785	9	0	1752905	2152928
"/kernel/arch/sparc/ap1000/bnet.c"	25843	36	33	17850231	54830695
"/kernel/arch/sparc/ap1000/dma.c"	1766	2	0	392148	894089
"/kernel/arch/sparc/ap1000/hw.c"	4350	11	10	3563567	7451351
"/kernel/arch/sparc/ap1000/irq.c"	1507	5	5	1425575	3035547
"/kernel/arch/sparc/ap1000/kgdb.c"	1681	3	2	1051855	1973428
"/kernel/arch/sparc/ap1000/mpp.c"	1686	4	3	1295396	2422446
"/kernel/arch/sparc/ap1000/msc.c"	34296	32	29	18104174	60912925
"/kernel/arch/sparc/ap1000/sync.c"	1202	1	0	385770	896926
"/kernel/arch/sparc/ap1000/timer.c"	2844	6	5	1740380	3474760
"/kernel/arch/sparc/ap1000/tnet.c"	15683	25	16	9174825	20331852
"/kernel/arch/sparc/ap1000/util.c"	8874	27	18	6824280	18716446
"/kernel/arch/sparc/boot/btfixupprep.c"	11448	4	1	2854068	4414234
"/kernel/arch/sparc/boot/piggyback.c"	3614	2	1	715882	1670837
"/kernel/arch/sparc/kernel/auxio.c"	2533	2	2	944160	4668963
"/kernel/arch/sparc/kernel/cpu.c"	4969	1	1	1162739	2181890
"/kernel/arch/sparc/kernel/devices.c"	2896	1	0	936004	1523128

"/kernel/arch/sparc/kernel/ebus.c"	10091	4	2	2290817	4001848
"/kernel/arch/sparc/kernel/errtbls.c"	4335	0	0	548534	1082172
"/kernel/arch/sparc/kernel/idprom.c"	3393	3	2	1150161	2171797
"/kernel/arch/sparc/kernel/init_task.c"	685	0	0	306481	768813
"/kernel/arch/sparc/kernel/ioport.c"	4251	3	0	755453	1220128
"/kernel/arch/sparc/kernel/irq.c"	18073	9	6	3937550	9453452
"/kernel/arch/sparc/kernel/muldiv.c"	5715	7	1	1869312	2680942
"/kernel/arch/sparc/kernel/pcic.c"	25927	3	0	478205	1439325
"/kernel/arch/sparc/kernel/process.c"	20184	19	13	7983854	18356917
"/kernel/arch/sparc/kernel/ptrace.c"	22525	15	5	6085387	11364523
"/kernel/arch/sparc/kernel/setup.c"	13653	9	5	5398662	8622674
"/kernel/arch/sparc/kernel/signal.c"	37229	21	4	10115582	22086107
"/kernel/arch/sparc/kernel/smp.c"	7474	22	16	7510113	20235369
"/kernel/arch/sparc/kernel/solaris.c"	865	1	0	240482	744311
"/kernel/arch/sparc/kernel/sparc_ksyms.c"	8149	0	0	7242317	7496356
"/kernel/arch/sparc/kernel/sparc-stub.c"	18440	15	7	5681962	11137198
"/kernel/arch/sparc/kernel/sun4c_irq.c"	6137	7	7	2944723	5800877
"/kernel/arch/sparc/kernel/sun4d_irq.c"	13858	14	11	5921582	17448958
"/kernel/arch/sparc/kernel/sun4d_smp.c"	14924	12	10	7405020	13918253
"/kernel/arch/sparc/kernel/sun4m_irq.c"	11506	11	9	4731690	11842545
"/kernel/arch/sparc/kernel/sun4m_smp.c"	13326	13	11	7080182	13928138
"/kernel/arch/sparc/kernel/sun4setup.c"	2191	1	1	1084461	1852644
"/kernel/arch/sparc/kernel/sunos_ioctl.c"	5963	1	0	1215211	1591102
"/kernel/arch/sparc/kernel/sys_solaris.c"	1168	2	0	375280	866454
"/kernel/arch/sparc/kernel/sys_sparc.c"	8513	13	0	3007630	3511348
"/kernel/arch/sparc/kernel/sys_sunos.c"	33912	45	2	12633200	16515033
"/kernel/arch/sparc/kernel/tadpole.c"	2818	6	5	1791486	3558895
"/kernel/arch/sparc/kernel/tick14.c"	1873	3	3	1113216	2253537
"/kernel/arch/sparc/kernel/time.c"	16164	10	7	4871439	12889564
"/kernel/arch/sparc/kernel/traps.c"	11936	19	7	6234970	10234982
"/kernel/arch/sparc/kernel/unaligned.c"	13737	17	6	4315479	8308158
"/kernel/arch/sparc/kernel/windows.c"	3278	4	4	1718367	2882804
"/kernel/arch/sparc/lib/debuglocks.c"	4525	0	0	21086	543027
"/kernel/arch/sparc/math-emu/math.c"	17883	3	0	2208847	3100834
"/kernel/arch/sparc/mm/asyncd.c"	5934	5	3	2083461	3697218
"/kernel/arch/sparc/mm/btfixup.c"	10389	2	2	2988373	4736414
"/kernel/arch/sparc/mm/extable.c"	1943	2	0	601666	1074233
"/kernel/arch/sparc/mm/fault.c"	12681	5	0	2836119	3688348
"/kernel/arch/sparc/mm/generic.c"	3196	5	1	1124650	1943399
"/kernel/arch/sparc/mm/init.c"	10373	9	4	4062430	202103982
"/kernel/arch/sparc/mm/io-unit.c"	8420	11	5	3349594	5837946
"/kernel/arch/sparc/mm/iommu.c"	9109	13	8	4260705	8541236
"/kernel/arch/sparc/mm/loadmmu.c"	1235	1	1	706394	1434672
"/kernel/arch/sparc/mm/nosrmmu.c"	1294	9	6	2128682	4097331
"/kernel/arch/sparc/mm/nosun4c.c"	1579	11	8	3284597	5119954
"/kernel/arch/sparc/mm/srmmu.c"	98329	162	95	97155071	472790080
"/kernel/arch/sparc/mm/sun4c.c"	76367	56	43	43668664	177574474
"/kernel/arch/sparc/prom/bootstr.c"	1492	1	0	403265	916725
"/kernel/arch/sparc/prom/console.c"	5112	6	1	1506838	2473244
"/kernel/arch/sparc/prom/devmap.c"	1447	2	1	529714	1232039
"/kernel/arch/sparc/prom/devops.c"	1908	3	1	740903	1453546
"/kernel/arch/sparc/prom/init.c"	2385	1	1	964918	1733099
"/kernel/arch/sparc/prom/memory.c"	7510	3	1	2090412	4743825
"/kernel/arch/sparc/prom/misc.c"	3052	9	5	1916228	3707095
"/kernel/arch/sparc/prom/mp.c"	2301	4	0	647088	1140821
"/kernel/arch/sparc/prom/palloc.c"	1625	2	1	667914	1244984
"/kernel/arch/sparc/prom/printf.c"	782	1	1	379029	1045447
"/kernel/arch/sparc/prom/ranges.c"	5234	7	3	2532885	3659800
"/kernel/arch/sparc/prom/segment.c"	779	1	1	379080	1058139
"/kernel/arch/sparc/prom/sun4prom.c"	3790	8	0	1612242	2152548
"/kernel/arch/sparc/prom/tree.c"	8046	21	1	3513044	4302622
"/kernel/arch/sparc64/boot/piggyback.c"	3800	2	1	792174	1575130
"/kernel/arch/sparc64/kernel/auxio.c"	2098	1	1	586920	1308684
"/kernel/arch/sparc64/kernel/binfmt_aout32.c"	11314	7	1	2689572	3921545
"/kernel/arch/sparc64/kernel/binfmt_elf32.c"	4074	0	0	628521	1120160
"/kernel/arch/sparc64/kernel/central.c"	11071	8	4	3456470	6094440
"/kernel/arch/sparc64/kernel/cpu.c"	2857	1	1	893323	1697606
"/kernel/arch/sparc64/kernel/devices.c"	2490	1	0	642678	1193370
"/kernel/arch/sparc64/kernel/ebus.c"	10846	5	2	2682064	4471053
"/kernel/arch/sparc64/kernel/idprom.c"	1392	2	1	610769	1326662
"/kernel/arch/sparc64/kernel/init_task.c"	695	0	0	310723	771072
"/kernel/arch/sparc64/kernel/ioctl32.c"	89275	32	1	17104157	23843706
"/kernel/arch/sparc64/kernel/ioport.c"	3403	3	0	649417	1177372
"/kernel/arch/sparc64/kernel/irq.c"	37541	19	13	10459186	30463934
"/kernel/arch/sparc64/kernel/power.c"	2453	1	1	372647	1163356
"/kernel/arch/sparc64/kernel/process.c"	22514	22	15	8879655	22709557
"/kernel/arch/sparc64/kernel/psycho.c"	73540	3	0	605393	2626208
"/kernel/arch/sparc64/kernel/ptrace.c"	28883	20	7	7706130	17054184
"/kernel/arch/sparc64/kernel/setup.c"	15466	10	5	5581078	9368808
"/kernel/arch/sparc64/kernel/signal.c"	27762	17	8	8309452	17100130
"/kernel/arch/sparc64/kernel/signal32.c"	44071	20	5	10816322	22698104
"/kernel/arch/sparc64/kernel/smp.c"	23980	30	23	15027901	35810701
"/kernel/arch/sparc64/kernel/sparc64_ksyms.c"	9355	0	0	7313781	6450564
"/kernel/arch/sparc64/kernel/starfire.c"	2731	4	1	1228265	1802196
"/kernel/arch/sparc64/kernel/sunos_ioctl32.c"	7782	1	0	1375962	2061914
"/kernel/arch/sparc64/kernel/sys_sparc.c"	11940	18	1	4151113	5692560
"/kernel/arch/sparc64/kernel/sys_sparc32.c"	107657	121	4	36950040	59972348
"/kernel/arch/sparc64/kernel/sys_sunos32.c"	38797	51	3	14931033	22913214
"/kernel/arch/sparc64/kernel/time.c"	14569	5	3	3325277	7419367
"/kernel/arch/sparc64/kernel/traps.c"	22300	40	32	13948352	48888928
"/kernel/arch/sparc64/kernel/unaligned.c"	19649	20	7	5877750	12019378

"/kernel/arch/sparc64/lib/debuglocks.c"	6101	0	0	21139	432608
"/kernel/arch/sparc64/lib/PeeCeeI.c"	4337	0	0	20934	545088
"/kernel/arch/sparc64/math-emu/math.c"	14163	2	0	2074657	2973874
"/kernel/arch/sparc64/mm/asyncd.c"	5941	5	3	2638396	3804026
"/kernel/arch/sparc64/mm/extable.c"	1945	2	0	592653	1093215
"/kernel/arch/sparc64/mm/fault.c"	7927	3	1	1730991	2881704
"/kernel/arch/sparc64/mm/generic.c"	4381	5	1	1301458	2174841
"/kernel/arch/sparc64/mm/init.c"	49538	36	25	24856626	64970710
"/kernel/arch/sparc64/mm/modutil.c"	1597	2	0	439562	948545
"/kernel/arch/sparc64/prom/bootstr.c"	712	1	0	288027	746979
"/kernel/arch/sparc64/prom/console.c"	3554	7	2	1442254	2473873
"/kernel/arch/sparc64/prom/devops.c"	1050	3	1	539060	1239144
"/kernel/arch/sparc64/prom/init.c"	2226	1	1	911208	1662877
"/kernel/arch/sparc64/prom/memory.c"	4758	3	1	1434129	2375319
"/kernel/arch/sparc64/prom/misc.c"	7760	21	8	4484939	7972543
"/kernel/arch/sparc64/prom/p1275.c"	11318	1	0	2242060	2755959
"/kernel/arch/sparc64/prom/printf.c"	930	1	1	437202	1121103
"/kernel/arch/sparc64/prom/ranges.c"	6498	10	4	2618862	4567949
"/kernel/arch/sparc64/prom/tree.c"	8619	21	1	3193100	4488327
"/kernel/arch/sparc64/solaris/fs.c"	22436	35	0	8221326	9129051
"/kernel/arch/sparc64/solaris/ioctl.c"	20321	15	0	5013352	6367940
"/kernel/arch/sparc64/solaris/ipc.c"	3866	1	0	747848	1294000
"/kernel/arch/sparc64/solaris/misc.c"	22471	24	1	6589532	8764677
"/kernel/arch/sparc64/solaris/signal.c"	11070	20	0	4029029	4961476
"/kernel/arch/sparc64/solaris/socket.c"	12124	26	1	5058288	9364786
"/kernel/arch/sparc64/solaris/socksys.c"	5261	5	1	1551783	2628842
"/kernel/arch/sparc64/solaris/timod.c"	23610	14	7	7730019	13944202
"/kernel/Documentation/networking/ip_masq/ip_masq-API-ex.c"	1969	2	0	400628	920674
"/kernel/drivers/acorn/block/fd1772.c"	46781	40	31	23549793	86743254
"/kernel/drivers/acorn/block/ide-ics.c"	7945	9	0	2205052	2821584
"/kernel/drivers/acorn/block/ide-rapide.c"	1648	2	0	772793	964384
"/kernel/drivers/acorn/block/mfmhd.c"	43075	31	18	15722390	51398664
"/kernel/drivers/acorn/char/keyb_arc.c"	12191	8	6	3513326	10173569
"/kernel/drivers/acorn/char/keyb_ps2.c"	9239	8	7	3603421	6667588
"/kernel/drivers/acorn/char/mouse_rpc.c"	1823	2	1	900276	1508336
"/kernel/drivers/acorn/char/serial-atomwide.c"	597	0	0	17548	453500
"/kernel/drivers/acorn/char/serial-card.c"	2602	2	0	383252	721400
"/kernel/drivers/acorn/char/serial-dualsp.c"	477	0	0	16795	428966
"/kernel/drivers/acorn/net/ether1.c"	28896	23	0	7106706	7813198
"/kernel/drivers/acorn/net/ether3.c"	24907	22	8	7691862	15613061
"/kernel/drivers/acorn/net/etherh.c"	13518	14	4	3837172	6862288
"/kernel/drivers/acorn/net/net-probe.c"	625	1	0	290246	781492
"/kernel/drivers/acorn/scsi/acornscsi.c"	89746	53	26	33894272	304736504
"/kernel/drivers/acorn/scsi/arxescsi.c"	9683	10	2	2491484	4165521
"/kernel/drivers/acorn/scsi/cumana_1.c"	8966	8	2	2237077	3885670
"/kernel/drivers/acorn/scsi/cumana_2.c"	15843	13	6	4672708	10758908
"/kernel/drivers/acorn/scsi/ecoscsi.c"	5781	6	2	1628938	2657140
"/kernel/drivers/acorn/scsi/eesox.c"	15784	13	6	4713700	9310499
"/kernel/drivers/acorn/scsi/fas216.c"	71738	48	24	27095334	89252614
"/kernel/drivers/acorn/scsi/msgqueue.c"	3725	8	4	2033296	6416194
"/kernel/drivers/acorn/scsi/oak.c"	5495	5	0	1184984	1787384
"/kernel/drivers/acorn/scsi/powertec.c"	13139	12	6	4303856	8363412
"/kernel/drivers/acorn/scsi/queue.c"	9467	10	0	2642537	3233659
"/kernel/drivers/ap1000/ap.c"	6535	8	4	2425689	4511265
"/kernel/drivers/ap1000/apfddi.c"	16834	20	10	6627731	17022538
"/kernel/drivers/ap1000/bif.c"	6285	12	1	2466584	3414510
"/kernel/drivers/ap1000/ddv_util.c"	2808	4	1	1156400	1974006
"/kernel/drivers/ap1000/ddv.c"	23839	24	15	12873198	25415871
"/kernel/drivers/ap1000/mac.c"	32122	21	13	11428677	28623282
"/kernel/drivers/ap1000/plc.c"	9601	12	11	4687577	11668131
"/kernel/drivers/ap1000/ringbuf.c"	7669	4	2	1871204	3438931
"/kernel/drivers/block/acsi_slm.c"	26542	17	4	7542631	11884348
"/kernel/drivers/block/acsi.c"	48803	33	17	17793456	47558061
"/kernel/drivers/block/ali14xx.c"	6251	6	0	1540024	2195386
"/kernel/drivers/block/alim15x3.c"	22831	9	3	3355448	7893277
"/kernel/drivers/block/amiflop.c"	49078	49	17	19116367	48139901
"/kernel/drivers/block/ataflop.c"	55516	40	29	25930424	87101753
"/kernel/drivers/block/buddha.c"	4058	3	0	835989	1408715
"/kernel/drivers/block/cciss.c"	59286	33	18	20649370	65928801
"/kernel/drivers/block/cmd640.c"	23552	14	0	3979909	4256904
"/kernel/drivers/block/cmd646.c"	7174	10	3	2753589	5912278
"/kernel/drivers/block/cpqarray.c"	50579	38	21	20745018	59639387
"/kernel/drivers/block/cs5530.c"	14330	5	0	2196158	3122019
"/kernel/drivers/block/DAC960.c"	267391	96	42	100840646	560832755
"/kernel/drivers/block/dtc2278.c"	3491	3	0	571136	1092618
"/kernel/drivers/block/falconide.c"	1781	1	0	242303	731333
"/kernel/drivers/block/floppy.c"	114843	111	67	74587476	365325022
"/kernel/drivers/block/gayle.c"	4331	3	0	689554	1257262
"/kernel/drivers/block/genhd.c"	49455	10	4	4012510	12029816
"/kernel/drivers/block/hd.c"	21326	27	14	9844931	21767701
"/kernel/drivers/block/ht6560b.c"	7582	4	0	1375540	1687004
"/kernel/drivers/block/ide-cd.c"	84286	65	1	17573799	26442278
"/kernel/drivers/block/ide-disk.c"	27419	24	1	6763160	8827102
"/kernel/drivers/block/ide-dma.c"	16058	8	0	2307879	3242202
"/kernel/drivers/block/ide-floppy.c"	57203	38	2	13205187	19389748
"/kernel/drivers/block/ide-pci.c"	18506	5	0	2563012	4577952
"/kernel/drivers/block/ide-pmac.c"	29148	7	3	2862158	7113578
"/kernel/drivers/block/ide-probe.c"	26920	11	0	4665001	5687746
"/kernel/drivers/block/ide-proc.c"	19398	26	4	6432254	11146410
"/kernel/drivers/block/ide-tape.c"	140671	77	1	26796352	36248282
"/kernel/drivers/block/ide.c"	96165	87	6	25974137	50304257

