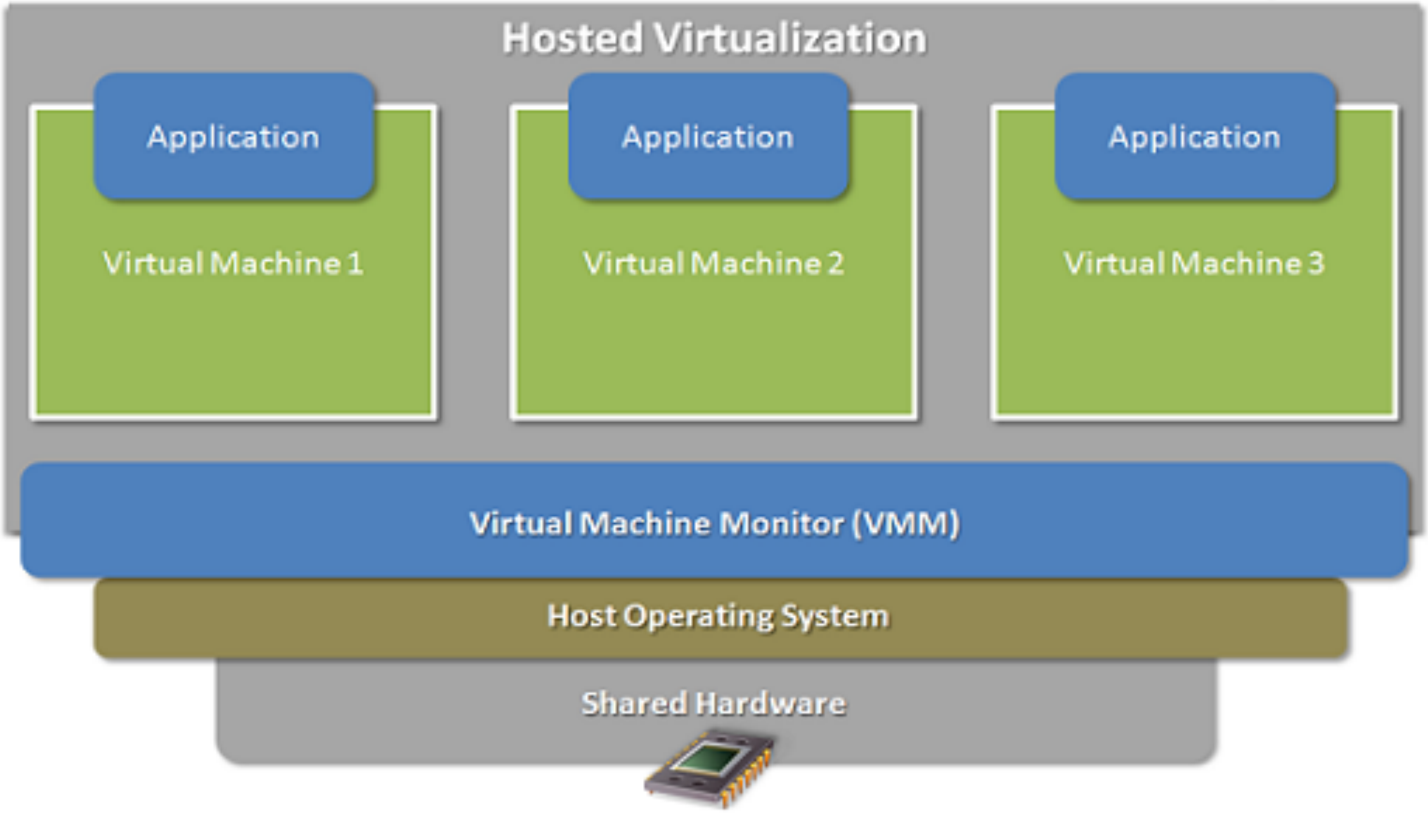
LECTURE 2: LINUX BASICS

Computer Systems and Networks

Dr. Pallipuram
(vpallipuramkrishnamani@pacific.edu)

