CS 1XC3 Lab 2: The Unix Filesystem

Jan. 30th - Feb. 3rd, 2023

Contents

1 Getting Set Up

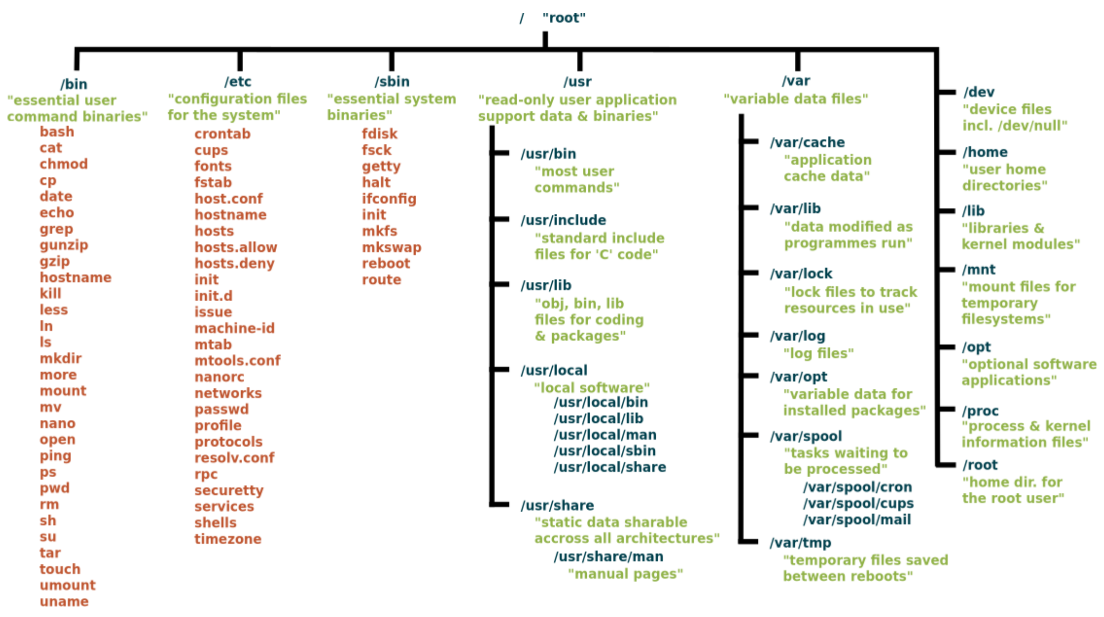
Create a directory within your CS1XC3 private github repository named "L02".

This lab is designed for command line interaction using a bash shell. You will need to clone your assignments repository to either your local file system or the `pascal.cas.mcmaster.ca` server in order to perform this lab as intended. If you need help, try this tutorial: `https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository-from-github/cloning-a-repository`

Please note that changing your github repository to public status at any time will constitute a violation of McMaster's academic integrity policy.

- You'll be assigned a single mark for each completed step in the lab procedure.
- You must submit your work via github commit operation.
- **Important Information for Mac Users**: This lab requires you to perform operations using a Unix/Linux file system. This means that opening up a bash prompt in OSx will not work for this lab. It is recommended that you:
  - ssh into pascal.cas.mcmaster.ca, instructions are on Avenue.
  - Set up a Linux virtual machine and perform the lab there.

2 Introduction: Unix FileSystem



Unix has a **standard** (specification of conventions) of directory names and layouts (starting at the root directory /). Some Un*x operating systems conform closer to the standard closer than others (for instance, Ubuntu conforms quite closely, while MacOS has alot of changes). The above figure shows the generalized standard as a diagram and provides a short description of the intended uses of certain directories. See `https://en.wikipedia.org/wiki/Unix_filesystem` for more detail.

3 Activity: Let's Create A FileSystem!

In this lab activity, we will recreate the Unix filesystem using the standard commands for basic file navigation / manipulation.

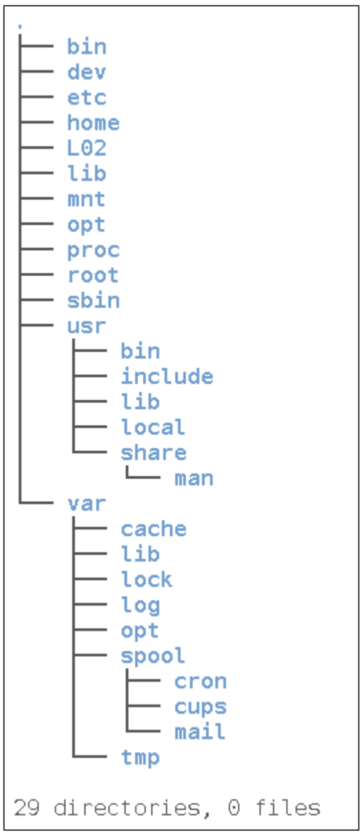
Command	**Description**
`ls <file path>`	Display the contents of the active directory
`cd <file path>`	Changes the active directory
`cd ..`	Ascend one directory in the file system hierarchy.
`cat <file path>`	Display the contents of a file
`pwd`	Display the path to the active directory
`mkdir <file path>`	Create a new directory
`cp <file path> <file path>`	Copy a file
`cp -r <file path> <file path>`	Copy a directory
`mv <file path> <file path>`	Move a file or directory
`rm <file path>`	Delete a file
`rm -r <directory path>`	Recursively delete a directory and all nested contents
`nano <file path>`	A basic text editor.
`emacs <file path>`	A better text editor.
`man <command>`	Display the manual entry for the given command.