”/kernel/drivers/block/linear.c”	4620	5	0	1303652	1690002
”/kernel/drivers/block/ll_rw_blk.c”	27006	24	7	8156710	18117770
”/kernel/drivers/block/loop.c”	18210	18	2	5035879	7593226
”/kernel/drivers/block/macide.c”	2982	2	0	428536	968757
”/kernel/drivers/block/md.c”	31609	31	2	8819543	13029878
”/kernel/drivers/block/nbd.c”	12130	10	4	3633556	6271478
”/kernel/drivers/block/ns87415.c”	5270	4	0	1156154	1824110
”/kernel/drivers/block/opti621.c”	10779	7	3	2176894	4340335
”/kernel/drivers/block/paride/aten.c”	3590	9	6	2386784	4733112
”/kernel/drivers/block/paride/bpck.c”	9390	14	6	3725286	10053707
”/kernel/drivers/block/paride/comm.c”	5149	9	6	2575021	5175882
”/kernel/drivers/block/paride/dstr.c”	5390	9	6	2604154	5248858
”/kernel/drivers/block/paride/epat.c”	6422	10	6	2877023	8327068
”/kernel/drivers/block/paride/epia.c”	8279	10	6	3284693	9069118
”/kernel/drivers/block/paride/fit2.c”	3501	9	6	2989868	4500666
”/kernel/drivers/block/paride/fit3.c”	4729	9	6	2447286	5035366
”/kernel/drivers/block/paride/friq.c”	6529	11	7	3213851	6479432
”/kernel/drivers/block/paride/frpw.c”	7675	12	7	3471166	7100504
”/kernel/drivers/block/paride/kbic.c”	7357	13	7	3892312	6939487
”/kernel/drivers/block/paride/ktti.c”	3029	9	6	17710620	6081149
”/kernel/drivers/block/paride/on20.c”	3343	9	6	2315626	4492362
”/kernel/drivers/block/paride/on26.c”	7753	10	6	3062088	6023004
”/kernel/drivers/block/paride/paride.c”	10735	19	13	5854877	12678884
”/kernel/drivers/block/paride/pcd.c”	24345	31	10	10112954	22597612
”/kernel/drivers/block/paride/pd.c”	31889	31	17	14494037	33687350
”/kernel/drivers/block/paride/pf.c”	29968	33	16	12947932	30731376
”/kernel/drivers/block/paride/pg.c”	16142	16	4	4886770	8252463
”/kernel/drivers/block/paride/pt.c”	24284	25	8	8740288	18435070
”/kernel/drivers/block/pdc4030.c”	11597	8	0	2083642	2932242
”/kernel/drivers/block/ps2esdi.c”	34191	0	0	57173	1163786
”/kernel/drivers/block/q40ide.c”	2539	3	0	606218	1099838
”/kernel/drivers/block/qd6580.c”	1724	2	0	285512	766305
”/kernel/drivers/block/raid0.c”	7337	6	0	1876440	2302448
”/kernel/drivers/block/raid1.c”	24047	19	2	6307392	9141984
”/kernel/drivers/block/raid5.c”	47637	56	24	22227060	66048316
”/kernel/drivers/block/rd.c”	19749	18	10	7504789	18149713
”/kernel/drivers/block/rz1000.c”	2790	2	0	421265	958038
”/kernel/drivers/block/sl82c105.c”	984	1	1	378183	1057856
”/kernel/drivers/block/swim_iop.c”	16111	21	10	8118476	20335660
”/kernel/drivers/block/swim3.c”	29399	29	15	13491468	35199578
”/kernel/drivers/block/trm290.c”	9427	4	0	1068152	1759405
”/kernel/drivers/block/umc8672.c”	4666	5	0	1023967	1581954
”/kernel/drivers/block/via82c586.c”	2371	2	0	401752	894953
”/kernel/drivers/block/xd.c”	37923	36	2	10590722	16679910
”/kernel/drivers/block/z2ram.c”	9482	6	3	2583622	4550822
”/kernel/drivers/cdrom/aztcd.c”	73277	35	16	58369600	67817696
”/kernel/drivers/cdrom/cdrom.c”	78857	40	4	17352123	29491405
”/kernel/drivers/cdrom/cdu31a.c”	101368	63	19	33942319	102892360
”/kernel/drivers/cdrom/cm206.c”	46092	58	22	18043967	55278439
”/kernel/drivers/cdrom/gscd.c”	24221	29	0	8415883	9248568
”/kernel/drivers/cdrom/isp16.c”	9498	6	0	2085570	2871394
”/kernel/drivers/cdrom/mcd.c”	38729	29	12	13957435	30668342
”/kernel/drivers/cdrom/mcdx.c”	51446	37	6	15497650	29639358
”/kernel/drivers/cdrom/optcd.c”	51358	49	13	52880019	42916374
”/kernel/drivers/cdrom/sbpcd.c”	163336	86	19	46543692	164793258
”/kernel/drivers/cdrom/sbpcd2.c”	98	0	0	16162	459842
”/kernel/drivers/cdrom/sbpcd3.c”	98	0	0	15696	264376
”/kernel/drivers/cdrom/sbpcd4.c”	98	0	0	14998	281474
”/kernel/drivers/cdrom/sjcd.c”	43531	33	12	15506132	36673675
”/kernel/drivers/cdrom/sonycd535.c”	49328	29	9	14499978	32779932
”/kernel/drivers/char/acquirewdt.c”	4153	8	1	1512492	2455319
”/kernel/drivers/char/adbmouse.c”	7407	9	2	2716804	4019666
”/kernel/drivers/char/advantechwdt.c”	4926	8	1	1615860	2750480
”/kernel/drivers/char/agp/agpgart_be.c”	82692	24	10	12326635	43746296
”/kernel/drivers/char/agp/agpgart_fe.c”	24057	45	11	10903597	21672996
”/kernel/drivers/char/amiga_ser.c”	13235	21	14	7674153	22131692
”/kernel/drivers/char/amigamouse.c”	7119	8	1	2150423	5657482
”/kernel/drivers/char/amikeyb.c”	13403	4	2	2868399	5188407
”/kernel/drivers/char/atari_MFPser.c”	27513	25	14	11344479	29484343
”/kernel/drivers/char/atari_MIDI.c”	14416	15	9	6867366	14495382
”/kernel/drivers/char/atari_SCC.c”	74765	19	10	14698389	42031894
”/kernel/drivers/char/atarimouse.c”	4788	9	2	2062443	3379292
”/kernel/drivers/char/atixlmouse.c”	6185	8	1	1758269	2798838
”/kernel/drivers/char/bttv.c”	104850	72	39	45108952	192024776
”/kernel/drivers/char/busmouse.c”	6567	9	2	2392852	3515396
”/kernel/drivers/char/buz.c”	88737	58	29	34906888	127415381
”/kernel/drivers/char/bw-qcam.c”	25315	26	6	7307719	13848536
”/kernel/drivers/char/c-qcam.c”	18524	24	5	6106821	10573710
”/kernel/drivers/char/conmakehash.c”	6121	4	2	1658636	3013646
”/kernel/drivers/char/console.c”	67443	93	73	54440956	445249566
”/kernel/drivers/char/consolemap.c”	21517	20	5	6507261	11051024
”/kernel/drivers/char/cpia_pp.c”	46117	18	5	6707606	18146144
”/kernel/drivers/char/cpia_usb.c”	15456	17	6	6018569	9956614
”/kernel/drivers/char/cpia.c”	103381	37	10	21803450	62291502
”/kernel/drivers/char/cyclades.c”	191258	58	25	53677971	239636984
”/kernel/drivers/char/defkeymap.c”	11111	0	0	1965839	2765099
”/kernel/drivers/char/dn_keyb.c”	23085	12	4	5691517	10141562
”/kernel/drivers/char/drm/agpsupport.c”	12137	11	1	3024096	4440556
”/kernel/drivers/char/drm/auth.c”	4439	6	0	1018238	1577222
”/kernel/drivers/char/drm/bufs.c”	14544	7	0	3360685	3572140
”/kernel/drivers/char/drm/context.c”	8847	9	0	1930314	2645252

"/kernel/drivers/char/drm/ctxbitmap.c"	2603	4	2	888840	1935364
"/kernel/drivers/char/drm/dma.c"	13990	11	6	3815879	10455973
"/kernel/drivers/char/drm/drawable.c"	1895	2	0	270068	778982
"/kernel/drivers/char/drm/drm_syms.c"	4105	0	0	3906000	4358460
"/kernel/drivers/char/drm/ffb_context.c"	17472	12	3	4485840	7615061
"/kernel/drivers/char/drm/ffb_drv.c"	22857	16	4	8032888	12085924
"/kernel/drivers/char/drm/fops.c"	7179	7	0	1449078	2113276
"/kernel/drivers/char/drm/gamma_dma.c"	21144	17	6	5886662	11577164
"/kernel/drivers/char/drm/gamma_drv.c"	15650	11	1	3764659	5468946
"/kernel/drivers/char/drm/i810_bufs.c"	9176	5	0	1656582	2378580
"/kernel/drivers/char/drm/i810_context.c"	6203	10	0	1667504	2334864
"/kernel/drivers/char/drm/i810_dma.c"	37691	41	11	13216948	28125082
"/kernel/drivers/char/drm/i810_drv.c"	18807	9	1	4071866	6078209
"/kernel/drivers/char/drm/init.c"	3651	3	2	847329	1946734
"/kernel/drivers/char/drm/ioctl.c"	3229	3	0	575150	2053710
"/kernel/drivers/char/drm/lists.c"	6127	9	0	1882403	2156481
"/kernel/drivers/char/drm/lock.c"	6070	10	0	1980150	4677055
"/kernel/drivers/char/drm/memory.c"	12316	12	5	3648836	6662490
"/kernel/drivers/char/drm/mga_bufs.c"	17541	7	0	3154114	4209891
"/kernel/drivers/char/drm/mga_context.c"	6264	10	0	1661118	2387211
"/kernel/drivers/char/drm/mga_dma.c"	30159	26	9	9578156	22135952
"/kernel/drivers/char/drm/mga_drv.c"	19404	9	1	4311658	6302935
"/kernel/drivers/char/drm/mga_state.c"	29386	24	9	9189512	21825568
"/kernel/drivers/char/drm/proc.c"	17141	11	0	2700712	3734516
"/kernel/drivers/char/drm/r128_bufs.c"	8826	2	0	683098	1387934
"/kernel/drivers/char/drm/r128_context.c"	6189	10	0	1931027	2357889
"/kernel/drivers/char/drm/r128_dma.c"	25627	26	2	6731640	11006424
"/kernel/drivers/char/drm/r128_drv.c"	21422	10	1	4564427	6727885
"/kernel/drivers/char/drm/tdfx_context.c"	6249	10	0	1713511	2384738
"/kernel/drivers/char/drm/tdfx_drv.c"	20043	10	1	4360764	6377276
"/kernel/drivers/char/drm/vm.c"	10840	8	2	2782203	4471968
"/kernel/drivers/char/dsp56k.c"	13758	8	0	2442043	3305607
"/kernel/drivers/char/dtlk.c"	17162	16	3	41894344	10732507
"/kernel/drivers/char/dz.c"	40946	36	1	8929996	14484533
"/kernel/drivers/char/epca.c"	126090	71	58	66042178	342430876
"/kernel/drivers/char/esp.c"	73750	50	26	29029760	103463106
"/kernel/drivers/char/ftape/compressor/lzrw3.c"	41547	3	1	4666610	8217654
"/kernel/drivers/char/ftape/compressor/zftape-compress.c"	40943	19	4	9167320	16554923
"/kernel/drivers/char/ftape/lowlevel/fc-10.c"	5550	1	0	658004	1258228
"/kernel/drivers/char/ftape/lowlevel/fdc-io.c"	39255	42	12	14528948	33698942
"/kernel/drivers/char/ftape/lowlevel/fdc-isr.c"	32208	18	9	9583408	20735118
"/kernel/drivers/char/ftape/lowlevel/ftape_syms.c"	3131	0	0	1114550	1624648
"/kernel/drivers/char/ftape/lowlevel/ftape-bsm.c"	13046	13	6	4331139	8022555
"/kernel/drivers/char/ftape/lowlevel/ftape-buffer.c"	3503	6	2	1264628	2333178
"/kernel/drivers/char/ftape/lowlevel/ftape-calibr.c"	7752	8	3	2093092	3899510
"/kernel/drivers/char/ftape/lowlevel/ftape-ctl.c"	25904	21	7	7958073	15113160
"/kernel/drivers/char/ftape/lowlevel/ftape-ecc.c"	26595	15	0	4823172	5614295
"/kernel/drivers/char/ftape/lowlevel/ftape-format.c"	9942	4	0	2049988	2427034
"/kernel/drivers/char/ftape/lowlevel/ftape-init.c"	5845	1	0	540888	1104226
"/kernel/drivers/char/ftape/lowlevel/ftape-io.c"	27312	28	4	7797990	15417274
"/kernel/drivers/char/ftape/lowlevel/ftape-proc.c"	11582	0	0	26820	694459
"/kernel/drivers/char/ftape/lowlevel/ftape-read.c"	18568	6	1	3518076	5315596
"/kernel/drivers/char/ftape/lowlevel/ftape-rw.c"	31666	21	1	7354073	10130594
"/kernel/drivers/char/ftape/lowlevel/ftape-setup.c"	3139	1	1	592706	1388486
"/kernel/drivers/char/ftape/lowlevel/ftape-tracing.c"	3481	3	3	1421718	2614174
"/kernel/drivers/char/ftape/lowlevel/ftape-write.c"	9857	6	1	2058769	3333860
"/kernel/drivers/char/ftape/zftape/zftape_syms.c"	1763	0	0	213324	713312
"/kernel/drivers/char/ftape/zftape/zftape-buffers.c"	3780	9	4	1890420	3510185
"/kernel/drivers/char/ftape/zftape/zftape-ctl.c"	42621	35	2	9137062	13559099
"/kernel/drivers/char/ftape/zftape/zftape-eof.c"	6199	4	2	1430793	5179928
"/kernel/drivers/char/ftape/zftape/zftape-init.c"	14119	10	0	3012932	3992878
"/kernel/drivers/char/ftape/zftape/zftape-read.c"	11495	7	1	2358572	3795534
"/kernel/drivers/char/ftape/zftape/zftape-rw.c"	9916	9	2	3477972	4657395
"/kernel/drivers/char/ftape/zftape/zftape-vtbl.c"	22777	21	9	7579082	15779197
"/kernel/drivers/char/ftape/zftape/zftape-write.c"	14847	9	1	3069439	4788808
"/kernel/drivers/char/generic_serial.c"	26082	20	9	7583756	18642780
"/kernel/drivers/char/h8.c"	32590	25	16	13929123	35311592
"/kernel/drivers/char/hfmodem/gentbl.c"	2258	2	1	550676	1214874
"/kernel/drivers/char/hfmodem/main.c"	23669	15	8	6417126	42750312
"/kernel/drivers/char/hfmodem/modem.c"	22535	25	7	7336741	13892305
"/kernel/drivers/char/hfmodem/refclock.c"	4739	3	2	1064139	9910593
"/kernel/drivers/char/hfmodem/sbc.c"	22244	25	17	10146733	29474280
"/kernel/drivers/char/hfmodem/wss.c"	14874	15	8	5306380	12295043
"/kernel/drivers/char/i2c.c"	9561	17	6	4157899	7484790
"/kernel/drivers/char/i810_rng.c"	7481	10	1	2624061	3488752
"/kernel/drivers/char/i810-tco.c"	7694	11	0	2353068	6765834
"/kernel/drivers/char/ip2.c"	3918	2	1	637284	1475807
"/kernel/drivers/char/ip2/i2cmd.c"	12639	4	0	3256036	3781254
"/kernel/drivers/char/ip2/i2ellis.c"	47172	31	11	11512342	41360302
"/kernel/drivers/char/ip2/i2lib.c"	66565	24	12	16017771	54414569
"/kernel/drivers/char/ip2/ip2mkdev.c"	2759	2	0	733220	3344193
"/kernel/drivers/char/ip2/ip2stat.c"	3834	1	0	974112	1313947
"/kernel/drivers/char/ip2/ip2trace.c"	7737	2	1	1726231	2941973
"/kernel/drivers/char/ip2main.c"	116812	42	14	66798438	99563390
"/kernel/drivers/char/isicom.c"	57977	48	25	25240254	85424714
"/kernel/drivers/char/istallion.c"	145346	98	49	70418402	330646248
"/kernel/drivers/char/joystick/joy-amiga.c"	3645	3	0	777778	1330320
"/kernel/drivers/char/joystick/joy-analog.c"	7275	3	0	1252822	1879008
"/kernel/drivers/char/joystick/joy-assassin.c"	10872	5	0	1963650	5039302
"/kernel/drivers/char/joystick/joy-console.c"	21315	8	3	4563932	7630258
"/kernel/drivers/char/joystick/joy-creative.c"	6310	4	0	1319423	1737617

"/kernel/drivers/char/joystick/joy-db9.c"	10807	3	0	2021763	2762543
"/kernel/drivers/char/joystick/joy-gravis.c"	9122	5	0	1564970	2288611
"/kernel/drivers/char/joystick/joy-lightning.c"	8832	7	1	2311550	3274386
"/kernel/drivers/char/joystick/joy-logitech.c"	13874	6	2	2664654	4674047
"/kernel/drivers/char/joystick/joy-magellan.c"	9796	12	4	3052492	5539974
"/kernel/drivers/char/joystick/joy-pci.c"	7821	7	4	2579525	4898576
"/kernel/drivers/char/joystick/joy-sidewinder.c"	22041	12	3	4565201	8102493
"/kernel/drivers/char/joystick/joy-spaceball.c"	8996	7	3	2413355	4444405
"/kernel/drivers/char/joystick/joy-spaceorb.c"	7606	7	3	2611182	4121183
"/kernel/drivers/char/joystick/joy-thrustmaster.c"	6964	4	0	1269103	1930037
"/kernel/drivers/char/joystick/joy-turbografx.c"	6456	3	0	1405126	2034006
"/kernel/drivers/char/joystick/joy-warrior.c"	7755	7	3	2251762	4169220
"/kernel/drivers/char/joystick/joystick.c"	20334	16	6	6055010	13250518
"/kernel/drivers/char/keyboard.c"	21262	49	41	19893034	62472304
"/kernel/drivers/char/lp_intern.c"	7665	8	2	2331700	3816660
"/kernel/drivers/char/lp_m68k.c"	13243	10	2	2859853	4835830
"/kernel/drivers/char/lp_mfc.c"	3873	9	3	2205620	3603116
"/kernel/drivers/char/lp.c"	25618	17	4	5443573	12266158
"/kernel/drivers/char/m68kserial.c"	51015	29	16	14755622	43262375
"/kernel/drivers/char/mac_SCC.c"	42272	40	22	18340704	57649996
"/kernel/drivers/char/machzwd.c"	10899	17	6	4665494	8190165
"/kernel/drivers/char/mem.c"	14680	21	0	4125144	5199942
"/kernel/drivers/char/misc.c"	7001	4	0	2432590	3134016
"/kernel/drivers/char/mixcomwd.c"	5706	9	3	2216866	3820608
"/kernel/drivers/char/moxa.c"	89004	65	30	35564764	131626931
"/kernel/drivers/char/msbusmouse.c"	5915	9	2	2042405	3449890
"/kernel/drivers/char/msp3400.c"	33591	19	10	8974614	23543068
"/kernel/drivers/char/mxser.c"	63162	39	16	22780209	63786708
"/kernel/drivers/char/n_hdlc.c"	28810	15	3	6239485	12979470
"/kernel/drivers/char/n_tty.c"	29504	27	14	11410531	28038918
"/kernel/drivers/char/nvram.c"	17089	11	2	2576989	4636593
"/kernel/drivers/char/pc_keyb.c"	25537	13	6	5510595	13959894
"/kernel/drivers/char/pc110pad.c"	14899	21	13	7841102	18709015
"/kernel/drivers/char/pcwd.c"	14954	14	4	4200260	7353789
"/kernel/drivers/char/pcxx.c"	57728	43	31	28342403	89388206
"/kernel/drivers/char/planb.c"	63797	41	21	25004087	75817214
"/kernel/drivers/char/pms.c"	19987	29	17	11334201	24035430
"/kernel/drivers/char/pty.c"	13357	11	4	3871081	6850747
"/kernel/drivers/char/q40_keyb.c"	15092	7	2	3088095	5243100
"/kernel/drivers/char/qpmouse.c"	8422	11	1	2591384	3871762
"/kernel/drivers/char/radio-aimslab.c"	8894	13	7	3689039	7334234
"/kernel/drivers/char/radio-aztech.c"	7598	11	1	2676929	3537204
"/kernel/drivers/char/radio-cadet.c"	13366	16	6	5235663	9196652
"/kernel/drivers/char/radio-gemtek.c"	6501	10	5	2621888	5105414
"/kernel/drivers/char/radio-maestro.c"	9350	12	5	3505447	6180127
"/kernel/drivers/char/radio-miropcm20.c"	4941	8	2	2261473	3187632
"/kernel/drivers/char/radio-rtrack2.c"	5407	10	5	2513792	6706494
"/kernel/drivers/char/radio-sf16fmi.c"	7222	9	4	2591310	4708336
"/kernel/drivers/char/radio-trust.c"	7246	13	8	4308688	8829758
"/kernel/drivers/char/radio-typhoon.c"	11614	10	4	3781818	8480200
"/kernel/drivers/char/radio-zoltrix.c"	8268	11	4	2889064	5181686
"/kernel/drivers/char/random.c"	61490	24	12	12767062	38335796
"/kernel/drivers/char/rio/rio_linux.c"	44169	33	0	9825428	14241154
"/kernel/drivers/char/rio/rioboot.c"	41196	7	1	7467838	10712320
"/kernel/drivers/char/rio/riocmd.c"	31641	20	4	7461181	12975290
"/kernel/drivers/char/rio/rioctrl.c"	60181	4	0	11646059	14371190
"/kernel/drivers/char/rio/rioinit.c"	43402	10	3	4276757	10358982
"/kernel/drivers/char/rio/riointr.c"	31905	6	5	5400703	12379186
"/kernel/drivers/char/rio/rioparam.c"	21809	6	3	3852845	14362642
"/kernel/drivers/char/rio/riopcicopy.c"	149	1	1	297912	964703
"/kernel/drivers/char/rio/rioroute.c"	35767	11	2	7397392	11175722
"/kernel/drivers/char/rio/riotable.c"	31170	6	0	5244947	6704458
"/kernel/drivers/char/rio/riotty.c"	37895	5	1	4233110	7305148
"/kernel/drivers/char/riscom8.c"	49311	53	30	25072723	78664554
"/kernel/drivers/char/rocket.c"	91723	46	22	26951300	101238964
"/kernel/drivers/char/rtc.c"	21114	15	6	6764419	13203600
"/kernel/drivers/char/saa5249.c"	19432	13	2	4302823	6896870
"/kernel/drivers/char/saa7111.c"	10184	7	0	2272639	5384411
"/kernel/drivers/char/saa7185.c"	8668	6	0	1765156	4930456
"/kernel/drivers/char/sbc60xxwdt.c"	7717	9	3	2255436	4023404
"/kernel/drivers/char/selection.c"	8113	10	3	2902519	4700582
"/kernel/drivers/char/ser_hpdca.c"	31240	16	7	7892909	16559164
"/kernel/drivers/char/ser_hypercom1.c"	19172	17	10	7268988	19094782
"/kernel/drivers/char/ser_mfc.c"	19003	16	8	7117444	13577342
"/kernel/drivers/char/ser_whippet.c"	21119	16	8	7555594	14153794
"/kernel/drivers/char/serial.c"	93765	51	25	31020190	113734052
"/kernel/drivers/char/serial167.c"	82366	50	24	28525089	97080539
"/kernel/drivers/char/softdog.c"	4183	7	3	1748035	3239806
"/kernel/drivers/char/specialix.c"	63342	60	30	28498192	101822072
"/kernel/drivers/char/stallion.c"	141599	99	57	78697369	373622343
"/kernel/drivers/char/sx.c"	77716	51	2	17229596	24957540
"/kernel/drivers/char/synclink.c"	237623	116	66	113323850	740068993
"/kernel/drivers/char/sysrq.c"	6292	6	4	2324691	4323300
"/kernel/drivers/char/toshiba.c"	13022	10	1	2948572	4472576
"/kernel/drivers/char/tpqic02.c"	87346	38	13	22865010	64694523
"/kernel/drivers/char/tty_io.c"	56878	55	16	20459215	57591863
"/kernel/drivers/char/tty_ioctl.c"	13291	8	3	2929154	5135354
"/kernel/drivers/char/tuner.c"	7073	10	3	2565580	4318098
"/kernel/drivers/char/vc_screen.c"	7484	6	0	2227590	2290759
"/kernel/drivers/char/videodev.c"	9432	10	1	2032804	3305906
"/kernel/drivers/char/vino.c"	5274	14	4	3184277	4730104