Hosted Virtualization

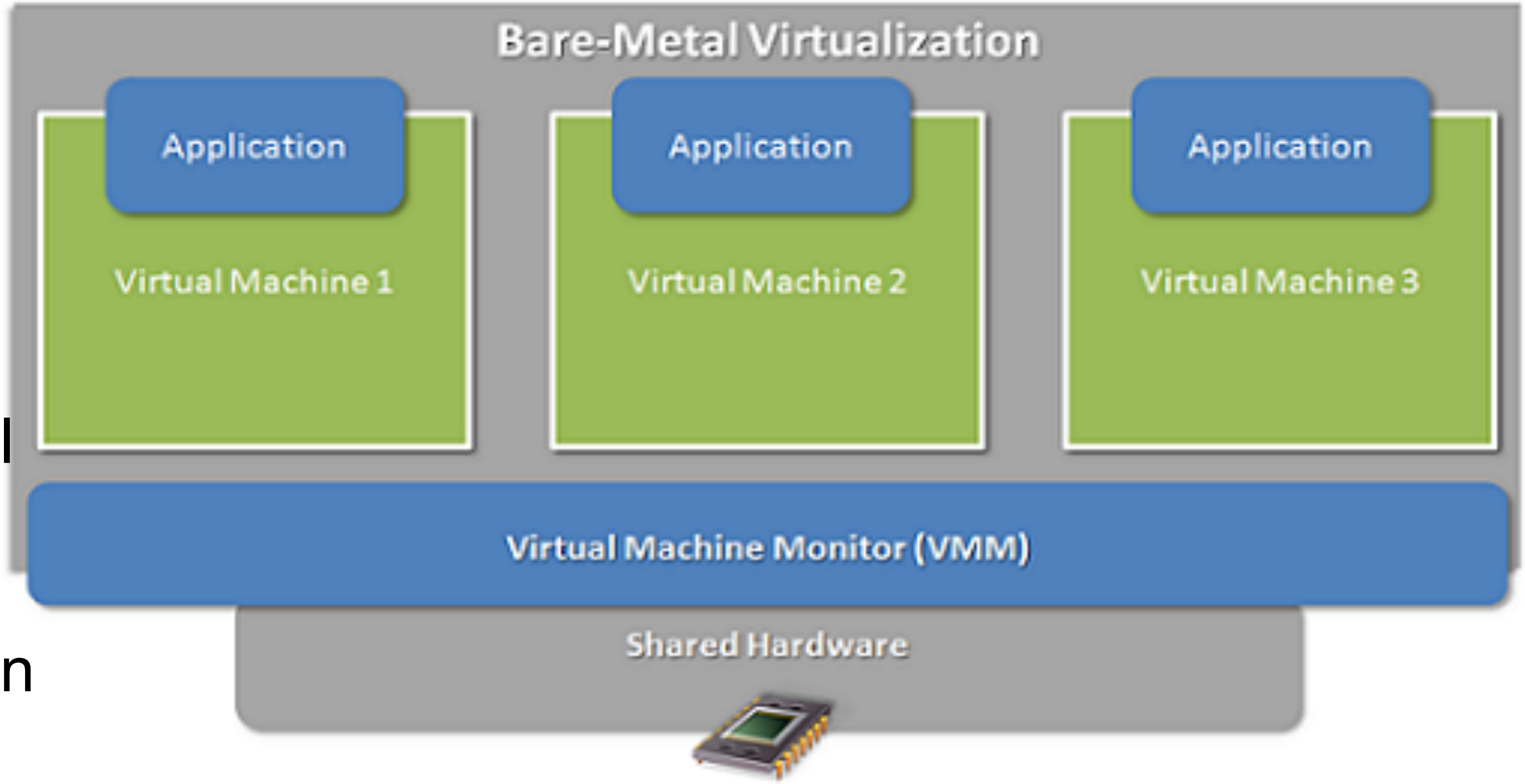
Recommended technique for ECPE 170



Bare-Metal Virtualization

More efficient, but not as easy to install.

