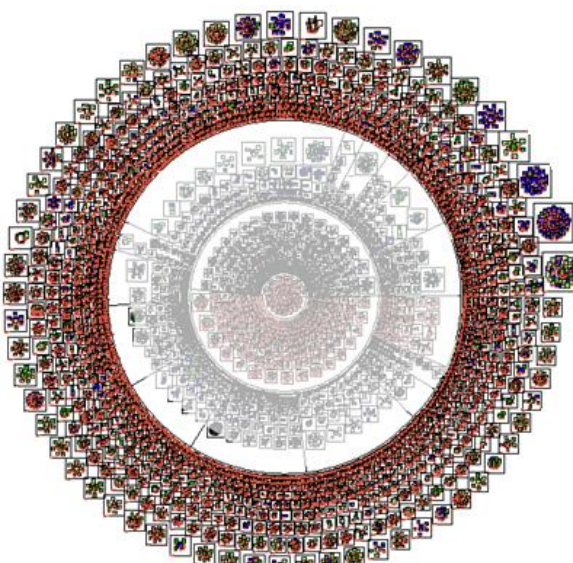


The Chair for Technology and Innovation Management (Prof. Dr. Joachim Henkel) at the TUM School of Management is offering an interdisciplinary research project (IDP):

Linux device drivers: Open or closed source, or both?

Background

Device drivers contribute the major part to the Linux Kernel's source code (see figure below). But not all device drivers are open source, many drivers are available only as binaries to be loaded during runtime as a module.



```

/*
 * abissd/abissd.c - ABISS user-space daemon
 *
 * Written 2004-2005 by Werner Almesberger
 *
 * (C) Copyright Koninklijke Philips Electronics NV 2004-2005. All rights
 * reserved. For licensing and warranty information, see the file
 * ../../COPYING.LGPLv21
 */

#include <stdint.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <fcntl.h>
#include <signal.h>
#include <errno.h>
#include <assert.h>
#include <sys/types.h>

#include <abiss_msg.h>
#include <abissd_msg.h>

#include "common.h"
#include "rm.h"
#include "var.h"
#include "stream.h"

    nt verbose = 0;

    tatic int dev_fd;

    tatic void daemonize(void)
    pid_t pid;

    pid = fork();
    if (pid < 0) {
        perror("fork");
        exit(1);
    }
    if (pid) {
        if (verbose)
            fprintf(stderr, "\nABISS daemor
        exit(0);
    }
    if (setsid() < 0) {
        perror("setsid");
    }
    
```

Left: Illustration of the Linux kernel, the complete outer ring (dark colored) is made of device drivers

Right: Source-code excerpt from a mixed open/closed-source device driver. The Philips ABISS driver enables real-time hard disk reading and writing under Linux (e.g. critical for video playback)

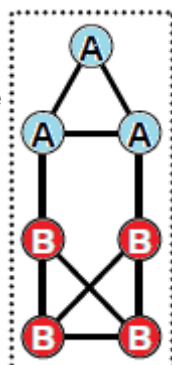
For the developing firm open source has several advantages like enabling further external enhancements and better system integration by other programmers. On the other hand, if a driver contains know-how that differentiates the company from the competition then closed source is often preferable.

There is a third approach combining the best of both worlds (like the above example of ABISS). The driver can be made modular by splitting it into an open source part and a binary containing the IP critical code, thus protecting the competitive advantage of the company. Thus, benefits from open source and closed source can both be attained. This is called **selective revealing** through an **IP-modular** architecture.

Integrated system

Selective revealing impossible

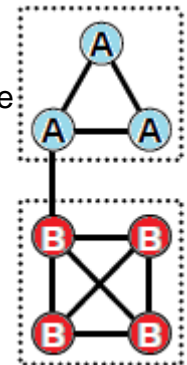
A and B can only be together
(open source)
or
(closed source)



Modular system

Selective revealing possible

A can be revealed
(open source),
and the competitive
advantage B is protected
(closed source)





Project

You would support our exploratory research. We will start with descriptive questions such as, “How many Linux device drivers are open, closed or both?”, and later analyze the drivers of developing firms in more detail including characteristics of the originator and the product category.

Main project tasks:

1. Develop a method to identify, collect and organize a large set of relevant drivers
2. Obtain drivers and relevant meta-information
3. Develop a method for automatic analysis of drivers (as far as helpful) regarding the research questions (most difficult and creative part), and collect data on the originator of the driver and the product category.

The project is an explorative study where a systematic and cautious working style is useful. Unknown issues can occur, but adaptations of the topic in coordination with the chair are possible.

Your profile

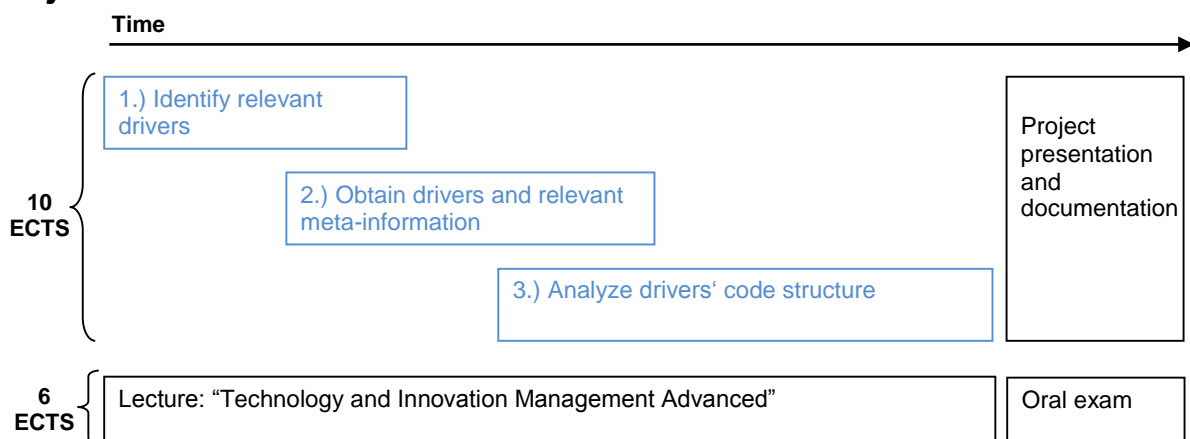
You are

- very interested in topics of Open Source, Linux and why proprietary software companies engage in Open Source
- keen to learn about the management perspective regarding Open Source, and bring the required computer skills and interest
- creative in using your knowledge, like to work structured and result-driven, and are most motivated when you understand the meaning of what you are doing

and know about

- C programming language (nice would be: coding tools and code analysis)
- Linux operating system basics (nice would be: Linux Kernel and device drivers)

Project schedule



Following IDP rules the project can be done with one student, but a group of up to 3 students would be more suitable due to the project scope.

Contact:



Dipl.-Ing. Robert Lüttke, Phone: +49 (0)89 289-25758

E-Mail: luettke@wi.tum.de,

Web: <http://www.tim.wi.tum.de/index.php?id=170>