NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

OPERATING SYSTEM (LAB)

Lab Report: 04

Submitted to
Mr.Meesum Raza

Submitted By

Adeel Naeem
(BSAI-146)
Junaid Asif
(BSAI-144)
Syed Qasim Ali
(BSAI-161)

Question # 01

- Use the df command to display the amount of used and available space on your hard drive.

```
junaidasifdev@Joni:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.4G     0  3.4G   0% /usr/lib/modules/5.15.167.4-microsoft-standard-WSL2
none            3.4G  4.0K  3.4G   1% /mnt/wsl
drivers         204G   77G  128G  38% /usr/lib/wsl/drivers
/dev/sdc       1007G  1.6G  955G   1% /
none            3.4G   72K  3.4G   1% /mnt/wslg
none            3.4G     0  3.4G   0% /usr/lib/wsl/lib
rootfs          3.4G  2.2M  3.4G   1% /init
none            3.4G     0  3.4G   0% /dev
none            3.4G  804K  3.4G   1% /run
none            3.4G     0  3.4G   0% /run/lock
none            3.4G     0  3.4G   0% /run/shm
tmpfs           3.4G     0  3.4G   0% /sys/fs/cgroup
none            3.4G   92K  3.4G   1% /mnt/wslg/versions.txt
none            3.4G   92K  3.4G   1% /mnt/wslg/doc
C:\             204G   77G  128G  38% /mnt/c
D:\             272G  4.6G  268G   2% /mnt/d
snapfuse         64M   64M     0 100% /snap/core20/1891
snapfuse         92M   92M     0 100% /snap/lxd/24061
snapfuse         92M   92M     0 100% /snap/lxd/29619
snapfuse         54M   54M     0 100% /snap/snapd/19122
snapfuse         39M   39M     0 100% /snap/snapd/21759
snapfuse         64M   64M     0 100% /snap/core20/2434
```

- Check the man page for df, and use it to find an option to the command which will display the free space in a more human-friendly form. Try both the single-letter and long-style options.

```
junaidasifdev@Joni:~$ man df
junaidasifdev@Joni:~$ df --human-readable
Filesystem      Size  Used Avail Use% Mounted on
none            3.4G     0  3.4G   0% /usr/lib/modules/5.15.167.4-microsoft-standard-WSL2
none            3.4G  4.0K  3.4G   1% /mnt/wsl
drivers         204G   75G  130G  37% /usr/lib/wsl/drivers
/dev/sdc       1007G  1.6G  955G   1% /
none            3.4G   76K  3.4G   1% /mnt/wslg
none            3.4G     0  3.4G   0% /usr/lib/wsl/lib
rootfs          3.4G  2.2M  3.4G   1% /init
none            3.4G     0  3.4G   0% /dev
none            3.4G  792K  3.4G   1% /run
none            3.4G     0  3.4G   0% /run/lock
none            3.4G     0  3.4G   0% /run/shm
tmpfs           3.4G     0  3.4G   0% /sys/fs/cgroup
none            3.4G   96K  3.4G   1% /mnt/wslg/versions.txt
none            3.4G   96K  3.4G   1% /mnt/wslg/doc
C:\             204G   75G  130G  37% /mnt/c
D:\             272G  4.6G  268G   2% /mnt/d
snapfuse         64M   64M     0 100% /snap/core20/1891
snapfuse         92M   92M     0 100% /snap/lxd/24061
snapfuse         92M   92M     0 100% /snap/lxd/29619
snapfuse         39M   39M     0 100% /snap/snapd/21759
snapfuse         64M   64M     0 100% /snap/core20/2434
snapfuse         45M   45M     0 100% /snap/snapd/23258
```

Question # 02

- Try ls with the -a and -A options. What is the difference between them?

```
junaidasifdev@Joni:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .landscape  .motd_shown  .profile  dog  junaid
junaidasifdev@Joni:~$ ls -A
.bash_history  .bash_logout  .bashrc  .cache  .landscape  .motd_shown  .profile  dog  junaid
junaidasifdev@Joni:~$
```

Difference:

- ls -a includes **all entries** (even . and ..).
- ls -A excludes the . and .. directories, making the output cleaner and more focused on actual files and subdirectories

- Write a for loop which goes through all the files in a directory and prints out their names with echo. If you write the whole thing on one line, then it will be easy to repeat it using the command line history.

```
junaidasifdev@Joni:~$ for file in *; do echo "$file"; done
dog
junaid
```

- Change the loop so that it goes through the names of the people in the room (which needn't be the names of files) and print greetings to them.

```
junaidasifdev@Joni:~$ for person in Alice Bob Charlie; do echo "Hello, $person!"; done
Hello, Alice!
Hello, Bob!
Hello, Charlie!
```

- Of course, a simpler way to print a list of filenames is echo *. Why might this be useful, when we usually use the ls command?

```
junaidasifdev@Joni:~$ ls
dog  junaid
junaidasifdev@Joni:~$ echo *
dog junaid
```

Using echo * to list filenames can be useful in several scenarios where ls may not behave as desired or may introduce complexities. Here are the key advantages:

- ls formats its output to make it human-readable, often arranging files in columns or adding colors (depending on your configuration).
- echo * gives a **simple, raw, space-separated list** of filenames, which can be more suitable for scripting or direct use in other commands.

Question # 03

- Use the find command to list all the files and directories under your home directory. Try the -type d and -type f criteria to show just files and just directories.

```
junaidasifdev@Joni:~$ find ~
/home/junaidasifdev
/home/junaidasifdev/.profile
/home/junaidasifdev/.landscape
/home/junaidasifdev/.landscape/sysinfo.log
/home/junaidasifdev/.bashrc
/home/junaidasifdev/.cache
/home/junaidasifdev/.cache/motd.legal-displayed
/home/junaidasifdev/.bash_logout
/home/junaidasifdev/.motd_shown
/home/junaidasifdev/dog
/home/junaidasifdev/dog/cat
/home/junaidasifdev/dog/cat/mouse
/home/junaidasifdev/junaid
/home/junaidasifdev/.bash_history
junaidasifdev@Joni:~$ find ~ -type d
/home/junaidasifdev
/home/junaidasifdev/.landscape
/home/junaidasifdev/.cache
/home/junaidasifdev/dog
/home/junaidasifdev/dog/cat
/home/junaidasifdev/dog/cat/mouse
junaidasifdev@Joni:~$ find ~ -type f
/home/junaidasifdev/.profile
/home/junaidasifdev/.landscape/sysinfo.log
/home/junaidasifdev/.bashrc
/home/junaidasifdev/.cache/motd.legal-displayed
/home/junaidasifdev/.bash_logout
/home/junaidasifdev/.motd_shown
/home/junaidasifdev/junaid
/home/junaidasifdev/.bash_history
```

- Use ‘locate’ to find files whose name contains the string ‘bashbug’. Try the same search with find, looking over all files on the system. You’ll need to use the * wildcard at the end of the pattern to match files with extensions.

```
junaidasifdev@Joni:~$ find / -type f -name "*bashbug*" 2>/dev/null
/snap/core20/1891/usr/bin/bashbug
/snap/core20/2434/usr/bin/bashbug
/mnt/c/Program Files/Git/usr/bin/bashbug
```

- Find out what the find criterion -iname does?

```
junaidasifdev@Joni:~$ find <path> -iname "<pattern>"
-bash: path: No such file or directory
junaidasifdev@Joni:~$
```

The -iname criterion in the find command is used to search for files or directories by name in a case-insensitive manner. This means it will match names regardless of whether they are in uppercase, lowercase, or a mix of both.