The virtual machine monitor acts like an operating system itself!



Fire-up your VMs for Linux Tutorial!

COLLABORATE WITH ONE STUDENT

Common goals of an Operating System

File Management

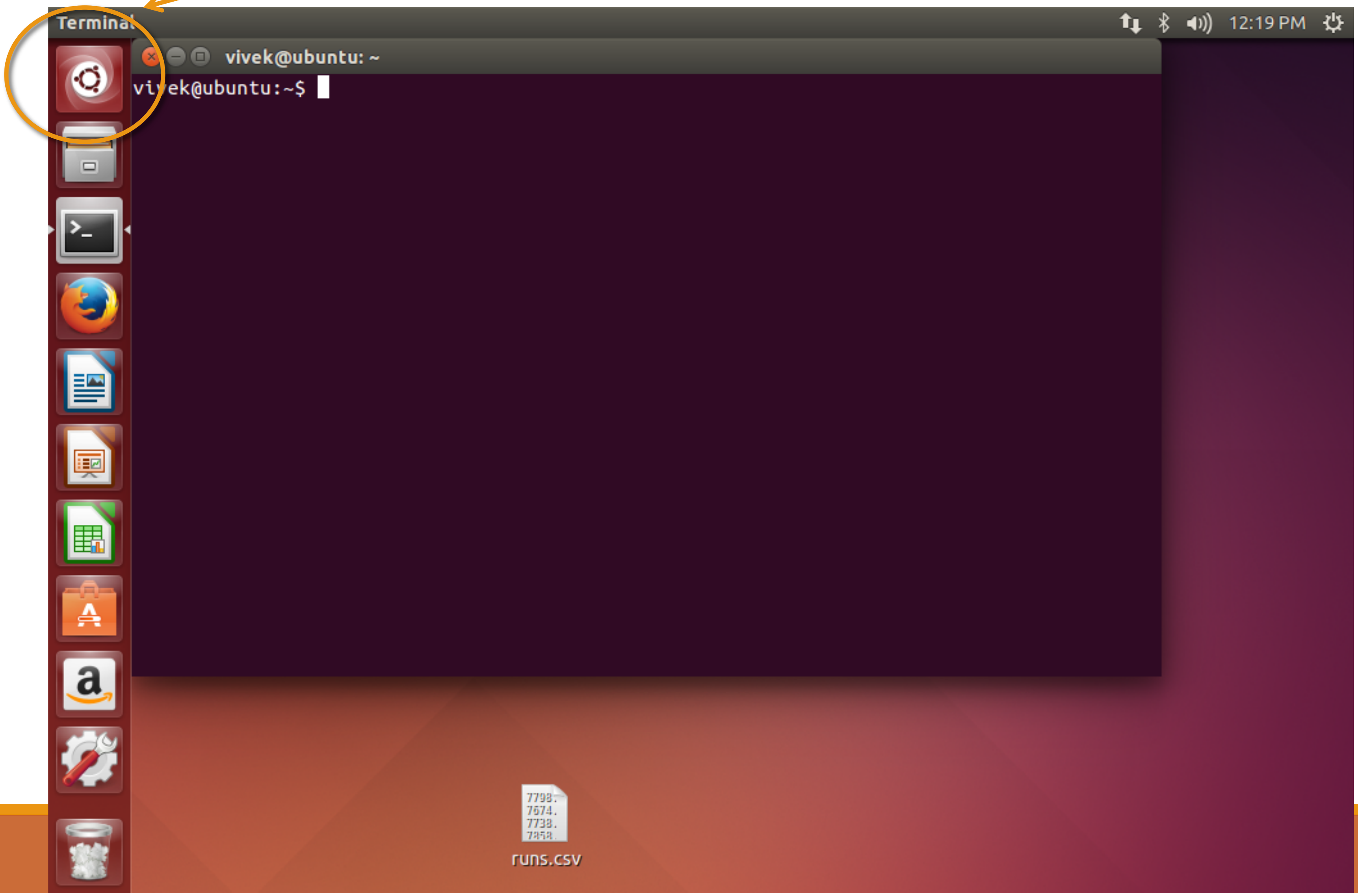
Process Management

Memory Management

1. The command line

Terminal: A text-based interface that accepts your commands.

