Assignment on Linux - Day 5

Create an alias: Add an alias to your .bashrc file that creates a shortcut for a long command that you frequently use. Test the alias by opening a new terminal session and running the alias name.

```
vim .bashrc
alias infoc='lscpu; ls -al'
source ~/.bashrc
alias
infoc
```

```
Editor    Tab 1    Tab 2    Tab 3    +
ubuntu $ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias infoc='lscpu; ls -al'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias python3='/usr/bin/python3.8'
ubuntu $ infoc
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   46 bits physical, 48 bits vir
CPU(s):                          1
On-line CPU(s) list:             0
Thread(s) per core:              1
Core(s) per socket:              1
```

Set an environment variable: Add an environment variable to your .bashrc file that sets the default text editor to Vim. Test the variable by running the command "echo $EDITOR".

```
vim .bashrc
```

```
export EDITOR=vim
source ~/.bashrc
echo $EDITOR
```

```
ubuntu $ vim .bashrc
ubuntu $ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias infoc='lscpu; ls -al'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias python3='/usr/bin/python3.8'
ubuntu $ source ~/.bashrc
ubuntu $ echo $EDITOR
vim
ubuntu $
```

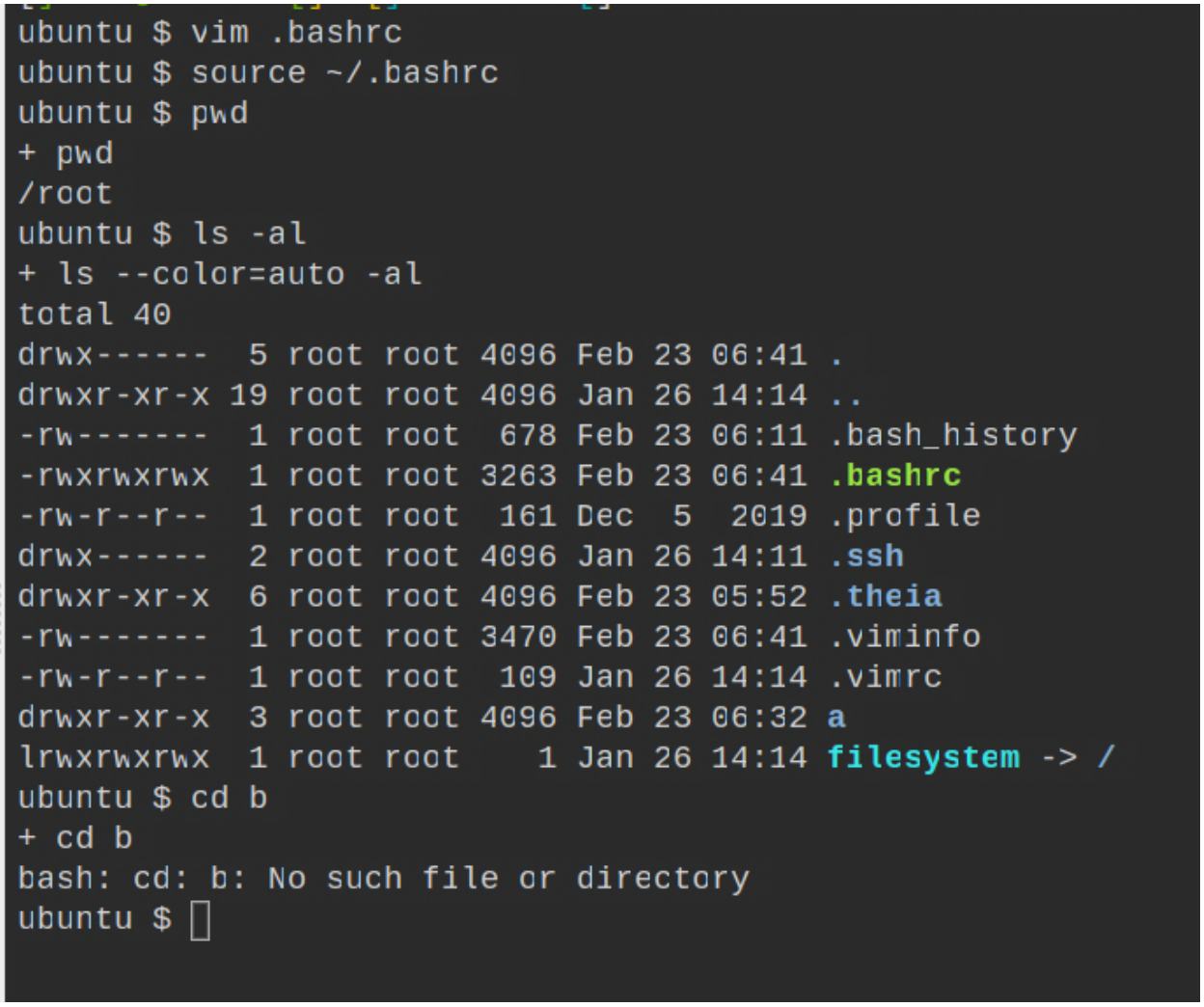
Customize your prompt: Add a custom prompt to your .bashrc file that displays the current directory and time. Use the command "export PS1='[\e[32m]\u@\h [\e[33m]\w [\e[36m]\t [\e[0m]$ '". Test the prompt by opening a new terminal session and navigating to different directories.

```
vim .bashrc
export PS1='[\e[32m]\u@\h [\e[33m]\w [\e[36m]\t [\e[0m]$
source ~/.bashrc
```

```
[]root@ubuntu []~ []06:36:26 []$vim .bashrc
[]root@ubuntu []~ []06:38:28 []$source ~/.bashrc
```