"/kernel/drivers/char/vt.c"	31104	11	4	7189996	14696232
"/kernel/drivers/char/wdt_pci.c"	12971	12	4	3525952	6515501
"/kernel/drivers/char/wdt.c"	7668	14	5	3257794	8112688
"/kernel/drivers/dio/dio.c"	9793	9	3	2570183	4595146
"/kernel/drivers/fc4/fc_syms.c"	804	0	0	16918	478245
"/kernel/drivers/fc4/fc.c"	32366	27	11	10924475	28207480
"/kernel/drivers/fc4/soc.c"	21503	12	7	5994784	14186511
"/kernel/drivers/fc4/socal.c"	24835	13	8	7377741	15088793
"/kernel/drivers/i2o/i2o_block.c"	41222	25	11	13509586	31526527
"/kernel/drivers/i2o/i2o_config.c"	21384	19	1	6322556	8221750
"/kernel/drivers/i2o/i2o_core.c"	70921	57	11	23696373	50674870
"/kernel/drivers/i2o/i2o_pci.c"	7591	8	4	2272821	4511888
"/kernel/drivers/i2o/i2o_scsi.c"	21174	15	6	7218993	12215554
"/kernel/drivers/isdn/act2000/act2000_isa.c"	13439	15	7	5643000	9534321
"/kernel/drivers/isdn/act2000/capi.c"	32843	27	14	10902621	28246099
"/kernel/drivers/isdn/act2000/module.c"	22472	21	7	7695802	16661498
"/kernel/drivers/isdn/avmb1/b1.c"	17396	14	7	6100833	11736115
"/kernel/drivers/isdn/avmb1/b1dma.c"	24410	31	14	10979399	25239250
"/kernel/drivers/isdn/avmb1/b1isa.c"	6341	5	3	2094762	3789751
"/kernel/drivers/isdn/avmb1/b1pci.c"	13940	6	3	2659662	5137440
"/kernel/drivers/isdn/avmb1/b1pcmcia.c"	7045	9	3	4047386	6727366
"/kernel/drivers/isdn/avmb1/c4.c"	35028	32	15	12977179	31751613
"/kernel/drivers/isdn/avmb1/capi.c"	45414	19	6	7617212	18114062
"/kernel/drivers/isdn/avmb1/capidrv.c"	64874	51	20	26686027	79548414
"/kernel/drivers/isdn/avmb1/capifs.c"	11585	13	6	5033733	8706375
"/kernel/drivers/isdn/avmb1/capiutil.c"	26571	16	8	6640738	15349368
"/kernel/drivers/isdn/avmb1/kcapi.c"	41449	56	19	19516687	50176269
"/kernel/drivers/isdn/avmb1/t1isa.c"	14804	10	6	4859502	11018685
"/kernel/drivers/isdn/avmb1/t1pci.c"	7809	4	2	2268792	3573021
"/kernel/drivers/isdn/divert/divert_init.c"	2485	1	1	825005	1606162
"/kernel/drivers/isdn/divert/divert_procfs.c"	8793	9	1	2120017	3337817
"/kernel/drivers/isdn/divert/isdn_divert.c"	26155	13	2	5754852	9172684
"/kernel/drivers/isdn/eicon/bri.c"	18059	11	2	5011611	7330802
"/kernel/drivers/isdn/eicon/common.c"	16027	34	20	10289082	27997142
"/kernel/drivers/isdn/eicon/Divas_mod.c"	2598	2	1	1160630	1786820
"/kernel/drivers/isdn/eicon/eicon_idi.c"	88137	24	7	15391460	39841347
"/kernel/drivers/isdn/eicon/eicon_io.c"	20366	9	7	3978080	10105824
"/kernel/drivers/isdn/eicon/eicon_isa.c"	12171	0	0	76387	758308
"/kernel/drivers/isdn/eicon/eicon_mod.c"	45293	22	10	12509415	28595128
"/kernel/drivers/isdn/eicon/eicon_pci.c"	2483	0	0	67549	577712
"/kernel/drivers/isdn/eicon/fourbri.c"	12534	12	3	4074036	6360954
"/kernel/drivers/isdn/eicon/fpga.c"	4441	1	0	754132	1329222
"/kernel/drivers/isdn/eicon/idi.c"	19707	35	24	12317544	38003222
"/kernel/drivers/isdn/eicon/kprintf.c"	10451	7	2	2612570	4394720
"/kernel/drivers/isdn/eicon/lincfg.c"	9759	3	0	2081727	2870432
"/kernel/drivers/isdn/eicon/linchr.c"	5325	6	1	1766088	2787145
"/kernel/drivers/isdn/eicon/linio.c"	14702	39	23	12384762	28164816
"/kernel/drivers/isdn/eicon/linsys.c"	3099	6	1	1127024	1990987
"/kernel/drivers/isdn/eicon/log.c"	2733	5	2	1252692	2361916
"/kernel/drivers/isdn/eicon/pri.c"	10847	10	2	3243470	5069045
"/kernel/drivers/isdn/eicon/xlog.c"	2946	3	2	948881	1971762
"/kernel/drivers/isdn/hisax/amd7930.c"	21682	24	18	9406232	30011540
"/kernel/drivers/isdn/hisax/arcofi.c"	3685	6	5	1882004	3806574
"/kernel/drivers/isdn/hisax/asuscom.c"	9969	20	14	7352065	18648360
"/kernel/drivers/isdn/hisax/avm_a1.c"	8837	14	9	5088321	11299834
"/kernel/drivers/isdn/hisax/avm_a1p.c"	7503	11	7	5080001	6849542
"/kernel/drivers/isdn/hisax/avm_pci.c"	22490	29	17	11565395	26346680
"/kernel/drivers/isdn/hisax/bkm_a4t.c"	9373	16	11	5766296	13461310
"/kernel/drivers/isdn/hisax/bkm_a8.c"	12615	1	0	1093472	1853558
"/kernel/drivers/isdn/hisax/callc.c"	49940	69	58	38399944	153994534
"/kernel/drivers/isdn/hisax/cert.c"	1393	1	0	250866	761474
"/kernel/drivers/isdn/hisax/config.c"	38242	21	10	9768288	22840988
"/kernel/drivers/isdn/hisax/diva.c"	26743	37	28	16667812	49865244
"/kernel/drivers/isdn/hisax/elsa_ser.c"	17369	20	13	8548792	20208680
"/kernel/drivers/isdn/hisax/elsa.c"	32665	26	16	12908002	34267741
"/kernel/drivers/isdn/hisax/fsm.c"	3898	9	5	2607141	4253150
"/kernel/drivers/isdn/hisax/gazel.c"	16303	25	15	9235612	20182824
"/kernel/drivers/isdn/hisax/hfc_2bds0.c"	28621	34	19	13436618	36288721
"/kernel/drivers/isdn/hisax/hfc_2bs0.c"	15884	17	8	5833033	11710538
"/kernel/drivers/isdn/hisax/hfc_pci.c"	54350	1	0	1209331	2947466
"/kernel/drivers/isdn/hisax/hfc_sx.c"	44039	34	23	18428690	56494682
"/kernel/drivers/isdn/hisax/hfcscard.c"	5744	6	4	2269885	6248064
"/kernel/drivers/isdn/hisax/hscx_irq.c"	7319	7	7	3060300	6435278
"/kernel/drivers/isdn/hisax/hscx.c"	7562	10	4	3396934	5227768
"/kernel/drivers/isdn/hisax/icc.c"	18166	14	12	6986093	16533876
"/kernel/drivers/isdn/hisax/isac.c"	18360	14	14	7639916	21066950
"/kernel/drivers/isdn/hisax/isar.c"	47924	35	24	21634637	65083338
"/kernel/drivers/isdn/hisax/isdnl1.c"	20903	36	32	15226217	46259707
"/kernel/drivers/isdn/hisax/isdnl2.c"	42568	107	82	53474956	414026751
"/kernel/drivers/isdn/hisax/isdnl3.c"	13675	32	23	11087007	26090272
"/kernel/drivers/isdn/hisax/isurf.c"	6219	12	7	3481973	6662314
"/kernel/drivers/isdn/hisax/ix1_micro.c"	6860	15	10	4755643	9164956
"/kernel/drivers/isdn/hisax/jade_irq.c"	6423	7	7	2953243	6193002
"/kernel/drivers/isdn/hisax/jade.c"	8693	10	5	3339366	6104278
"/kernel/drivers/isdn/hisax/l3_1tr6.c"	22635	33	33	17294656	53690338
"/kernel/drivers/isdn/hisax/l3dss1.c"	83007	84	62	58199463	258653714
"/kernel/drivers/isdn/hisax/l3ni1.c"	81005	89	67	60606895	274382490
"/kernel/drivers/isdn/hisax/lmgr.c"	1038	3	3	856559	1969218
"/kernel/drivers/isdn/hisax/mic.c"	6052	14	9	4093768	7958017
"/kernel/drivers/isdn/hisax/netjet.c"	27294	22	16	11055445	29272274
"/kernel/drivers/isdn/hisax/niccy.c"	9786	15	10	5340236	10634001

"/kernel/drivers/isdn/hisax/nj_s.c"	7244	6	3	2368362	4188168
"/kernel/drivers/isdn/hisax/nj_u.c"	7358	6	3	2374224	6337564
"/kernel/drivers/isdn/hisax/q931.c"	32140	21	3	7860741	16699368
"/kernel/drivers/isdn/hisax/rawhdlc.c"	16545	3	1	1936338	3396622
"/kernel/drivers/isdn/hisax/s0box.c"	7235	14	9	4361130	10523616
"/kernel/drivers/isdn/hisax/saphir.c"	7827	16	10	4932505	9663868
"/kernel/drivers/isdn/hisax/sedlbauer.c"	22652	23	16	10131560	25841227
"/kernel/drivers/isdn/hisax/sportster.c"	6574	15	9	4276926	8340289
"/kernel/drivers/isdn/hisax/tei.c"	11325	20	17	7796383	17187464
"/kernel/drivers/isdn/hisax/teleint.c"	8037	16	11	5401563	10522962
"/kernel/drivers/isdn/hisax/teles0.c"	8722	19	12	6056542	11776900
"/kernel/drivers/isdn/hisax/teles3.c"	11866	16	10	5845491	11510789
"/kernel/drivers/isdn/hisax/telespci.c"	9324	18	12	5645538	12112376
"/kernel/drivers/isdn/hisax/w6692.c"	27934	29	23	16464364	41215059
"/kernel/drivers/isdn/hysdn/boardergo.c"	15795	10	5	3758236	7788978
"/kernel/drivers/isdn/hysdn/hycapi.c"	23376	22	11	7988231	20302221
"/kernel/drivers/isdn/hysdn/hysdn_boot.c"	13907	7	2	2618178	4618794
"/kernel/drivers/isdn/hysdn/hysdn_init.c"	8123	6	4	2385410	4582042
"/kernel/drivers/isdn/hysdn/hysdn_net.c"	10340	11	2	3155361	4153140
"/kernel/drivers/isdn/hysdn/hysdn_procconf.c"	14630	8	1	2948408	4538742
"/kernel/drivers/isdn/hysdn/hysdn_proclog.c"	13215	11	4	3429694	6515480
"/kernel/drivers/isdn/hysdn/hysdn_sched.c"	7186	3	0	1035463	1691824
"/kernel/drivers/isdn/icn/icn.c"	42616	34	17	16184802	41941314
"/kernel/drivers/isdn/isdn_audio.c"	20702	18	9	7442250	15089338
"/kernel/drivers/isdn/isdn_bsdcomp.c"	24271	13	2	4956896	8095632
"/kernel/drivers/isdn/isdn_common.c"	57337	39	13	18626066	48535798
"/kernel/drivers/isdn/isdn_concap.c"	3267	4	0	744992	1292770
"/kernel/drivers/isdn/isdn_net.c"	86478	62	28	35139752	123560379
"/kernel/drivers/isdn/isdn_ppp.c"	74490	44	10	19771185	48191084
"/kernel/drivers/isdn/isdn_tty.c"	101016	61	37	39676180	177721375
"/kernel/drivers/isdn/isdn_ttyfax.c"	25790	8	2	5277146	8711663
"/kernel/drivers/isdn/isdn_v110.c"	16676	12	2	4228121	6623118
"/kernel/drivers/isdn/isdn_x25iface.c"	10324	10	2	3029361	4773566
"/kernel/drivers/isdn/isdnloop/isdnloop.c"	39603	34	17	15967955	43839144
"/kernel/drivers/isdn/pcbit/callbacks.c"	8345	12	0	2947996	3357789
"/kernel/drivers/isdn/pcbit/capi.c"	15373	18	0	3789272	8840589
"/kernel/drivers/isdn/pcbit/drv.c"	22711	15	5	6693039	11725326
"/kernel/drivers/isdn/pcbit/edss1.c"	9787	3	1	1641161	2845924
"/kernel/drivers/isdn/pcbit/layer2.c"	14613	13	12	6249859	16167593
"/kernel/drivers/isdn/pcbit/module.c"	2391	2	2	1456494	2294862
"/kernel/drivers/isdn/sc/command.c"	13162	22	0	4649842	5535386
"/kernel/drivers/isdn/sc/debug.c"	1201	4	3	1044489	2133178
"/kernel/drivers/isdn/sc/event.c"	1769	1	0	300540	791252
"/kernel/drivers/isdn/sc/init.c"	14034	3	1	3502021	4762880
"/kernel/drivers/isdn/sc/interrupt.c"	6689	2	1	1718102	2728894
"/kernel/drivers/isdn/sc/ioctl.c"	14266	2	0	2624081	4500958
"/kernel/drivers/isdn/sc/message.c"	5982	3	0	1335342	1946478
"/kernel/drivers/isdn/sc/packet.c"	7279	4	1	2183637	2983533
"/kernel/drivers/isdn/sc/shmem.c"	3867	3	0	864430	1411616
"/kernel/drivers/isdn/sc/timer.c"	4725	4	4	2036996	3290060
"/kernel/drivers/macintosh/adb.c"	14201	16	7	6440095	10113387
"/kernel/drivers/macintosh/adbhid.c"	24632	17	14	10825983	24286646
"/kernel/drivers/macintosh/mac_hid.c"	17854	2	0	967649	1836305
"/kernel/drivers/macintosh/mac_keyb.c"	32763	19	14	11190520	29130635
"/kernel/drivers/macintosh/macio-adb.c"	5486	7	4	2549823	4475854
"/kernel/drivers/macintosh/macserial.c"	82133	60	34	34802341	132778248
"/kernel/drivers/macintosh/mediabay.c"	16855	10	5	4383670	8498882
"/kernel/drivers/macintosh/nvram.c"	2886	7	0	1155035	1704428
"/kernel/drivers/macintosh/rtc.c"	3207	7	2	1820174	2589999
"/kernel/drivers/macintosh/via-cuda.c"	12236	13	7	5146210	9342353
"/kernel/drivers/macintosh/via-pmu.c"	48396	32	3	12497840	21381223
"/kernel/drivers/misc/parport_amiga.c"	7582	24	14	6107700	15093007
"/kernel/drivers/misc/parport_arc.c"	3214	7	5	1873695	3746206
"/kernel/drivers/misc/parport_atari.c"	5506	17	11	4992360	9622253
"/kernel/drivers/misc/parport_ax.c"	13574	41	20	10672912	26263430
"/kernel/drivers/misc/parport_ieee1284.c"	2384	2	0	429872	948440
"/kernel/drivers/misc/parport_init.c"	3893	4	3	2127034	3512931
"/kernel/drivers/misc/parport_mfc3.c"	10402	25	14	6605716	14597873
"/kernel/drivers/misc/parport_pc.c"	26844	50	20	13996726	37815292
"/kernel/drivers/misc/parport_procfs.c"	8821	4	1	592381	4613201
"/kernel/drivers/misc/parport_share.c"	13438	10	6	4305629	10630993
"/kernel/drivers/net/3c501.c"	20977	10	4	6022629	8775168
"/kernel/drivers/net/3c503.c"	21668	10	4	5100596	9490298
"/kernel/drivers/net/3c505.c"	49002	31	11	13615259	32748635
"/kernel/drivers/net/3c507.c"	28759	11	5	6330082	12656381
"/kernel/drivers/net/3c509.c"	28234	8	3	5734342	10360320
"/kernel/drivers/net/3c515.c"	48706	10	3	9731257	20984646
"/kernel/drivers/net/3c523.c"	38904	17	8	10160551	21277332
"/kernel/drivers/net/3c527.c"	43496	25	13	11749677	34559790
"/kernel/drivers/net/3c59x.c"	82445	25	13	24020864	65452822
"/kernel/drivers/net/7990.c"	22401	13	0	4353933	5611463
"/kernel/drivers/net/8139too.c"	68704	36	3	16242756	28049622
"/kernel/drivers/net/82596.c"	40687	21	12	13322248	30206360
"/kernel/drivers/net/8390.c"	34860	16	9	8838834	22783060
"/kernel/drivers/net/a2065.c"	21272	14	0	4170389	7445691
"/kernel/drivers/net/ac3200.c"	12185	8	2	2944647	4815938
"/kernel/drivers/net/acenic.c"	78923	25	15	20099950	62578362
"/kernel/drivers/net/am79c961a.c"	16703	15	5	4586377	8688658
"/kernel/drivers/net/apne.c"	17757	10	4	5467702	8349441
"/kernel/drivers/net/arc-rimi.c"	22674	11	5	5933878	10564079
"/kernel/drivers/net/arcnet.c"	58555	19	8	11483118	29616439

