UNIX LAB

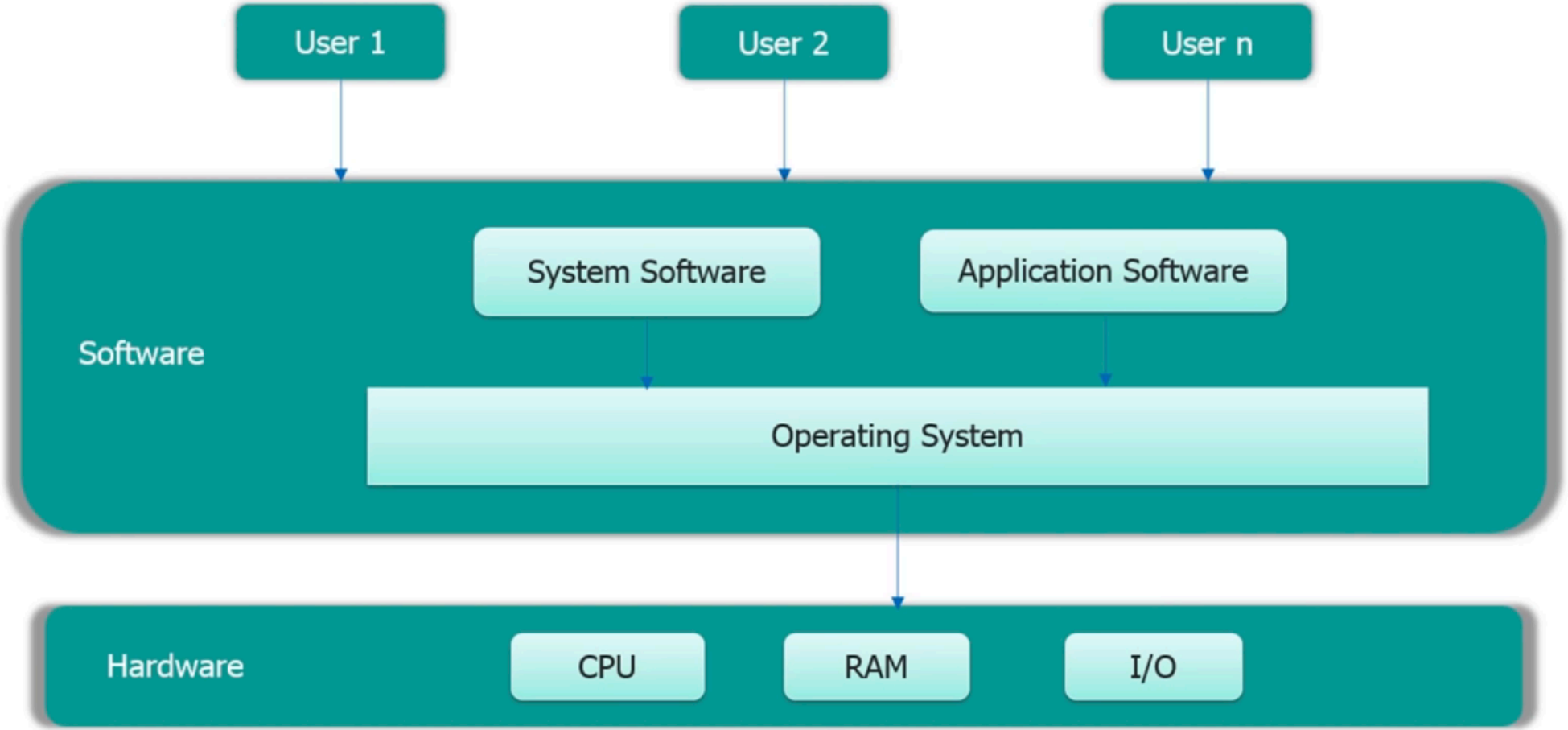
By,

Himani Deshpande

Unix Lab

- **LO 1 -:** Students will be able to understand the architecture and functioning of Unix
- **LO 2**-: Students will be able to identify the basic Unix general purpose commands.
- **LO 3-:** Students will be able to execute unix commands for system administrative tasks such as file management and user management.
- **LO 4-:** Students will be able to execute unix commands for system administrative tasks such as process management and memory management.
- **LO 5-:** Students will be able to implement shell scripts for different applications.
- **LO 3-:** Students will be able to implement **advance scripts** using the awk, grep, sed and perl scripts for performing various tasks.

UNIX OS


User 1
User 2
User n
System Software
Application Software
Software
Operating System
Hardware
CPU
RAM
I/O