To open terminal: Dash -> Search for Terminal or Activities (18LTS)

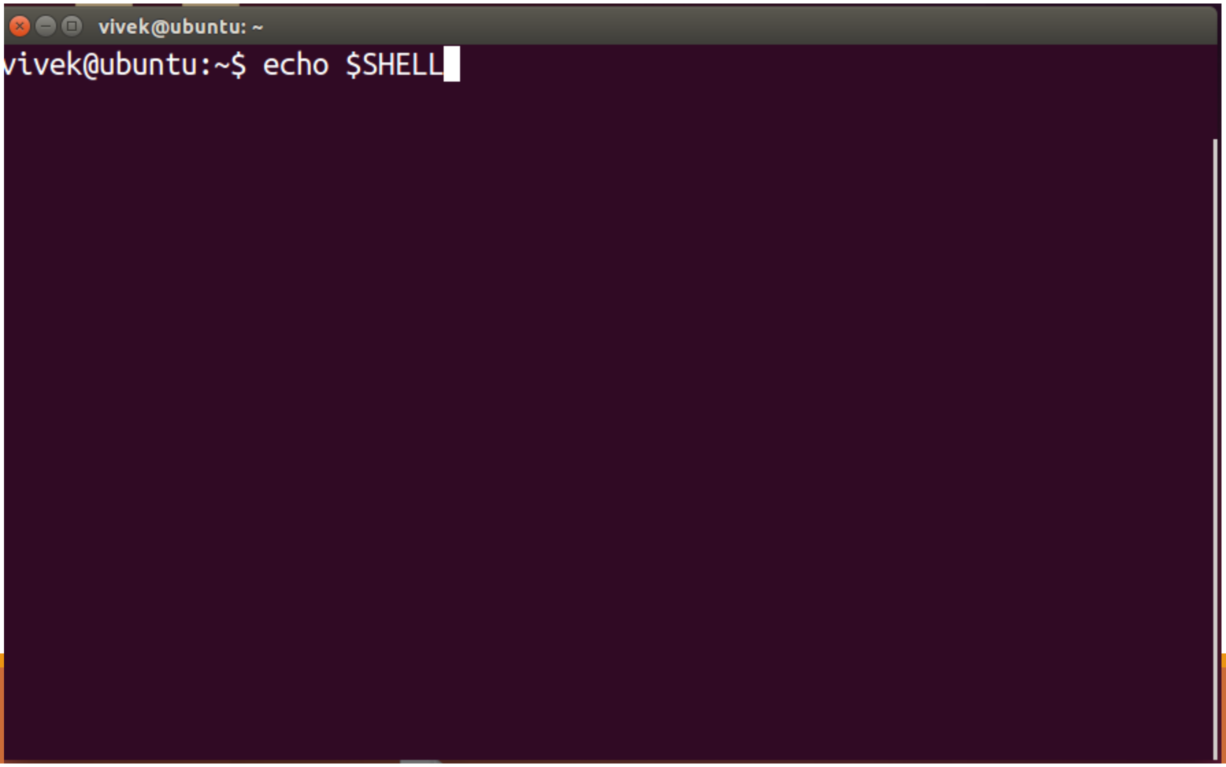


```
vivek@ubuntu:~$ 
```

Problem 1 - Which shell are you using?

A shell is a user program that defines how your terminal is going to behave for input commands

Many types: sh, bash (Bourne again), C syntax motivated: csh, tsh, etc.



