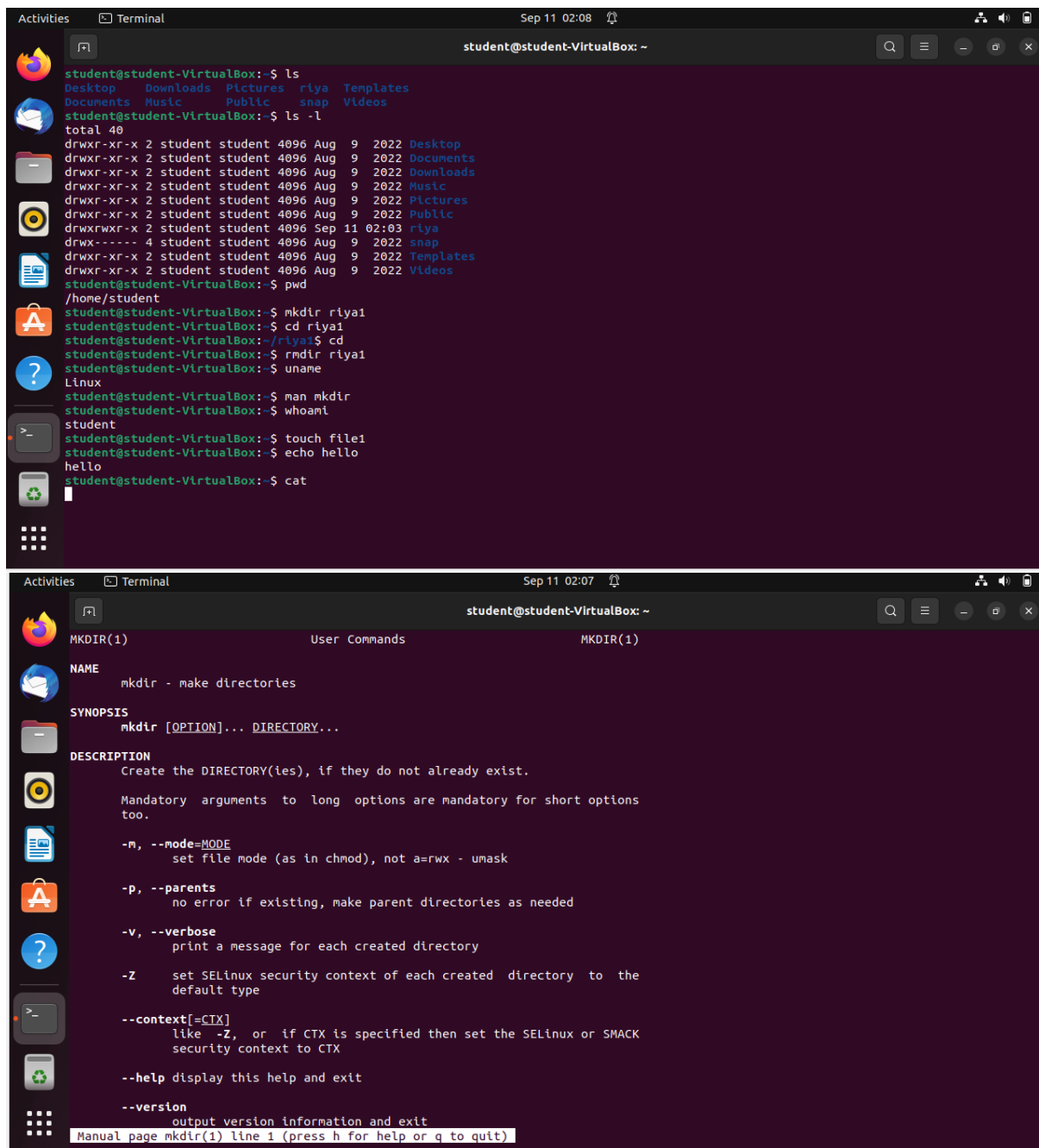


Activity 1 – Basic Unix Commands



The image displays two screenshots of a Linux terminal window, likely from a virtual machine named 'student-VirtualBox'.