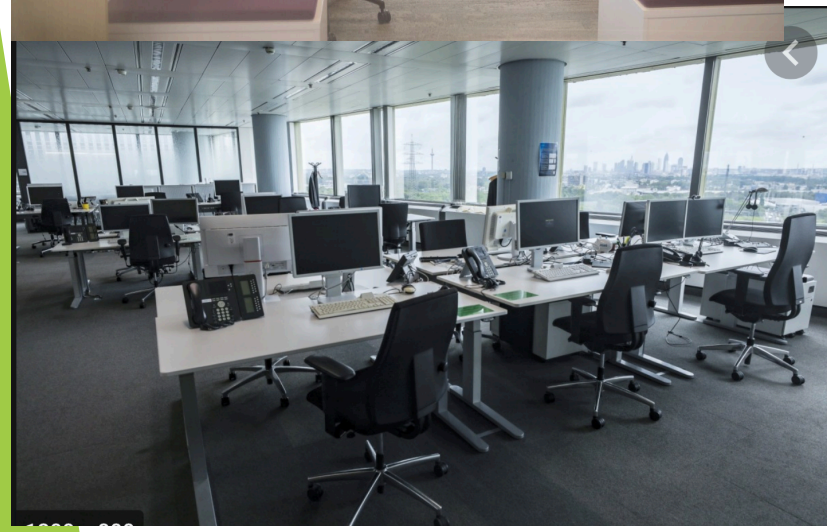
Operating System

- Operating system is a set of programs which acts as an interface between between users and computer.
- It is the heart of any machine's software and provides environment in which a user can execute programs.
- It controls the allocation of resources and services such as memory, processor, devices , information etc.

- Popular OS are
  DOS
  Windows
  MAC OS
  Unix/Linux

OS example

- Employees are application software they need to perform.
- Office building provides the environment.
- If the office has facilities like cleanliness, proper ventilation , lift, canteen, welding machine, other resources.
- Environment effects efficiency.



Without OS

If you want to play a game you need to design it to suit to your hardware.

You need to design whole environment.

Your code cant port that game to another machine .

Power of OS

- IBM launched its DOS coded by Bill Gates.
- Bill gates became the richest man, developing Windows Operating systems.

Assignment 1

- Q1. List operating system services.
- Q2. List and explain with a diagram different operating system structures with an example for each.

UNIX OS

- Unix is a multi-user, multi tasking and multi processing OS.
- It is a Command line Interpreter
- Developed at AT&T Bell Laboratories Research Center, USA in 1969 by Ken Thompson and Denis Ritchie.
- Earlier Unix was written in Assembly language and originally spelled as UNICS.
- Later it was re-written in ‘C’ language and named as Unix.
- Some Unix OS are open source and some are not.

Why UNIX

- Completely Free
- Comparatively secured
- Could run smoothly on any machine(hardware), even on your old computer .

• What Is UNIX?

- UNIX is a computer operating system, a control program that works with users to
  - run programs,
  - manage resources, and
  - communicate with other computer systems.

- Several people can use a UNIX computer at the same time; hence UNIX is called a multiuser system. Any of these users can also run multiple programs at the same time; hence UNIX is called multitasking.

Unix variants

- Unix has a number of variants but they all follow almost all the Unix terminal commands

Unix Operating System supports the following basic features and capabilities :

- **Multi-user** : More than one user can use the machine simultaneously supported via terminals
- **Multi-tasking** : Multiple programs can be run at a time
- **Multi-process** : Each user can execute several processes simultaneously
- **Hierarchical Structure** : Unix directories are present like a tree structure to support the organization and maintenance of files
- **Open System** : Some of the Unix OS are open-source. Users can modify the Unix source code
- **Portability** : It is the ability of the software that operates from one machine to another machine having different configuration Unix allows users to transfer data from one system to another
- **Programming Facility** : Unix Shell can be used as a Programming/Scripting Language
- **Communication Facility** : Unix allows communication between different users by providing some information
- **Security** : Unix has system level security controlled by system administrator and file level security controlled by owner of the file
- **Tools and Utilities** : Supports many of the tools, libraries and utilities to aid software development
- **Piping** : In Piping concept, the output of the first command becomes the input of the next command/process
- **Help Facility** : In Unix, 'man' command is used to view help content on any command
- **Modularity** : Unix consists of multiple number of independent modules or programs which perform different elementary tasks

Installation and Execution Options

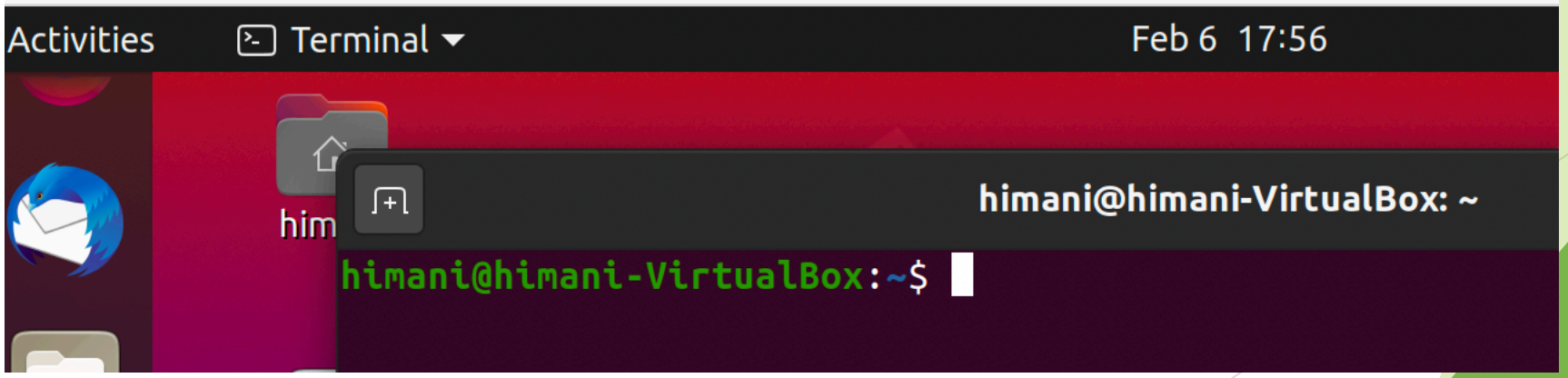
1. Host OS :-
   a. Single Boot - Ubuntu 20.04 LTS / BOSS Linux 8.0
   b. Dual Boot along with Windows - Ubuntu 20.04 LTS
2. Guest OS/Virtual OS
   a. VirtualBox
      Installation
      ISO IMAGE
   b.. VMware
      Installing VMware on Windows
      Ubuntu ISO image
      How to create ubuntu VM in VMware
3. Ubuntu Terminal App for Windows
4. Cloud OS - onworks.net
5. Termux App for Android - Google Play, F-Droid, Kali Nethunter Store
6. Virtual Labs