"/kernel/drivers/net/ariadne.c"	25597	10	4	5881951	10671348
"/kernel/drivers/net/ariadne2.c"	12793	8	1	2924592	4400766
"/kernel/drivers/net/arlan-proc.c"	27038	1	0	370210	1399562
"/kernel/drivers/net/arlan.c"	58375	30	10	18344134	39979238
"/kernel/drivers/net/at1700.c"	27008	11	3	6492256	9797608
"/kernel/drivers/net/atari_bionet.c"	18281	13	4	4597957	8444669
"/kernel/drivers/net/atari_pamsnet.c"	24119	19	3	6462929	9906646
"/kernel/drivers/net/atarilance.c"	34840	13	3	7752784	12354150
"/kernel/drivers/net/atp.c"	24741	16	9	8663640	16570498
"/kernel/drivers/net/auto_irq.c"	2093	2	1	446902	1213319
"/kernel/drivers/net/bagetlance.c"	38601	14	3	8172556	13842664
"/kernel/drivers/net/bmac.c"	41280	44	23	19101648	55284762
"/kernel/drivers/net/bonding.c"	8802	10	1	2166016	3398642
"/kernel/drivers/net/bsd_comp.c"	30335	15	1	6631093	10812515
"/kernel/drivers/net/c101.c"	6162	9	6	2616008	5227420
"/kernel/drivers/net/com20020.c"	28812	17	7	8573293	18188499
"/kernel/drivers/net/com90io.c"	24747	15	5	6657504	12009858
"/kernel/drivers/net/com90xx.c"	32383	11	6	7586889	15488980
"/kernel/drivers/net/comx-hw-comx.c"	38026	25	7	11086546	23609384
"/kernel/drivers/net/comx-hw-locomx.c"	12424	13	2	3432170	7644789
"/kernel/drivers/net/comx-hw-mixcom.c"	24833	26	12	9705579	21849348
"/kernel/drivers/net/comx-proto-fr.c"	25785	25	5	7805912	13971010
"/kernel/drivers/net/comx-proto-lapb.c"	13466	20	6	5447656	9558330
"/kernel/drivers/net/comx-proto-ppp.c"	5939	10	2	2258297	3885285
"/kernel/drivers/net/comx.c"	32641	36	6	11521934	19693626
"/kernel/drivers/net/cops.c"	32042	18	6	8414281	16625533
"/kernel/drivers/net/cosa.c"	58309	49	12	20126891	50683822
"/kernel/drivers/net/cs89x0.c"	34666	24	5	9386529	16234645
"/kernel/drivers/net/daynaport.c"	25812	23	12	10790248	23090687
"/kernel/drivers/net/de4x5.c"	176537	115	43	75598432	361188804
"/kernel/drivers/net/de600.c"	22935	12	3	4780923	8194608
"/kernel/drivers/net/de620.c"	27816	20	8	8641954	19559426
"/kernel/drivers/net/declance.c"	31484	18	8	8138852	18101618
"/kernel/drivers/net/defxx.c"	111601	32	1	16808204	25177009
"/kernel/drivers/net/depca.c"	63484	26	9	15769230	36362364
"/kernel/drivers/net/dgrs_firmware.c"	359131	0	0	71860557	96707541
"/kernel/drivers/net/dgrs.c"	39619	17	3	10391687	13484357
"/kernel/drivers/net/dlci.c"	12582	16	1	3734832	5142538
"/kernel/drivers/net/dmfe.c"	57974	34	22	22964698	72046663
"/kernel/drivers/net/dummy.c"	3794	6	1	1084128	1968302
"/kernel/drivers/net/e2100.c"	14252	10	4	3561334	6729104
"/kernel/drivers/net/eepro.c"	50978	16	5	9176569	19727833
"/kernel/drivers/net/eepro100.c"	74207	29	13	20917883	56431076
"/kernel/drivers/net/eexpress.c"	43991	29	16	14624702	44250502
"/kernel/drivers/net/epic100.c"	42651	19	8	13016129	25081481
"/kernel/drivers/net/eql.c"	24477	32	5	9338356	14395432
"/kernel/drivers/net/es3210.c"	13046	8	2	2967484	4731204
"/kernel/drivers/net/eth16i.c"	43812	22	9	10652836	31463967
"/kernel/drivers/net/ethertap.c"	8306	7	1	2427171	3265068
"/kernel/drivers/net/ewrk3.c"	54087	24	7	13608142	28391123
"/kernel/drivers/net/farsync.c"	64878	27	0	12442020	14170203
"/kernel/drivers/net/fc/iph5526_novram.c"	6740	3	2	1112857	2404636
"/kernel/drivers/net/fc/iph5526.c"	150903	100	66	88653514	457821032
"/kernel/drivers/net/fmv18x.c"	19943	9	3	4412508	7736129
"/kernel/drivers/net/gmac.c"	35225	27	17	14074660	38507103
"/kernel/drivers/net/hamachi.c"	70351	18	7	14784917	35351315
"/kernel/drivers/net/hamradio/6pack.c"	27309	32	15	11656558	29807593
"/kernel/drivers/net/hamradio/baycom_epp.c"	45722	24	8	10873585	24173176
"/kernel/drivers/net/hamradio/baycom_par.c"	17778	11	5	4003523	8339821
"/kernel/drivers/net/hamradio/baycom_ser_fdx.c"	21825	11	3	4654537	7894916
"/kernel/drivers/net/hamradio/baycom_ser_hdx.c"	21630	12	4	5066892	9046935
"/kernel/drivers/net/hamradio/bpqether.c"	15160	17	0	3598354	4601420
"/kernel/drivers/net/hamradio/dmascc.c"	35054	23	12	11806547	46807048
"/kernel/drivers/net/hamradio/hdlcdrv.c"	26967	20	6	6906794	13839026
"/kernel/drivers/net/hamradio/mkiss.c"	31109	37	14	13403167	32996680
"/kernel/drivers/net/hamradio/pi2.c"	47185	30	16	16828832	53946574
"/kernel/drivers/net/hamradio/pt.c"	50056	28	14	16596894	48169722
"/kernel/drivers/net/hamradio/scc.c"	57333	54	38	31452811	134842474
"/kernel/drivers/net/hamradio/soundmodem/gentbl.c"	23169	13	10	6994727	16068349
"/kernel/drivers/net/hamradio/soundmodem/sm_afsk1200.c"	7940	9	4	2895071	5357056
"/kernel/drivers/net/hamradio/soundmodem/sm_afsk2400_7.c"	8878	9	5	3129197	7943492
"/kernel/drivers/net/hamradio/soundmodem/sm_afsk2400_8.c"	8873	9	5	3342277	5953486
"/kernel/drivers/net/hamradio/soundmodem/sm_afsk2666.c"	10192	9	5	3220600	7168962
"/kernel/drivers/net/hamradio/soundmodem/sm_fsk9600.c"	12014	9	9	5425752	12753803
"/kernel/drivers/net/hamradio/soundmodem/sm_hapn4800.c"	17299	13	13	7728948	19357884
"/kernel/drivers/net/hamradio/soundmodem/sm_psk4800.c"	11911	8	6	4143304	7579416
"/kernel/drivers/net/hamradio/soundmodem/sm_sbc.c"	24909	18	4	6327190	10856976
"/kernel/drivers/net/hamradio/soundmodem/sm_wss.c"	28394	20	4	7167511	13844780
"/kernel/drivers/net/hamradio/soundmodem/sm.c"	23480	12	4	5708202	9936830
"/kernel/drivers/net/hamradio/yam.c"	33341	27	10	9886661	24164034
"/kernel/drivers/net/hd6457x.c"	20432	21	10	7580558	20093908
"/kernel/drivers/net/hdlc.c"	29730	30	11	10800186	23024232
"/kernel/drivers/net/hostess_sv11.c"	8556	10	2	2431874	4005302
"/kernel/drivers/net/hp-plus.c"	15300	11	5	4630070	8351163
"/kernel/drivers/net/hp.c"	13999	9	4	3762546	6822898
"/kernel/drivers/net/hp100.c"	98918	31	18	28344074	94638890
"/kernel/drivers/net/hplance.c"	8293	7	2	2303399	3927815
"/kernel/drivers/net/hydra.c"	19084	8	2	3729624	6371826
"/kernel/drivers/net/ibmtr.c"	62304	22	11	16821432	41510339
"/kernel/drivers/net/ipddp.c"	9845	9	0	2616598	3124918
"/kernel/drivers/net/irda/actisys.c"	8784	5	3	2158504	3897025

”/kernel/drivers/net/irda/esi.c”	3707	5	3	1782618	4939869
”/kernel/drivers/net/irda/girbil.c”	6386	5	3	1810090	3464129
”/kernel/drivers/net/irda/irport.c”	24291	21	6	7427390	16261626
”/kernel/drivers/net/irda/irtty.c”	26636	21	4	7379628	12706463
”/kernel/drivers/net/irda/litelink.c”	4769	5	3	1644041	3107667
”/kernel/drivers/net/irda/nsc_fir.c”	44744	26	7	12454722	27257507
”/kernel/drivers/net/irda/nsc-ircc.c”	49902	27	6	13412311	25296424
”/kernel/drivers/net/irda/old_belkin.c”	5013	5	3	1443161	2942702
”/kernel/drivers/net/irda/smc-ircc.c”	25618	15	6	7133652	13686481
”/kernel/drivers/net/irda/tekram.c”	7182	5	3	1920814	3647628
”/kernel/drivers/net/irda/toshoboe.c”	22482	14	0	4258966	5382505
”/kernel/drivers/net/irda/w83977af_ir.c”	33419	20	5	8943104	16413952
”/kernel/drivers/net/jazzsonic.c”	7048	2	0	1199398	1909302
”/kernel/drivers/net/lance.c”	41363	13	5	8813256	21540966
”/kernel/drivers/net/lanstreamer.c”	60297	12	6	13059058	26743737
”/kernel/drivers/net/lapbether.c”	12307	19	4	4342212	7481686
”/kernel/drivers/net/lmc/lmc_debug.c”	1447	2	2	713062	1640588
”/kernel/drivers/net/lmc/lmc_main.c”	74134	25	6	17928512	36589306
”/kernel/drivers/net/lmc/lmc_media.c”	50498	44	0	14484820	18678774
”/kernel/drivers/net/lmc/lmc_proto.c”	6819	9	7	3098548	8018724
”/kernel/drivers/net/lne390.c”	13470	8	2	3761858	5071126
”/kernel/drivers/net/loopback.c”	3987	4	0	721017	3393348
”/kernel/drivers/net/lp486e.c”	35073	24	9	10494985	23524406
”/kernel/drivers/net/ltpc.c”	32801	30	11	10417862	23798384
”/kernel/drivers/net/mac8390.c”	16786	11	5	5659139	11905644
”/kernel/drivers/net/mac89x0.c”	20251	14	3	4797914	10470091
”/kernel/drivers/net/mace.c”	25371	18	11	9679843	24200830
”/kernel/drivers/net/macmace.c”	18147	21	14	8395356	21417434
”/kernel/drivers/net/macsonic.c”	17359	8	1	3702636	5296754
”/kernel/drivers/net/mvme147.c”	5976	6	2	2302105	3329900
”/kernel/drivers/net/myri_sbus.c”	30113	29	15	12372702	28272364
”/kernel/drivers/net/n2.c”	11319	11	7	3828877	7744308
”/kernel/drivers/net/ncr885e.c”	36109	18	12	11296290	28108830
”/kernel/drivers/net/ne.c”	25845	8	3	5020272	10848950
”/kernel/drivers/net/ne2.c”	20309	9	1	4163446	5916352
”/kernel/drivers/net/ne2k-pci.c”	19955	8	3	4722633	8012614
”/kernel/drivers/net/ne3210.c”	12862	8	2	2942729	4834232
”/kernel/drivers/net/net_init.c”	19711	9	3	2242020	5219955
”/kernel/drivers/net/ni5010.c”	25122	17	9	8158166	17004633
”/kernel/drivers/net/ni52.c”	38300	16	7	9136030	21106830
”/kernel/drivers/net/ni65.c”	30001	17	8	8786837	17526132
”/kernel/drivers/net/old_tulip.c”	91540	22	11	22241421	56742251
”/kernel/drivers/net/olympic.c”	59946	15	6	12594760	26382998
”/kernel/drivers/net/pc300.c”	109238	62	43	45815960	213939518
”/kernel/drivers/net/pcnet32.c”	47157	28	4	11946200	22395137
”/kernel/drivers/net/plip.c”	33764	25	5	9499554	16960300
”/kernel/drivers/net/ppp_deflate.c”	17724	16	7	6085757	11966481
”/kernel/drivers/net/ppp.c”	73492	58	14	25326515	65894436
”/kernel/drivers/net/ptifddi.c”	5738	20	12	4969577	10300647
”/kernel/drivers/net/rclanmtl.c”	71123	28	2	12664617	22778198
”/kernel/drivers/net/rcpci45.c”	44913	18	4	10443623	22256758
”/kernel/drivers/net/rrunner.c”	34483	18	4	7945082	16290750
”/kernel/drivers/net/rtl8139.c”	47078	18	7	12663311	27061602
”/kernel/drivers/net/sb1000.c”	36121	30	5	11494325	19013552
”/kernel/drivers/net/sbni.c”	38067	23	12	12045876	32956492
”/kernel/drivers/net/sdla_chdlc.c”	81654	41	2	19038179	26657933
”/kernel/drivers/net/sdla_fr.c”	106459	58	8	27120283	61398726
”/kernel/drivers/net/sdla_ppp.c”	82664	49	8	24231494	45316621
”/kernel/drivers/net/sdla_x25.c”	67837	56	1	16953818	24056182
”/kernel/drivers/net/sdla.c”	39155	30	10	48391655	32854768
”/kernel/drivers/net/sdladrv.c”	59811	41	0	13838725	15892394
”/kernel/drivers/net/sdlamain.c”	26592	12	0	5181633	6201596
”/kernel/drivers/net/sealevel.c”	9252	10	2	3049546	4213165
”/kernel/drivers/net/seeq8005.c”	22099	12	6	6042282	11878335
”/kernel/drivers/net/sgiseeq.c”	20158	29	9	9139480	16191951
”/kernel/drivers/net/shaper.c”	16279	22	7	5566668	10889149
”/kernel/drivers/net/sis900.c”	52145	36	14	19315254	46157010
”/kernel/drivers/net/sk_g16.c”	57961	20	10	11965721	37920562
”/kernel/drivers/net/sk_mca.c”	28578	24	10	8425343	19844499
”/kernel/drivers/net/sk98lin/skaddr.c”	34608	8	0	4733728	6292114
”/kernel/drivers/net/sk98lin/skcsum.c”	28546	0	0	42685	1045347
”/kernel/drivers/net/sk98lin/skge.c”	116332	47	5	25540742	49487840
”/kernel/drivers/net/sk98lin/skgehwt.c”	4781	5	0	827018	1421236
”/kernel/drivers/net/sk98lin/skgeinit.c”	56823	23	0	7796436	9923126
”/kernel/drivers/net/sk98lin/skgepnmi.c”	205955	42	0	33138300	43481394
”/kernel/drivers/net/sk98lin/skgesirq.c”	59485	17	0	9517787	14475578
”/kernel/drivers/net/sk98lin/ski2c.c”	32670	10	0	3811991	4881223
”/kernel/drivers/net/sk98lin/sklm80.c”	7314	1	0	568373	1183468
”/kernel/drivers/net/sk98lin/skproc.c”	11493	3	0	2105472	2782890
”/kernel/drivers/net/sk98lin/skqueue.c”	5381	3	0	720528	1289494
”/kernel/drivers/net/sk98lin/skrlmt.c”	100175	33	0	15941801	20041248
”/kernel/drivers/net/sk98lin/sktimer.c”	6863	5	0	1246523	1919732
”/kernel/drivers/net/sk98lin/skvpd.c”	31048	20	0	5351101	6988207
”/kernel/drivers/net/sk98lin/skxmac2.c”	61741	27	0	11915250	13209267
”/kernel/drivers/net/skeleton.c”	16757	9	3	3837868	9042441
”/kernel/drivers/net/skfp/can.c”	2470	0	0	269837	769632
”/kernel/drivers/net/skfp/cfm.c”	16547	9	3	4102100	7144786
”/kernel/drivers/net/skfp/drvfbi.c”	34700	19	17	9035456	33239131
”/kernel/drivers/net/skfp/ecm.c”	12919	6	6	3398621	7447596
”/kernel/drivers/net/skfp/ess.c”	19050	0	0	34470	833188
”/kernel/drivers/net/skfp/fplustm.c”	41513	45	37	24171844	89816429

"/kernel/drivers/net/skfp/hwmtm.c"	56352	25	16	16336149	241894058
"/kernel/drivers/net/skfp/hwt.c"	6596	5	4	1600916	5733388
"/kernel/drivers/net/skfp/lnkstat.c"	4667	3	0	880301	1446268
"/kernel/drivers/net/skfp/pcmplc.c"	49508	27	23	18576768	62831178
"/kernel/drivers/net/skfp/pmf.c"	40625	9	2	8148636	12681276
"/kernel/drivers/net/skfp/queue.c"	4159	5	4	1480054	3187022
"/kernel/drivers/net/skfp/rmt.c"	16447	13	13	7200972	18581204
"/kernel/drivers/net/skfp/skfddi.c"	75314	39	19	23395702	74644278
"/kernel/drivers/net/skfp/smt.c"	53293	52	36	32321584	101217370
"/kernel/drivers/net/skfp/smtdef.c"	9915	6	4	2691299	7457650
"/kernel/drivers/net/skfp/smtinit.c"	3003	2	1	800490	1621624
"/kernel/drivers/net/skfp/smtparse.c"	9981	0	0	1885850	2594176
"/kernel/drivers/net/skfp/smttimer.c"	3534	6	6	2198163	4404190
"/kernel/drivers/net/skfp/srf.c"	10415	6	5	3159292	6166264
"/kernel/drivers/net/sktr.c"	72781	49	30	32533379	110157964
"/kernel/drivers/net/slhc.c"	20477	6	1	1297106	3099701
"/kernel/drivers/net/slip.c"	36418	24	10	9601737	25530708
"/kernel/drivers/net/smc-mca.c"	15736	6	3	3082520	5921418
"/kernel/drivers/net/smc-ultra.c"	15695	11	3	3977555	6933482
"/kernel/drivers/net/smc-ultra32.c"	13583	8	1	3340292	4349942
"/kernel/drivers/net/smc9194.c"	49285	19	9	12233424	30051577
"/kernel/drivers/net/sonic.c"	17259	8	3	5075349	7040690
"/kernel/drivers/net/Space.c"	29538	2	0	7347022	12713152
"/kernel/drivers/net/starfire.c"	63223	17	8	15976012	34440022
"/kernel/drivers/net/strip.c"	101508	58	25	35687599	134380738
"/kernel/drivers/net/sunbmac.c"	39555	36	20	16846912	49733849
"/kernel/drivers/net/sunhme.c"	100693	43	14	25781619	79445040
"/kernel/drivers/net/sunlance.c"	33475	16	0	5848579	12026986
"/kernel/drivers/net/sunqe.c"	26609	16	6	6940240	13858755
"/kernel/drivers/net/syncppp.c"	34822	28	5	10252226	19704850
"/kernel/drivers/net/tlan.c"	82134	43	24	24729773	93563056
"/kernel/drivers/net/tulip.c"	111003	28	16	30162045	90990973
"/kernel/drivers/net/via-rhine.c"	44353	22	14	14788292	35910769
"/kernel/drivers/net/wanxl.c"	29788	26	6	9551823	18327770
"/kernel/drivers/net/wavelan.c"	121258	51	12	23240574	76487012
"/kernel/drivers/net/wd.c"	16732	8	3	3726943	6819119
"/kernel/drivers/net/x25_asy.c"	22132	26	14	9440365	22480153
"/kernel/drivers/net/xpds/xpds-encap-fr.c"	23384	7	1	5075450	7138044
"/kernel/drivers/net/xpds/xpds-fsm.c"	23449	8	0	4149314	5241550
"/kernel/drivers/net/xpds/xpds-sdsl.c"	27458	24	0	6352151	7581079
"/kernel/drivers/net/xpds/xpds.c"	121509	46	0	25508699	31237558
"/kernel/drivers/net/yellowfin.c"	45982	17	7	11416494	23933326
"/kernel/drivers/net/z85230.c"	28533	36	21	15644685	40916806
"/kernel/drivers/net/zlib.c"	178328	67	22	51101545	208098088
"/kernel/drivers/net/znet.c"	25390	11	6	7032171	12922282
"/kernel/drivers/nubus/nubus_syms.c"	711	0	0	551176	1033032
"/kernel/drivers/nubus/nubus.c"	23155	21	3	7084911	13274012
"/kernel/drivers/nubus/proc.c"	4699	7	2	1908470	2808566
"/kernel/drivers/pci/compat.c"	1765	3	0	535738	1020282
"/kernel/drivers/pci/oldproc.c"	45743	0	0	61913	1427871
"/kernel/drivers/pci/pci.c"	13133	14	3	4072999	8792280
"/kernel/drivers/pci/pcisyms.c"	1310	0	0	1014116	1502676
"/kernel/drivers/pci/proc.c"	9212	9	2	3244245	4232688
"/kernel/drivers/pci/quirks.c"	9569	6	5	2624918	5414283
"/kernel/drivers/pnp/parport_probe.c"	7827	8	4	2556978	4725786
"/kernel/drivers/s390/block/dasd_3370_erp.c"	3348	1	0	400202	924642
"/kernel/drivers/s390/block/dasd_3990_erp.c"	37197	23	0	6120940	7683180
"/kernel/drivers/s390/block/dasd_9336_erp.c"	1829	1	0	156036	660525
"/kernel/drivers/s390/block/dasd_9343_erp.c"	636	1	0	146174	518696
"/kernel/drivers/s390/block/dasd_diag.c"	18935	16	1	4750644	6364614
"/kernel/drivers/s390/block/dasd_eckd_erp.c"	397	1	0	115197	599832
"/kernel/drivers/s390/block/dasd_eckd.c"	35371	20	1	7564596	12950156
"/kernel/drivers/s390/block/dasd_fba.c"	13221	13	1	3970850	4758824
"/kernel/drivers/s390/block/dasd.c"	116517	58	1	26426702	39296795
"/kernel/drivers/s390/block/mdisk.c"	18856	16	5	5820000	9986272
"/kernel/drivers/s390/block/xpram.c"	33604	19	2	6854656	12458445
"/kernel/drivers/s390/ccwcache.c"	11323	5	0	2303098	3128180
"/kernel/drivers/s390/char/con3215.c"	33868	33	22	14966333	43928565
"/kernel/drivers/s390/char/hwc_con.c"	1905	1	0	210520	743572
"/kernel/drivers/s390/char/hwc_rw.c"	47853	50	0	12887168	14366725
"/kernel/drivers/s390/char/hwc_tty.c"	8732	12	0	2839019	3615616
"/kernel/drivers/s390/char/tape.c"	34442	20	0	6829978	8383084
"/kernel/drivers/s390/char/tape3480.c"	5797	3	0	1280754	1713131
"/kernel/drivers/s390/char/tape3490.c"	5797	3	0	1094979	1702162
"/kernel/drivers/s390/char/tape34xx.c"	70447	58	2	56771666	27380628
"/kernel/drivers/s390/char/tapeblock.c"	18591	13	3	5100850	7848122
"/kernel/drivers/s390/char/tapechar.c"	21709	9	0	4729132	5224991
"/kernel/drivers/s390/idals.c"	1049	0	0	16983	463185
"/kernel/drivers/s390/net/ctc.c"	58201	35	10	16365095	36487494
"/kernel/drivers/s390/net/iucv.c"	70993	31	0	12373878	17166677
"/kernel/drivers/s390/net/lcs.c"	113744	75	0	24118646	27379586
"/kernel/drivers/s390/net/netiucv.c"	30589	15	0	5503132	8930965
"/kernel/drivers/sbus/audio/amd7930.c"	48089	52	13	13817346	36757908
"/kernel/drivers/sbus/audio/audio.c"	69859	22	5	16801694	30679119
"/kernel/drivers/sbus/audio/cs4231.c"	77442	78	17	25760184	74826706
"/kernel/drivers/sbus/audio/dbri.c"	77591	85	31	34285676	127161472
"/kernel/drivers/sbus/audio/dmy.c"	22120	61	5	10609679	16248695
"/kernel/drivers/sbus/char/aurora.c"	62015	50	28	27334218	96844751
"/kernel/drivers/sbus/char/bpp.c"	32809	15	2	5029070	8945064
"/kernel/drivers/sbus/char/cpwatchdog.c"	21931	20	8	7357128	15259802
"/kernel/drivers/sbus/char/display7seg.c"	6232	6	1	1564572	2601959