The top screenshot shows a series of commands and their outputs:

```
student@student-VirtualBox:~$ ls
Desktop  Downloads  Pictures  riya  Templates
Documents Music      Public   snap  Videos
student@student-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 student student 4096 Aug  9 2022 Desktop
drwxr-xr-x 2 student student 4096 Aug  9 2022 Documents
drwxr-xr-x 2 student student 4096 Aug  9 2022 Downloads
drwxr-xr-x 2 student student 4096 Aug  9 2022 Music
drwxr-xr-x 2 student student 4096 Aug  9 2022 Pictures
drwxr-xr-x 2 student student 4096 Aug  9 2022 Public
drwxrwxr-x 2 student student 4096 Sep 11 02:03 riya
drwx----- 4 student student 4096 Aug  9 2022 snap
drwxr-xr-x 2 student student 4096 Aug  9 2022 Templates
drwxr-xr-x 2 student student 4096 Aug  9 2022 Videos
student@student-VirtualBox:~$ pwd
/home/student
student@student-VirtualBox:~$ mkdir riya1
student@student-VirtualBox:~$ cd riya1
student@student-VirtualBox:~/riya1$ cd
student@student-VirtualBox:~$ rmdir riya1
student@student-VirtualBox:~$ uname
Linux
student@student-VirtualBox:~$ man mkdir
student@student-VirtualBox:~$ whoami
student
student@student-VirtualBox:~$ touch file1
student@student-VirtualBox:~$ echo hello
hello
student@student-VirtualBox:~$ cat
```

The bottom screenshot shows the manual page for the 'mkdir' command, titled 'MKDIR(1) User Commands MKDIR(1)'. It includes sections for NAME, SYNOPSIS, DESCRIPTION, and a list of options:

```
NAME
mkdir - make directories

SYNOPSIS
mkdir [OPTION]... DIRECTORY...

DESCRIPTION
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.

-m, --mode=MODE
    set file mode (as in chmod), not a=rwx - umask

-p, --parents
    no error if existing, make parent directories as needed

-v, --verbose
    print a message for each created directory

-Z
    set SELinux security context of each created directory to the default type

--context[=CTX]
    like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX

--help
    display this help and exit

--version
    output version information and exit

Manual page mkdir(1) line 1 (press h for help or q to quit)
```

Activity-2: Creating a Directory Structure with mkdir and touch in Unix

```
student@student-VirtualBox: ~  
$ pwd  
/home/student  
$ mkdir dir3  
$ ls  
datefile  dir3  file1  mydir  r1ya  r1ya4  r1yac++  Videos  
Desktop  Documents  file2  Pictures  r1ya2  r1ya6  snap  
dir1  Downloads  Music  Public  r1ya3  r1ya.c++  Templates  
$ mkdir dir4 dir5 dir6  
$ ls  
dir1  Documents  Music  r1ya  r1ya6  Templates  
datefile  dir3  Downloads  mydir  r1ya2  r1ya.c++  Videos  
Desktop  dir4  file1  Pictures  r1ya3  r1yac++  
dir  dir6  file2  Public  r1ya4  snap  
$ mkdir dir3/dir12 dir6/dir41  
$ ls dir3  
dir12  
$ ls dir6  
dir41  
$
```

```
student@student-VirtualBox: ~  
$ mkdir -p mydir/{colors/{red,blue,green},blended/{orange,pink,yellow},shape/{circle,square,cube},animals/{mammals,reptiles}}  
$ tree mydir  
mydir  
├── animals  
│   ├── mammals  
│   └── reptiles  
├── animalsmammals  
├── animalsreptiles  
├── blended  
│   ├── orange,shape  
│   │   ├── circle,animals  
│   │   │   ├── mammals  
│   │   │   └── reptiles  
│   │   ├── cube,animals  
│   │   │   ├── mammals  
│   │   │   └── reptiles  
│   │   └── square,animals  
│   │       ├── mammals  
│   │       └── reptiles  
│   ├── pink,shape  
│   │   ├── circle,animals  
│   │   │   ├── mammals  
│   │   │   └── reptiles  
│   │   ├── cube,animals  
│   │   │   ├── mammals  
│   │   │   └── reptiles  
│   │   └── square,animals  
│   │       ├── mammals  
│   │       └── reptiles  
│   └── yellow,shape  
│       ├── circle,animals  
│       │   ├── mammals  
│       │   └── reptiles  
│       └── cube,animals  
│           ├── mammals  
│           └── reptiles  
└── mydir
```