Terminal

Ctrl+Alt+T

Terminal provides an interface into which users can type commands and that can print text.

The terminal outputs the results of **commands** which are specified by the user itself. Execution of typed **command** is done only after you press the Enter key.



- Unix is also case-sensitive. This means that ***cat*** and ***Cat*** are different commands.
- The prompt is displayed by a special program called the **shell**.
- **Shells accept** commands, and **run** those commands.
- They can also be programmed in their own language. These programs are called "**shell scripts**".

BASIC UNIX COMMANDS

- echo,
- clear,
- exit,
- date,
- time,
- uptime: **Uptime** is a **command** that returns information about how long your system has been running together with the current time
- Cal
- man,

Cal

```
himani@himani-VirtualBox:~$ cal
   February 2021
Su Mo Tu We Th Fr Sa
     1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28
```

```
himani@himani-VirtualBox:~$ cal 2 2021
   February 2021
Su Mo Tu We Th Fr Sa
     1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28
```

```
himani@himani-VirtualBox:~$ date "+Date: %m%d%y%n Time: %H%M%S"
Date: 020621
 Time: 182005
```

```
himani@himani-VirtualBox:~$ who
himani   :0           2021-02-06 17:40 (:0)
himani@himani-VirtualBox:~$ whoami
himani
himani@himani-VirtualBox:~$ who am i
himani@himani-VirtualBox:~$ passwd
Changing password for himani.
Current password:
New password:
Retype new password:
```

Cat Command

- **cat filename**
  It will show content of given filename
- **$cat file1 file2**
  This will show the content of file1 and file2.
- **$cat -n filename**
  It will show content with line number example:-cat-n geeks.txt
  1)This is geeks
  2)A unique array
- **$ cat >newfile**
  Will create and a file named newfile
- **$cat [filename-whose-contents-is-to-be-copied] > [destination-filename]>**
  The content will be copied in destination file
- **$cat -e file**
  '$' is shows at the end of line
- **$cat file1 > file2**
  Will append the contents of one file to another file

tty command

- Since UNIX treats even terminals as files. It is tty (teletype) command, that tells you the filename of the terminal you are using. This command is simple and need no arguments:

  $tty <enter>

  /dev/tty01

- The terminal filename tty01 resident in the /dev directory. If the user logs in from another terminal next time, his terminal device name will be different.

tty – – version : Prints the version information

- which : which returns the pathnames of the files which would be executed in the current environment. Eg which cal
- history : history of executed commands
- pwd : present working directory
- whoami : user logged in

passwd

- With the passwd command, you can change the password associated with your individual account name.

- For example,

```
sariyer:~> passwd
Changing password for dag.
Old password:
New passwd:
Retype new passwd:
sariyer:~>
```

- pr: convert text file for printing
- lp: The lp command is used to print **files** on Unix and Linux systems. The name "lp" stands for "line printer".
- lpr: lpr submits **files** for printing.
- lpstat: **lpstat** displays status information about the current classes, jobs, and printers.
- lpq: shows printer queue status
- lprm: cancels printer queued job
- Cancel: cancel printer current job.
- mail, etc.

- id : id command in **Linux** is used to find out user and group names and numeric ID's (UID or group ID) of the current user or any other user in the server.
  This command is useful to find out the following information as listed below:
  1. User name and real user id.
  2. Find out the specific Users UID.
  3. Show the UID and all groups associated with a user.
  4. List out all the groups a user belongs to.
  5. Display security context of the current user.
- ping: **PING** (Packet Internet Groper) **command** is used to check the network connectivity
- ifconfig,

Shell Commands of UNIX

Unix Commands

- When you first log into a unix system, you are presented with something that looks like the following:

```
/home/larry#
```

- That "something" is called a **prompt**. As its name would suggest, it is prompting you to enter a command.
- Every unix command is a sequence of **letters**, **numbers** and **characters**. But there are no spaces.

- There are two major types of shells in **unix**:
  - Bourne shells
  - C shells
- *Steven Bourne* wrote the original unix shell sh and most shells since then end in the letters sh to indicate they are extentions on the original idea
- Linux comes with a Bourne shell called bash written by the Free Software Foundation.
- bash stands for **B**ourne **A**gain **Sh**ell and is the default shell to use running linux

- When you first login, the prompt is displayed by bash, and you are running your first unix program, the bash shell.

- As long as you are logged in, the *bash shell* will constantly be running.

