RIPHAH INTERNATIONAL UNIVERSITY, ISLAMABAD

Lab # 7

Bachelors of Computer Science – 6th Semester

Subject: Operating System

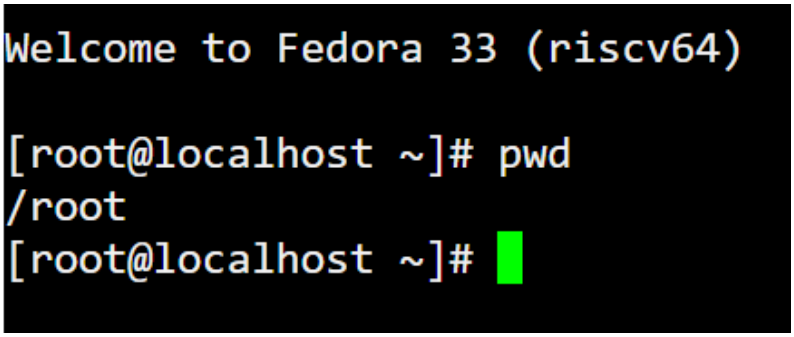
Submitted to: Ms. Kausar

Submitted by: Ayesha Noor _ 41379

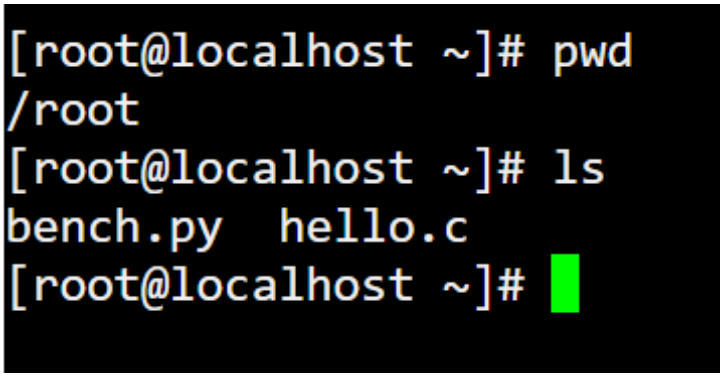
Date of Submission: 07- Oct -2024

Lab Task

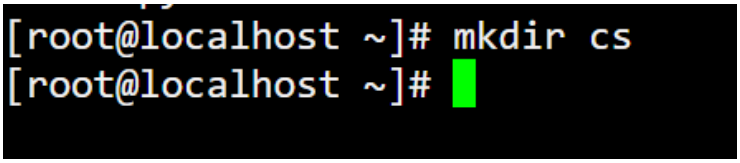
pwd (Print Working Directory): Shows the full path of the current directory you are in.



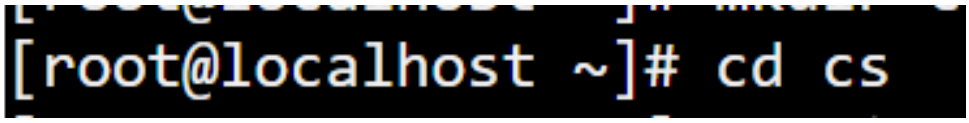
ls (list) : Lists files and directories in the current directory.



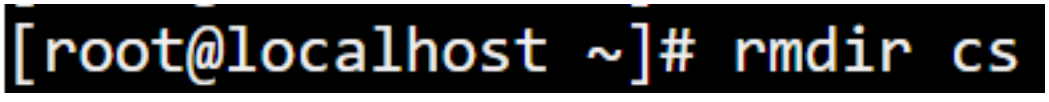
mkdir (make directory) : Creates a new directory.



cd (change directory) : Changes the current directory to another.



rmdir (remove directory) : Deletes an empty directory.



rm -r (Remove Recursively) : Deletes a directory and all its contents.

```
[root@localhost ~]# rm -r cs
```

cp (copy) : Copies files or directories to a new location.

```
[root@localhost ~]# ls
bench.py  books  file1  file2  hello.c
[root@localhost ~]# cp -r file1 file2
[root@localhost ~]# cd file2
[root@localhost file2]# ls
file1
[root@localhost file2]#
```

mv (move):

```
[root@localhost ~]# mv /root/A/MidTerm /root/B/Task
```

touch: Creates an empty file or updates the timestamp.

```
[root@localhost D11]# touch F1
[root@localhost D11]# touch F2
[root@localhost D11]# touch F3
[root@localhost D11]# ls
F1  F2  F3
```

Absolute Path:

```
[root@localhost Lab_Practice]# pwd
/root/OS_Lab/Lab_Practice
[root@localhost Lab_Practice]# cd /root/OS_Lab/Lab_Activities
```

Relative Path:

```
[root@localhost Lab_Practice]# pwd
/root/OS_Lab/Lab_Practice
[root@localhost Lab_Practice]# cd ../Lab_Activities
```