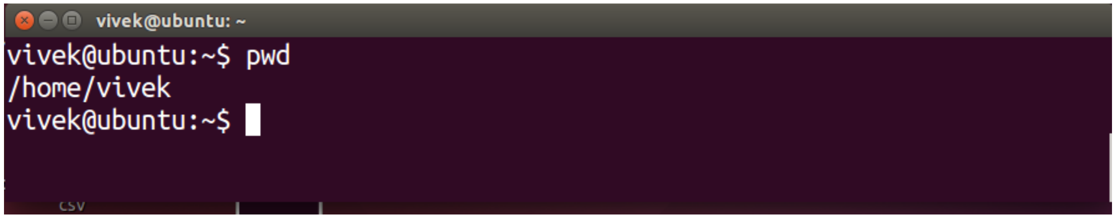
```
vivek@ubuntu:~$ echo $SHELL
```

2. Basic Navigation

Problem 2 – Navigations skills

a. Where are we?

`pwd` – print working directory



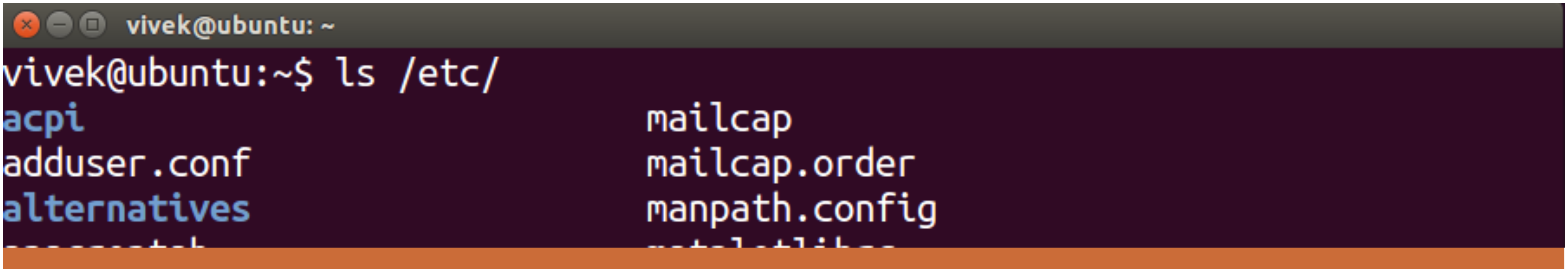
```
vivek@ubuntu:~$ pwd
/home/vivek
vivek@ubuntu:~$ 
```

b. Listing items in the current location

`ls`– list working directory

`ls [options] [location]`

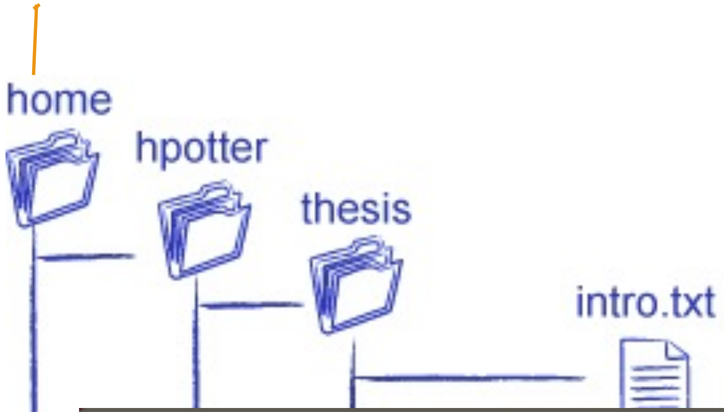
List the first 5 files/directories of /etc/



```
vivek@ubuntu:~$ ls /etc/
acpi                      mailcap
adduser.conf              mailcap.order
alternatives              manpath.config
```

