1. The ***echo*** command can be used simply to display text. Piping the output of this command to count the number of lines can be done simply by ***wc -l*** command.

```
ibab@IBAB-RA-Comp203:~$ echo 'I am completing my lab work' | wc
      1       6      28
ibab@IBAB-RA-Comp203:~$ echo 'I am completing my lab work' | wc -l
1
ibab@IBAB-RA-Comp203:~$
```

2. ***cat .bashrc***: Reads the contents of the .bashrc file, which is a shell script that runs every time you start a new terminal session in bash.
| Pipes the output of cat .bashrc to the next command.
grep 'PS1': Filters the lines from .bashrc that contain the string PS1.

debian_chroot – this variable is used to show the current environment in the prompt.
[\033[01;32m\]- gives the colour code of the command.
\u@\h[\033[00m]:[\033[01;34m\]\w\ - this is the normal BASH prompt.

```
ibab@IBAB-RA-Comp203:~$ cat .bashrc | grep 'PS1'
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
ibab@IBAB-RA-Comp203:~$
```

3. Using ***grep 'HIST'*** will filter out all commands in bash environment which begins with HIST.
HISTSIZE – number of commands to remember in history.
HISTFILE – file where command history is saved.
HISTCONTROL – controls what is saved in history (e.g., ignoring duplicates).

```
ibab@IBAB-RA-Comp203:~$ cat .bashrc | grep 'HIST'
HISTCONTROL=ignoreboth
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
ibab@IBAB-RA-Comp203:~$
```

4. ***whereis*** shows all locations related to a command ie it shows the entire pathway where the command could be present.
which shows the exact path of the command that will run when you execute the command. It depends on the default of the system.

```
ibab@IBAB-RA-Comp203:~$ type ls
ls is aliased to `ls --color=auto'
ibab@IBAB-RA-Comp203:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
ibab@IBAB-RA-Comp203:~$ which ls
/usr/bin/ls
ibab@IBAB-RA-Comp203:~$
```

5. Here, we use ***dirname*** along with the absolute path as it strips last component (filename) and returns the directory path.