"/kernel/drivers/sbus/char/envctrl.c"	29932	29	10	9734553	26028036
"/kernel/drivers/sbus/char/flash.c"	5701	6	0	1763032	2163453
"/kernel/drivers/sbus/char/openprom.c"	14934	13	0	3376074	4323742
"/kernel/drivers/sbus/char/pcikbd.c"	31148	38	14	12990669	31392550
"/kernel/drivers/sbus/char/rtc.c"	3384	7	2	1537574	2656622
"/kernel/drivers/sbus/char/sab82532.c"	68222	47	24	25201024	89627524
"/kernel/drivers/sbus/char/su.c"	76568	49	29	30590316	107244316
"/kernel/drivers/sbus/char/sunkbd.c"	37531	61	44	28143914	92566766
"/kernel/drivers/sbus/char/sunkbdmap.c"	872	0	0	17539	483055
"/kernel/drivers/sbus/char/sunkeymap.c"	11114	0	0	1958400	2572430
"/kernel/drivers/sbus/char/sunmouse.c"	13851	16	3	3990216	6486675
"/kernel/drivers/sbus/char/sunserial.c"	8780	26	6	5477516	8222636
"/kernel/drivers/sbus/char/vfc_dev.c"	16460	24	5	5790161	12224380
"/kernel/drivers/sbus/char/vfc_i2c.c"	7484	13	2	2519550	4055992
"/kernel/drivers/sbus/char/zs.c"	73604	53	32	31642824	116377866
"/kernel/drivers/sbus/dvma.c"	3806	3	1	974464	1883534
"/kernel/drivers/sbus/sbus.c"	13445	3	1	2898126	4329449
"/kernel/drivers/scsi/3w-xxxx.c"	108723	52	14	28642602	86950864
"/kernel/drivers/scsi/53c78xx.c"	207134	62	4	40895682	83875238
"/kernel/drivers/scsi/53c7xx.c"	194473	53	2	35295482	64725954
"/kernel/drivers/scsi/a2091.c"	6481	5	0	1541985	2223328
"/kernel/drivers/scsi/a3000.c"	5338	5	0	1294460	1903370
"/kernel/drivers/scsi/advansys.c"	741751	130	16	153480509	702871124
"/kernel/drivers/scsi/aha152x.c"	89492	31	16	24014959	74055142
"/kernel/drivers/scsi/aha1542.c"	48841	25	7	13714492	25223556
"/kernel/drivers/scsi/aha1740.c"	17958	12	3	4367425	7440370
"/kernel/drivers/scsi/aic7xxx.c"	415540	87	43	142772576	1071945822
"/kernel/drivers/scsi/aic7xxx/aic7xxx_proc.c"	13708	2	0	1996344	2882446
"/kernel/drivers/scsi/aic7xxx/aic7xxx_seq.c"	20747	16	0	4944710	6040565
"/kernel/drivers/scsi/AM53C974.c"	92222	28	16	22281855	80921130
"/kernel/drivers/scsi/amiga7xx.c"	4162	1	0	713184	1268591
"/kernel/drivers/scsi/atari_dma_emul.c"	11027	3	2	2092601	5882832
"/kernel/drivers/scsi/atari_NCR5380.c"	94623	21	6	16094291	37025782
"/kernel/drivers/scsi/atari_scsi.c"	36132	21	8	10020648	20892379
"/kernel/drivers/scsi/atp870u.c"	42599	17	7	11144384	22113446
"/kernel/drivers/scsi/blz1230.c"	7155	11	6	3498714	6389090
"/kernel/drivers/scsi/blz2060.c"	6408	13	8	4009520	7496966
"/kernel/drivers/scsi/BusLogic.c"	183962	45	13	42217493	139216508
"/kernel/drivers/scsi/bvme6000.c"	1329	1	0	325429	851016
"/kernel/drivers/scsi/constants.c"	26832	8	4	2678029	7298396
"/kernel/drivers/scsi/cpqfcTScontrol.c"	72718	16	5	14968820	30534448
"/kernel/drivers/scsi/cpqfcTSi2c.c"	12649	13	6	3821752	7870536
"/kernel/drivers/scsi/cpqfcTSinit.c"	58856	24	7	13427956	31443769
"/kernel/drivers/scsi/cpqfcTStrigger.c"	846	1	1	352850	1048705
"/kernel/drivers/scsi/cpqfcTSworker.c"	202886	36	12	46617076	148345889
"/kernel/drivers/scsi/cpqioctl.c"	1471	1	0	339576	822156
"/kernel/drivers/scsi/cyberstorm.c"	8816	13	8	4463091	8600141
"/kernel/drivers/scsi/cyberstormII.c"	6809	13	8	4428517	7768666
"/kernel/drivers/scsi/dec_esp.c"	9185	16	11	6301866	12371750
"/kernel/drivers/scsi/dtc.c"	12775	5	1	2020458	3484463
"/kernel/drivers/scsi/eata_dma_proc.c"	13884	3	1	2664541	3898741
"/kernel/drivers/scsi/eata_dma.c"	48329	20	10	13137521	32032622
"/kernel/drivers/scsi/eata_pio_proc.c"	3965	2	0	628331	1214357
"/kernel/drivers/scsi/eata_pio.c"	30400	16	8	8397520	22606251
"/kernel/drivers/scsi/eata.c"	78143	30	10	19012604	49331673
"/kernel/drivers/scsi/esp.c"	124003	85	28	49116006	178551933
"/kernel/drivers/scsi/fastlane.c"	9083	15	10	5114049	10157596
"/kernel/drivers/scsi/fcal.c"	9488	5	1	2110548	3341396
"/kernel/drivers/scsi/fd_mcs.c"	44531	17	8	10501906	29541046
"/kernel/drivers/scsi/fdomain.c"	66894	21	9	14852118	35640514
"/kernel/drivers/scsi/FlashPoint.c"	315496	100	69	124998006	1171979847
"/kernel/drivers/scsi/g_NCR5380.c"	24956	12	1	3704090	5860378
"/kernel/drivers/scsi/gdth_proc.c"	53714	14	5	12454302	23089616
"/kernel/drivers/scsi/gdth.c"	169130	51	13	41284383	123595138
"/kernel/drivers/scsi/gvp11.c"	10032	6	0	2401692	3132803
"/kernel/drivers/scsi/hosts.c"	19349	5	2	2736824	5259615
"/kernel/drivers/scsi/i60uscsi.c"	31720	27	9	9545252	20372256
"/kernel/drivers/scsi/i91uscsi.c"	84083	72	23	31798334	99651602
"/kernel/drivers/scsi/ibmmca.c"	116385	32	1	22005285	30732448
"/kernel/drivers/scsi/ide-scsi.c"	25769	32	3	7718390	12079402
"/kernel/drivers/scsi/imm.c"	31714	32	8	10025822	24627050
"/kernel/drivers/scsi/in2000.c"	79258	22	5	17413561	33627313
"/kernel/drivers/scsi/ini9100u.c"	32971	5	2	6718076	10587393
"/kernel/drivers/scsi/inia100.c"	31499	5	2	6773745	10351260
"/kernel/drivers/scsi/ips.c"	228345	94	29	69446536	343439050
"/kernel/drivers/scsi/jazz_esp.c"	7567	18	8	5007074	8735711
"/kernel/drivers/scsi/mac_esp.c"	16006	23	14	8685901	22494207
"/kernel/drivers/scsi/mac_scsi.c"	12896	10	5	4154124	7520486
"/kernel/drivers/scsi/mac53c94.c"	14009	12	7	5311995	11875535
"/kernel/drivers/scsi/mca_53c9x.c"	12250	14	8	4685073	9442019
"/kernel/drivers/scsi/megaraid.c"	76842	34	3	14273491	24324274
"/kernel/drivers/scsi/mesh.c"	51683	34	27	23891786	73439658
"/kernel/drivers/scsi/mvme16x.c"	1429	1	0	341090	873312
"/kernel/drivers/scsi/NCR5380.c"	104966	24	15	23378180	78243442
"/kernel/drivers/scsi/NCR53c406a.c"	31729	17	8	9304641	19377615
"/kernel/drivers/scsi/ncr53c8xx.c"	230414	67	22	63759759	257463685
"/kernel/drivers/scsi/NCR53C9x.c"	107424	70	25	39816287	134910591
"/kernel/drivers/scsi/oktagon_esp.c"	14793	23	16	8812494	19689062
"/kernel/drivers/scsi/osst.c"	177928	62	5	40253267	83196006
"/kernel/drivers/scsi/pas16.c"	17581	8	1	2938601	4784108
"/kernel/drivers/scsi/pci2000.c"	23512	15	1	4879536	7980908

"/kernel/drivers/scsi/pci2220i.c"	94903	43	2	18368681	32302363
"/kernel/drivers/scsi/pluto.c"	8838	8	4	2926729	5394342
"/kernel/drivers/scsi/ppa.c"	28027	32	11	10850812	22329800
"/kernel/drivers/scsi/psi240i.c"	22273	14	2	4582314	9505244
"/kernel/drivers/scsi/qlogicfas.c"	20137	15	6	5806580	10787794
"/kernel/drivers/scsi/qlogicfc_asm.c"	506102	0	0	98451641	129428650
"/kernel/drivers/scsi/qlogicfc.c"	62869	20	6	13609674	32237794
"/kernel/drivers/scsi/qlogicisp_asm.c"	131538	0	0	345747	3804159
"/kernel/drivers/scsi/qlogicisp.c"	49700	19	6	9673812	25642328
"/kernel/drivers/scsi/qlogicpti_asm.c"	75238	0	0	11039421	15378326
"/kernel/drivers/scsi/qlogicpti.c"	41021	24	11	13925442	31194064
"/kernel/drivers/scsi/scsi_debug.c"	19575	11	4	5299084	8788834
"/kernel/drivers/scsi/scsi_error.c"	61400	27	4	12894705	23036165
"/kernel/drivers/scsi/scsi_ioctl.c"	14234	6	0	2572623	3222462
"/kernel/drivers/scsi/scsi_module.c"	2587	2	1	421421	1210888
"/kernel/drivers/scsi/scsi_obsolete.c"	32682	10	3	43748992	11097290
"/kernel/drivers/scsi/scsi_proc.c"	10336	1	1	348651	1537138
"/kernel/drivers/scsi/scsi_queue.c"	8635	2	0	1228049	1926990
"/kernel/drivers/scsi/scsi_syms.c"	2120	0	0	18423	553666
"/kernel/drivers/scsi/scsi.c"	109805	25	12	22115116	71314424
"/kernel/drivers/scsi/scsicam.c"	7247	3	0	1115220	1627733
"/kernel/drivers/scsi/scsiiom.c"	52277	40	23	21131442	64707540
"/kernel/drivers/scsi/sd_ioctl.c"	3980	1	0	571764	1159242
"/kernel/drivers/scsi/sd.c"	57934	18	3	12751349	21713325
"/kernel/drivers/scsi/seagate.c"	51820	10	1	9738862	13570202
"/kernel/drivers/scsi/sg.c"	62630	36	13	22150932	49176477
"/kernel/drivers/scsi/sgiwd93.c"	7744	10	4	2889175	7262452
"/kernel/drivers/scsi/sim710.c"	45808	23	5	10898208	20187469
"/kernel/drivers/scsi/sr_ioctl.c"	18639	16	1	4584953	6456520
"/kernel/drivers/scsi/sr_vendor.c"	8265	3	1	1427878	2644300
"/kernel/drivers/scsi/sr.c"	38485	15	9	10195256	23527102
"/kernel/drivers/scsi/st.c"	103455	38	5	23461520	46308354
"/kernel/drivers/scsi/sun3x_esp.c"	8179	20	10	6103845	13217279
"/kernel/drivers/scsi/sym53c416.c"	26347	17	7	6866772	14417370
"/kernel/drivers/scsi/sym53c8xx.c"	376129	115	38	122079218	705964152
"/kernel/drivers/scsi/t128.c"	10969	5	1	1715454	3029924
"/kernel/drivers/scsi/tmscsim.c"	92608	54	10	23027800	55266819
"/kernel/drivers/scsi/u14-34f.c"	66119	26	8	15087605	33561648
"/kernel/drivers/scsi/ultrastor.c"	35372	13	4	7862528	15783840
"/kernel/drivers/scsi/wd33c93.c"	63947	21	6	14865479	31664168
"/kernel/drivers/scsi/wd7000.c"	54743	28	1	10164743	14732001
"/kernel/drivers/sgi/char/ds1286.c"	14621	11	2	4223643	6215372
"/kernel/drivers/sgi/char/graphics_syms.c"	855	0	0	482100	948022
"/kernel/drivers/sgi/char/graphics.c"	9898	7	0	2043628	2859404
"/kernel/drivers/sgi/char/newport.c"	4792	3	0	1007312	1533574
"/kernel/drivers/sgi/char/rrm.c"	1509	5	0	687088	1172138
"/kernel/drivers/sgi/char/sgicons.c"	907	3	0	489948	952704
"/kernel/drivers/sgi/char/sgiserial.c"	57589	57	34	29517948	103040226
"/kernel/drivers/sgi/char/shmiq.c"	11431	14	0	2902404	3766575
"/kernel/drivers/sgi/char/streamable.c"	8521	13	0	2978196	5478034
"/kernel/drivers/sgi/char/usema.c"	5024	6	1	1480812	2435964
"/kernel/drivers/sound/ac97_codec.c"	21855	11	1	5380596	7300358
"/kernel/drivers/sound/ac97.c"	11193	14	0	3095230	3725136
"/kernel/drivers/sound/ad1816.c"	32984	0	0	47063	1140142
"/kernel/drivers/sound/ad1848.c"	65852	0	0	81129	1913694
"/kernel/drivers/sound/adlib_card.c"	1390	3	2	704503	1583156
"/kernel/drivers/sound/audio_syms.c"	478	0	0	410929	883536
"/kernel/drivers/sound/audio.c"	24378	4	1	2771845	4715388
"/kernel/drivers/sound/bin2hex.c"	708	1	0	196749	582535
"/kernel/drivers/sound/cmpci.c"	76866	45	17	51588432	76975538
"/kernel/drivers/sound/cs4232.c"	9690	8	6	3468119	6382405
"/kernel/drivers/sound/cs4281.c"	90268	38	14	25293558	74674390
"/kernel/drivers/sound/cs46xx.c"	72092	56	32	33462238	117705928
"/kernel/drivers/sound/dev_table.c"	13408	24	11	6535741	13599789
"/kernel/drivers/sound/dmabuf.c"	34858	0	0	48637	1206266
"/kernel/drivers/sound/dmasound.c"	137723	25	6	15982782	49383600
"/kernel/drivers/sound/emu10k1/audio.c"	37999	12	4	8539006	17457908
"/kernel/drivers/sound/emu10k1/cardmi.c"	19687	24	1	5196106	7160868
"/kernel/drivers/sound/emu10k1/cardmo.c"	6146	5	1	1343205	2308420
"/kernel/drivers/sound/emu10k1/cardwi.c"	9181	12	9	4131402	8723998
"/kernel/drivers/sound/emu10k1/cardwo.c"	13952	16	11	6319996	12949804
"/kernel/drivers/sound/emu10k1/ecard.c"	5189	2	2	1108662	2352749
"/kernel/drivers/sound/emu10k1/emu_wrapper.c"	3830	8	1	1711680	2555520
"/kernel/drivers/sound/emu10k1/emuadxmg.c"	3187	2	1	669954	1458058
"/kernel/drivers/sound/emu10k1/hwaccess.c"	13695	20	8	5311904	10358870
"/kernel/drivers/sound/emu10k1/irqmgr.c"	2776	1	1	524364	1290226
"/kernel/drivers/sound/emu10k1/main.c"	21241	0	0	3264878	4313078
"/kernel/drivers/sound/emu10k1/midi.c"	10543	8	1	2485104	3763416
"/kernel/drivers/sound/emu10k1/mixer.c"	34788	7	2	5012804	8967776
"/kernel/drivers/sound/emu10k1/recmgr.c"	3579	3	3	1186452	2537038
"/kernel/drivers/sound/emu10k1/timer.c"	4345	5	5	1845025	3870604
"/kernel/drivers/sound/emu10k1/voicemgr.c"	9242	5	4	2437476	5989190
"/kernel/drivers/sound/es1370.c"	82787	45	13	22271697	61929731
"/kernel/drivers/sound/es1371.c"	102686	53	16	27581407	86428436
"/kernel/drivers/sound/esssolo1.c"	70502	44	15	22479159	61396535
"/kernel/drivers/sound/gus_card.c"	6591	0	0	21653	594251
"/kernel/drivers/sound/gus_midi.c"	4718	0	0	20095	422879
"/kernel/drivers/sound/gus_vol.c"	3403	0	0	17962	369426
"/kernel/drivers/sound/gus_wave.c"	79642	0	0	96738	2128433
"/kernel/drivers/sound/hex2hex.c"	1939	2	0	494659	968210
"/kernel/drivers/sound/i810_audio.c"	51877	37	16	18217541	49654627

”/kernel/drivers/sound/ics2101.c”	4612	0	0	20055	550152
”/kernel/drivers/sound/lowlevel/aci.c”	19440	0	0	33563	713064
”/kernel/drivers/sound/lowlevel/aedsp16.c”	37008	0	0	52084	1197600
”/kernel/drivers/sound/lowlevel/awe_wave.c”	167081	0	0	246328	4753260
”/kernel/drivers/sound/lowlevel/soundlow.c”	1346	3	3	1381823	4812145
”/kernel/drivers/sound/mad16.c”	23925	0	0	39479	924804
”/kernel/drivers/sound/maestro.c”	104906	66	34	42050310	173013116
”/kernel/drivers/sound/maestro3.c”	87627	64	27	32854084	118739400
”/kernel/drivers/sound/maui.c”	9740	0	0	24516	657935
”/kernel/drivers/sound/midi_syms.c”	849	0	0	860394	1238458
”/kernel/drivers/sound/midi_synth.c”	14941	0	0	31332	707324
”/kernel/drivers/sound/midibuf.c”	8701	0	0	23330	567900
”/kernel/drivers/sound/mpu401.c”	36290	0	0	50662	1126635
”/kernel/drivers/sound/msnd_classic.c”	122	0	0	16639	467795
”/kernel/drivers/sound/msnd_pinnacle.c”	48755	50	14	18101570	47150436
”/kernel/drivers/sound/msnd.c”	8162	18	5	3478592	6001926
”/kernel/drivers/sound/nm256_audio.c”	43543	38	2	11505784	15441338
”/kernel/drivers/sound/opl3.c”	26855	25	12	10179465	24312749
”/kernel/drivers/sound/opl3sa.c”	6554	0	0	21597	604165
”/kernel/drivers/sound/opl3sa2.c”	16834	0	0	30753	691424
”/kernel/drivers/sound/pas2_card.c”	9408	0	0	25184	597762
”/kernel/drivers/sound/pas2_midi.c”	4693	0	0	21029	458976
”/kernel/drivers/sound/pas2_mixer.c”	8269	0	0	22798	480844
”/kernel/drivers/sound/pas2_pcm.c”	9138	0	0	23550	526387
”/kernel/drivers/sound/pss.c”	31051	0	0	45099	1097709
”/kernel/drivers/sound/sb_audio.c”	25648	0	0	39322	942086
”/kernel/drivers/sound/sb_card.c”	5235	0	0	21440	511960
”/kernel/drivers/sound/sb_common.c”	28591	0	0	42963	932420
”/kernel/drivers/sound/sb_ess.c”	51892	48	2	12494854	19881534
”/kernel/drivers/sound/sb_midi.c”	4140	0	0	19445	555520
”/kernel/drivers/sound/sb_mixer.c”	19951	0	0	33424	718548
”/kernel/drivers/sound/sequencer_syms.c”	743	0	0	630576	1082477
”/kernel/drivers/sound/sequencer.c”	35234	0	0	49861	1158883
”/kernel/drivers/sound/sgalaxy.c”	3599	0	0	19030	533534
”/kernel/drivers/sound/skeleton.c”	4396	4	1	1062461	1808716
”/kernel/drivers/sound/softoss_rs.c”	2938	0	0	19028	491682
”/kernel/drivers/sound/softoss.c”	33137	0	0	47315	1150966
”/kernel/drivers/sound/sonicvibes.c”	78425	47	17	24456170	71745256
”/kernel/drivers/sound/sound_core.c”	9333	17	7	5237433	9045378
”/kernel/drivers/sound/sound_firmware.c”	1026	2	0	542730	901163
”/kernel/drivers/sound/sound_syms.c”	1649	0	0	1875388	2109156
”/kernel/drivers/sound/sound_timer.c”	5894	0	0	21315	560273
”/kernel/drivers/sound/soundcard.c”	25096	25	6	8331470	15265833
”/kernel/drivers/sound/sscape.c”	35280	0	0	48988	1225410
”/kernel/drivers/sound/sys_timer.c”	5271	0	0	20485	557083
”/kernel/drivers/sound/trident.c”	84394	64	25	33088006	107014524
”/kernel/drivers/sound/trix.c”	12122	0	0	27279	713636
”/kernel/drivers/sound/uart401.c”	10087	0	0	24538	596266
”/kernel/drivers/sound/uart6850.c”	6508	0	0	21293	501562
”/kernel/drivers/sound/v_midi.c”	6036	0	0	20558	468413
”/kernel/drivers/sound/via82cxxx_audio.c”	61023	45	2	13908930	20152432
”/kernel/drivers/sound/vidc_synth.c”	2007	0	0	18046	531175
”/kernel/drivers/sound/vidc.c”	11482	20	11	6157159	15148166
”/kernel/drivers/sound/vwsnd.c”	97988	74	27	38561298	130006366
”/kernel/drivers/sound/waveartist.c”	42934	41	18	16999784	47865346
”/kernel/drivers/sound/wavfront.c”	89698	0	0	175711	2715142
”/kernel/drivers/sound/wf_midi.c”	18756	0	0	32318	827488
”/kernel/drivers/sound/ymfpci.c”	70071	54	18	23297540	72656310
”/kernel/drivers/sound/yss225.c”	18009	0	0	3278036	3973550
”/kernel/drivers/tc/tc.c”	5259	9	5	2877627	4890084
”/kernel/drivers/tc/tcsyms.c”	287	0	0	233705	703492
”/kernel/drivers/tc/zs.c”	53831	47	24	21283628	71322314
”/kernel/drivers/telephony/ixj.c”	265345	114	43	105917255	566694722
”/kernel/drivers/telephony/phonedev.c”	3216	4	1	1082815	1903010
”/kernel/drivers/usb/acm.c”	19141	17	10	7058873	15335036
”/kernel/drivers/usb/audio.c”	119062	72	26	43203655	154945089
”/kernel/drivers/usb/bluetooth.c”	36602	22	3	9286524	16512882
”/kernel/drivers/usb/dabusb.c”	19983	23	0	5567439	6366821
”/kernel/drivers/usb/dc2xx.c”	13622	6	1	2802558	4341566
”/kernel/drivers/usb/devices.c”	16645	17	1	4607752	8336788
”/kernel/drivers/usb/devio.c”	30604	41	6	10923609	18184188
”/kernel/drivers/usb/drivers.c”	3312	2	0	512837	1048505
”/kernel/drivers/usb/dsbr100.c”	9252	9	3	3026698	7143631
”/kernel/drivers/usb/evdev.c”	9240	10	2	3015219	4694272
”/kernel/drivers/usb/hid.c”	41250	40	14	15535029	40615796
”/kernel/drivers/usb/hub.c”	24572	25	9	8897982	17809742
”/kernel/drivers/usb/ibmcam.c”	92332	60	33	41489998	155236219
”/kernel/drivers/usb/inode.c”	19142	21	10	7762282	15607636
”/kernel/drivers/usb/input.c”	9457	13	10	5321058	10582344
”/kernel/drivers/usb/joydev.c”	13455	11	2	3956848	7978120
”/kernel/drivers/usb/kaweth.c”	26781	22	7	7824629	15204989
”/kernel/drivers/usb/keybdev.c”	6747	4	3	1559791	3190488
”/kernel/drivers/usb/mdc800.c”	21964	14	0	4464323	5412634
”/kernel/drivers/usb/microtek.c”	27187	22	13	10150916	25517186
”/kernel/drivers/usb/mousedev.c”	12600	10	3	3849228	6226698
”/kernel/drivers/usb/ov511.c”	89344	46	8	23090830	53335938
”/kernel/drivers/usb/pegasus.c”	22811	31	13	10322741	21869096
”/kernel/drivers/usb/plusb.c”	29746	17	6	7003816	13919079
”/kernel/drivers/usb/printer.c”	17666	12	2	4568903	6745488
”/kernel/drivers/usb/rio500.c”	12143	9	2	3629647	5114447
”/kernel/drivers/usb/scanner.c”	28241	9	2	4178271	7428719

