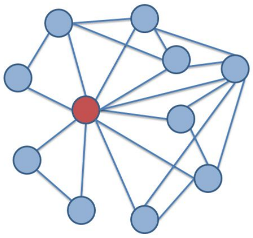


# CMPE 150 - Prelab 1

Kevin Romero Peces-Barba - 1635745

kromero8@ucsc.edu

**1. What command will show you which groups you are member of? 3**

**2. What does the environmental variable “$?” hold? 3**

**3. What key combination will suspend a currently running process and place it as a background process? 3**

**4. With what command (and arguments) can you find out your kernel version and the “nodename”? 3**

**5. What is the difference between the paths “.”, “..” and “~”? What does the path “/” refer to when not preceded by anything? 3**

**6. What is a pid? Which command would you use to find the “pid” for a running process? 3**

**7. Write a single command that will return every user’s default shell. 4**

**8. What is the difference between “sudo” and “su root”? 4**

**9. How would you tell your computer to run a program or script on a schedule or set interval on Linux? 4**

**10. Write a shell Script that only prints the even numbered lines of each file in the current directory. 5**

## 1. What command will show you which groups you are member of?

The command “groups <username>” will show to what groups a *username* belongs to.

## 2. What does the environmental variable “$?” hold?

It holds the return number of the last executed program.

## 3. What key combination will suspend a currently running process and place it as a background process?

You need to press Control+Z to suspend a running process. This won’t, however, place it as a background process. Instead the process will stop.

To run a process in background the command should be run with the “&” flag at the end OR press Control+Z and then run the command “bg”.

## 4. With what command (and arguments) can you find out your kernel version and the “nodename”?

With “uname -s” we’ll get the kernel name, with “uname -v” the kernel version and with “uname -n” the nodename.

## 5. What is the difference between the paths “.”, “..” and “~”? What does the path “/” refer to when not preceded by anything?

“.” refers to the current directory. “..” refers to the parent directory of the current one. “~” refers to . “/” refers to the root directory of the filesystem.

## 6. What is a pid? Which command would you use to find the “pid” for a running process?

“pid” stands for *process ID*. Each process has a unique ID which identifies it.

The command “ps” will show a list of current running processes and their ids. To see all running processes the command is “ps -e”.

## 7. Write a single command that will return every user's default shell.

The default shell for every user is stored in /etc/passwd so we need to access this file. We can do it using " cat /etc/passwd ".

For every user this file contains 7 columns -> *username:password:UID:GID:UserInfo:Home:shellPath*. We need to get the first and the seventh to get username and the path to shell. We achieve this by using " cat /etc/passwd | cut -d ":" -f 1,7 ". We'll something like *username:shellPath*. This could be the final solution.

If we want only to get the name of the shell and not the path we could run the command " cat /etc/passwd | cut -d ":" -f 1,7 | cut -d "/" -f 1,3 --output-delimiter="" ". The output will be *username:shell*.

## 8. What is the difference between "sudo" and "su root"?

You use "sudo" to run a single command with *root* privileges. "su root" will open a new terminal with *root* privileges for every command. This happens because "su" stands for "substitute" and is used to change to another user. So when you use "su root" you are changing the user.

## 9. How would you tell your computer to run a program or script on a schedule or set interval on Linux?

The best way to schedule a program to run *once* is to use "at" command. The syntax is "at <time>" where you type the time you want your program to be executed. This will start the *at prompt*. Now you can specify the command or program you want to be scheduled.

If you want a command or program to be periodically repeated, the best way is to use *cron*.