Unix Commands

obtaining help

- The man command displays reference pages for the command you specify.
- The UNIX man pages (man is short for manual) cover every command available.
- To search for a man page, enter man followed by the name of the command to find .
- For example:

```
bagriy@sariyer:~> man ls
```

To exit
Press “q”

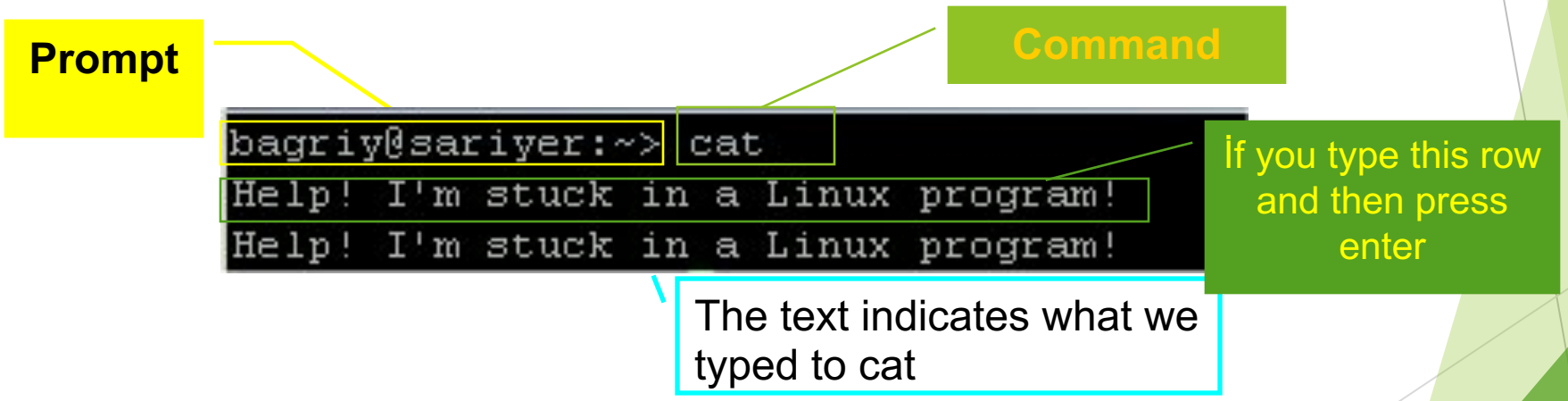
• man (*obtaining help*)

- There is also a keyword function in man.
- For example;
  - If you are interested in any commands that deal with Postscript, the printer control language for Adobe
  - Type **man -k ps** or **man -k Postscript**,

    you'll get a listing of all commands, system calls, and other documented parts of unix that have the word "ps" (or "Postscript") in their name or short description.
- This can be very useful when you're looking for a tool to do something, but you don't know it's name-or if it even exists!

• cat

- cat command is used to concatenate or displays the contents of a file.
- To use it, type cat, and then press enter key:



• To end many unix command, type end-of-file command (EOF) [*hold down the key labeled* "Ctrl" *and press* "d" **(Ctrl+d)**]

- To display the contents of a file, type
  cat *filename*

```
bagriy@sariyer:~/EST_guz_2003/hafta_1> cat program1.c
/* C programlama
ilk program */
#include<stdio.h>
int main()
{
printf("ilk C programimiz \n");
return 0;
}
bagriy@sariyer:~/EST_guz_2003/hafta_1>
```

- To see linux commands press **Tab** key,
- If you want to learn commands beginning with c you can write c then press Tab key

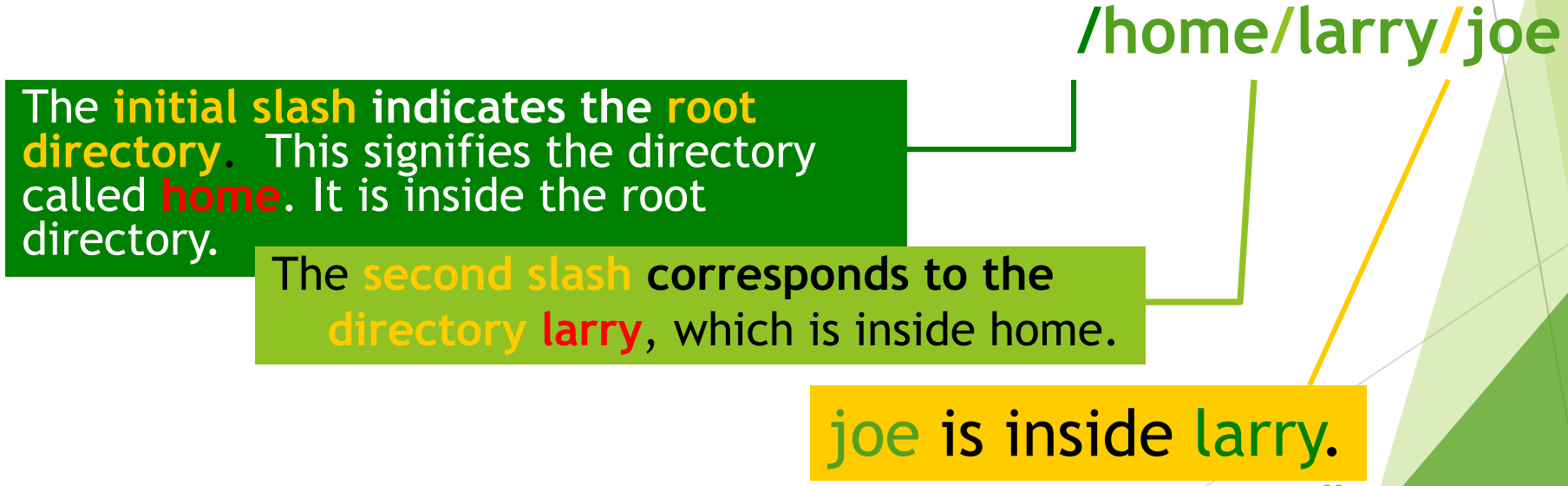
/home/larry# c

```
c++             chage           codepage        continue
c++decl         charset         col             control-panel
c++filt         chattr          colcrt          convert_smbpasswd
c2ph            checkalias      collateindex.pl cp
c_rehash        chfn            colrm           cpio
cal             chgrp           column          cpp
calibrate_ppa   chmod           comm            cproto
cancel          chown           command         crontab
captoinfo       chsh            comp            csh
card            chvt            comp_err        csplit
case            ci              compgen         ctags
cat             cjpeg           compile_et      cut
catchsegv       cksum           complete        cvs
cc              clear           composeglyphs   cvsbug
cd              cmp             compress        cxpm
cdecl           cmuwmtopbm      consolechars    cytune
chacl           co              consolehelper
```

