

CMSI 387-01

OPERATING SYSTEMS

Spring 2013

Assignment 0124

Alright, let's get started—we start by setting up version control, then finish up our command line work.

Outcomes

This assignment will affect your proficiency measures for outcomes *1a* and *4d–4f*.

Not for Submission

If you have not already done so in a previous course, acquire a private `git` repository. Although technically “not for submission,” it is a prerequisite for doing everything else *so there's no escape!*

Free Private GitHub Repository for Students

Acquire a GitHub account then connect to <https://github.com/edu> and request for a student account. This activity is listed first in the assignment because you might need to wait a few days for the GitHub folks to process it—so make your request as soon as possible.

Once you have gotten an account:

- Create a private repository called *cmsi387*
- Add my GitHub account, *dondi*, as a collaborator for that private repository

Git Software Setup

Install *git* on the computer(s) where you will be doing work for this course. There are many ways to install the software, so no single set of instructions can be provided. This GitHub help page can get you going:

<https://help.github.com/articles/set-up-git>

If you get stuck, ask me.

Sample Code Bazaar (optional)

If you like (and for additional `git` practice), fork <https://github.com/dondi/bazaar> and clone it to your computer. Having a fork will allow you to play freely with the sample code; you can always re-clone or revert any changes if you mess things up.

For Submission

DIY Private `git` Repository

Because GitHub provides slick web access and a bunch of additional neat services to its private `git` repositories, you will most likely prefer to use that for your coursework. However, this is operating systems after all, and it is worthwhile to learn how to set up your own personal `git` repository:

1. If you have not already done so, acquire a *my.cs.lmu.edu* account.
2. While `ssh`'ed into your *my.cs.lmu.edu* account, follow the instructions given in the [How to use `git` over `ssh`](#) article that is linked from the course website.
3. Put one or more files in that repository.
4. Send me an email telling me where I can find your repository, and what I should find there.

If you are so thrilled by this experience that you prefer to host your coursework here, just say the word and I will use that instead of GitHub.

Special Names and Symbols

The `bash` command-line shell attaches special meanings to certain names and symbols. For at least ten (10) of them, provide a brief description of what that name or symbol does or represents.

Some names and symbols to consider, in no particular order: filenames beginning with `.` (period), `~`, `/`, `\`, `.bash_profile`, `$`, `PATH`, `SHELL`, `TERM`, `USER`, `HOME`, `LANG`, `/dev/null`, `<`, `>`, `<<`, `>>`, `|`, `||`, `&`, `&&`, `;`, `..`, `!`, `!!`, `*`, `?`, `"`, `'` (to name a few). Note also that some symbols may have more than one meaning, based on their context or location. If so, pick any.

Commit your “name and symbol guide” as a document in any widely readable format as *homework/shell-guide/names-and-symbols.____* (using the appropriate file format extension).

Command “Crib Sheet”

Finally, round out the Unix command glossary that we started in class. For at least ten (10) commands, write up a document that includes:

- A brief description of what each command does
- A sample invocation of each command
- A plain English description of what that invocation does

Commit your “crib sheet” as a document in any widely readable format under the file/location *homework/shell-guide/command-crib-sheet.____* (using the appropriate file format extension).

Some commands to consider (in no particular order): cd, ls, ps, pwd, history, cp, mv, sh, mkdir, rm, rmdir, ln, fg, bg, jobs, cat, more, less, ping, traceroute, tcpdump, wget, curl, ssh, scp, telnet, cut, paste, tail, vi, emacs, nano, pico, ed, uptime, date, env, export, time, uname, top, kill, killall, du, df, hexdump, dd, od, which, locate, mount, umount, chgrp, chown, users, groups, netstat, arp, nc, echo, find, grep, whoami, open, who, wc, sort, sed, awk, rev, dig, nslookup, diff, patch, nice, make, tar, split, man, crontab, xargs, screen, sudo, script, ssh-keygen, clear