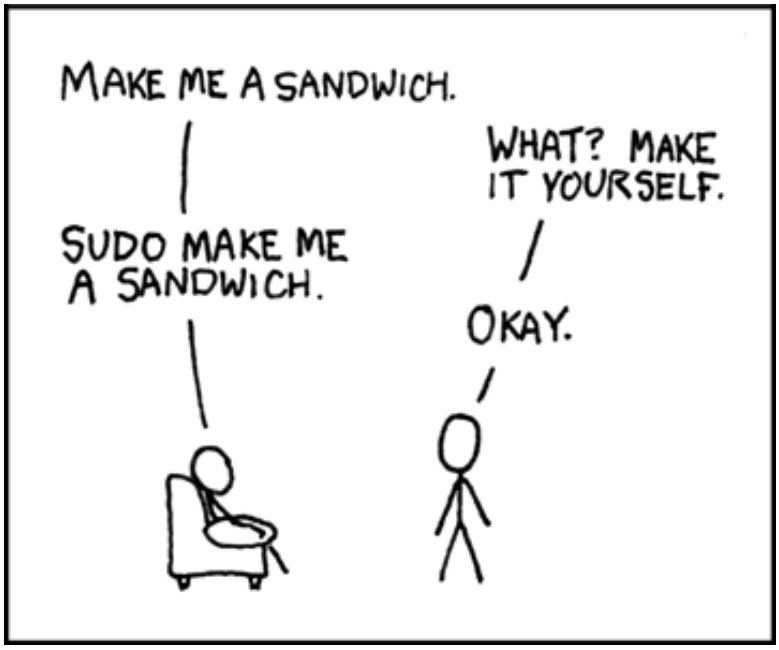
CS 251: InLab 05 [Code Warrior]: Makefile, Linux V2

Overview

The purpose of this lab is to get you more comfortable using the Linux command line interface. We also use makefiles for NOT building programs.



http://xkcd.com/149/

Tasks

1. (Action needed). Read about the `watch` command and learn how to use it. Also read about the `head` command. Simulate the `top` command using the `ps` command and the `watch` command. You should only show the top 10 cpu utilizing processes. Put this inside a script. Name the script `mytop.sh`. We will need it later. [20 Marks]
2. (Action needed). Run the following command on your terminal

   ```
   type ping
   ```

   You will get as a result the path to the executable of the ping program. Every time you issue the ping command, this executable is executed in the same way, for example, as you execute the `cs251\_base` executable to run the dominoes simulation.

   However, there is a difference in the actual invocation (simply) `ping` versus `./bin/cs251_base`. In the latter case, you have to give the entire path to get something executed. Why do you think is this the case? After all, `ping` is also yet another executable.

   Read up about environment variables and more specifically the PATH variable in Linux. Set things up (for the current terminal session) such that you can call an executable of your choice from any directory without specifying the complete verbose path to the executable.

   What did you do to achieve this? Write it in `q2.txt` [10 Marks]
3. (Action needed). In the previous lab, you did bash scripting. In Linux, the terminal is an indispensable tool for a user wishing to do anything significant. As a result, bash provides many customization options to the user. The `.bashrc` file (what does the `.` at the start of the file name signify?) is an important entity in this respect. We want you to create a well commented bashrc file in this question to achieve the following.
   (a) Using the bashrc file, display your favourite message each time you open up a new terminal window/tab.
   (b) The default prompt on your machine may look something like this

   ```
   <username>@<machine name>:<current directory path>$<space>
   ```

We want you to change this to the following

`[<current_date_time>][<current directory>] -><space>`

For example,

`[2016-05-24 11:55:23][Desktop] ->  command_typed_here`

Here the current directory should have the leading directory components stripped (see `basename`)

(c) Navigating the directory structure in bash may be frustrating. for example, I have to type `cd ../../../` to go 3 directories up. I want to be able to do the following instead:

`cd...` to mean `cd ../../`
`cd....` to mean `cd ../../../`

Achieve this using modifications to the `.bashrc` file only.

(d) Make a folder called `bashrc_gXX`, where `gXX` is the group number. Put the script you wrote in task 1 in this folder. Also put your .bashrc file in this folder. We want to be able to run the scripts you have just made in this lab from any directory you might be in.

Now write a `makefile` that has two targets, install and uninstall.

install moves any existing `.bashrc` to a file called `bashrc.bak`, copies the new .bashrc (from this lab) in place of the original and sets up the PATH variable so that `mytop.sh` script from question 1 can be called from anywhere. You *MUST* leave the script inside the folder `bashrc_gXX`.

Note: The makefile should work regardless of where in your home directory tree the `bashrc_gXX` folder is placed.

uninstall removes the new .bashrc file and restores the original .bashrc backup from `bashrc.bak`

[40 Marks]

Tasks for extra credit

1. (Action needed) Consider the situation where you maintain two sub directories in your projects directory as follows:

```
projects
    | - Computer Science
         | - <Some CS project files>
    | - Biology
         | - <Some biology project files>
```

Turns out you have completed a great project in computational biology (which is roughly both CS and biology). Where would you put that project file? Ideally you want just one copy of the file as if you have two, maintaining them is a tough job. What to do then? Links come to our rescue at this point. Create for a file of your choice hard and soft links and check if the links still work after you do each of the following

(a) Edit the original file

(b) Rename the original file

(c) Delete the original file and create a new one with the same one in its place

(d) Move the original file to some other directory

Write down your observations in the readme.txt file

[10 Marks]

2. (Action needed) You used the `/dev/null` file in your last lab. There are two other files called `/dev/zero` and `/dev/random`. Try executing the following commands on the terminal (understand what they do first)

```
dd if=/dev/zero of=~/Documents/zeros.txt bs=1K count=1 && hexdump zeros.txt

dd if=/dev/random of=~/Documents/rand.txt bs=1K count=1 && hexdump rand.txt
```

Write down what the files `/dev/zero` and `/dev/random` are for in the readme.txt file.

[10 Marks]

3. Learn how to use tmux and show your TA that you can run two separate terminals using one `ssh` command.

Challenge Questions (No credit)

WARNING: Do not run the following code on any machine. The system will crash and you would have to force reboot it.
Look at the piece of bash code below:

```
fun(){ fun|fun& }; fun
```

Try to guess why the above piece of code leads to a system crash. After figuring out how it works, see if there is any way in which you can prevent your computer from crashing if someone executes this code. [For example, this does not work on a Mac.]

And finally some fun!

Have you seen the Lord of Rings trilogy? Or even better, read the books? If not, read up about the fantasy on the internet and know who Gandalf and Bilbo are in the story. In all the files in the folder `/usr` of your Linux machine, search (remember `grep`?) for the string Gandalf or Bilbo and look at the search results. Investigate further and write down what you see.
Note: The search might take some time. You might want to stop the search after the first 10-20 results