Debugging: Add the "set -x" command to the beginning of your .bashrc file to enable debugging output. Test the debugging by opening a new terminal session and watching the output as the file is executed. Look for any errors or unexpected behavior.

```
vim .bashrc
set -x
source ~/.bashrc
```



Display all block devices:

```
lsblk
```

```
ubuntu $ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0 63.2M  1 loop /snap/core20/1634
loop1     7:1    0 67.8M  1 loop /snap/lxd/22753
loop2     7:2    0   48M  1 loop /snap/snapd/17336
vda     252:0    0   20G  0 disk
|-vda1  252:1    0 19.9G  0 part /
|-vda14 252:14   0    4M  0 part
`-vda15 252:15   0  106M  0 part /boot/efi
ubuntu $
```

Show device partitions:

```
fdisk -l /dev/vda
```

```
ubuntu $ fdisk -l /dev/vda
+ fdisk -l /dev/vda
Disk /dev/vda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: D75B0394-501D-41E7-94C7-29E9642F257A

Device      Start      End  Sectors  Size Type
/dev/vda1  227328 41943006 41715679 19.9G Linux filesystem
/dev/vda14   2048    10239     8192    4M BIOS boot
/dev/vda15  10240   227327   217088  106M EFI System

Partition table entries are not in disk order.
ubuntu $
```

Schedule a script to run daily: Use the crontab command to schedule a script to run at a specific time every day. For example, schedule a script named backup.sh to run at 1:00 AM every day.

Created shell script and check functionality.

```
ubuntu $ ls -l
total 4
-rw-r--r-- 1 root root 16 Feb 23 06:54 backup.sh
lrwxrwxrwx 1 root root  1 Jan 26 14:14 filesystem -> /
ubuntu $ chmod +x backup.sh
ubuntu $ ls -l
total 4
-rwxr-xr-x 1 root root 16 Feb 23 06:54 backup.sh
lrwxrwxrwx 1 root root  1 Jan 26 14:14 filesystem -> /
ubuntu $ ./backup.sh > output.txt
ubuntu $ cat output.txt
Thu Feb 23 06:55:57 UTC 2023
/root
.
|-- backup.sh
|-- filesystem -> /
`-- output.txt

1 directory, 2 files
ubuntu $
```

Create cron job for the script.

```
#
# For more information see the manual pages of crontab(5) and
#
# m h  dom mon dow   command
00 1 * * * (cd /root; ./backup.sh >> output.txt)
~
~
~
~
~
~
~
~
~
~
"/tmp/crontab.hXzrq8/crontab" 24L, 938C
```

```
ubuntu $ cat output.txt
Thu Feb 23 07:05:01 UTC 2023
/root
.
├── backup.sh
├── filesystem -> /
├── output
├── output.txt
└── temp.txt

1 directory, 4 files
ubuntu $ cat output.txt
Thu Feb 23 07:05:01 UTC 2023
/root
.
├── backup.sh
├── filesystem -> /
├── output
├── output.txt
└── temp.txt

1 directory, 4 files
Thu Feb 23 07:06:01 UTC 2023
/root
.
├── backup.sh
├── filesystem -> /
├── output.txt
└── temp.txt

1 directory, 3 files
ubuntu $
```