4 Procedure

4.1 Recreate the Unix FileSystem Inside of Your Private Directory

1. Using a bash prompt, inside of the `L02` directory of your `CS1XC3` repository, recreate the Unix filesystem (just the directories, i.e none of the red files in the figure in §2).

   - You must do this without copying your system's existing directory structure stored in root, you'll get more than you bargained for!
   - When you're finished, the `tree` command should return the following output.



```
.
├── bin
├── dev
├── etc
├── home
├── L02
├── lib
├── mnt
├── opt
├── proc
├── root
├── sbin
├── usr
│   ├── bin
│   ├── include
│   ├── lib
│   ├── local
│   └── share
│       └── man
└── var
    ├── cache
    ├── lib
    ├── lock
    ├── log
    ├── opt
    ├── spool
    │   ├── cron
    │   ├── cups
    │   └── mail
    └── tmp

29 directories, 0 files
```

2. Folders with the name `bin` hold **binary files** which are **executable**. The root bin folder (i.e `/bin`) is used to store fundamental programs like those in the table in §3.

   - copy the `pwd` and `touch` programs into your `/bin` directory.
   - Here are some file path shortcuts:
     - `cd` will bring you to your user's home directory, regardless of where you are in the file system.
     - `cd /` will bring you to root, no matter where you are in the file system.
     - You can navigate multiple folders/by/putting/forward/slashes/between/them.

3. Not all programs are located in `/bin`. The `which` command will tell you where an executable is located.

   - If you don't intuitively know how `which` should probably be used, try checking out it's `man` page.
   - Find out where the `unzip` command is located.
   - Copy `unzip` into the correct **bin** directory in your simulated file system.

4. The `/home` directory contains the home directories of all the systems users. Easily enough, the name of the folder is the username of the associated user. Create a directory in `/home` for a user with your MacID as their username.

5. Some files/directories are hidden. Prefixing a file or directory name with a `.` character sets it as hidden in Unix. Files are sometimes hidden if they don't need to be modified by the user. This includes program files, such as the files and directories generated by `git init`.

   - Go to your home directory and type `ls -a` to see all files/directories, including the hidden ones.
   - Create a file `.bash_history` in the fake home directory for the user you made in the previous step.
   - Copy the first line (and only the first line) of the `.bash_history` in your real home directory (use command "cd ~" to get into your real home directory) into the one you just created.

6. The `/boot` directory contains important files for the operating system to boot, including the kernel image and configuration files. If you take a look inside, you'll find show multiple kernel images. The purpose of having multiple kernel images is so that the operating system can be booted from an older version of the kernel if the new version causes problems. Kernel problems can cause a variety of issues, including an inoperable operating system.

   - Create a boot directory in the root directory of your facsimile file system.
   - Create a kernel image file `linux-1` in this directory. You can leave this file blank.
   - Create a config file `config-1` in the same directory, and give it the following contents:

     ```
     CONFIG_DUMBOS=y
     CONFIG_x86=n
     # this is a comment
     ```

7. The `/etc` directory contains important system-wide configuration files. This includes the user database (`/etc/passwd`), the list of trusted shells (`/etc/shells`) and much more besides.

   - Copy the directory `/etc/default` into your facsimile file system. This directory contains many of your system's default parameters.

8. Directories with the name **lib** (i.e `/lib`, `/usr/lib`) are used to hold shared object files (libraries). There's generally a **lib** folder to correspond to every **bin** folder.

   - Navigate to `/usr/lib/gcc/x86_64-linux-gnu/7.5.0/include`. Gaze upon the untrammelled majesty of the C standard library.
   - Create an empty file `/lib/systemlib.so`
   - Create an empty file `/usr/lib/userlib.so`

9. Directories with the name **include** (i.e `/usr/include`) are used to hold **header files** for **C Programs**. Unix systems are built in C, so it's important to provide access to C libraries system-wide.

   - Create a file `/usr/include/stdfake.h` and give it the following contents:

     ```
     extern int fake_print (void) ;
     ```

10. The directory `/var/log` is used to hold various system logs.

    - Create a file `/var/log/syslog` and fill it with the following:

      ```
      DONE!
      ```

5 Completing this Lab

In order to submit your work, add all of the files and directories you've created to your CS1XC3 git repository.

- In order for a folder to be added to a git repository, there has to be something in it.
- For every folder that doesn't have anything in it, add a blank file `blank.txt` to that directory.
- Remember! blank files can be easily created using `touch`.

Commit and push your lab to finish this laboratory exercise. Remember, work not committed will have grades omitted!