First of all, these notes will cover only a small subset of the available commands and utilities, and will cover most of those in a shallow fashion.

**Read the relevant material in Sobell!**

If you want to follow along with the examples that follow, and you do, open a Linux terminal.

Second, most of the Linux commands have features that are enabled by using command-line switches; check the **man** pages for the commands for details!
Getting Started

The Linux terminal (or command shell) allows you to enter commands and execute programs.

A terminal displays a prompt and a cursor indicating it’s waiting for user input:

![Terminal Prompt](image)

The prompt itself is configurable; precisely how depends on the particular type of shell you are running.

It is likely that by default you will run the `bash` shell.
What’s Running?

The `ps` command displays information about processes the shell is currently running:

```
williammcquain@MSI-Ubuntu:~$ ps
PID TTY     TIME CMD
1594 pts/0  00:00:00 bash
1650 pts/0  00:00:00 ps
williammcquain@MSI-Ubuntu:~$ 
```

We see that two processes are executing, `bash` and `ps`.

Moreover, we see that:
- each is assigned a unique numeric identifier called a process ID or PID
- each is associated with a terminal (TTY) named pts/0

Try executing `ps` a second time… you’ll notice that the PID for `bash` is the same as before but the PID for `ps` has changed.

Why? (That’s two questions.)
More Information

Try running `ps` with the `–l` (that’s ell, not one) switch:

![Example output of `ps -l` command]

Don’t worry about the meaning of all that just yet, but do notice that the results of the `ps` command were altered by the use of a “switch” on the command line.

This is typical of Linux commands and many user programs.
The **man** (manual) command can be used to obtain more information about Linux commands:

![Linux User's Manual](image)

**NAME**

`ps` - report a snapshot of the current processes.

**SYNOPSIS**

`ps [options]`

**DESCRIPTION**

`ps` displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use `top(1)` instead.

This version of `ps` accepts several kinds of options:

1. UNIX options, which may be grouped and must be preceded by a dash.
2. BSD options, which may be grouped and must not be used with a dash.
3. GNU long options, which are preceded by two dashes.

Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many standards and `ps` implementations that this `ps` is compatible with.

Note that "ps -aux" is distinct from "ps aux". The POSIX and UNIX standards require that

---

The **man** pages are often the first resort for discovering options. Try running **man man**…
The file system is a set of data structures that organizes collections of files.

Files are grouped into *directories* (although directories are themselves files).

Here’s one possible file system organization:
File Names

Each file and directory has a name:
   - names are case-sensitive
   - names within the same directory must be unique
   - the use of characters other than letters, digits, underscores and periods tends to cause extra work when using many commands

File names are often of the form <name.ext>, such as BashTerminal.jpg.

While file extensions are not mandatory, it is standard (and good) practice to employ them.

You are required to use appropriate file extensions in this course.

It is bad practice to employ extensions incorrectly. Common ones include:
   - c  C language source files
   - h  C language header files
   - txt plain text files
   - gz file compressed with gzip
   - tar archive file created with tar
   - html hypertext markup language file
Absolute Pathnames

Each file (and directory) can be specified by a unique absolute pathname:

```
/home/hls/bin/log
```
When you open a terminal, by default you will be in your *home directory*.

Typically, this will be /home/<userid>, but you can check the path to your current directory by using the **pwd** command:

```
williammcquain@MSI-Ubuntu:~$ pwd
/home/williammcquain
williammcquain@MSI-Ubuntu:~$
```
The `ls` command lists the files in the current directory:
You can display a map of the directory tree rooted at your current directory:
Directory Navigation

You can use the **cd** command to change your current (or *working*) directory:

```
williammcquain@MSI-Ubuntu:~/Documents
williammcquain@MSI-Ubuntu:~/Documents$ cd /home/williammcquain/Documents
williammcquain@MSI-Ubuntu:~/Documents$ ls -l
```

Using **cd** with no destination moves you back to your home directory:

```
williammcquain@MSI-Ubuntu:~/Documents
williammcquain@MSI-Ubuntu:~/Documents$ cd
williammcquain@MSI-Ubuntu:~/Documents$ pwd
/home/williammcquain
williammcquain@MSI-Ubuntu:~/Documents$ cd /home/williammcquain/Documents/2505/examples
williammcquain@MSI-Ubuntu:~/Documents/2505/examples$ ls -l
```
Relative Pathnames

You can also specify a pathname that’s relative to the current (working) directory.