Using Symbolic Method:

```
Welcome to Fedora 33 (riscv64)

[root@localhost ~]# touch LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rw-r--r-- 1 root root   0 Sep  8 16:54 LINUXOS
[root@localhost ~]# chmod u+rwx LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rwxr--r-- 1 root root   0 Sep  8 16:54 LINUXOS
[root@localhost ~]# chmod g+rw LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rwxrw-r-- 1 root root   0 Sep  8 16:54 LINUXOS
[root@localhost ~]# chmod o+r LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rwxrw-r-- 1 root root   0 Sep  8 16:54 LINUXOS
```

Using Numerical Method:

```
Welcome to Fedora 33 (riscv64)

[root@localhost ~]# touch LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rw-r--r-- 1 root root   0 Sep  8 16:56 LINUXOS
[root@localhost ~]# chmod 764 LINUXOS
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
-rwxrw-r-- 1 root root   0 Sep  8 16:56 LINUXOS
[root@localhost ~]#
```

cat:

```
Welcome to Fedora 33 (riscv64)

[root@localhost ~]# mkdir OSLAB
[root@localhost ~]# mkdir OSTHEORY
[root@localhost ~]# ls
bench.py  hello.c  OSLAB  OSTHEORY
[root@localhost ~]# cd OSLAB
[root@localhost OSLAB]# cat > overview.txt
Overview of Operating System^C
[root@localhost OSLAB]# cat > details.txt
Detailed study of key OS concept^C
[root@localhost OSLAB]# cat > applications.txt
Applications and Examples of OS concepts^C
[root@localhost OSLAB]# ls
applications.txt  details.txt  overview.txt
[root@localhost OSLAB]# cat overview.txt details.txt applications.txt > Combined
Text
[root@localhost OSLAB]# cat CombinedText
Overview of Operating SystemDetailed study of key OS conceptApplications and Exa
mples of OS concepts[root@localhost OSLAB]#
```

The time of the log entry in the 12-hour format with AM/PM.

```
[root@localhost ~]# date +%r
01:28:39 PM
```

The date of the log entry in the "Month Day Year" format.

```
[root@localhost ~]# date +"%B %d %Y"
September 15 2024
```

OR

```
[root@localhost ~]# date +"%m %d %Y"
09 15 2024
```

The full weekday name for the log entry.

```
[root@localhost ~]# date +%A
Sunday
```

The numerical representation of the month.

```
[root@localhost ~]# date +%m
09
```

The last two digits of the year.

```
[root@localhost ~]# date +%y
24
```

24-hour format with only hours and minutes

```
[root@localhost ~]# date +"%H:%M"
14:03
```

current year in four-digit format

```
[root@localhost ~]# date +%Y
2024
```

current date in a combined format of numerical day, month, and year and Time

```
[root@localhost ~]# date +"%d-%m-%Y %H:%M:%S"
15-09-2024 14:12:50
```

cal

```
[root@localhost ~]# cal
    October 2024
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Yesterday date:

```
[root@localhost ~]# date -d "yesterday" +"%d-%m-%Y"
06-10-2024
[root@localhost ~]#
```

Tomorrow date:

```
[root@localhost ~]# date -d "tomorrow" +"%d-%m-%Y"
08-10-2024
```

10 days ago:

```
[root@localhost ~]# date -d "10 days ago" +"%d-%m-%Y"
27-09-2024
```

Calculator of month December 2024:

```
[root@localhost ~]# cal 12 2024
    December 2024
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

print a welcoming message that includes the current date and time:

```
[root@localhost ~]# echo "Current time and date is: $(date)"
Current time and date is: Mon Oct  7 12:26:41 PM UTC 2024
```

Clear screen:

```
[root@localhost ~]# clear
```

sort strings in reverse alphabetical order:

```
[root@localhost ~]# cat > data.txt
Hungry Lady
Beauty and The Beast
Thirsty Crow
Cindrella
Zoo and The Monkey
^C
[root@localhost ~]# sort -r data.txt
Zoo and The Monkey
Thirsty Crow
Hungry Lady
Cindrella
Beauty and The Beast
```

sort file based on the values in the second column:

```
[root@localhost ~]# cat > records.txt
cat  3
city 4
lab  7
bag  9
form 2
[root@localhost ~]# sort -k 2 -n records.txt
form 2
cat  3
city 4
lab  7
bag  9
```

-o Option:
example.c:

```
[root@localhost ~]# nano example.c
```

```
  GNU nano 5.3                          example.c
#include<stdio.h>
int main() {
 printf("Hello, Example!\n");
 return 0;
}
```

To compile this program with a custom output name, we would use the -o option like this:

gcc -o my_example example.c

```
[root@localhost ~]# gcc -o myexample example.c
```

Chmod command:

```
chmod +x myfile
```

nano hello.cpp

```
[root@localhost ~]# nano hello.cpp
```

```
  GNU nano 5.3                                  hello.cpp
#include<iostream>
int main() {
 std::cout << "Hello, World!" << std::endl;
return 0;
}
```