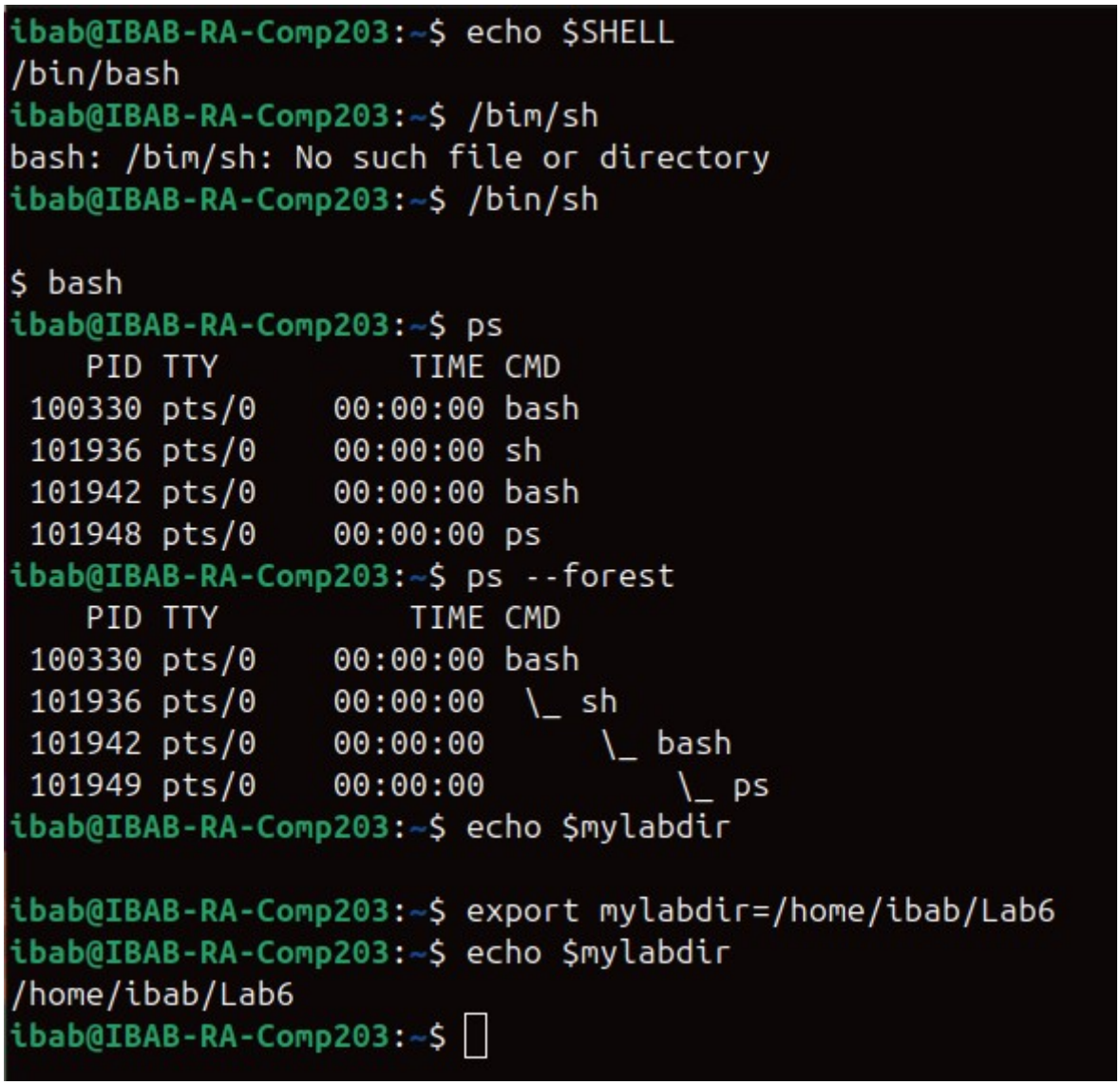
```
ibab@IBAB-RA-Comp203:~$ ls
 bash                   Lab4            Public
 col1_sorted_gnu.out    Lab5            scripts
 Desktop                Music           snap
 Documents              'old students'  Templates
 Downloads              Pictures        Videos
ibab@IBAB-RA-Comp203:~$ cd Lab4
ibab@IBAB-RA-Comp203:~/Lab4$ ls
age_restbp_sort.out  heart1.tar.gz
age_sorted_gnu.out   Heart.csv
age_sorted.out       heart_hlink
col1_sorted_gnu.out  heart_slink
col1_sorted.out      restbp_revsort.out
heart1_both.tar.gz   sex_age_chpain_sort.out
ibab@IBAB-RA-Comp203:~/Lab4$ dirname Heart.csv
.
ibab@IBAB-RA-Comp203:~/Lab4$ dirname /home/ibab/Lab4/Heart.csv
/home/ibab/Lab4
ibab@IBAB-RA-Comp203:~/Lab4$
```

6. This creates a shell variable mylabdir that holds the directory path /home/ibab/Lab6.
e***cho $mylabdir*** prints the value of mylabdir.
dirname $mylabdir prints the directory portion of the path (i.e., /home/ibab)

```
ibab@IBAB-RA-Comp203:~$ mkdir Lab6
ibab@IBAB-RA-Comp203:~$ mylabdir="/home/ibab/Lab6"
ibab@IBAB-RA-Comp203:~$ echo $mylabdir
/home/ibab/Lab6
ibab@IBAB-RA-Comp203:~$ dirname $mylabdir
/home/ibab
ibab@IBAB-RA-Comp203:~$
```

7. The ***bash*** command creates a subshell within the parent shell . We check if it is created using the ***ps*** command along with the ps –forest command.

On using ***echo with mylabdir,*** nothing is executes ie i**t is empty.** But after exporting it into the subshell, we make it global and are able to execute it.