Navigation Skills continued

/ is root

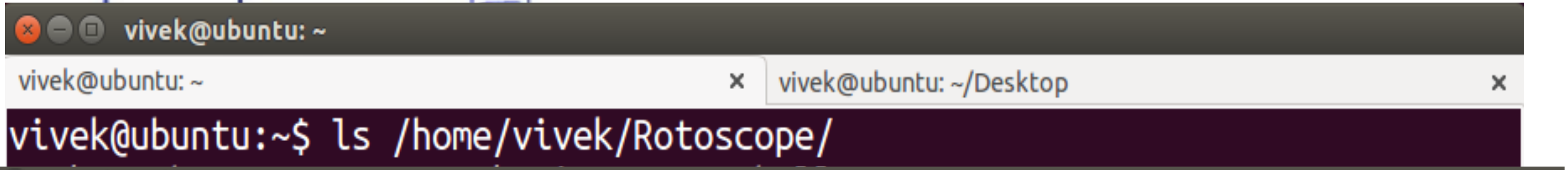


Absolute path:

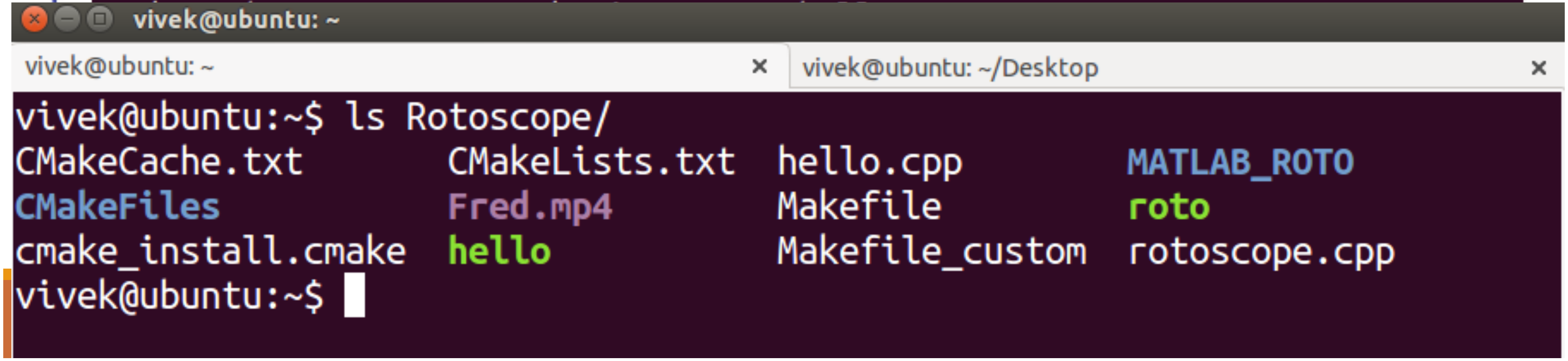
- /home/hpotter/thesis/intro.txt

Relative path:

- If I am already in /home/hpotter/
- addresses.html



```
vivek@ubuntu:~$ ls /home/vivek/Rotoscope/
```



```
vivek@ubuntu:~$ ls Rotoscope/
CMakeCache.txt       CMakeLists.txt  hello.cpp        MATLAB_ROTO
CMakeFiles           Fred.mp4        Makefile         roto
cmake_install.cmake  hello           Makefile_custom  rotoscope.cpp
vivek@ubuntu:~$ █
```

Navigation Skills continued

~: Tilde sign refers to your home directory. You can perform either:

```
you@Ubuntu:~$: ls /home/you/Documents
```

or

```
you@Ubuntu:~$: ls ~/Documents
```

.: Dot sign refers to current directory. Try:

```
you@Ubuntu:~$: ls .
```

..: Double dot refers to the parent directory of your current directory. Try:

```
you@Ubuntu:~$: ls ..
```

Problem 3: move around

cd: Change directory

```
cd [location]
```

Exercise: The `root` directory's `etc` directory `has an init.d` directory. Go to that directory. List top 4 items and return back to your `home` directory.

Problem 4: Know more about Linux commands

hidden files have a “dot” before them. Eg: .secret, .bashrc

type `ls`. Do you see any hidden files?

