Questions for lab 1

Answer in complete sentences,

- What is a “shell program”?

Shell program interprets user’s commands, which are directly entered by a user or which can be read from a file called shell script aka shell program. Important that shell script is interpreted and not compiled.

- What can a shell program be used for(give examples)?

It can be used for outputting text with command echo
Give or remove permission with command chmod + or - x, same thing w for write and r for read
Create variable name=name
Condition statements if [-e <nameOfFile] # if this file exists
Then echo else echo

- How do you make a *shell(.sh) file* executable (what is the procedure)?

#! /bin/bash - we tell location of our bash, it helps interpreter to know that this is bash shell script
We name our file with <name>.sh , where sh stands for shell, it is not necessary to have extension <sh> but it is good practice and editor will know it shell file and apply useful features like styling with colors.

- What is the difference between Interpreted code VS. Compiled code(in your own words)?

Important that shell script is interpreted and not compiled. So when you write your shell script it interpreted by your

operating system and you do need to compile your shell script in order to execute it. You write shell script in your editor (in our case it is VIM) and you right away execute it without compiling this shell script.

- What is the difference between running a program with and without *exec?*

If we run ./<nameOfFile>.sh it won't execute because permission is denied. So we need to give permission to the file in order to execute it with command chmod +x and <nameOfTheFile>.sh. Because when we created file with touch <nameOfTheFile>.sh it created without permission to execute it. When we use exec it will replace it and used same PID number.

- What does the ps command do, and why is it useful?

PS command shows what processing running on our server, so we can see what's taking memory.

PID is process id and it is unique id because no 2 processes can be the same and underneath each line is a process itself.

When a process is run it's assigned a pid.

TTY is a terminal where it is running

Time refers to CPU time, how much time process is utilizing CPU, it is not a time of process's run

CMD is command that running in that process

- Why do we need 3 sets of file/folder permission for a file/folder?

Rwx rw- r—

R stands for read, w stands for write and x for execute and - means not available

First three - related to you (like admin)

Second three - related to group

Third three - related to user

```
#!/bin/bash
echo Starting b
if [ -e ./b.sh ] # Check if the file b.sh exitsts
then
        ./b.sh
        ps #list running processes
else #give error message indicating it as such
echo "---file b.sh does not exist in current directory---"
echo "current files in directory:"
ls
echo " "
fi
echo " "
sleep 5
#x stands to give permission to access that file otherwise permission will be deny,
so in oder to access it we have to use chmode +x
~
~
~
~
~
~
~
~
"./a.sh" 15L, 425C                                          15,1          All
```

```
if [ -z "$1" ]
then
        echo "no file list returned from b.sh exiting"
else
        echo "Number of files in directory: `echo $1 | wc -w`"
#-z ---check if it = to 0 then execute below echo; $1 is argument; wc- word count; `
` used for execute it; fi = like } workingDir -- name of variable and assigning valu
e in `` if use '' it will treat as string
fi
if [ -e ./d.sh ]
then
        ./d.sh
else
        workingDir = `pwd`
echo "---file d.sh does not exist in current directory---"
echo "you are currently working int:" $workingDir
echo files in $workingDir
ls
echo " "
fi
ps
echo " "
sleep 5

"./c.sh" 24L, 599C                                          19,8          Bot
```

```
#!/bin/bash
echo starting e
if [ -e ./e.sh ]
then
        chmod +x ./e.sh
        ./e.sh
        #-e --- stand if it exists name of file then do smth or else
        #chmod +x -- means give permission to execute: name of file
        # ps shows all running commands
        # sleep 5 - means give 5 seconds before close
else
        echo "--- file e.sh does not exist in the current directoryecho current file
s in directory:"
        ls
        echo " "
        workingDir=`pwd`
        echo "---file c.sh does not exist in current directory --"
        echo "you are currently woring int:" $workingDir
        echo files in $workingDir
        ls
        echo " "
fi
ps
echo " "
"./d.sh" 25L, 584C                                          12,65-72      Top
```

```
#!/bin/bash
echo f is running
cd /
echo all files in root directory:
ls -l
#first line means tells interpreter what language to use to intepret
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"./f.sh" 6L, 144C                                             5,5           All
```

```
1265229 pts/0    00:00:00 d.sh
1265261 pts/0    00:00:00 e.sh
1265263 pts/0    00:00:00 ps

All files in current directory
/home/prymak
a.sh
b.sh
c.sh
d.sh
e.sh
f.sh
unix.txt
vivaLinux

    PID TTY          TIME CMD
1265139 pts/0    00:00:00 bash
1265229 pts/0    00:00:00 d.sh
1265266 pts/0    00:00:00 ps

[prymak@csdept ~]$ n a.sh
bash: n: command not found...
[prymak@csdept ~]$ n ./a.sh
bash: n: command not found...
[prymak@csdept ~]$
```