

Programming Assignment 5 – UNIX File System Programming

Assigned: Tuesday, 4/25/2006, 4pm

Due: Tuesday, 5/9/2006, 4pm

Part 1. Read chapters 2, 3 and 4 of the required textbook by Bruce Molay (2002) *Understanding Unix/Linux Programming: A Guide to Theory and Practice (1st Edition)*. Try out (compile and run) the programs and make sure you understand how they work. Sources can be downloaded from the book's companion Web site (link in the CS 446/646 Web page). You might have to modify a few details in the code, depending on the platform and compiler you are using (e.g., adding `void` types, moving function prototypes out of the main, etc.) Do *not* turn in a report about Part 1.

Part 2. Complete the following programming exercises from Molay's book:

- Exercise 2.13, p70: start from `utmplib.c` and `logout_tty.c` and name your new programs `utmplib213.c` and `logout_tty213.c`, respectively.
- Exercise 3.11, p105: start from `ls2.c` and name your new program `ls311.c`
- Custom improvement: start from `ls311.c` and modify it so that it also lists one info line for any filepath argument that is *not* a directory; name the new version `ls311f.c`
- Exercise 3.18, p106: start from `ls311f.c` and name the improved version `ls318.c`
- Exercise 3.19, p106: start from `ls318.c` and name the improved version `ls319.c`
- Exercise 4.20, p138: start from `ls319.c` and name the final version `ls420.c`

Project Requirements, Submission & Required Documentation

- ✓ Every new version of a source code must include and preserve the older version as exactly as possible. When improving a program, introduce only the minimum amount of disturbance to the existing code so that a simple `>diff` command will only show the relevant changes (for example, `>diff ls311f.c ls318.c`).
- ✓ Follow all the requirement and submission guidelines exactly as prescribed in Stallings' book on pages 154 to 156, replacing any mention of shell and shell features with the above programs. These guidelines include: appropriately structuring and properly commenting your code (3.), *not* submitting executables, only the source codes above (*all* of them) and a doc file (5.), naming the makefile exactly `makefile` (which must compile *all* of the above programs in one) (6.), writing a reasonably detailed UNIX-like `readme` doc file (which must describe *only* the last command `ls420`) (2.). This documentation should describe `ls420`'s usage, options and parameters, and give a sample of a typical outcome.
- ✓ Your code must compile without any errors or warnings and run properly under Gentoo Linux, the system used in the ECC lab. You may develop and test your programs on your own UNIX machine, but it is your responsibility to ensure that they work properly on the Gentoo installation of ECC, under the default ECC Lab shell.
- ✓ Place all necessary files and documents in a compressed tarball using `tar` and `gzip` (`.tar.gz` or `.tgz`) or, alternatively, a zip file (`.zip`) (please do not use the `bzip2` format). Email this file to the instructor doursat@unr.edu before the deadline above. Do not turn in printouts in any form. Late assignments will be marked down according to the late policy published on the course Web page.