Lesson 02: Command Line Interface (W01D2)

Balboa High School

Michael Ferraro

<mferraro@balstaff.org>

Do Now #1

- Once you start Linux, start a terminal shell by pressing `[F12]`
- Change your server account password by issuing this command and pressing `[ENTER]`:

```
smbpasswd -r apcs02 -U username
```

  - replace `username` with your GitHub username
  - the command is case sensitive, so use lowercase '`r`' and uppercase '`U`'
  - new password **must** have at least 5 characters

proceed to next slide →

Do Now #2

- Make sure you're able to connect to your locker folder using the desktop shortcut *Mount APCS/AOIT Locker*
- Start `gedit`, a text editor[a], by issuing this command at the terminal shell[b]:
  ```
  gedit &
  ```
- In `gedit`, type definitions for the following, consulting the Internet as necessary:
  1. GUI
  2. Command Prompt or Terminal Shell
  3. Wildcard
  4. Linux

---

[a] `Notepad` is an example of a Windows text editor

[b] Remember, `[F12]` starts the terminal shell

Aim

Students will learn the basics of the command line interface in Linux.

Getting Started

- In case your accounts were not prepared yesterday or you're having issues logging in, let me know as I come around.
- Partner with a neighbor and...
  - open up these lesson slides
    - browse to `http://feromax.com`
    - click the *lessons slides* link under APCS category
    - open up `L02`
  - proceed on to the slides that follow this one

History: PCs and the Command Prompt

- Most of your computer experience revolves having a GUI [graphical user interface] available to you in which all programs run inside of windows, each window with its control buttons and standard look and feel.

[a] **D**isk **O**perating **S**ystem

History: PCs and the Command Prompt

- Most of your computer experience revolves having a GUI [graphical user interface] available to you in which all programs run inside of windows, each window with its control buttons and standard look and feel.
- Before there was Windows on PCs, there was DOS[a]. Early versions of Windows were graphical programs that you would start up from DOS by typing `win`. Nowadays a PC boots directly into Windows (no command prompt).

[a] **D**isk **O**perating **S**ystem

History: PCs and the Command Prompt

- Most of your computer experience revolves having a GUI [graphical user interface] available to you in which all programs run inside of windows, each window with its control buttons and standard look and feel.
- Before there was Windows on PCs, there was DOS[a]. Early versions of Windows were graphical programs that you would start up from DOS by typing `win`. Nowadays a PC boots directly into Windows (no command prompt).
- Microsoft has retained a way for you to run a command line session called the *Command Prompt*:

  Windows button + R → `cmd`

[a] **D**isk **O**perating **S**ystem

About CLIs

- The command line interface — or CLI for short — gives you a means of interacting with (or “talking to”) the operating system using typed commands.

About CLIs

- The command line interface — or CLI for short — gives you a means of interacting with (or “talking to”) the operating system using typed commands.
- You can start programs, delete/rename/copy files, create folders, etc.

About CLIs

- The command line interface — or CLI for short — gives you a means of interacting with (or “talking to”) the operating system using typed commands.
- You can start programs, delete/rename/copy files, create folders, etc.
- Most operating systems have a CLI:

Operating System	**CLI Name**
Windows	Command Prompt
Linux	Terminal Shell
Mac OS/X	Terminal

Hands-On Activity

- Start a terminal shell.

[a] `pwd` is short for *print working directory*

Hands-On Activity

- Start a terminal shell.
- Notice the text to the left of the blinking cursor?
  `student@rm127-53:~$` ← that's the **prompt.**
  - user logged in = `student`
  - name of the workstation = `rm127-53`
  - directory/folder you're currently in = `~`

[a] `pwd` is short for *print working directory*

Hands-On Activity

- Start a terminal shell.
- Notice the text to the left of the blinking cursor? `student@rm127-53:~$` ← that's the **prompt.**
  - user logged in = `student`
  - name of the workstation = `rm127-53`
  - directory/folder you're currently in = `~`
- The directory "~" is really a shorthand for the current user's home directory. The home directory is really. . . type `pwd` to see.[a] Make sure you and your partner know the result of that command!

[a] `pwd` is short for *print working directory*

Hands-On Activity

- Let's play with the `ls` command, which lists files and folders. On one computer, type "`ls`" to see what's in the student's home directory.