"/kernel/drivers/usb/serial/belkin_sa.c"	21389	8	1	5079812	7118516
"/kernel/drivers/usb/serial/cyberjack.c"	14375	8	1	4094256	5414483
"/kernel/drivers/usb/serial/digi_acceleport.c"	61460	24	9	14635445	36880184
"/kernel/drivers/usb/serial/empeg.c"	16303	13	0	4636328	5624299
"/kernel/drivers/usb/serial/ftdi_sio.c"	31006	15	1	7738033	10396943
"/kernel/drivers/usb/serial/io_edgeport.c"	98870	44	5	22016516	43204103
"/kernel/drivers/usb/serial/ir-usb.c"	15915	9	1	4472306	5957762
"/kernel/drivers/usb/serial/keyspan_pda.c"	26410	20	3	6831768	11073010
"/kernel/drivers/usb/serial/keyspan.c"	47593	38	21	19939980	56921202
"/kernel/drivers/usb/serial/mct_u232.c"	27876	14	2	6963640	11446310
"/kernel/drivers/usb/serial/omninet.c"	10753	7	0	2707296	3568414
"/kernel/drivers/usb/serial/pl2303.c"	22788	14	0	5798008	7176705
"/kernel/drivers/usb/serial/usbserial.c"	45546	27	4	10673874	18525270
"/kernel/drivers/usb/serial/visor.c"	26344	13	0	6155106	7759142
"/kernel/drivers/usb/serial/whiteheat.c"	19530	15	1	5263951	7352190
"/kernel/drivers/usb/uhci.c"	60780	60	28	28618166	347390406
"/kernel/drivers/usb/usb-debug.c"	6798	10	9	3867189	7728724
"/kernel/drivers/usb/usb-ohci.c"	61270	34	1	11958095	18684169
"/kernel/drivers/usb/usb-storage.c"	62386	31	9	17229375	41845310
"/kernel/drivers/usb/usb-uhci.c"	75739	56	11	21339291	53316536
"/kernel/drivers/usb/usb.c"	60285	66	22	25257268	76775241
"/kernel/drivers/usb/usbkbd.c"	7908	7	4	3021417	4894682
"/kernel/drivers/usb/usbmouse.c"	5338	5	3	1975274	3546260
"/kernel/drivers/usb/uss720.c"	17681	29	10	8039108	18129802
"/kernel/drivers/usb/wacom.c"	10601	6	4	2821480	5372880
"/kernel/drivers/usb/wmforce.c"	5542	5	3	2007846	3597314
"/kernel/drivers/video/acornfb.c"	50650	28	3	9716705	18872884
"/kernel/drivers/video/amifb.c"	100569	49	19	32725056	102165664
"/kernel/drivers/video/atafb.c"	85869	62	20	32122888	89321244
"/kernel/drivers/video/aty128fb.c"	75869	53	11	22593684	56490210
"/kernel/drivers/video/atyfb.c"	118161	61	16	34810780	100762964
"/kernel/drivers/video/bwtwofb.c"	6371	3	0	1234831	1879394
"/kernel/drivers/video/cgfourteenfb.c"	12632	11	0	3039032	3634707
"/kernel/drivers/video/cgsixfb.c"	21072	16	3	5151293	10403936
"/kernel/drivers/video/cgthreefb.c"	7043	5	0	1628142	2336320
"/kernel/drivers/video/chipsfb.c"	20332	20	7	7842258	13634982
"/kernel/drivers/video/clgenfb.c"	63731	43	20	24885598	69683798
"/kernel/drivers/video/controlfb.c"	35791	33	13	14517465	35553072
"/kernel/drivers/video/creatorfb.c"	22856	17	6	6388258	12303693
"/kernel/drivers/video/cyber2000fb.c"	33766	33	8	11532698	22566001
"/kernel/drivers/video/cyberfb.c"	64298	44	9	19249865	44272399
"/kernel/drivers/video/dnfb.c"	9715	14	3	3870226	7840892
"/kernel/drivers/video/dummycon.c"	1730	3	1	662742	1397684
"/kernel/drivers/video/fbcmap.c"	6655	7	3	3696743	4763204
"/kernel/drivers/video/fbcon-afb.c"	13776	6	2	3250197	5346399
"/kernel/drivers/video/fbcon-cfb16.c"	8586	8	2	2992048	4590357
"/kernel/drivers/video/fbcon-cfb2.c"	5232	6	2	2110530	5231366
"/kernel/drivers/video/fbcon-cfb24.c"	8940	9	3	3168854	5159294
"/kernel/drivers/video/fbcon-cfb32.c"	8550	8	2	3385226	4513068
"/kernel/drivers/video/fbcon-cfb4.c"	5409	6	2	2156518	3471031
"/kernel/drivers/video/fbcon-cfb8.c"	8063	8	2	2909984	4431349
"/kernel/drivers/video/fbcon-ilbm.c"	6893	6	2	2525062	3647082
"/kernel/drivers/video/fbcon-iplan2p2.c"	12191	14	5	4583725	7950166
"/kernel/drivers/video/fbcon-iplan2p4.c"	12746	14	5	4674790	8127976
"/kernel/drivers/video/fbcon-iplan2p8.c"	14410	14	5	4879418	8788864
"/kernel/drivers/video/fbcon-mac.c"	12029	8	3	3546373	5943924
"/kernel/drivers/video/fbcon-mfb.c"	5006	7	2	2398322	3523612
"/kernel/drivers/video/fbcon-vga-planes.c"	8760	16	12	5878651	13391862
"/kernel/drivers/video/fbcon-vga.c"	4918	11	5	2981046	5160317
"/kernel/drivers/video/fbcon.c"	65347	44	16	23823317	66618454
"/kernel/drivers/video/fbgen.c"	9004	13	3	3398914	4994466
"/kernel/drivers/video/fbmem.c"	22630	16	2	9376838	11447746
"/kernel/drivers/video/fm2fb.c"	15988	17	4	5387207	8737339
"/kernel/drivers/video/font_6x11.c"	68155	0	0	10276162	12361574
"/kernel/drivers/video/font_8x16.c"	95863	0	0	14334494	19528542
"/kernel/drivers/video/font_8x8.c"	50801	0	0	7462628	9113188
"/kernel/drivers/video/font_acorn_8x8.c"	16008	0	0	1979121	2852108
"/kernel/drivers/video/font_pearl_8x8.c"	55669	0	0	7806519	11915236
"/kernel/drivers/video/font_sun12x22.c"	186491	0	0	33124845	44003778
"/kernel/drivers/video/font_sun8x16.c"	22518	0	0	2989028	4166784
"/kernel/drivers/video/fonts.c"	2328	2	0	465131	956843
"/kernel/drivers/video/g364fb.c"	14699	17	4	6007186	10585575
"/kernel/drivers/video/hpfb.c"	9445	20	5	4732554	7480024
"/kernel/drivers/video/igafb.c"	22061	20	6	6290006	12174472
"/kernel/drivers/video/imsttfb.c"	54252	39	0	12360804	14915022
"/kernel/drivers/video/leofb.c"	19525	17	3	5630262	8973734
"/kernel/drivers/video/macfb.c"	32319	18	5	8104567	14187139
"/kernel/drivers/video/macmodes.c"	11423	3	0	2357848	3203325
"/kernel/drivers/video/matroxfb.c"	188996	69	32	54923206	280192862
"/kernel/drivers/video/mdacon.c"	14135	27	13	8787693	16973946
"/kernel/drivers/video/newport_con.c"	17133	22	10	7533198	16804268
"/kernel/drivers/video/offb.c"	29261	20	5	9242356	15870346
"/kernel/drivers/video/platinumfb.c"	26300	27	7	10160037	19707560
"/kernel/drivers/video/pm2fb.c"	59557	42	15	17088216	50639074
"/kernel/drivers/video/promcon.c"	12188	20	7	5363236	9843622
"/kernel/drivers/video/q40fb.c"	9262	16	3	4215554	6365318
"/kernel/drivers/video/retz3fb.c"	41986	29	8	14964871	29941608
"/kernel/drivers/video/S3triofb.c"	24210	25	9	9560030	17841067
"/kernel/drivers/video/sbusfb.c"	33689	28	9	12222899	25302218
"/kernel/drivers/video/sgivwfb.c"	36684	29	7	13394420	23307777
"/kernel/drivers/video/skeletonfb.c"	10160	16	6	4656499	8123089

”/kernel/drivers/video/tcxfb.c”	9992	12	0	2617194	3433084
”/kernel/drivers/video/tgafb.c”	31577	18	7	10169344	20924284
”/kernel/drivers/video/valkyriefb.c”	26103	29	9	10910622	20314634
”/kernel/drivers/video/vesafb.c”	17408	18	5	6577948	12888342
”/kernel/drivers/video/vfb.c”	16024	20	5	6900312	12865406
”/kernel/drivers/video/vga_font.c”	25111	0	0	3614120	4482408
”/kernel/drivers/video/vga16fb.c”	29426	28	11	10472273	24957236
”/kernel/drivers/video/vgacon.c”	30028	25	13	12704377	27034255
”/kernel/drivers/video/virgefb.c”	35610	31	10	12696996	27801354
”/kernel/drivers/zorro/proc.c”	4117	5	1	1728571	2300962
”/kernel/drivers/zorro/zorro.c”	5851	6	4	2004694	3825929
”/kernel/drivers/zorro/zorrosyms.c”	456	0	0	240586	705297
”/kernel/fs/adfs/dir_f.c”	9961	16	5	3901990	6987071
”/kernel/fs/adfs/dir_fplus.c”	4285	5	2	1580347	2681756
”/kernel/fs/adfs/dir.c”	5551	8	0	1724537	2236860
”/kernel/fs/adfs/file.c”	867	0	0	158890	608168
”/kernel/fs/adfs/inode.c”	8013	10	3	2407687	4117688
”/kernel/fs/adfs/map.c”	5519	5	0	1266869	1915462
”/kernel/fs/adfs/super.c”	11622	11	2	3304599	5227765
”/kernel/fs/affs/amigaffs.c”	11909	14	4	3865290	8535560
”/kernel/fs/affs/bitmap.c”	9235	8	2	2882388	4173040
”/kernel/fs/affs/dir.c”	4666	2	0	967568	1560668
”/kernel/fs/affs/file.c”	25302	11	2	5629264	8772761
”/kernel/fs/affs/inode.c”	11597	8	4	3502934	8360196
”/kernel/fs/affs/namei.c”	14216	16	0	3692028	6777678
”/kernel/fs/affs/super.c”	19733	7	2	4369260	6971795
”/kernel/fs/affs/symlink.c”	3645	2	0	822750	1368754
”/kernel/fs/attr.c”	2616	3	1	1068912	1644202
”/kernel/fs/autofs/dir.c”	2087	2	0	517320	991460
”/kernel/fs/autofs/dirhash.c”	6242	11	6	3379192	7929843
”/kernel/fs/autofs/init.c”	1075	1	0	176140	668106
”/kernel/fs/autofs/inode.c”	8151	8	5	2965412	5448798
”/kernel/fs/autofs/root.c”	15037	12	0	3612794	4589543
”/kernel/fs/autofs/symlink.c”	1639	2	0	405826	892044
”/kernel/fs/autofs/waitq.c”	5685	5	2	1647898	2916442
”/kernel/fs/bad_inode.c”	2434	4	1	894264	1628216
”/kernel/fs/binfmt_aout.c”	14407	6	1	3328820	4569554
”/kernel/fs/binfmt_elf.c”	35399	7	2	5403558	9284868
”/kernel/fs/binfmt_em86.c”	2939	1	0	554008	1091424
”/kernel/fs/binfmt_java.c”	4017	2	0	931599	1455945
”/kernel/fs/binfmt_misc.c”	12817	16	5	4480709	7791162
”/kernel/fs/binfmt_script.c”	2604	1	0	532624	1049486
”/kernel/fs/block_dev.c”	7423	3	0	1822921	2159531
”/kernel/fs/buffer.c”	51402	51	24	21591221	66688328
”/kernel/fs/coda/cache.c”	7204	12	9	3670678	9974449
”/kernel/fs/coda/cnode.c”	7032	6	1	1912537	3005804
”/kernel/fs/coda/coda_linux.c”	8990	13	4	3475338	5647679
”/kernel/fs/coda/dir.c”	24954	18	2	6624101	9941720
”/kernel/fs/coda/file.c”	8676	8	0	2455954	3079354
”/kernel/fs/coda/inode.c”	6415	7	3	2345856	4016223
”/kernel/fs/coda/pioctl.c”	3903	2	0	846000	1392936
”/kernel/fs/coda/psdev.c”	10877	8	0	2313200	3210422
”/kernel/fs/coda/symlink.c”	3017	2	0	672282	1202553
”/kernel/fs/coda/sysctl.c”	13640	20	7	36039497	9924512
”/kernel/fs/coda/upcall.c”	26830	22	0	6665726	7403636
”/kernel/fs/dcache.c”	23936	27	10	8901602	17495633
”/kernel/fs/devices.c”	8795	15	0	2942776	3512890
”/kernel/fs/devpts/inode.c”	7842	9	5	3361953	5613269
”/kernel/fs/devpts/root.c”	4453	3	0	1056081	1690030
”/kernel/fs/dquot.c”	42063	59	28	23095030	69585483
”/kernel/fs/efs/dir.c”	3351	1	0	718540	1273100
”/kernel/fs/efs/file.c”	1541	1	0	374969	845512
”/kernel/fs/efs/inode.c”	8467	4	2	2015775	3545714
”/kernel/fs/efs/namei.c”	1767	2	0	453737	931294
”/kernel/fs/efs/super.c”	6460	6	1	1766491	2821056
”/kernel/fs/efs/symlink.c”	2436	3	0	782016	1284564
”/kernel/fs/exec.c”	23208	20	5	6493457	11416044
”/kernel/fs/ext2/acl.c”	1430	1	0	242066	744349
”/kernel/fs/ext2/balloc.c”	22285	11	0	4172534	5397922
”/kernel/fs/ext2/bitmap.c”	614	1	0	202356	674358
”/kernel/fs/ext2/dir.c”	6188	3	0	1891887	1828664
”/kernel/fs/ext2/file.c”	9838	5	1	2439956	3490648
”/kernel/fs/ext2/fsync.c”	4316	7	0	1329546	1713242
”/kernel/fs/ext2/ialloc.c”	16860	6	0	2951493	3772994
”/kernel/fs/ext2/inode.c”	23383	15	0	4736784	8153514
”/kernel/fs/ext2/ioctl.c”	2313	1	0	417319	930984
”/kernel/fs/ext2/namei.c”	24465	14	0	5158369	6149632
”/kernel/fs/ext2/super.c”	24714	13	0	5197593	7755827
”/kernel/fs/ext2/symlink.c”	2334	2	0	604470	1104462
”/kernel/fs/ext2/truncate.c”	11783	6	0	1776789	2600230
”/kernel/fs/fat/buffer.c”	5157	7	0	1472844	2093761
”/kernel/fs/fat/cache.c”	8380	9	5	2898410	5615969
”/kernel/fs/fat/cvf.c”	3618	4	1	1160176	2044570
”/kernel/fs/fat/dir.c”	16073	10	0	3361568	4204702
”/kernel/fs/fat/fatfs_syms.c”	1750	1	0	2324530	2606183
”/kernel/fs/fat/file.c”	13093	5	1	2689353	6271405
”/kernel/fs/fat/inode.c”	27209	18	9	8203013	17788310
”/kernel/fs/fat/misc.c”	17480	19	6	5191235	10170228
”/kernel/fs/fat/mmap.c”	3280	3	0	729751	1263426
”/kernel/fs/fat/tables.c”	2795	0	0	337488	813502
”/kernel/fs/fcntl.c”	6276	7	2	2297556	3453098

"/kernel/fs/fifo.c"	3922	2	1	1032496	1896446
"/kernel/fs/file_table.c"	3597	7	5	2087794	3979025
"/kernel/fs/file.c"	4899	6	2	1540209	2789625
"/kernel/fs/filesystems.c"	3420	1	0	534375	1060370
"/kernel/fs/hfs/balloc.c"	12101	7	1	1907405	3282223
"/kernel/fs/hfs/bdelete.c"	13355	6	2	2367786	4398850
"/kernel/fs/hfs/bfind.c"	10590	3	1	1185038	2444180
"/kernel/fs/hfs/bins_del.c"	6755	4	0	993778	1605156
"/kernel/fs/hfs/binsert.c"	16261	8	2	2868131	4870817
"/kernel/fs/hfs/bitmap.c"	11233	4	0	1446018	2220972
"/kernel/fs/hfs/bitops.c"	2827	2	0	433811	919025
"/kernel/fs/hfs/bnode.c"	15743	6	4	2815511	8186164
"/kernel/fs/hfs/brec.c"	7153	6	1	1233532	2305922
"/kernel/fs/hfs/btree.c"	9664	4	3	1847083	3843039
"/kernel/fs/hfs/catalog.c"	40085	46	20	17074606	303227490
"/kernel/fs/hfs/dir_cap.c"	11060	3	1	2009727	3325128
"/kernel/fs/hfs/dir_dbl.c"	12667	10	1	2869641	4307060
"/kernel/fs/hfs/dir_nat.c"	14146	6	1	2666970	4219222
"/kernel/fs/hfs/dir.c"	11869	11	2	3131080	4629127
"/kernel/fs/hfs/extent.c"	19670	21	12	7624929	16897212
"/kernel/fs/hfs/file_cap.c"	8053	4	1	1718122	2876971
"/kernel/fs/hfs/file_hdr.c"	26593	10	1	6822356	9394771
"/kernel/fs/hfs/file.c"	13833	10	3	3767318	5796707
"/kernel/fs/hfs/inode.c"	13330	7	5	3027710	6489476
"/kernel/fs/hfs/mdb.c"	10250	3	2	1913224	5900386
"/kernel/fs/hfs/part_tbl.c"	6508	3	0	1105030	1716468
"/kernel/fs/hfs/string.c"	5325	4	1	1157342	2087717
"/kernel/fs/hfs/super.c"	13423	7	3	3172660	5381100
"/kernel/fs/hfs/sysdep.c"	2973	5	1	1354263	2056441
"/kernel/fs/hfs/trans.c"	16033	12	4	2992682	6191112
"/kernel/fs/hfs/version.c"	252	0	0	79608	520918
"/kernel/fs/hpfs/hpfs_caps.c"	5692	7	0	935146	1366211
"/kernel/fs/hpfs/hpfs_fs.c"	40955	38	4	11802606	20991014
"/kernel/fs/inode.c"	22629	35	19	58315887	30881800
"/kernel/fs/ioctl.c"	2453	2	0	581235	3228352
"/kernel/fs/isofs/dir.c"	7329	4	0	1492790	2153415
"/kernel/fs/isofs/file.c"	1370	0	0	197285	672074
"/kernel/fs/isofs/inode.c"	36420	20	4	8965396	15259042
"/kernel/fs/isofs/joliet.c"	1840	3	0	535680	1033402
"/kernel/fs/isofs/namei.c"	4132	3	0	827050	1362110
"/kernel/fs/isofs/rock.c"	13425	4	0	2030152	2872668
"/kernel/fs/isofs/symlink.c"	1747	2	0	523456	1011690
"/kernel/fs/isofs/util.c"	3759	9	0	1243547	1794146
"/kernel/fs/lockd/clntlock.c"	4737	4	1	1291142	2213549
"/kernel/fs/lockd/clntproc.c"	16324	20	7	6451863	11511388
"/kernel/fs/lockd/host.c"	9171	13	4	3597923	5713070
"/kernel/fs/lockd/lockd_syms.c"	1311	0	0	18010	484977
"/kernel/fs/lockd/mon.c"	5588	8	0	1751864	2220932
"/kernel/fs/lockd/svc.c"	9601	4	2	2549232	4164896
"/kernel/fs/lockd/svc4proc.c"	14880	19	1	4409105	5955473
"/kernel/fs/lockd/svclock.c"	17839	17	6	5895034	12631068
"/kernel/fs/lockd/svcproc.c"	15533	20	1	4740148	6263272
"/kernel/fs/lockd/svcshare.c"	2586	4	0	724858	1239108
"/kernel/fs/lockd/svcsubs.c"	7232	10	5	2775274	5223022
"/kernel/fs/lockd/xdr.c"	15215	38	4	6808450	10241641
"/kernel/fs/lockd/xdr4.c"	14188	35	0	5306825	8545834
"/kernel/fs/locks.c"	34424	34	8	10323187	22840310
"/kernel/fs/minix/bitmap.c"	7386	10	3	2562496	4370474
"/kernel/fs/minix/dir.c"	2381	2	0	629082	1134193
"/kernel/fs/minix/file.c"	2988	1	0	585882	3402560
"/kernel/fs/minix/fsync.c"	6773	14	0	2231415	2873107
"/kernel/fs/minix/inode.c"	24655	30	8	8994841	19175660
"/kernel/fs/minix/namei.c"	14754	14	0	3440887	4394727
"/kernel/fs/minix/symlink.c"	1642	2	0	531218	1014486
"/kernel/fs/minix/truncate.c"	9348	10	3	2670422	4657396
"/kernel/fs/msdos/msdosfs_syms.c"	797	1	0	536111	1021947
"/kernel/fs/msdos/namei.c"	17687	15	2	5781122	6989204
"/kernel/fs/namei.c"	33749	42	4	10334739	15431471
"/kernel/fs/ncpfs/dir.c"	30042	29	7	10306813	18706876
"/kernel/fs/ncpfs/file.c"	7123	5	0	1551148	2297095
"/kernel/fs/ncpfs/inode.c"	20600	13	7	6606954	14794074
"/kernel/fs/ncpfs/ioctl.c"	14857	1	0	2347450	3267035
"/kernel/fs/ncpfs/mmap.c"	2763	3	0	660642	1174039
"/kernel/fs/ncpfs/ncplib_kernel.c"	23789	40	11	11269732	21205142
"/kernel/fs/ncpfs/ncpsign_kernel.c"	3296	0	0	18949	532896
"/kernel/fs/ncpfs/sock.c"	11776	9	2	2889002	4603687
"/kernel/fs/ncpfs/symlink.c"	5276	0	0	22294	547589
"/kernel/fs/nfs/dir.c"	35470	29	3	9381654	16181709
"/kernel/fs/nfs/file.c"	7445	8	0	2155713	2685182
"/kernel/fs/nfs/flushd.c"	6485	10	8	3220688	6677787
"/kernel/fs/nfs/inode.c"	29286	27	8	9614748	18767788
"/kernel/fs/nfs/mount_clnt.c"	4384	8	0	1767162	2400418
"/kernel/fs/nfs/nfs2xdr.c"	19199	29	0	5222336	6392236
"/kernel/fs/nfs/nfs3proc.c"	14341	20	1	4358346	6135160
"/kernel/fs/nfs/nfs3xdr.c"	25821	45	0	8127111	9310669
"/kernel/fs/nfs/nfsroot.c"	13816	12	4	3325111	6444436
"/kernel/fs/nfs/proc.c"	11238	19	1	3833924	5107349
"/kernel/fs/nfs/read.c"	12512	18	6	4924964	8747669
"/kernel/fs/nfs/symlink.c"	2665	3	0	734721	1264280
"/kernel/fs/nfs/unlink.c"	5107	8	7	2703200	5704298
"/kernel/fs/nfs/write.c"	36618	37	12	11607502	28607728

"/kernel/fs/nfsd/auth.c"	1457	1	1	476537	1205244
"/kernel/fs/nfsd/export.c"	20648	25	9	10148354	20197192
"/kernel/fs/nfsd/lockd.c"	1988	5	4	1312647	2741089
"/kernel/fs/nfsd/nfs3proc.c"	17717	23	1	5028560	7026845
"/kernel/fs/nfsd/nfs3xdr.c"	19909	49	0	7646892	9161829
"/kernel/fs/nfsd/nfscache.c"	7866	7	5	2780382	5341468
"/kernel/fs/nfsd/nfsctl.c"	8045	12	2	2969764	4537456
"/kernel/fs/nfsd/nfsfh.c"	20178	11	3	4631298	10169606
"/kernel/fs/nfsd/nfsproc.c"	15041	18	1	4552510	5894064
"/kernel/fs/nfsd/nfssvc.c"	7383	3	1	1784455	4054809
"/kernel/fs/nfsd/nfsxdr.c"	9664	28	0	4070379	4982304
"/kernel/fs/nfsd/stats.c"	1779	3	2	894501	1823145
"/kernel/fs/nfsd/vfs.c"	42228	23	7	10089898	20961046
"/kernel/fs/nls/nls_base.c"	13565	14	5	5728595	8380428
"/kernel/fs/nls/nls_cp437.c"	23108	4	4	4473833	10963104
"/kernel/fs/nls/nls_cp737.c"	18865	4	4	3606825	7505239
"/kernel/fs/nls/nls_cp775.c"	18864	4	4	3612273	7485394
"/kernel/fs/nls/nls_cp850.c"	16742	4	4	3851988	6944937
"/kernel/fs/nls/nls_cp852.c"	16742	4	4	3268139	6832388
"/kernel/fs/nls/nls_cp855.c"	16742	4	4	3263809	6838830
"/kernel/fs/nls/nls_cp857.c"	14620	4	4	3258948	6182828
"/kernel/fs/nls/nls_cp860.c"	20986	4	4	3975851	8116858
"/kernel/fs/nls/nls_cp861.c"	23108	4	4	4646056	8833774
"/kernel/fs/nls/nls_cp862.c"	25230	4	4	4662983	11820775
"/kernel/fs/nls/nls_cp863.c"	23108	4	4	5071100	8824364
"/kernel/fs/nls/nls_cp864.c"	22768	4	4	4411914	8780188
"/kernel/fs/nls/nls_cp865.c"	23108	4	4	4617661	10955151
"/kernel/fs/nls/nls_cp866.c"	18864	4	4	3656194	9548941
"/kernel/fs/nls/nls_cp869.c"	16742	4	4	3483512	7772951
"/kernel/fs/nls/nls_cp874.c"	14620	4	4	2951060	6129930
"/kernel/fs/nls/nls_cp932.c"	633062	6	6	144605448	303315731
"/kernel/fs/nls/nls_cp936.c"	1030126	4	4	162104721	394068510
"/kernel/fs/nls/nls_cp949.c"	1205618	4	4	263898883	474331657
"/kernel/fs/nls/nls_cp950.c"	833435	4	4	130919816	317327412
"/kernel/fs/nls/nls_iso8859-1.c"	10132	4	4	2252870	4729924
"/kernel/fs/nls/nls_iso8859-14.c"	12653	4	4	2902224	5757134
"/kernel/fs/nls/nls_iso8859-15.c"	12374	4	4	3031508	5796431
"/kernel/fs/nls/nls_iso8859-2.c"	14376	4	4	2893714	6065667
"/kernel/fs/nls/nls_iso8859-3.c"	14376	4	4	2882317	6055530
"/kernel/fs/nls/nls_iso8859-4.c"	14376	4	4	3076260	8316141
"/kernel/fs/nls/nls_iso8859-5.c"	14640	4	4	2914580	6090846
"/kernel/fs/nls/nls_iso8859-6.c"	12254	4	4	2787218	5470768
"/kernel/fs/nls/nls_iso8859-7.c"	16762	4	4	3227768	6838983
"/kernel/fs/nls/nls_iso8859-8.c"	14640	4	4	3409938	6199086
"/kernel/fs/nls/nls_iso8859-9.c"	12254	4	4	2597167	5433864
"/kernel/fs/nls/nls_koi8-r.c"	18918	4	4	3602817	7517428
"/kernel/fs/nls/nls_koi8-ru.c"	18872	4	4	3628688	7765074
"/kernel/fs/noquot.c"	342	1	0	262882	749050
"/kernel/fs/ntfs/attr.c"	15060	13	1	4281577	5268961
"/kernel/fs/ntfs/dir.c"	26187	26	0	6184769	7458664
"/kernel/fs/ntfs/fs.c"	27609	15	4	6449295	11287238
"/kernel/fs/ntfs/inode.c"	32855	36	6	10099280	20718978
"/kernel/fs/ntfs/super.c"	15691	16	2	4352764	9254116
"/kernel/fs/ntfs/support.c"	6963	13	6	3130311	5989129
"/kernel/fs/ntfs/sysctl.c"	1380	0	0	17580	471069
"/kernel/fs/ntfs/util.c"	8246	17	4	3280166	8837336
"/kernel/fs/open.c"	18523	25	1	9496359	7720208
"/kernel/fs/pipe.c"	11762	18	0	3656788	4532156
"/kernel/fs/proc/array.c"	41809	35	0	11077531	12119310
"/kernel/fs/proc/base.c"	5136	2	2	1798558	3037907
"/kernel/fs/proc/fd.c"	4242	2	0	883322	1511080
"/kernel/fs/proc/generic.c"	7946	8	2	2549944	4063053
"/kernel/fs/proc/inode.c"	9062	12	5	3267523	6495700
"/kernel/fs/proc/kmsg.c"	1652	4	0	816964	1321116
"/kernel/fs/proc/link.c"	3984	3	0	1020729	1602102
"/kernel/fs/proc/mem.c"	5286	4	0	978040	1562058
"/kernel/fs/proc/net.c"	3380	1	0	480200	1013121
"/kernel/fs/proc/omirr.c"	6117	12	3	2940168	4616290
"/kernel/fs/proc/openpromfs.c"	27962	12	0	6123347	7766470
"/kernel/fs/proc/proc_devtree.c"	5521	5	2	1770655	3078423
"/kernel/fs/proc/proc_tty.c"	4824	5	3	1850605	3420866
"/kernel/fs/proc/procfs_syms.c"	1479	1	0	983576	1482325
"/kernel/fs/proc/root.c"	24591	15	3	7352443	10318636
"/kernel/fs/proc/scsi.c"	5826	4	0	1334150	1966592
"/kernel/fs/qnx4/bitmap.c"	4163	4	0	789232	1126336
"/kernel/fs/qnx4/dir.c"	2516	1	0	533253	3405604
"/kernel/fs/qnx4/file.c"	3932	0	0	159055	669472
"/kernel/fs/qnx4/fsync.c"	3028	0	0	18307	424569
"/kernel/fs/qnx4/inode.c"	12341	10	2	3105691	5011060
"/kernel/fs/qnx4/namei.c"	5654	3	0	711354	1287218
"/kernel/fs/qnx4/symlinks.c"	1812	2	0	554358	1041562
"/kernel/fs/qnx4/truncate.c"	738	0	0	16886	440109
"/kernel/fs/read_write.c"	8196	11	0	2489343	3066931
"/kernel/fs/readdir.c"	4258	4	0	1141188	1712037
"/kernel/fs/romfs/inode.c"	15906	14	2	4390298	6554528
"/kernel/fs/select.c"	10600	8	2	2468919	4231150
"/kernel/fs/smbfs/cache.c"	6712	9	4	2980020	6887014
"/kernel/fs/smbfs/dir.c"	10725	15	2	3905008	5722242
"/kernel/fs/smbfs/file.c"	8457	13	1	2849866	3869150
"/kernel/fs/smbfs/inode.c"	14484	15	7	5369057	10099671
"/kernel/fs/smbfs/ioctl.c"	967	1	0	192442	693384

"/kernel/fs/smbfs/proc.c"	52607	60	8	17553044	37808742
"/kernel/fs/smbfs/sock.c"	19982	19	4	5661824	9932196
"/kernel/fs/stat.c"	6155	10	0	1936103	2624292
"/kernel/fs/super.c"	30798	33	6	9603376	17159130
"/kernel/fs/sysv/balloc.c"	9949	3	1	2196028	3232626
"/kernel/fs/sysv/dir.c"	2987	2	0	705608	1240247
"/kernel/fs/sysv/file.c"	6697	2	0	1244284	1891634
"/kernel/fs/sysv/fsync.c"	4153	7	0	1201317	1758273
"/kernel/fs/sysv/ialloc.c"	7016	4	1	1742986	2624120
"/kernel/fs/sysv/inode.c"	30210	31	4	8807811	14206708
"/kernel/fs/sysv/namei.c"	15474	14	0	3606228	4461841
"/kernel/fs/sysv/symlink.c"	1896	2	0	550648	1042571
"/kernel/fs/sysv/truncate.c"	7144	6	1	1712833	5080614
"/kernel/fs/ufs/acl.c"	1536	1	0	239743	720470
"/kernel/fs/ufs/balloc.c"	24275	8	0	5203344	8562100
"/kernel/fs/ufs/cylinder.c"	6151	3	0	1081098	1706129
"/kernel/fs/ufs/dir.c"	6342	2	0	1132339	1754854
"/kernel/fs/ufs/file.c"	6322	4	1	1601712	2608387
"/kernel/fs/ufs/ialloc.c"	7972	2	0	1096416	1762888
"/kernel/fs/ufs/inode.c"	18934	12	0	4095009	4873292
"/kernel/fs/ufs/namei.c"	24633	14	0	5061391	8412453
"/kernel/fs/ufs/super.c"	28106	12	0	5245549	6024742
"/kernel/fs/ufs/symlink.c"	2720	2	0	530811	1032176
"/kernel/fs/ufs/truncate.c"	11235	5	0	2612479	2763286
"/kernel/fs/ufs/util.c"	4188	13	0	1940400	2576666
"/kernel/fs/umsdos/check.c"	5924	4	0	552258	1143377
"/kernel/fs/umsdos/dir.c"	22167	14	2	4886173	7739796
"/kernel/fs/umsdos/emd.c"	17234	17	0	3899720	4649463
"/kernel/fs/umsdos/file.c"	4291	3	0	1040407	1612919
"/kernel/fs/umsdos/inode.c"	13561	13	2	7646683	8274842
"/kernel/fs/umsdos/ioctl.c"	12822	2	0	2095370	5323886
"/kernel/fs/umsdos/mangle.c"	15914	3	0	2140692	3009819
"/kernel/fs/umsdos/namei.c"	29433	20	0	5827598	7032616
"/kernel/fs/umsdos/rdir.c"	7558	5	0	1536216	2022567
"/kernel/fs/umsdos/symlink.c"	3183	3	0	796842	3656787
"/kernel/fs/vfat/namei.c"	30939	29	2	8331478	12041965
"/kernel/fs/vfat/vfatfs_syms.c"	675	1	0	485169	972689
"/kernel/init/main.c"	42338	4	0	9035758	10161007
"/kernel/init/version.c"	614	0	0	180342	636812
"/kernel/ipc/msg.c"	11754	9	0	3189829	3985352
"/kernel/ipc/sem.c"	22451	15	0	5111930	6057391
"/kernel/ipc/shm.c"	19930	17	1	6082857	8151009
"/kernel/ipc/util.c"	2543	14	1	1986487	2804474
"/kernel/kernel/acct.c"	9485	1	0	153029	797211
"/kernel/kernel/capability.c"	6341	4	0	1582708	2036764
"/kernel/kernel/context.c"	4352	6	1	1529226	2425126
"/kernel/kernel/dma.c"	2843	3	1	534757	1296440
"/kernel/kernel/exec_domain.c"	3013	4	0	1037440	1541405
"/kernel/kernel/exit.c"	13048	20	13	6600814	14678226
"/kernel/kernel/fork.c"	17729	17	6	6381814	11017912
"/kernel/kernel/info.c"	734	1	0	205932	694572
"/kernel/kernel/itimer.c"	4041	7	2	1594964	2653396
"/kernel/kernel/kmod.c"	5867	4	1	1641944	2268942
"/kernel/kernel/ksyms.c"	12075	0	0	14173648	17207561
"/kernel/kernel/module.c"	22736	6	0	844648	1744987
"/kernel/kernel/panic.c"	2009	1	1	1223466	1731901
"/kernel/kernel/printk.c"	11639	8	4	3621649	6265377
"/kernel/kernel/resource.c"	4061	5	2	1310834	2448037
"/kernel/kernel/sched.c"	51891	73	37	31852524	106243258
"/kernel/kernel/signal.c"	24884	29	4	9201569	13153455
"/kernel/kernel/softirq.c"	1719	2	2	815084	1749950
"/kernel/kernel/sys.c"	25927	39	1	8607539	10617866
"/kernel/kernel/sysctl.c"	28772	9	1	1477251	6304274
"/kernel/kernel/time.c"	12413	10	3	3542900	5876678
"/kernel/lib/ctype.c"	1321	0	0	182308	671376
"/kernel/lib/errno.c"	89	0	0	60769	448347
"/kernel/lib/inflate.c"	36614	10	1	6598852	9048394
"/kernel/lib/string.c"	5598	22	0	2803809	3445224
"/kernel/lib/vsprintf.c"	6572	7	0	1598137	2236058
"/kernel/mm/filemap.c"	44581	34	9	13280992	27487944
"/kernel/mm/memory.c"	24447	31	11	9231745	22443721
"/kernel/mm/mlock.c"	6445	11	0	2406224	2780220
"/kernel/mm/mmap_avl.c"	13175	4	0	1541871	2412780
"/kernel/mm/mmap.c"	24546	16	4	6512057	11003062
"/kernel/mm/mprotect.c"	5866	9	1	2069115	3194512
"/kernel/mm/mremap.c"	6623	7	0	1755847	2420631
"/kernel/mm/page_alloc.c"	13279	12	10	5000903	11327378
"/kernel/mm/page_io.c"	9581	5	4	2456316	5046285
"/kernel/mm/slab.c"	59852	28	15	16495473	54158111
"/kernel/mm/swap_state.c"	8290	8	3	2272316	4147794
"/kernel/mm/swap.c"	2250	0	0	441090	955670
"/kernel/mm/swapfile.c"	18093	13	2	4675722	8464348
"/kernel/mm/vmalloc.c"	5452	10	4	2789888	4413241
"/kernel/mm/vmscan.c"	13253	8	0	2439039	5741405
"/kernel/net/802/cl2llc.c"	14012	6	4	3157822	6182342
"/kernel/net/802/fc.c"	2531	2	0	440095	937402
"/kernel/net/802/fddi.c"	4484	3	0	691099	1213751
"/kernel/net/802/hippi.c"	3861	4	1	914230	1745808
"/kernel/net/802/llc_macinit.c"	4809	4	3	1749730	3199902
"/kernel/net/802/llc_sendpdu.c"	8279	8	5	2674797	5480770
"/kernel/net/802/llc_utility.c"	5358	9	7	2673960	5477064

