Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320202
Date: 2015-05-08
Deadline: 2015-05-15

# Problem Sheet[1] #6

**Problem 6.1:** *filtered fuse filesystem* (5+5 = 10 points)

The Filesystem in Userspace (FUSE)[2] software enables the implementation of filesystems in user space. FUSE consists of a kernel module and an user-space library. Your task is to write a FUSE filesystem that mirrors files in an underlying filesystem but also filters the files that are visible. The FUSE distribution contains an example filesystem in `fusexmp.c` that can be used as a starting point.

a) Starting from `fusexmp.c`, create a filesystem (lets call the executable `fff`) that mirrors only a portion of the underlying real filesystem. To do this properly, read about the FUSE library support for processing options. The `fff` program should take two mandatory arguments: The first is the path in the underlying filesystem that defines the root of the mirrored filesystem and the second specifies the mount point.

b) Extend your implementation so that names of regular files visible in the `fff` are filtered by regular expressions. You should use the Perl Compatible Regular Expressions (PCRE) library[3] instead of the regular expression library that is part of the C library in order to support more powerful expressions. Make sure all system calls behave in a sensible way. The regular expression is passed as a command line option to `fff`. The option `-r <regex>` defines a regular expression constraint that file names must satisfy. The option `-x <regex>` defines a regular expression constraint that excludes any matching file names. If both options are present, the `-r` option takes precedence.

A simple example execusion may look like this:

```
$ mkdir a
$ touch a/b a/b~
$ mkdir x
$ ./fff -x '~$' ./a ./x
$ ls x
b
```

[1] See the course web page for submission instructions and grading details.
[2] http://fuse.sourceforge.net/
[3] http://www.pcre.org/