`man` command stands for manual. It provides information on a particular command

```
man <command to look up>
```

Using the `man` command, find out what option you need with ls to list the hidden files

How can you list files sorted on file size using the `ls` command?

Making directory, copying and moving files

```
mkdir [options] <Directory’s Name>
```

Create a directory called `Linux_tutorial`. Change to that directory.

1. Create a blank file using the `touch` command: `touch` example1

2. Fill something in it by executing this command:

```
You@Ubuntu:~/Linux_tutorial$: echo “Bleh ble bleh”
>example1
```

redirecting command: put to file

3. Copy example1 to example2:

```
You@Ubuntu:~/Linux_tutorial$: cp example1 example2
```

4. Move example2 to home directory using `mv` command:

```
You@Ubuntu:~/Linux_tutorial$:
```

Removing files and directories

```
rmdir [options] <Directory’s Name>
```

```
rm [options] <file’s Name>
```

You can also force remove a directory using rm:

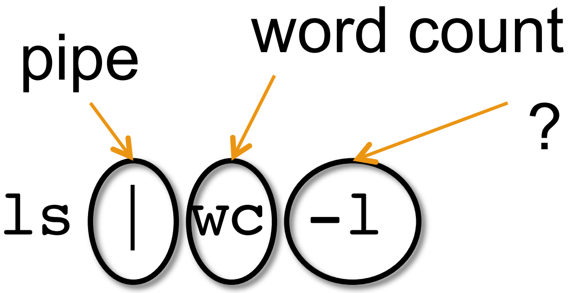
```
rm –rf <directory>
```

Remove example2.

DO NOT TRY: `rm –rf ~`

Problem 7: Fun with piping and some wildcards

Change to `/etc/` directory and count the number of files in that directory. You have only 60 seconds. Tick tock!



pipe

word count

?

Try this: `you@ubuntu:/etc$ ls | wc -l`

Sometimes you can recall only a few letters of a file's name
Use wildcards to let the terminal simplify it for you.

- * -- represents zero or more characters
- ? – represents a single character
- [] – represents a range of characters

Problem 5 -- Wildcards

Example: In `etc` directory, to list all files with extension conf:

```
you@ubuntu:/etc$ ls *.conf
```

Example: In `etc` directory, to list all files such that second letter is d and extension conf: `you@ubuntu:/etc$ ls ?d*.conf`

Problem: List all files with `.conf` extension that contain the word ‘switch’ in their names.

Permissions

Linux provides you privacy with files via permissions:

r read – the contents of the file can be viewed
w write – something can be written to the file
x execute – the file can be executed if an executable or script

Permission is granted to three types of people:
owner – the one who created the file, also called user (u)
group – the file belongs to a single group (g)
others – everyone else (o) but the group or the owner