Activity-3: Use of Date, Who and Sed Command

```
Activities Terminal Sep 19 21:51 student@student-VirtualBox: ~
student@student-VirtualBox:~$ date
Tuesday 19 September 2023 09:42:41 PM IST
student@student-VirtualBox:~$ date -u
Tuesday 19 September 2023 04:12:53 PM UTC
student@student-VirtualBox:~$ date --date="2 year ago"
Sunday 19 September 2021 09:43:12 PM IST
student@student-VirtualBox:~$ date --date="next tue"
Tuesday 26 September 2023 12:00:00 AM IST
student@student-VirtualBox:~$ cat >> datefile
sep 23 2018
nov 03 2019
^C
student@student-VirtualBox:~$
student@student-VirtualBox:~$ date --file=datefile
date: invalid date 'unix is a great operating system. unix is quick .'
Sunday 23 September 2018 12:00:00 AM IST
Sunday 03 November 2019 12:00:00 AM IST
Sunday 23 September 2018 12:00:00 AM IST
Sunday 03 November 2019 12:00:00 AM IST
student@student-VirtualBox:~$ date -r datefile
Tuesday 19 September 2023 09:44:35 PM IST
student@student-VirtualBox:~$ who
student tty2 2023-09-19 21:41 (tty2)
student@student-VirtualBox:~$ cat > geekfile.txt
unix is user friendly operating system.unix is great
unix is free os.learn operating system
unix linux which one you choose
^C
student@student-VirtualBox:~$ ^C
student@student-VirtualBox:~$ sed 's/unix/linux/' geekfile.txt
linux is user friendly operating system.unix is great
linux is free os.learn operating system
linux linux which one you choose
student@student-VirtualBox:~$ sed 's/unix/linux/2' geekfile.txt
unix is user friendly operating system.linux is great
unix is free os.learn operating system
```

```
Activities Terminal Sep 19 21:52 student@student-VirtualBox: ~
Sunday 03 November 2019 12:00:00 AM IST
student@student-VirtualBox:~$ date -r datefile
Tuesday 19 September 2023 09:44:35 PM IST
student@student-VirtualBox:~$ who
student tty2 2023-09-19 21:41 (tty2)
student@student-VirtualBox:~$ cat > geekfile.txt
unix is user friendly operating system.unix is great
unix is free os.learn operating system
unix linux which one you choose
^C
student@student-VirtualBox:~$ ^C
student@student-VirtualBox:~$ sed 's/unix/linux/' geekfile.txt
linux is user friendly operating system.unix is great
linux is free os.learn operating system
linux linux which one you choose
student@student-VirtualBox:~$ sed 's/unix/linux/2' geekfile.txt
unix is user friendly operating system.linux is great
unix is free os.learn operating system
unix linux which one you choose
student@student-VirtualBox:~$ sed 's/unix/linux/g' geekfile.txt
linux is user friendly operating system.linux is great
linux is free os.learn operating system
linux linux which one you choose
student@student-VirtualBox:~$ sed 's/unix/linux/p' geekfile.txt
linux is user friendly operating system.unix is great
linux is user friendly operating system.unix is great
linux is free os.learn operating system
linux is free os.learn operating system
linux linux which one you choose
linux linux which one you choose
student@student-VirtualBox:~$ sed '1,3s/unix/linux/' geekfile.txt
linux is user friendly operating system.unix is great
linux is free os.learn operating system
linux linux which one you choose
student@student-VirtualBox:~$
```

```
Activities Terminal Sep 20 13:48 student@student-VirtualBox: ~
student@student-VirtualBox:~$ date
Wednesday 20 September 2023 01:21:17 PM IST
student@student-VirtualBox:~$ date; who>hello.txt
Wednesday 20 September 2023 01:22:12 PM IST
student@student-VirtualBox:~$ cat hello.txt
student tty2 2023-09-20 13:18 (tty2)
student@student-VirtualBox:~$ cat newfile.txt
cat: newfile.txt: No such file or directory
student@student-VirtualBox:~$ cat > newfile.txt
This is unix operating system.
This is linux operating system.
This is MIET>
^C
student@student-VirtualBox:~$ sed -e "s/([^\ ]*) *([^\ ]*)/2 \1 /g" newfile.txt
is This unix operating system.
is This linux operating system.
is This MIET>
student@student-VirtualBox:~$ echo "Welcome To College" | sed -E 's/(<.)
> ^C
student@student-VirtualBox:~$ echo "Welcome To College" | sed -E 's/(<.)/(|)/g'
(|)elcome (|)o (|)ollege
student@student-VirtualBox:~$ echo "Welcome To College" | sed -E 's/(<.)/(1)/g'
(W)elcome (T)o (C)ollege
student@student-VirtualBox:~$
```