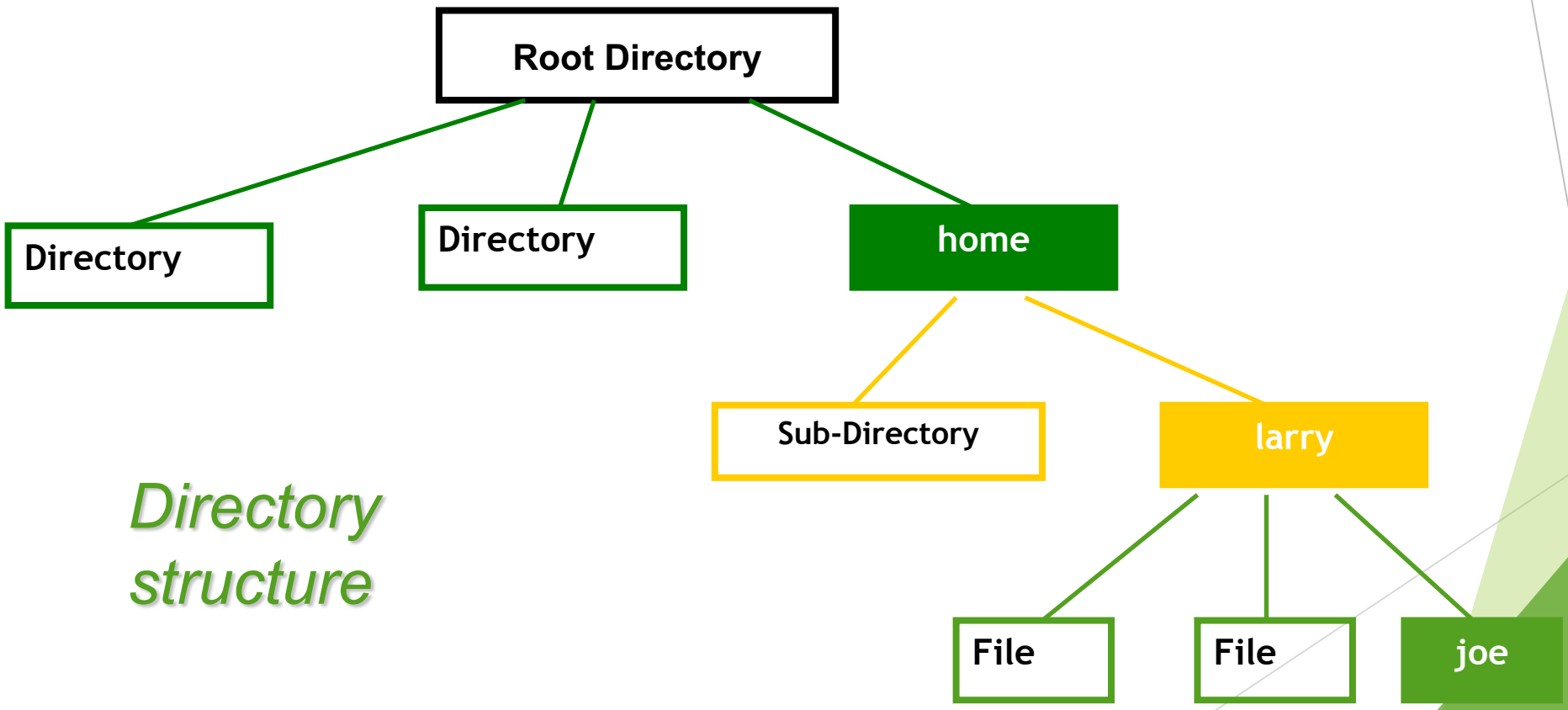
Storing information

- Unix provides files and directories.
- A directory is like a folder: it contains pieces of paper, or files.
- A large folder can even hold other folders-*directories can be inside directories*.
- In unix, the collection of directories and files is called the file system. Initially, the file system consists of one directory, called the **"root"** directory
- Inside "root" directory, there are more directories, and inside those directories are files and yet more directories.