[a] That's a hyphen followed by the letter 'l' and not the number 1

Hands-On Activity

- Let's play with the `ls` command, which lists files and folders. On one computer, type "`ls`" to see what's in the student's home directory.
- On the other computer, type "`ls -l`".[a]

[a] That's a hyphen followed by the letter 'l' and not the number 1

Hands-On Activity

- Let's play with the `ls` command, which lists files and folders. On one computer, type "`ls`" to see what's in the student's home directory.
- On the other computer, type "`ls -l`".[a]
- By using the **command line switch** "`-l`," you instructed the `ls` command to give you a more detailed file/folder listing with owners, dates, permissions, etc.

[a] That's a hyphen followed by the letter 'l' and not the number 1

Hands-On Activity

- Let's play with the `ls` command, which lists files and folders. On one computer, type "`ls`" to see what's in the student's home directory.
- On the other computer, type "`ls -l`".[a]
- By using the **command line switch** "`-l`," you instructed the `ls` command to give you a more detailed file/folder listing with owners, dates, permissions, etc.
- Try "`ls -la`" to see *all* files, even the ones that were hidden from view earlier.

[a] That's a hyphen followed by the letter 'l' and not the number 1

Hands-On Activity

- Many commands have a help feature. Try "`ls --help`" to see all the available options for the `ls` command.

Hands-On Activity

- Many commands have a help feature. Try "`ls --help`" to see all the available options for the `ls` command.
- By now you've seen a listing of files and folders in user `student`'s home directory. You're probably used to a more graphical view, however. All common operating systems have graphical file managers:

Operating System	**File Manager Name**
Windows	Explorer
Linux	Nautilus or Konqueror
Mac OS/X	Finder

Hands-On Activity

- Many commands have a help feature. Try "`ls --help`" to see all the available options for the `ls` command.
- By now you've seen a listing of files and folders in user `student`'s home directory. You're probably used to a more graphical view, however. All common operating systems have graphical file managers:

Operating System	File Manager Name
Windows	Explorer
Linux	Nautilus or Konqueror
Mac OS/X	Finder

- Let's see `student`'s home directory in Nautilus. In your terminal shell:

```
nautilus /home/student &
```

Hands-On Activity

- Close the Nautilus window. Open another:
  `nautilus ~ &`

Hands-On Activity

- Close the Nautilus window. Open another:
  `nautilus ~ &`
- Remember, “~” is a shorthand for `/home/student`, so Nautilus opens up the same folder.

Hands-On Activity

- Close the Nautilus window. Open another:
  `nautilus ~ &`
- Remember, "~" is a shorthand for `/home/student`, so Nautilus opens up the same folder.
- Your current working directory (recall `pwd`) is `/home/student`. Close Nautilus and issue this command:
  `nautilus . &`

Hands-On Activity

- Close the Nautilus window. Open another:
  `nautilus ~ &`
- Remember, "~" is a shorthand for `/home/student`, so Nautilus opens up the same folder.
- Your current working directory (recall `pwd`) is `/home/student`. Close Nautilus and issue this command:
  `nautilus . &`
- It turns out that the directory "." is a shorthand that always refers to the directory you're currently in. Close Nautilus and try this:
  ```
  cd /tmp
  nautilus .
  ```

Hands-On Activity

- The "`cd`" command **c**hanged **d**irectory to `/tmp`. You then started Nautilus, asking it to start in the `/tmp` folder.

Hands-On Activity

- The "`cd`" command **c**hanged **d**irectory to `/tmp`. You then started Nautilus, asking it to start in the `/tmp` folder.
- Type "`pwd`" to confirm you have, in fact, left `student`'s home directory. To quickly return to the home directory, type "`cd ~`". You can close Nautilus now.

Hands-On Activity

- The "`cd`" command **c**hanged **d**irectory to `/tmp`. You then started Nautilus, asking it to start in the `/tmp` folder.
- Type "`pwd`" to confirm you have, in fact, left `student`'s home directory. To quickly return to the home directory, type "`cd ~`". You can close Nautilus now.
- We're about to make a new folder in `student`'s home directory. Given how these workstations are configured, a reboot will remove any changes we make — so you won't be able to do any lasting damage `;)`

Hands-On Activity

- Try this:

```
mkdir animals
cd animals
touch ants.txt bats.txt beetles.txt
touch cats.txt dogs.txt
ls -l
```

Hands-On Activity

- Try this:

```
mkdir animals
cd animals
touch ants.txt bats.txt beetles.txt
touch cats.txt dogs.txt
ls -l
```

- `mkdir` is a command that **mak**es **dir**ectories

Hands-On Activity

- Try this:

```
mkdir animals
cd animals
touch ants.txt bats.txt beetles.txt
touch cats.txt dogs.txt
ls -l
```

- `mkdir` is a command that **mak**es **dir**ectories

- `touch` creates new files that are completely empty. Notice the zeroes printed out by "`ls -l`"? The zero refers to how many bytes are in each file.