”/kernel/net/802/p8022.c”	3546	6	2	1528922	2576866
”/kernel/net/802/p8023.c”	1654	3	1	633622	1330252
”/kernel/net/802/psnap.c”	3354	6	3	1691132	3026294
”/kernel/net/802/sysctl_net_802.c”	754	0	0	67829	528926
”/kernel/net/802/tr.c”	13687	7	4	3344465	6122781
”/kernel/net/appletalk/aarp.c”	26167	0	0	40307	1000220
”/kernel/net/appletalk/ddp.c”	53381	0	0	69275	1585386
”/kernel/net/appletalk/sysctl_net_atalk.c”	1602	2	2	686870	1567060
”/kernel/net/ax25/af_ax25.c”	47176	0	0	61294	1472338
”/kernel/net/ax25/ax25_addr.c”	6159	0	0	21373	546846
”/kernel/net/ax25/ax25_dev.c”	5116	0	0	19844	423540
”/kernel/net/ax25/ax25_ds_in.c”	7837	0	0	22598	453201
”/kernel/net/ax25/ax25_ds_subr.c”	5647	0	0	20335	440038
”/kernel/net/ax25/ax25_ds_timer.c”	5871	0	0	20500	440076
”/kernel/net/ax25/ax25_iface.c”	5545	0	0	20196	454828
”/kernel/net/ax25/ax25_in.c”	12513	0	0	26768	553522
”/kernel/net/ax25/ax25_ip.c”	4799	0	0	19774	476432
”/kernel/net/ax25/ax25_out.c”	9938	0	0	24583	522209
”/kernel/net/ax25/ax25_route.c”	11665	0	0	26087	585728
”/kernel/net/ax25/ax25_std_in.c”	12474	0	0	26857	614558
”/kernel/net/ax25/ax25_std_subr.c”	2902	0	0	17864	431705
”/kernel/net/ax25/ax25_std_timer.c”	4579	0	0	19338	366505
”/kernel/net/ax25/ax25_subr.c”	8359	0	0	22775	469024
”/kernel/net/ax25/ax25_timer.c”	6113	0	0	20825	463709
”/kernel/net/ax25/ax25_uid.c”	4005	0	0	18640	387290
”/kernel/net/ax25/sysctl_net_ax25.c”	5034	2	2	2383440	3243179
”/kernel/net/bridge/br_tree.c”	15102	8	1	3050062	4463404
”/kernel/net/bridge/br.c”	75009	93	51	49679144	199114720
”/kernel/net/bridge/sysctl_net_bridge.c”	279	0	0	63961	518122
”/kernel/net/core/datagram.c”	6789	7	2	1589524	2778673
”/kernel/net/core/dev_mcast.c”	5599	5	3	1377584	5215123
”/kernel/net/core/dev.c”	46447	32	11	13976105	34342050
”/kernel/net/core/dst.c”	3121	6	3	1695923	2977938
”/kernel/net/core/dv.c”	11684	8	3	3961415	5050260
”/kernel/net/core/filter.c”	9330	0	0	24719	628019
”/kernel/net/core/firewall.c”	2948	6	1	1517043	2259858
”/kernel/net/core/iovec.c”	5463	6	1	1386502	2345506
”/kernel/net/core/neighbour.c”	34527	30	11	9759054	28132770
”/kernel/net/core/profile.c”	6849	0	0	21961	589976
”/kernel/net/core/rtnetlink.c”	11657	3	2	794337	2297671
”/kernel/net/core/scm.c”	6326	7	2	1844129	3124409
”/kernel/net/core/skbuff.c”	10737	12	5	3820879	8983612
”/kernel/net/core/sock.c”	24871	41	16	11390819	26382503
”/kernel/net/core/sysctl_net_core.c”	1928	0	0	18025	503958
”/kernel/net/core/utils.c”	1533	3	1	702338	1252118
”/kernel/net/econet/econet.c”	24257	15	3	5014925	8954433
”/kernel/net/ethernet/eth.c”	7723	8	3	2417326	3870289
”/kernel/net/ethernet/pe2.c”	840	3	1	630278	1305764
”/kernel/net/ethernet/sysctl_net_ether.c”	279	0	0	65818	513266
”/kernel/net/ipv4/af_inet.c”	27667	22	6	7726530	14414004
”/kernel/net/ipv4/arp.c”	29566	18	4	6505562	11947538
”/kernel/net/ipv4/devinet.c”	25997	18	4	5281152	10111788
”/kernel/net/ipv4/fib_frontend.c”	14584	12	6	3930194	8169194
”/kernel/net/ipv4/fib_hash.c”	19615	16	2	4962683	6906596
”/kernel/net/ipv4/fib_rules.c”	10508	11	4	3175384	5906875
”/kernel/net/ipv4/fib_semantics.c”	24375	11	1	3571342	5780790
”/kernel/net/ipv4/icmp.c”	38172	15	11	8453996	25914476
”/kernel/net/ipv4/igmp.c”	16801	13	9	4012255	10568550
”/kernel/net/ipv4/ip_forward.c”	7049	1	0	916799	1558097
”/kernel/net/ipv4/ip_fragment.c”	16572	15	7	5792625	13737989
”/kernel/net/ipv4/ip_fw.c”	50672	30	1	10346202	17012966
”/kernel/net/ipv4/ip_gre.c”	30300	16	5	6612897	13132866
”/kernel/net/ipv4/ip_input.c”	15459	5	0	2277475	3162796
”/kernel/net/ipv4/ip_masq_app.c”	19634	18	3	5655802	8560107
”/kernel/net/ipv4/ip_masq_autofw.c”	10270	15	1	2815137	4186220
”/kernel/net/ipv4/ip_masq_cuseeme.c”	6732	6	0	1498811	2147536
”/kernel/net/ipv4/ip_masq_ftp.c”	19529	12	0	4040170	4821946
”/kernel/net/ipv4/ip_masq_irc.c”	12228	7	0	2863519	3143210
”/kernel/net/ipv4/ip_masq_mfw.c”	15403	19	3	4354221	9805821
”/kernel/net/ipv4/ip_masq_mod.c”	7233	14	1	3065648	3912572
”/kernel/net/ipv4/ip_masq_portfw.c”	11036	11	1	2835643	3998196
”/kernel/net/ipv4/ip_masq_quake.c”	6961	6	0	1870362	2260210
”/kernel/net/ipv4/ip_masq_raudio.c”	14425	6	0	2603226	3296300
”/kernel/net/ipv4/ip_masq_user.c”	9235	10	1	2336096	3413710
”/kernel/net/ipv4/ip_masq_vdolive.c”	7703	5	0	1573737	2277688
”/kernel/net/ipv4/ip_masq.c”	66924	38	6	17027438	33540088
”/kernel/net/ipv4/ip_nat_dumb.c”	4158	1	0	532815	1081870
”/kernel/net/ipv4/ip_options.c”	14329	8	3	3445046	5902013
”/kernel/net/ipv4/ip_output.c”	23139	12	4	5417754	9788285
”/kernel/net/ipv4/ip_sockglue.c”	17175	13	8	5540041	12013955
”/kernel/net/ipv4/ipconfig.c”	31870	1	1	3329030	7903776
”/kernel/net/ipv4/ipip.c”	21469	16	5	5661795	10743146
”/kernel/net/ipv4/ipmr.c”	36649	26	9	10086758	22599995
”/kernel/net/ipv4/proc.c”	12865	9	1	2778851	4348854
”/kernel/net/ipv4/protocol.c”	4735	3	1	1094006	2022272
”/kernel/net/ipv4/rarp.c”	12472	10	3	3022586	5387637
”/kernel/net/ipv4/raw.c”	13953	17	3	4028076	6698308
”/kernel/net/ipv4/route.c”	52979	33	14	16173243	50318828
”/kernel/net/ipv4/syncookies.c”	5387	0	0	21464	575004
”/kernel/net/ipv4/sysctl_net_ipv4.c”	7546	0	0	1671726	2174717
”/kernel/net/ipv4/tcp_input.c”	71941	45	30	30347187	110759573

”/kernel/net/ipv4/tcp_ipv4.c”	53520	42	17	20983678	53865457
”/kernel/net/ipv4/tcp_output.c”	36947	24	17	13089506	39697944
”/kernel/net/ipv4/tcp_timer.c”	17214	17	14	7707688	18099173
”/kernel/net/ipv4/tcp.c”	53313	27	9	11364172	31099345
”/kernel/net/ipv4/timer.c”	3588	3	0	568834	1130514
”/kernel/net/ipv4/udp.c”	30720	19	5	7602596	13745996
”/kernel/net/ipv4/utils.c”	1931	2	0	267770	801638
”/kernel/net/ipv6/addrconf.c”	43531	40	23	17327554	252098355
”/kernel/net/ipv6/af_inet6.c”	14274	7	1	2842632	4229020
”/kernel/net/ipv6/datagram.c”	10239	5	1	2423588	3490144
”/kernel/net/ipv6/exthdrs.c”	19371	24	3	5779210	11202456
”/kernel/net/ipv6/icmp.c”	15660	11	3	4105787	9299922
”/kernel/net/ipv6/ip6_fib.c”	23081	27	7	8489679	14756762
”/kernel/net/ipv6/ip6_flowlabel.c”	13594	19	8	5352943	10420887
”/kernel/net/ipv6/ip6_fw.c”	7448	14	5	3493305	6015565
”/kernel/net/ipv6/ip6_input.c”	5863	5	0	1391632	1814062
”/kernel/net/ipv6/ip6_output.c”	17536	9	0	3487640	4505456
”/kernel/net/ipv6/ipv6_sockglue.c”	9036	5	3	2499953	4385988
”/kernel/net/ipv6/mcast.c”	15335	20	10	6099257	13212914
”/kernel/net/ipv6/ndisc.c”	28811	22	8	9383796	18188833
”/kernel/net/ipv6/proc.c”	8179	6	0	1630506	2366664
”/kernel/net/ipv6/protocol.c”	2809	3	1	850971	1682265
”/kernel/net/ipv6/raw.c”	14422	18	3	4682799	7294343
”/kernel/net/ipv6/reassembly.c”	11343	10	5	4416630	7258581
”/kernel/net/ipv6/route.c”	43902	33	7	9833108	23819632
”/kernel/net/ipv6/sit.c”	18100	16	5	5808505	12224201
”/kernel/net/ipv6/sysctl_net_ipv6.c”	1003	0	0	66468	543788
”/kernel/net/ipv6/tcp_ipv6.c”	42554	34	10	15194944	31835195
”/kernel/net/ipv6/udp.c”	21549	15	3	5804270	9162543
”/kernel/net/ipx/af_ipx.c”	60173	0	0	76965	1791820
”/kernel/net/ipx/af_spx.c”	23735	0	0	37728	908517
”/kernel/net/ipx/sysctl_net_ipx.c”	268	0	0	61491	512414
”/kernel/net/irda/af_irda.c”	53204	32	5	12877948	25421138
”/kernel/net/irda/compressors/irda_deflate.c”	16499	16	8	6217532	11854914
”/kernel/net/irda/crc.c”	3170	1	0	442713	956739
”/kernel/net/irda/discovery.c”	7786	6	4	2123351	4246741
”/kernel/net/irda/ircomm/ircomm_core.c”	11958	15	3	3415186	5800984
”/kernel/net/irda/ircomm/ircomm_event.c”	7271	6	1	1903694	3011011
”/kernel/net/irda/ircomm/ircomm_lmp.c”	8729	10	1	2231594	3523682
”/kernel/net/irda/ircomm/ircomm_param.c”	13758	14	0	4273722	4588326
”/kernel/net/irda/ircomm/ircomm_ttp.c”	7935	10	1	2318822	3299456
”/kernel/net/irda/ircomm/ircomm_tty_attach.c”	25449	20	7	8257893	15478053
”/kernel/net/irda/ircomm/ircomm_tty_ioctl.c”	12516	7	1	3125979	3956303
”/kernel/net/irda/ircomm/ircomm_tty.c”	37368	23	12	12814651	30867206
”/kernel/net/irda/irda_device.c”	13403	19	8	5668484	10714953
”/kernel/net/irda/iriap_event.c”	12473	24	4	6077575	9135608
”/kernel/net/irda/iriap.c”	24707	23	14	9826686	25506301
”/kernel/net/irda/irias_object.c”	10692	18	7	4900612	8496134
”/kernel/net/irda/irlan/irlan_client_event.c”	13559	12	0	3666745	4416462
”/kernel/net/irda/irlan/irlan_client.c”	14982	12	7	5747200	10045443
”/kernel/net/irda/irlan/irlan_common.c”	29668	29	17	11919774	33291683
”/kernel/net/irda/irlan/irlan_eth.c”	10183	10	3	2637914	4864254
”/kernel/net/irda/irlan/irlan_event.c”	1714	2	2	708293	1657916
”/kernel/net/irda/irlan/irlan_filter.c”	6495	3	2	1663914	2826390
”/kernel/net/irda/irlan/irlan_provider_event.c”	6444	5	0	1472240	2152634
”/kernel/net/irda/irlan/irlan_provider.c”	10873	8	0	2228776	3030364
”/kernel/net/irda/irlap_comp.c”	8637	7	2	2145556	3725504
”/kernel/net/irda/irlap_event.c”	55179	19	3	12015894	20229181
”/kernel/net/irda/irlap_frame.c”	33855	33	13	12756318	28883816
”/kernel/net/irda/irlap.c”	32255	30	22	12832256	47464946
”/kernel/net/irda/irlmp_event.c”	22137	16	5	6551864	11867756
”/kernel/net/irda/irlmp_frame.c”	11919	9	1	2609235	4075900
”/kernel/net/irda/irlmp.c”	38889	36	15	12729818	39149254
”/kernel/net/irda/irmod.c”	13485	13	6	7988702	12280360
”/kernel/net/irda/irproc.c”	2834	2	2	1061734	2105669
”/kernel/net/irda/irqueue.c”	15570	18	5	4809461	8643530
”/kernel/net/irda/irsysctl.c”	3262	2	1	1133777	1996814
”/kernel/net/irda/irttp.c”	39088	25	10	11719877	33875742
”/kernel/net/irda/parameters.c”	14677	11	0	3656787	4352041
”/kernel/net/irda/qos.c”	18252	20	4	5640840	9591462
”/kernel/net/irda/timer.c”	5541	16	15	5647400	11667114
”/kernel/net/irda/wrapper.c”	9219	8	0	2670786	3022409
”/kernel/net/lapb/lapb_iface.c”	8656	0	0	23857	620445
”/kernel/net/lapb/lapb_in.c”	17507	0	0	31586	712851
”/kernel/net/lapb/lapb_out.c”	5410	0	0	20496	515596
”/kernel/net/lapb/lapb_subr.c”	7803	0	0	22115	459598
”/kernel/net/lapb/lapb_timer.c”	4635	0	0	19534	419582
”/kernel/net/netlink/af_netlink.c”	19255	27	12	8651316	18131465
”/kernel/net/netlink/netlink_dev.c”	4579	8	0	1399693	2024954
”/kernel/net/netrom/af_netrom.c”	31446	0	0	45078	1102590
”/kernel/net/netrom/nr_dev.c”	5587	0	0	20982	534390
”/kernel/net/netrom/nr_in.c”	8028	0	0	22404	462340
”/kernel/net/netrom/nr_loopback.c”	2300	0	0	17589	384016
”/kernel/net/netrom/nr_out.c”	6524	0	0	21509	396383
”/kernel/net/netrom/nr_route.c”	20393	0	0	35306	737778
”/kernel/net/netrom/nr_subr.c”	7151	0	0	22569	547220
”/kernel/net/netrom/nr_timer.c”	5900	0	0	20433	437574
”/kernel/net/netrom/sysctl_net_netrom.c”	3961	2	2	2212943	2849394
”/kernel/net/netsyms.c”	14987	0	0	4888307	4714614
”/kernel/net/packet/af_packet.c”	28077	21	4	7040060	12346163
”/kernel/net/protocols.c”	3322	0	0	242627	759310

”/kernel/net/rose/af_rose.c”	38448	0	0	57618	1260730
”/kernel/net/rose/rose_dev.c”	4759	0	0	20153	518142
”/kernel/net/rose/rose_in.c”	8715	0	0	23031	483481
”/kernel/net/rose/rose_link.c”	8222	0	0	22609	502048
”/kernel/net/rose/rose_loopback.c”	2664	0	0	17430	384725
”/kernel/net/rose/rose_out.c”	3304	0	0	19452	332502
”/kernel/net/rose/rose_route.c”	27790	0	0	41641	868326
”/kernel/net/rose/rose_subr.c”	13014	0	0	28367	659800
”/kernel/net/rose/rose_timer.c”	5296	0	0	20193	478025
”/kernel/net/rose/sysctl_net_rose.c”	3247	2	2	1678330	2420935
”/kernel/net/sched/cls_api.c”	10256	4	0	696395	2284026
”/kernel/net/sched/cls_fw.c”	7610	9	3	2322282	4093440
”/kernel/net/sched/cls_route.c”	12668	17	4	4283838	9458559
”/kernel/net/sched/cls_rsvp.c”	1099	0	0	17695	474786
”/kernel/net/sched/cls_rsvp6.c”	1126	0	0	16225	290356
”/kernel/net/sched/cls_u32.c”	15331	18	4	4568648	10183982
”/kernel/net/sched/estimator.c”	4811	3	2	1146224	2335475
”/kernel/net/sched/police.c”	5330	6	1	1431368	2345152
”/kernel/net/sched/sch_api.c”	26654	13	1	2842315	5179796
”/kernel/net/sched/sch_cbq.c”	48576	50	26	21760580	70503718
”/kernel/net/sched/sch_csz.c”	25332	20	4	5124151	9785198
”/kernel/net/sched/sch_fifo.c”	4044	9	1	1576285	2509926
”/kernel/net/sched/sch_generic.c”	9336	18	9	4814633	9445890
”/kernel/net/sched/sch_prio.c”	8590	18	4	4139431	5934541
”/kernel/net/sched/sch_red.c”	9540	7	2	1956336	3529179
”/kernel/net/sched/sch_sfq.c”	11324	14	6	3912632	7564722
”/kernel/net/sched/sch_tbf.c”	9413	9	3	2326222	4321266
”/kernel/net/sched/sch_teql.c”	10957	16	3	3707442	8195760
”/kernel/net/socket.c”	33015	43	2	10782639	16495582
”/kernel/net/sunrpc/auth_null.c”	2376	8	2	1405377	2444336
”/kernel/net/sunrpc/auth_unix.c”	5735	9	2	2010807	3322669
”/kernel/net/sunrpc/auth.c”	7180	22	11	5267993	10316622
”/kernel/net/sunrpc/clnt.c”	22132	27	20	11651272	31554696
”/kernel/net/sunrpc/ping.c”	5281	7	7	2860482	5637430
”/kernel/net/sunrpc/pmap_clnt.c”	6499	8	2	2217808	3587130
”/kernel/net/sunrpc/sched.c”	28688	48	29	17801706	52579460
”/kernel/net/sunrpc/stats.c”	4638	9	4	2028365	3818838
”/kernel/net/sunrpc/sunrpc_syms.c”	3024	0	0	3074609	3514552
”/kernel/net/sunrpc/svc.c”	9041	8	3	2501469	4503166
”/kernel/net/sunrpc/svcauth_des.c”	5489	2	1	1368974	2170940
”/kernel/net/sunrpc/svcauth.c”	3637	5	4	2093599	3625041
”/kernel/net/sunrpc/svcsock.c”	24661	30	15	10924654	25905218
”/kernel/net/sunrpc/sysctl.c”	3015	0	0	195322	733046
”/kernel/net/sunrpc/xdr.c”	3844	8	3	2329081	3324712
”/kernel/net/sunrpc/xprt.c”	39399	58	37	25243790	85185984
”/kernel/net/sysctl_net.c”	1598	0	0	199347	680184
”/kernel/net/unix/af_unix.c”	37540	42	15	15066593	35362836
”/kernel/net/unix/garbage.c”	7099	7	4	2000153	4129551
”/kernel/net/unix/sysctl_net_unix.c”	1589	0	0	17759	488321
”/kernel/net/wanrouter/wanmain.c”	16767	13	0	3398121	6745934
”/kernel/net/wanrouter/wanproc.c”	14156	4	1	714644	2080998
”/kernel/net/x25/af_x25.c”	31123	0	0	45468	1260333
”/kernel/net/x25/sysctl_net_x25.c”	1986	2	2	984926	1930629
”/kernel/net/x25/x25_dev.c”	5327	0	0	20824	545611
”/kernel/net/x25/x25_facilities.c”	5873	0	0	20554	426888
”/kernel/net/x25/x25_in.c”	9660	0	0	23864	622146
”/kernel/net/x25/x25_link.c”	9546	0	0	24223	550938
”/kernel/net/x25/x25_out.c”	5580	0	0	20243	464569
”/kernel/net/x25/x25_route.c”	6151	0	0	20606	423873
”/kernel/net/x25/x25_subr.c”	9031	0	0	23324	497754
”/kernel/net/x25/x25_timer.c”	4756	0	0	19529	437962
”/kernel/scripts/ksymoops/io.c”	2573	6	2	1211436	2088204
”/kernel/scripts/ksymoops/ksymoops.c”	15674	7	5	5002463	8609840
”/kernel/scripts/ksymoops/ksyms.c”	6994	6	6	2643338	5279964
”/kernel/scripts/ksymoops/map.c”	6461	4	3	2207342	5169326
”/kernel/scripts/ksymoops/misc.c”	2602	3	1	1086933	1576003
”/kernel/scripts/ksymoops/object.c”	5472	8	6	2420063	4981415
”/kernel/scripts/ksymoops/oops.c”	38395	22	6	10705224	22177810
”/kernel/scripts/ksymoops/re.c”	3523	5	3	1635573	2787828
”/kernel/scripts/ksymoops/symbol.c”	11037	18	9	5338608	12171332
”/kernel/scripts/lxdialog/checklist.c”	9783	4	1	2094603	3377000
”/kernel/scripts/lxdialog/inputbox.c”	6179	2	1	1283600	2327746
”/kernel/scripts/lxdialog/lxdialog.c”	6023	10	0	2263019	2725994
”/kernel/scripts/lxdialog/menubox.c”	12842	4	0	2043155	2930953
”/kernel/scripts/lxdialog/msgbox.c”	2529	1	0	368891	840146
”/kernel/scripts/lxdialog/textbox.c”	15584	6	0	3197702	4224740
”/kernel/scripts/lxdialog/util.c”	9604	10	0	2568287	3072866
”/kernel/scripts/lxdialog/yesno.c”	3067	2	1	726489	1552576
”/kernel/scripts/mkdep.c”	12287	11	9	4362977	9828652
”/kernel/scripts/split-include.c”	5565	1	0	780958	1391361
”/kernel/scripts/tkcond.c”	14760	6	3	3282962	5759869
”/kernel/scripts/tkgen.c”	45436	8	7	11764374	21731166
”/kernel/scripts/tkparse.c”	18947	11	3	5036462	7827524