

# basic\_linux\_commands

November 2, 2022

## 1 Basic Linux Commands

The main commands you need to learn to succeed in the Robolympics project are `cd`, `ls`, `cp`, and `pwd` along with the main `git` command: `git pull origin main`.

### 1.1 cd

`cd` means change directory. There are several forms we could use:

- `cd 345_lab_git`
  - `cd` followed by the name of a folder changes the directory into the folder
- `cd`
  - `cd` by itself takes you to your `home` folder
    - \* since your username is `pi`, your home folder is `/home/pi`
- `cd ..`
  - `..` means one level up, so this command takes you up one folder in the tree
  - if you were in `/home/pi/folder1/subfolder1`, this command would take you “up” to `/home/pi/folder1`
- `cd ~/345_lab_git`
  - `~` is a short-cut for your home folder, so this command would take you to `/home/pi/345_lab_git`
    - \* `cd 345_lab_git` only works correctly if you are already in your `home` folder
    - \* `cd ~/345_lab_git` would work from anywhere

### 1.2 ls

`ls` means list the contents of the current folder. There are many optional flags and one optional argument.

Here are some different we could use `ls`:

- `ls`
  - by itself, `ls` prints out the names of all the visible files and folders in the current directory
- `ls -a`
  - show all of the contents of the current folder, including hidden items
    - \* in Linux, any folder or filename that starts with a period `.` is hidden
      - `.git` is a hidden folder in all git repos
- `ls *.c`
  - show all `.c` files in the current folder
  - `*` is the wild card character that can match any number of characters

- \* \* can also match zero characters
- `ls -alh`
  - list the contents of the current folder showing *all* files and folders in a *long, human-readable* format
  - `-alh` is the most common flag I use
- `ls -alh *.c`
  - show all `.c` files in a long, human-readable format
  - this is how you check the modification or creation time stamp for all `.c` files

### 1.3 cp

The `cp` command copies a file:

```
cp old_file.c new_file.c
```

This can be used to create a new file with a different name in the same folder or to copy the file to a different folder:

```
cp source.c ~/myfolder/destination.c
```

### 1.4 pwd

- `pwd` means print working directory
  - tell me where I am in the file structure
  - what folder am I currently in?

### 1.5 main git command

The main `git` command used in this class is

```
git pull origin main
```

In `git` terminology, pulling means retrieving the latest files and other changes from the server. The opposite is called pushing and that refers to uploading changes from my computer to the server.

`origin` refers to the internet location of the source repository. `main` refers to the main branch of the repo.

### 1.6 At the start of lab each week

At the start of lab each week, you will probably want to execute the following commands:

```
cd 345_lab_git
```

```
git pull origin main
```

```
echo_performance.sh
```

- you should only have to type the first few letters and then hit the up arrow to cycle through possible matching commands in your command history

## 1.7 Compiling and running Raspberry Pi C code

If the gui generated a C file called `linefollow.c`, you would compile the code by going to the correct folder (using `cd`), and then executing the command

```
rpibd_build.py linefollow.c
```

That command would create an executable file called `linefollow.o`. If your terminal is in the same folder as `linefollow.o`, you would execute it using the command

```
./linefollow.o
```

## 1.8 Dr. Krauss' lab shell scripts

One powerful feature of linux is that you can write shell scripts that can be used like terminal commands. I have written several scripts to help with the lab.

```
launch_pybd_gui.py
```

```
echo_performance.sh
```

```
rpibd_build.py
```

```
upgrade_pybd_pip_stuff.sh
```

You can find all of them in the folder `~/345_lab_git/scripts_345`. You can also use the up arrow with nothing typed in to scroll through your entire command history.

[ ]: