Operating systems and multiprogramming

G-assignment 5

Alexander Worm Olsen - bdj816

Allan Martin Nielsen - jcl187

Troels Thompsen - qvw203

March 17 2014

Department of Computer Science
University of Copenhagen

System calls for the Buenos file system

This task required us to implement new file handling system call, as the previous implemented syscall_read and syscall_write from G-1. The system calls we are asked to extend the system with, are open, close, create, delete, seek. The two calls, read and write, is also need to be re-implemented, but we found it unnecessary to do so, as they works as they are.
For the system calls we use the corresponding vfs-functions and make sure our filehandles is added by 2, to prevent conflicts with the predifened filehandles.

A simple shell and directory listing support

In this task we were asked to extend a shell, giving it a broader selection of commands at its disposal. The commands are listed below with a brief description of how we implemented each one of them into the shell.

- exit: We call the system call syscall_exit(), which terminates the current process of the system and therefore exits the shell.
- rm: We use the tfs_delete through vfs_remove, which frees the blocks used by the given argument, if it exists.
- cp: We read through the argument to copied and load it's content into a buffer, followed by a vfs_create with the argument to be copied to, write the buffer content into this path.
- cmp: We read through each file, bit by bit, returning the first incident the two are not equal.
- ls: To implement this command, we are asked to implement two new system calls; syscall_filecount and syscall_file. We use these to list our volume, by iterating over the files in the current volume (the argument), if the filevount is greater than zero. As long as it i not zero, we iterate through the amount fo files found, finding the index of the corresponding file and copy its name to a buffer and print these out.