Hands-On Activity

- Try this:

```
mkdir animals
cd animals
touch ants.txt bats.txt beetles.txt
touch cats.txt dogs.txt
ls -l
```

- `mkdir` is a command that **mak**es **dir**ectories

- `touch` creates new files that are completely empty. Notice the zeroes printed out by "`ls -l`"? The zero refers to how many bytes are in each file.

- Type "`nautilus ~ &`" and open up your new `animals` folder. You should see your empty files in it.

Hands-On Activity

- Leaving the Nautilus window open, try this:
  `rm cats.txt`

Hands-On Activity

- Leaving the Nautilus window open, try this:
  `rm cats.txt`
- "`rm`" **rem**oves, or deletes, files and folders. You should have seen the file disappear in your Nautilus window almost immediately after running the `rm` command.

Hands-On Activity

- Leaving the Nautilus window open, try this:
  `rm cats.txt`
- "`rm`" **rem**oves, or deletes, files and folders. You should have seen the file disappear in your Nautilus window almost immediately after running the `rm` command.
- What if we only wanted to delete files starting with the letter 'b'? Try this:
  `rm b*`

Hands-On Activity

- Leaving the Nautilus window open, try this:
  `rm cats.txt`
- "`rm`" **rem**oves, or deletes, files and folders. You should have seen the file disappear in your Nautilus window almost immediately after running the `rm` command.
- What if we only wanted to delete files starting with the letter '`b`'? Try this:
  `rm b*`
- '`*`' is a **wildcard** character, similar to wildcards in card games (often the joker card). The pattern "`b*`" *matches* any file starting with '`b`' with any ending. Confirm in Nautilus that you have deleted `bats.txt` and `beetles.txt`.

Hands-On Activity

- Let's play with your Desktop folder:
  `cd /home/student/Desktop`

Hands-On Activity

- Let's play with your Desktop folder:
  `cd /home/student/Desktop`
- Note that if you didn't use a capital '`D`' in "`Desktop`" that Linux would tell you that there's *no such file or directory.* Just as in much of programming, you have to be *really* specific about things. In Linux, directory names have to be perfectly spelled, including use of upper- and lowercase.

Hands-On Activity

- Let's play with your Desktop folder:
  `cd /home/student/Desktop`
- Note that if you didn't use a capital '`D`' in "`Desktop`" that Linux would tell you that there's *no such file or directory.* Just as in much of programming, you have to be *really* specific about things. In Linux, directory names have to be perfectly spelled, including use of upper- and lowercase.
- ```
  mkdir mug
  mkdir coffee
  cd coffee
  touch Starbucks.txt
  ```
  ```

Hands-On Activity

- Check your Desktop; make sure you see your new folders, and that there's an empty file inside the `coffee` folder.

Hands-On Activity

- Check your Desktop; make sure you see your new folders, and that there's an empty file inside the `coffee` folder.
- Currently, you should only have two file folders on your Desktop. Let's see a text-based representation of your *filesystem* as it appears from your Desktop:

```
cd ~/Desktop
tree
```

Hands-On Activity

- Check your Desktop; make sure you see your new folders, and that there's an empty file inside the `coffee` folder.
- Currently, you should only have two file folders on your Desktop. Let's see a text-based representation of your *filesystem* as it appears from your Desktop:

```
cd ~/Desktop
tree
```

- Oops, we should have created folder `coffee` inside of `mug`. You *could* drag-and-drop `coffee` into `mug`, but we're learning Linux commands today. Try this:

```
mv coffee mug
tree
```

Hands-On Activity

- As you might infer, `mv` is the **move** command. It can move folders elsewhere, move files into different folders, and even rename a file. Try this:

```
cd mug/coffee
ls
mv Starbucks.txt Peets.txt
ls
```

Hands-On Activity

- As you might infer, `mv` is the **move** command. It can move folders elsewhere, move files into different folders, and even rename a file. Try this:

```
cd mug/coffee
ls
mv Starbucks.txt Peets.txt
ls
```

- What directory are we in? (`pwd`)

Hands-On Activity

- As you might infer, `mv` is the **move** command. It can move folders elsewhere, move files into different folders, and even rename a file. Try this:

```
cd mug/coffee
ls
mv Starbucks.txt Peets.txt
ls
```

- What directory are we in? (`pwd`)
- Run "`tree`" in this directory.

Hands-On Activity

- As you might infer, `mv` is the **move** command. It can move folders elsewhere, move files into different folders, and even rename a file. Try this:

```
cd mug/coffee
ls
mv Starbucks.txt Peets.txt
ls
```

- What directory are we in? (`pwd`)
- Run "`tree`" in this directory.
- Let's go one directory *up*: "`cd ..`"

Hands-On Activity

- As you might infer, `mv` is the **move** command. It can move folders elsewhere, move files into different folders, and even rename a file. Try this:

```
cd mug/coffee
ls
mv Starbucks.txt Peets.txt
ls
```

- What directory are we in? (`pwd`)
- Run "`tree`" in this directory.
- Let's go one directory *up*: "`cd ..`"
- Run "`tree`" here. It's showing all files and folders *deeper* or `below` the current directory in the filesystem.

Hands-On Activity

- You could keep moving *up* the filesystem tree, one directory at a time, all the way to the **root** of the filesystem. You could even go up two directory levels at once if you do this: "`cd ../..`"

Hands-On Activity

- You could keep moving *up* the filesystem tree, one directory at a time, all the way to the **root** of the filesystem. You could even go up two directory levels at once if you do this: "`cd ../..`"
- There's a faster way to get all the way up to the root directory, the directory that contains all others:
  `cd /`

Hands-On Activity

- You could keep moving *up* the filesystem tree, one directory at a time, all the way to the **root** of the filesystem. You could even go up two directory levels at once if you do this: "`cd ../..`"
- There's a faster way to get all the way up to the root directory, the directory that contains all others:
  `cd /`
- `tree` ← this is showing *all* files and folders in the Linux partition on the hard disk! This could take a while. Press `[CTRL][C]` to stop the `tree` command once you're tired of waiting for it to finish...

Hands-On Activity

Here's a summary of some special directories/shorthand names we've used:

`.`	current directory
`..`	one directory up, closer to root
`~`	home directory (e.g., `/home/student`)
`/`	root directory

Hands-On Activity

- Let's go back to our Desktop folder:
  `cd ~/Desktop`

Hands-On Activity

- Let's go back to our Desktop folder:
  `cd ~/Desktop`
- We still have our `mug` folder. Let's say we want to put a bunch of system configuration files into this folder as a backup in case a virus deletes the originals. Lots of config files live in the `/etc` folder in Linux and OS/X. We can see the config files in a faraway directory without `cd`-ing to that directory:
  `ls /etc/*.conf`

Hands-On Activity

- Let's go back to our Desktop folder:
  `cd ~/Desktop`
- We still have our `mug` folder. Let's say we want to put a bunch of system configuration files into this folder as a backup in case a virus deletes the originals. Lots of config files live in the `/etc` folder in Linux and OS/X. We can see the config files in a faraway directory without `cd`-ing to that directory:
  `ls /etc/*.conf`
- Let's copy those files to folder `mug`:
  `cp /etc/*.conf mug`
  `ls mug`

Hands-On Activity

- Wait, those config files should've been in `/home/student/Desktop/mug/coffee`! Let's move these files to the right place:

```
cd mug/coffee
mv ../*.conf .
```

Hands-On Activity

- Wait, those config files should've been in `/home/student/Desktop/mug/coffee`! Let's move these files to the right place:

```
cd mug/coffee
mv ../*.conf .
```

- Click on the `mug` folder on your desktop. Nautilus should show you that there's only one folder inside, called `coffee`. Open up the `coffee` folder. You should now see all those files ending in `conf`.

Hands-On Activity

- Wait, those config files should've been in `/home/student/Desktop/mug/coffee`! Let's move these files to the right place:

```
cd mug/coffee
mv ../*.conf .
```

- Click on the `mug` folder on your desktop. Nautilus should show you that there's only one folder inside, called `coffee`. Open up the `coffee` folder. You should now see all those files ending in `conf`.

- With your partner, discuss exactly what the `mv` command above did, including what "`../*.conf`" referred to and what the single period meant. (If you two are uncertain, ask neighboring teams!)

HW

Before the class after next[a], email mferraro@balstaff.org with the subject `L02 Period 5` or `L02 Period 6`. In the email, you need to include (a) your full name and (b) a list of commands.[b]

The list of commands must modify the filesystem under `/home/student` so that typing this command —

```
tree ~/L02-HW
```

— yields this output:

```
/home/student/L02-HW/
├── apples.txt
├── bananas.txt
└── Cars
    ├── Chevy
    └── Toyota
        └── Camry.png

3 directories, 3 files
```

You should complete this activity in the APCS lab or in Rm319.

[a] That's two classes from now.

[b] Ask me about the `history` command!