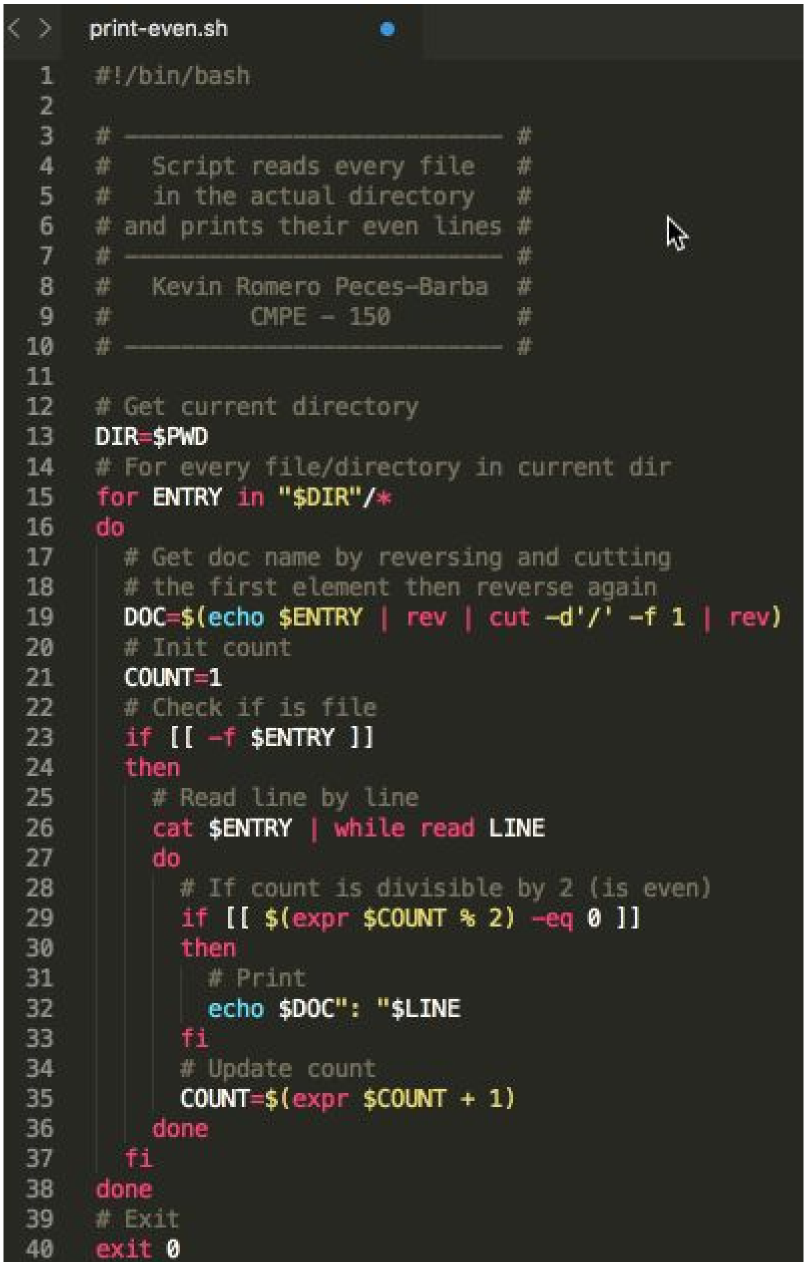
## 10. Write a shell Script that only prints the even numbered lines of each file in the current directory.

(The file is attached):

First it gets the current directory and iterates over the entries it has (files and directories). Then it gets the filename.

The method to do this is a bit tricky but you get used to it if you have need it a couple times. You have to reverse the string, split it by slashes, get the first element and then reverse it again. This way you'll get the filename no matter how long the path is.

Then the line count is initialized to 1 and if the entry is a file the script iterates over its lines. If the count for the line is even (divisible by 2), then it prints the docname and the line. Finally the count updates.



```bash
#!/bin/bash

# --------------------------- #
#   Script reads every file   #
#   in the actual directory   #
# and prints their even lines #
# --------------------------- #
#   Kevin Romero Peces-Barba  #
#          CMPE - 150         #
# --------------------------- #

# Get current directory
DIR=$PWD
# For every file/directory in current dir
for ENTRY in "$DIR"/*
do
  # Get doc name by reversing and cutting
  # the first element then reverse again
  DOC=$(echo $ENTRY | rev | cut -d'/' -f 1 | rev)
  # Init count
  COUNT=1
  # Check if is file
  if [[ -f $ENTRY ]]
  then
    # Read line by line
    cat $ENTRY | while read LINE
    do
      # If count is divisible by 2 (is even)
      if [[ $(expr $COUNT % 2) -eq 0 ]]
      then
        # Print
        echo $DOC": "$LINE
      fi
      # Update count
      COUNT=$(expr $COUNT + 1)
    done
  fi
done
# Exit
exit 0
```