- Each file and each directory has a name.
- A short name for a file could be joe,
- while it's "full name" would be **/home/larry/joe.** The full name is usually called the **path**.
- The path can be divide into a sequence of directories.
- For example, here is how **/home/larry/joe** is read:



- A path could refer to either a directory or a filename, so joe could be either.
- All the items before the short name must be directories.



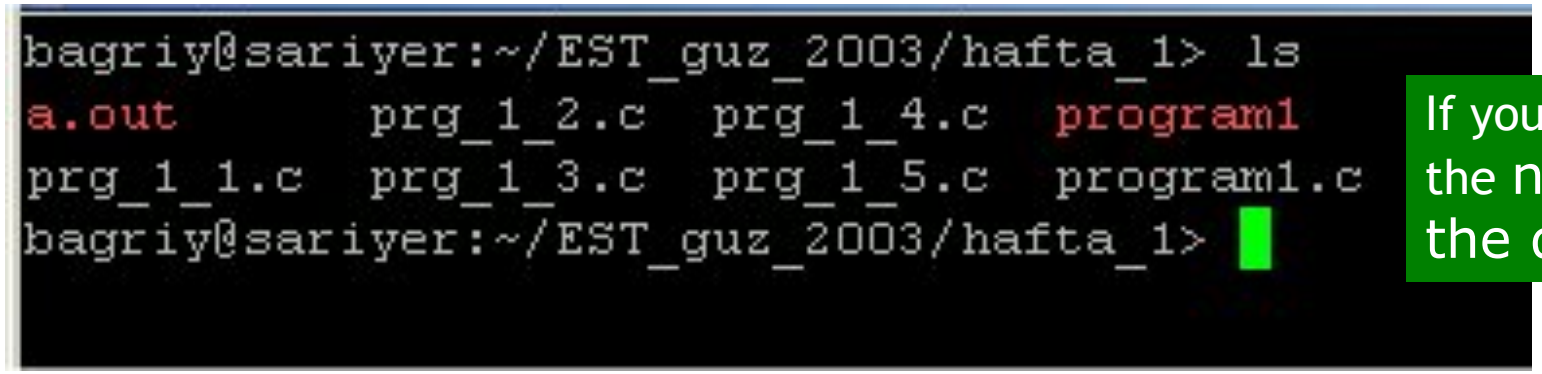
Directory structure

Looking at directories with ls

- The command ls lists files.
- If you try ls as a command, you'll see:

```
/home/larry#  ls
/home/larry#
```

That is right, you will see nothing.



If you have files, ls lists the names of files in the directory

- If you want a list of files of a more active directory, try the root directory.

```
/home/larry#  ls /
  bin   etc      install  mnt   root user var
  dev  home  lib      proc  tmp  usr   vmlinux
```

"/" is a parameter saying what directory you want a list for.

Some commands have special parameters called options or switches.
To see this try:

```
/home/larry#  ls -F /
  bin    etc/      install/ mnt/   root/  user/ var/
  dev/  home/  lib/       proc/  tmp/  usr/   vmlinux/
```

The **-F is an option**. It displays file types.

- An option is a special kind of parameter that starts with a **dash "-"**
- An option modifies how the program **runs**, but not what the program runs on.
- For **ls**, **-F** is an **option** that lets you see which ones are **directories**, which ones are special **files**, which are **programs**, and which are normal files.
- Anything with a **slash "/"** is a **directory**.
- ls -l file* displays files starting with "file"
- ls -l displays all details

```
bagriy@sariyer:~/EST_guz_2003/hafta_1> ls -l
total 56
-rwxr-xr-x    1 bagriy   users       13495 Eki  3 20:41 a.out
-rw-r--r--    1 bagriy   users         115 Eki  3 20:09 prg_1_1.c
-rw-r--r--    1 bagriy   users         424 Eki 11  2002 prg_1_2.c
-rw-r--r--    1 bagriy   users         215 Eki 11  2002 prg_1_3.c
-rw-r--r--    1 bagriy   users         201 Eki 11  2002 prg_1_4.c
-rw-r--r--    1 bagriy   users         324 Eki 11  2002 prg_1_5.c
-rwxr-xr-x    1 bagriy   users       13495 Eki  3 20:41 program1
-rw-r--r--    1 bagriy   users         107 Eki  3 20:41 program1.c
bagriy@sariyer:~/EST_guz_2003/hafta_1>
```

- Many unix commands are like ls.
- They have options, which are generally one character after a dash, and they have parameters.
- Unlike ls, some commands require certain parameters and/or options. You have to learn these commands.

- pwd
  - **pwd** (**p**resent **w**orking **d**irectory) tells you your current directory.
    - *Most commands act, by default, on the current directory. For instance, **ls** without any parameters displays the contents of the current directory.*
- **cd**
  - **cd** is used to **c**hange **d**irectories.
  - The format of this command :

    cd new-directory (where new-directory is the name of the new directory you want).

- For instance, try:

```
/home/larry#  cd /home
/home#
```

• If you **omit the optional parameter** directory, you're **returned to your home**, or original directory. Otherwise, cd will change you to the specified directory.

- There are two directories used only for relative pathnames:
  - The directory "." refers to the current directory
  - The directory ".." refers to the parent directory
- These are "shortcut" directories.
- The directory ".." is most useful in "backing up":

```
/usr/local/bin# cd ..
/usr/local#
```

- mkdir

mkdir (**m**a**k**e **dir**ectory) is used to create a new directory,

- It can take more than one parameter, interpreting each parameter as another directory to create.