Activity – 4 : Introduction to Shell Scripting

```
Activities Terminal Sep 25 23:29 student@student-VirtualBox: ~
student@student-VirtualBox:~$ vi scriptmass.sh
student@student-VirtualBox:~$ bash scriptmass.sh
5
datefile mydir riyac++ scriptsample.sh
mydiranimalsmammals scripttagain.sh scriptttake.sh
mydiranimalsreptiles scriptdead.sh scriptttea.sh
dir newfile.txt scripteat.sh scripttteddy.sh
dir1 nss.sh scriptevil.sh scripttted.sh
dir3 Pictures scriptfloor.sh scriptwalk.sh
dir4 Public scriptglare.sh scriptwest.sh
dir6 read.sh scriptground.sh scriptwrite.sh
Documents riya scriptmass.sh snap
Downloads riya2 scriptqwerty.sh tedtalk.sh
file1 riya21 scriptqwertyy.sh Templates
file2 riya3 scriptread.sh Videos
geekfile.txt riya4 scriptrise.sh
hello.txt riya6 scriptrock.sh
Music riya.c++ scriptrose.sh
student@student-VirtualBox:~$ vi scriptweight.sh
student@student-VirtualBox:~$ bash scriptweight.sh
/home/student
student@student-VirtualBox:~$ vi scripteditor.sh
student@student-VirtualBox:~$ vi scriptedit.sh
student@student-VirtualBox:~$ bash scriptedit.sh
What is your name?
riya
How do you do, riya?
excellent
I am excellent too!
student@student-VirtualBox:~$ vi scriptdoor.sh
student@student-VirtualBox:~$ vi scriptrat.sh
student@student-VirtualBox:~$ bash scriptrat.sh
Total value : 4
student@student-VirtualBox:~$
```

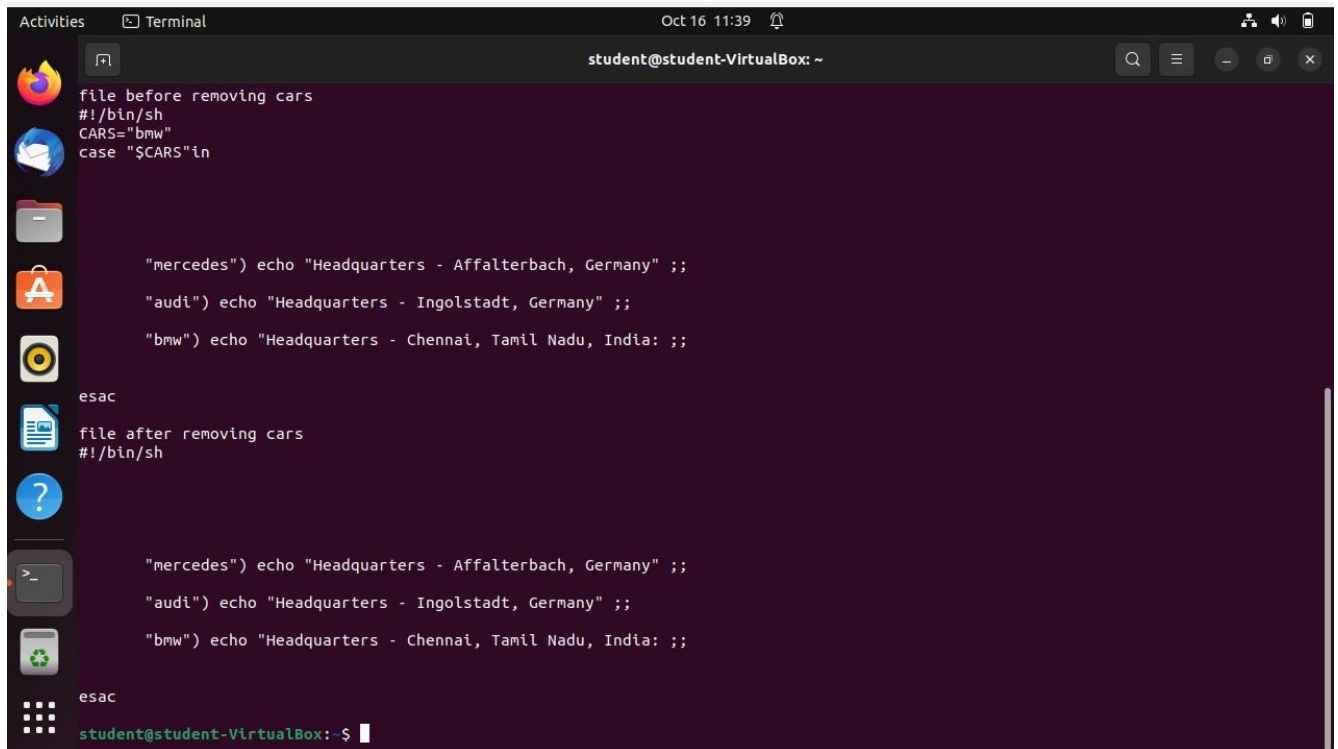
```
Activities Terminal Oct 2 11:53 student@student-VirtualBox: ~
student@student-VirtualBox:~$ vi editor.sh
student@student-VirtualBox:~$ bash editor.sh
First Index: Diya
Second Index: Prachi
Third Index: Smriti
student@student-VirtualBox:~$ vi link.sh
student@student-VirtualBox:~$ bash link.sh
a is not equal to b
student@student-VirtualBox:~$ vi seed.sh
student@student-VirtualBox:~$ bash seed.sh
a is equal to b
student@student-VirtualBox:~$ vi brand.sh
student@student-VirtualBox:~$ bash brand.sh
Headquarters - Chennai, Tamil Nadu, India
student@student-VirtualBox:~$
```