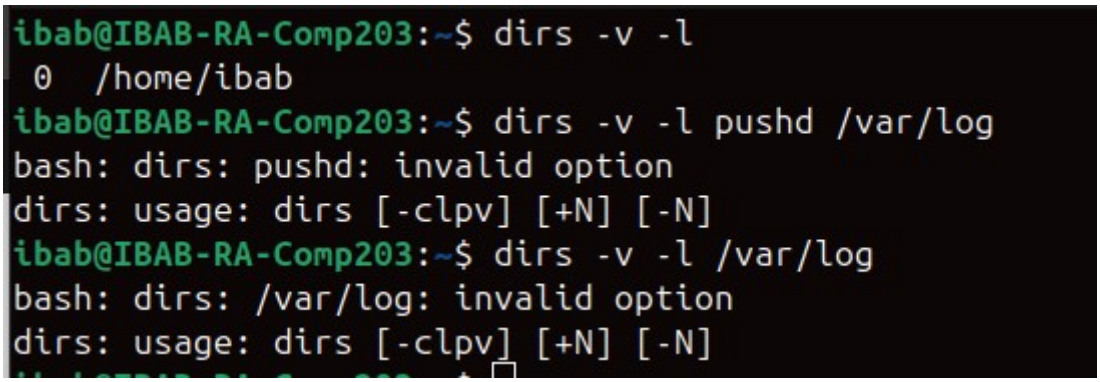


8. We use ***echo $PATH*** to display the path. We can check if its global or local through the ***type*** command or by creating a new bash and checking the path again. We use ***env | grep 'PATH'*** . As it is visible in both the subshell and env command we can tell that it is a global variable.

```
ibab@IBAB-RA-Comp203:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
ibab@IBAB-RA-Comp203:~$ type PATH
bash: type: PATH: not found
ibab@IBAB-RA-Comp203:~$ type $PATH
bash: type: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin: not found
ibab@IBAB-RA-Comp203:~$ bash
ibab@IBAB-RA-Comp203:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
ibab@IBAB-RA-Comp203:~$ env | grep 'PATH'
WINDOWPATH=2
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
ibab@IBAB-RA-Comp203:~$
```

9) I.

- `-v`: Shows stack entries with indexes.
- `-l`: Shows full pathnames.
- You'll see only the current directory at index 0 which is default stack as shown in man pages.



ii) ***pushd*** changes your working directory to the given one. It pushes the previous one into a **stack**. Each time you run pushd, it shows the current stack

```
ibab@IBAB-RA-Comp203:~$ pushd /var/log
/var/log ~
ibab@IBAB-RA-Comp203:/var/log$ pushd /tmp
/tmp /var/log ~
ibab@IBAB-RA-Comp203:/tmp$ pushd /etc
/etc /tmp /var/log ~
ibab@IBAB-RA-Comp203:/etc$ pushd ~/Downloads
~/Downloads /etc /tmp /var/log ~
ibab@IBAB-RA-Comp203:~/Downloads$ pushd ~/Documents
~/Documents ~/Downloads /etc /tmp /var/log ~
```

iii)After running ***dirs -v -l,*** we now see multiple entries with index 0 being your most recent and the original directory at the bottom. It shows the entire stack of directories.

```
ibab@IBAB-RA-Comp203:~/Documents$ dirs -v -l
 0  /home/ibab/Documents
 1  /home/ibab/Downloads
 2  /etc
 3  /tmp
 4  /var/log
 5  /home/ibab
ibab@IBAB-RA-Comp203:~/Documents$
```

iv) ***pushd +1*** makes the directory at index1 the current directory.
v) ***cd ~*** gets you out of all the directories back to the home directory.
vi) ***cd /tmp*** simply changed directory to an entry in the directory stack without reference to the stack in any way.
vii) ***popd*** is responsible for removing the topmost “plate” in the stack and changes to the directory entry against index 1.
viii) ***popd +2*** removes the entry at index 2, but **does not change** current directory but the stack shortens by one.
ix) Index `0` is always your **current working directory**.
As you pushd, it gets replaced. As you popd, the next one becomes current.

```
ibab@IBAB-RA-Comp203:~/Documents$ pushd +1
~/Downloads /etc /tmp /var/log ~ ~/Documents
ibab@IBAB-RA-Comp203:~/Downloads$ cd ~.
bash: cd: ~.: No such file or directory
ibab@IBAB-RA-Comp203:~/Downloads$ cd ~
ibab@IBAB-RA-Comp203:~$ cd /tmp
ibab@IBAB-RA-Comp203:/tmp$ popd
/etc /tmp /var/log ~ ~/Documents
ibab@IBAB-RA-Comp203:/etc$ popd +2
/etc /tmp ~ ~/Documents
ibab@IBAB-RA-Comp203:/etc$
```