- rmdir

rmdir (**r**e**m**ove **dir**ectory) is used to remove a directory,

- rmdir will refuse to remove a non-existant directory, as well as a **directory that has anything in it**.

Moving Information

- The primary commands for manipulating files under unix are cp, mv, and rm. They stand for **c**o**p**y, **m**o**v**e, and **r**e**m**ove, respectively.

cp

- cp is used to copy contents of file1 to file2

  cp file1 file2 (*contents of file1 is copied to file2 in the same directory*)

  cp folder1/file1 folder2 (*contents of file1 is copied to file1 in the inside of folder2 directory*)

- rm

- rm is used to remove a file.
  - rm *filename* ---> removes a file named *filename*

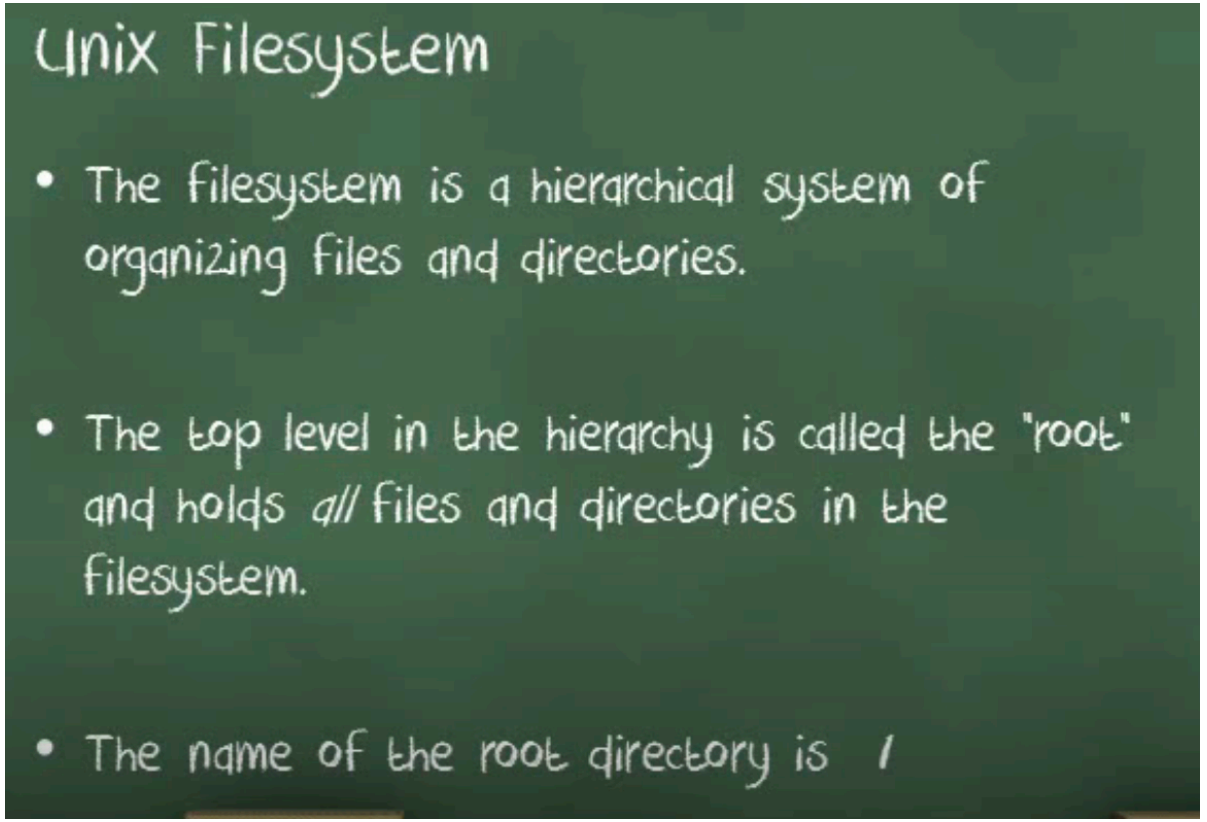
- mv

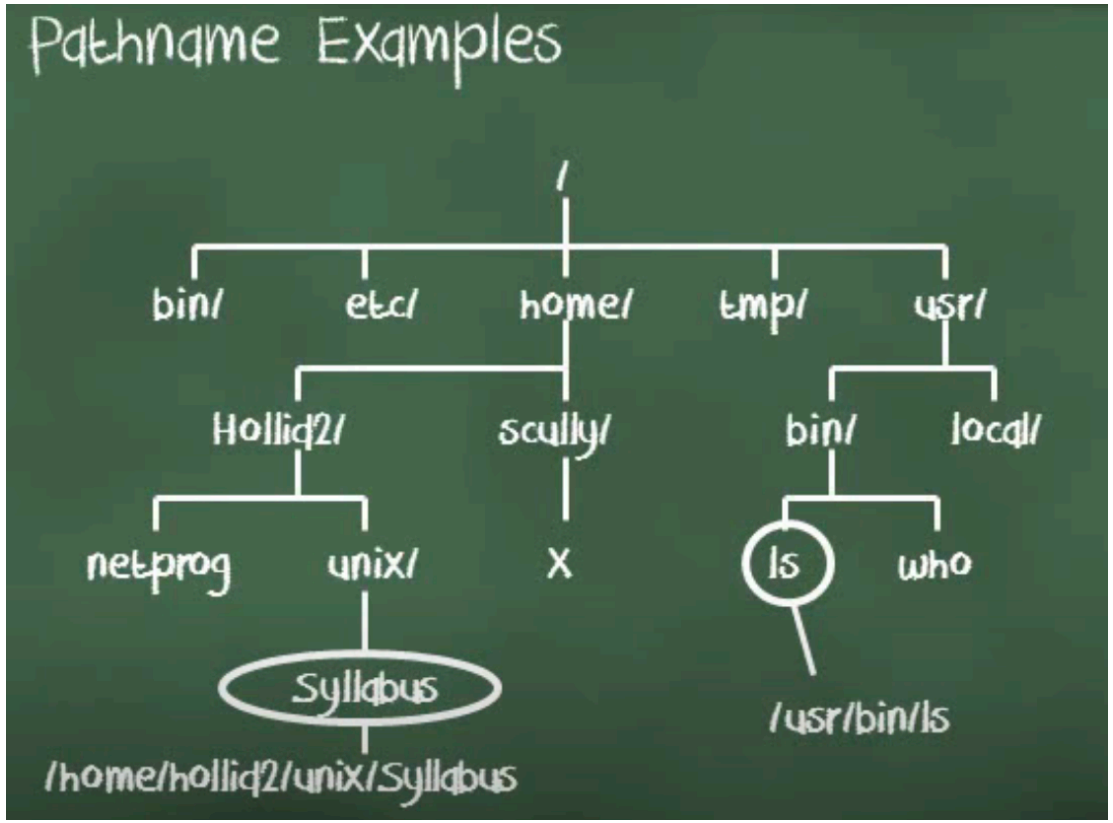
- mv is used to move a file.
  - rm *filename* ---> removes a file named *filename*
- looks like cp, except that it deletes the original file after copying it.
- mv will rename a file if the second parameter is a file. If the second parameter is a directory, mv will move the file to the new directory, keeping it's shortname the same.

- A file is a basic unit of storage (usually storage on a disk).
- Every file has a name.
- Filenames are case-sensitive!
- Unix file names can contain any characters (although some make it difficult to access the file) except the null character and the slash (/).
- Unix file names can be long!
  - how long depends on your specific flavor of Unix

Directories

- A directory is a special kind of file - Unix uses a directory to hold information about other files.
- We often think of a directory as a container that holds other files (or directories).
- A directory is the same idea as a *folder* on Windows.


Unix Filesystem
• The filesystem is a hierarchical system of organizing files and directories.
• The top level in the hierarchy is called the "root" and holds all files and directories in the filesystem.
• The name of the root directory is /


Pathname Examples
/
bin/
etc/
home/
tmp/
usr/
Hollid2/
scully/
bin/
local/
netprog
unix/
X
ls
who
Syllabus
/usr/bin/ls
/home/hollid2/unix/Syllabus

Unix command

- **Unix commands** are inbuilt programs that can be invoked in multiple ways.
- We work with these **commands** interactively from a **Unix** terminal.
- A **Unix** terminal is a graphical program that provides a **command**-line interface using a shell program.
- The **Unix command line** is a text interface to your computer. Often referred to as the shell, terminal, console, prompt .