Let’s say you’re in a directory at the top of the tree shown below:
Relative Pathnames

There are two special directory names:

. refers to the current directory
.. refers to the parent of the current directory

Suppose your working directory is Public:

.. refers to williammcquain
../Documents/2104
../Documents/2505/examples
../Music
Making/Removing a Directory

You can create a new directory with the `mkdir` command:

You can remove an empty directory with the `rmdir` command.
You can create a copy of a file with the `cp` command.

Assume we’re in a directory containing a file named `infloop.c`:

```bash
cp infloop.c infloop2.c
```

makes a copy of `infloop.c` named `infloop2.c` in the same directory.

```bash
cp infloop.c ..
```

makes a copy of `infloop.c` with the same name in the parent directory.

```bash
cp infloop.c ../infloop2.c
```

makes a copy of `infloop.c`, named `infloop2.c`, in the parent directory.
Renaming/Moving Files: mv

As before, assume we’re in a directory containing a file named `infloop.c`:

```
mv infloop.c infiniteloop.c
```

changes the name of the file `infloop.c` to `infinitefloop.c`

```
mv infloop.c ../attic
```

moves the file `infloop.c` to the subdirectory of the parent named `attic`

```
mv infloop.c ../infinitefloop.c
```

removes the file `infloop.c` from this directory, and creates a copy named `infinitefloop.c` in the parent directory
Viewing a File: cat and less

You can use the `cat` command to display the contents of a file to the terminal:

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int pass = 0;
    while (true) {
        printf(".");
        pass++;
        if (pass == 10) {
            pass = 0;
            printf("\n");
        }
    } 
    return 0;
}
```

You can use the `less` command to display the contents of a file to the terminal, one screenful at a time.
You can view the first few lines of a file by using the `head` command:

```
williammcquain@MSI-Ubuntu:~/Documents$ head GettysburgAddress.txt
Four score and seven years ago our fathers brought forth on this continent
a new nation, conceived in liberty, and dedicated to the proposition that
all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or
any nation, so conceived and so dedicated, can long endure. We are met on a
great battle-field of that war. We have come to dedicate a portion of that
field, as a final resting place for those who here gave their lives that
that nation might live. It is altogether fitting and proper that we should
do this.
williammcquain@MSI-Ubuntu:~/Documents$
```

You can control how many lines are shown; see the `man` page.

There is an analogous `tail` command for viewing the last few lines of a file.
Searching File Contents: grep

The `grep` command can be used to display lines of a file that match a pattern:

```
williammcquain@MSI-Ubuntu:~/Documents$ grep remember GettysburgAddress.txt
will little note, nor long remember what we say here, but it can never forget
williammcquain@MSI-Ubuntu:~/Documents$
williammcquain@MSI-Ubuntu:~/Documents$ grep -n field GettysburgAddress.txt
7:great battle-field of that war. We have come to dedicate a portion of that
8:field, as a final resting place for those who here gave their lives that
williammcquain@MSI-Ubuntu:~/Documents$
williammcquain@MSI-Ubuntu:~/Documents$ grep satisfaction GettysburgAddress.txt
```

We will examine the `grep` command in considerably more detail in a future assignment.
You can obtain information about a file with the `file` command:

```
williammcquain@MSI-Ubuntu:~ Pictures
File Edit View Search Terminal Help
williammcquain@MSI-Ubuntu:~ Pictures$ ls
fairyglen-280606-002177.jpg
williammcquain@MSI-Ubuntu:~ Pictures$ file fairyglen-280606-002177.jpg
fairyglen-280606-002177.jpg: JPEG image data, JFIF standard 1.01
williammcquain@MSI-Ubuntu:~ Pictures$
```
Compressing Files: bzip2

The **bzip2** utility can frequently reduce the amount of space a file occupies:

```
williammcquain@MSI-Ubuntu:~/Documents$ ls -l
 total 12
drwxr-xr-x 3 williammcquain williammcquain 4096 2011-08-24 12:52 2104
-drwxr-xr-x 6 williammcquain williammcquain 4096 2011-08-24 14:44 2505
-rw-r--r-- 1 williammcquain williammcquain 1482 2011-08-22 22:34 GettysburgAddress.txt
williammcquain@MSI-Ubuntu:~/Documents$ bzip2 -v GettysburgAddress.txt
   GettysburgAddress.txt: 1.976:1, 4.049 bits/byte, 49.39% saved, 1482 in, 750 out.
williammcquain@MSI-Ubuntu:~/Documents$ ls -l
 total 12
drwxr-xr-x 3 williammcquain williammcquain 4096 2011-08-24 12:52 2104
drwxr-xr-x 6 williammcquain williammcquain 4096 2011-08-24 14:44 2505
-rw-r--r-- 1 williammcquain williammcquain 750 2011-08-22 22:34 GettysburgAddress.txt.bz2
williammcquain@MSI-Ubuntu:~/Documents$
```
Uncompressing a bzip2'd file: `bunzip2`

You can uncompress a file created with bzip2 by using `bunzip2`:

```
williammcquain@MSI-Ubuntu:~/Documents$ ls -l
total 12
drw-xr-x 3 williammcquain williammcquain 4096 2011-08-24 12:52 2104
drw-xr-x 6 williammcquain williammcquain 4096 2011-08-24 14:44 2505
-rw-r--r-- 1 williammcquain williammcquain 750 2011-08-22 22:34 GettysburgAddress.txt.bz2
williammcquain@MSI-Ubuntu:~/Documents$ bunzip2 GettysburgAddress.txt.bz2
williammcquain@MSI-Ubuntu:~/Documents$ ls -l
total 12
drw-xr-x 3 williammcquain williammcquain 4096 2011-08-24 12:52 2104
drw-xr-x 6 williammcquain williammcquain 4096 2011-08-24 14:44 2505
-rw-r--r-- 1 williammcquain williammcquain 1482 2011-08-22 22:34 GettysburgAddress.txt
williammcquain@MSI-Ubuntu:~/Documents$
```
Bundling Files into an Archive: tar

You can create a single file that contains a collection of files, including a directory structure with the tar utility:

```
2017: Documents > ls
2104  2505
2018: Documents > tar -cvf 2505.tar 2505
2505/
2505/ice/
2505/infloop.c
2505/GettysburgAddress.txt
2505/ooc/
2505/examples/
2505/examples/infloop.c
2505/examples/sleeper2
2505/examples/gp.txt
2505/examples/sleeper2log.txt
2505/examples/cp2.txt
2505/examples/sleeper2.c
2505/examples/gp2.txt
2505/examples/infloop
2505/examples/sleeper.c
2505/examples/sleeper
2505/tests/
2019: Documents > ls
2104  2505  2505.tar
2020: Documents >
```

**-cvf**  
create, be verbose, write to a file

Note the name of the new tar file is listed before the target (files to be tar'd up).

DO NOT get that backwards!
Checking Contents

You can create check the contents of a tar file:

```
$ tar -tf 2505.tar
2021: Documents > tar -tf 2505.tar
2505/
2505/ice/
2505/infloop.c
2505/GettysburgAddress.txt
2505/ooc/
2505/examples/
2505/examples/infloop.c
2505/examples/sleeper2
2505/examples/gp.txt
2505/examples/sleeper2log.txt
2505/examples/cp2.txt
2505/examples/sleeper2.c
2505/examples/gp2.txt
2505/examples/infloop
2505/examples/sleeper.c
2505/examples/sleeper
2505/tests/
2022: Documents >
```

`-tf` table of contents of a file
Creating a Compressed tar File

You can also create a compressed tar file:

```
2104  2505  2505.tar
2027: Documents > tar -czf 2505.tgz 2505
2028: Documents > ls -l
  total 68
  -rw-r--r-- 1 williammcquain vboxusers 4096 2012-08-13 20:18 2505
  -rw-r--r-- 1 williammcquain vboxusers 51200 2012-09-04 20:23 2505.tar
  -rw-r--r-- 1 williammcquain vboxusers  5425 2012-09-04 20:38 2505.tgz
2029: Documents >
2029: Documents >
```

Note that the resulting file is about 11% as large as the uncompressed version.

The `z` switch causes `tar` to apply the `gzip` utility to compress the results.

Alternatively, the `j` switch causes `tar` to apply `bzip2`.
Extracting a tar File

Use the `x` switch to extract the contents of a `tar` file:

```
2037: tarexample > tar -xf 2505.tar
2038: tarexample > tree

2505
  examples
    cp2.txt
    gp2.txt
    gp.txt
    infloop
    infloop.c
    sleeper
    sleeper2
    sleeper2.c
    sleeper2log.txt
    sleeper.c
  GettysburgAddress.txt
    ice
    infloop.c
    ooc
    tests
  2505.tar

5 directories, 13 files
```