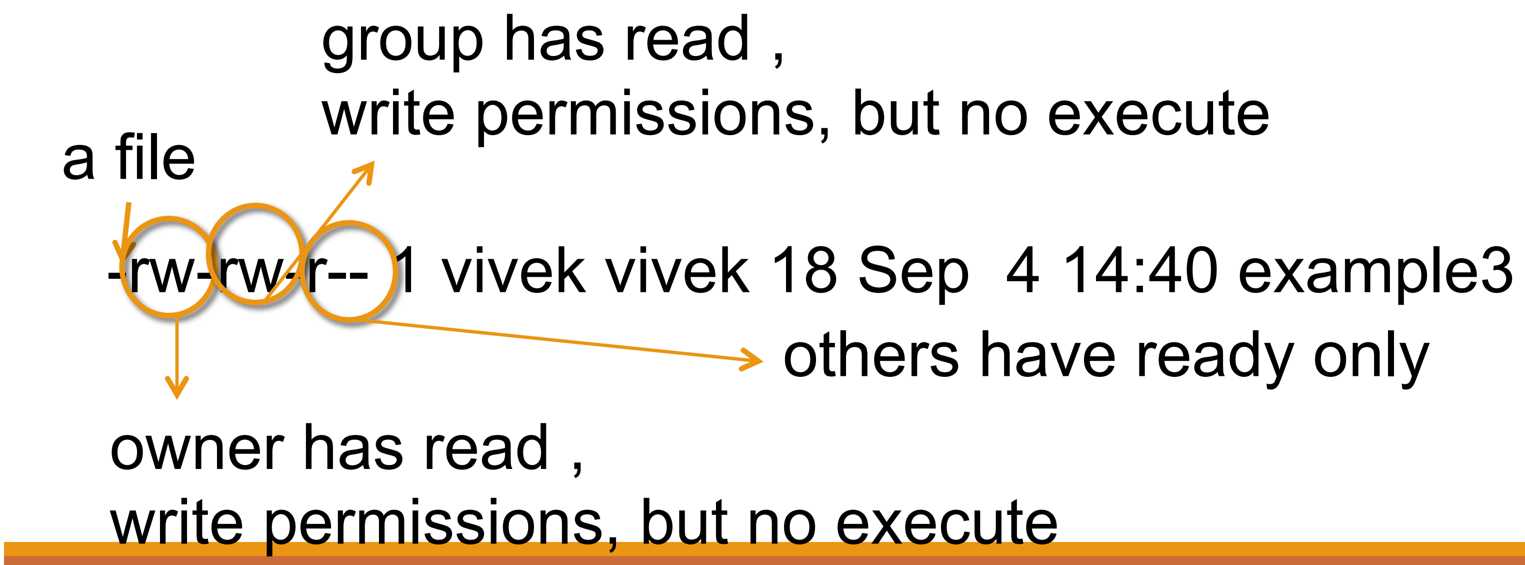
Problem 6

From wherever you are in your terminal, change to **Linux_tutorial** directory.

a. Touch a file example3

b. Put the string "`cat /proc/cpuinfo`" into it.

c. Perform `ls –l example3` and write output



group has read ,
write permissions, but no execute

a file

```
-rw-rw-r-- 1 vivek vivek 18 Sep  4 14:40 example3
```

others have ready only

owner has read ,
write permissions, but no execute

Problem 7 Changing permissions using `chmod`

try: ./example3

We change permissions using `chmod [permissions] [path]`

Whose permissions are we changing? `[ugoa]`:owner, group, others, or all

Grant or revoke? `+: providing -:` revoking

What are we providing? r (read), w (write), or x (execute)

example: `chmod u+rwx file` #provides read/write/execute to owner

example: `chmod g-x file` #removes executable for group

Problem: Provide yourself (owner) the execute privilege to example3. Type ls. Output? Do ./example3. Output?

Here is what happened

You executed/ran an executable

Your in-class participation problem (10 minutes):

a. what does `cat /proc/cpuinfo` perform? Use Linux tricks learned today and write it.
b. You executed `echo` command a lot today. What does it perform?
c. What are sudo and apt-get commands for?

You are ready to tackle Lab 1!

Note on Labs

Labs have (at most) two graded elements:

1. **Pre-Lab "checkpoint"** – quick verification that pre-lab *appears* to be done
   1. Due somewhere in the middle or the first day of the lab
2. **Lab Report**
   1. Submit all source code used with lab report
   2. Due by posted date after lab

Lab Reports

Not really “reports”, more like “worksheets”

Create in LibreOffice (aka *OpenOffice*) using example template on website

Export in **PDF format**

Submit

- Via Canvas *Assignments* section for Lab 1 only!
- Via Version control for Lab 2 and beyond

Upcoming Schedule

Today

- **Lab 1 – Linux Basics**

Thursday

- **Lab 2 – Version Control**

Deadlines

- **Lab 1 Report (Canvas) – 8th September 5 AM**
- **Lab 2 Report (Version Control) – 11th September 5 AM**

Now learn more by performing Lab 1

WORK AT LEAST FOR NEXT HALF HOUR. AFTER THAT YOU ARE FREE TO GO