- The **Unix s**hell has been around longer than most of its users have been alive. It has survived so long because it’s a power tool that allows people to do complex things with just a few keystrokes.

Unix file identification

- **White (No colour code)** Regular File or Normal File
- **Blue** Directory
- **Bright Green:** Executable File
- **Bright Red:** Archive file or Compressed File
- **Magenta** Image File
- **Cyan** Audio File
- **Sky Blue** Symbolic Link File
- **Yellow with black background** Device

Basic Unix Command

File Commands
ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1* *file2* – copy *file1* to *file2*
cp -r *dir1* *dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1* *file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file* *link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Basic Unix Command

System Info
date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger ***user*** – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man ***command*** – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis ***app*** – show possible locations of *app*
which ***app*** – show which *app* will be run by default

Basic Unix Command

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*

tar xf *file.tar* – extract the files from *file.tar*

tar czf *file.tar.gz files* – create a tar with Gzip compression

tar xzf *file.tar.gz* – extract a tar using Gzip

tar cjf *file.tar.bz2* – create a tar with Bzip2 compression

tar xjf *file.tar.bz2* – extract a tar using Bzip2

gzip *file* – compresses *file* and renames it to *file.gz*

gzip -d *file.gz* – decompresses *file.gz* back to *file*

Process Management

ps – display your currently active processes

top – display all running processes

kill *pid* – kill process id *pid*

killall *proc* – kill all processes named *proc* *

bg – lists stopped or background jobs; resume a stopped job in the background

fg – brings the most recent job to foreground

fg *n* – brings job *n* to the foreground

Basic Unix Command

File Permissions

chmod ***octal file*** – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:

chmod 777 – read, write, execute for all

chmod 755 – rwx for owner, rx for group and world

For more options, see **man chmod**.

Shortcuts

Ctrl+C – halts the current command

Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background

Ctrl+D – log out of current session, similar to **exit**

Ctrl+W – erases one word in the current line

Ctrl+U – erases the whole line

Ctrl+R – type to bring up a recent command

!! - repeats the last command

exit – log out of current session