Activity-5: Looping Statements

```
Activities Terminal Oct 13 13:16 student@student-VirtualBox: ~
student@student-VirtualBox:~$ vi yelling.sh
student@student-VirtualBox:~$ bash yelling.sh
Iteration no 1
Iteration no 2
Iteration no 3
Iteration no 4
Iteration no 5
Iteration no 6
Iteration no 7
Iteration no 8
Iteration no 9
Iteration no 10
student@student-VirtualBox:~$ vi continue.sh
student@student-VirtualBox:~$ bash continue.sh
Iteration no 1
Iteration no 2
Iteration no 3
Iteration no 4
Iteration no 6
Iteration no 7
Iteration no 8
Iteration no 9
Iteration no 10
student@student-VirtualBox:~$ vi lessop.sh
student@student-VirtualBox:~$ bash lessop.sh
0
1
2
3
4
5
6
7
8
9
student@student-VirtualBox:~$ vi greatrop.sh
```

```
Activities Terminal Oct 13 13:15 student@student-VirtualBox: ~
7
8
9
student@student-VirtualBox:~$ vi greatrop.sh
student@student-VirtualBox:~$ bash greater.sh
bash: greater.sh: No such file or directory
student@student-VirtualBox:~$ bash greatrop.sh
0
1
2
3
4
5
6
7
8
9
10
student@student-VirtualBox:~$ vi colors.sh
student@student-VirtualBox:~$ bash colors.sh
colors.sh: line 1: !#/bin/sh: No such file or directory
COLOR: red
COLOR: green
COLOR: blue
student@student-VirtualBox:~$ vi sun.sh
student@student-VirtualBox:~$ bash sun.sh
Enter your name:tom
Is tom correct? n
Enter your name:jerry
Is jerry correct? y
```

ACTIVITY: - Shell Scripting



The screenshot shows a terminal window titled "Terminal" with the date and time "Oct 16 11:39". The user is logged in as "student" on a machine named "student-VirtualBox". The terminal displays a shell script that defines car headquarters locations based on the value of the "CARS" variable. The script is divided into two sections: "file before removing cars" and "file after removing cars". Both sections use a case statement to handle three possible values for "CARS": "mercedes", "audi", and "bmw". Each case prints a specific headquarters location. The script ends with "esac" and a prompt for the user to run it.

```
file before removing cars
#!/bin/sh
CARS="bmw"
case "$CARS" in

    "mercedes") echo "Headquarters - Affalterbach, Germany" ;;

    "audi") echo "Headquarters - Ingolstadt, Germany" ;;

    "bmw") echo "Headquarters - Chennai, Tamil Nadu, India: " ;;

esac

file after removing cars
#!/bin/sh

    "mercedes") echo "Headquarters - Affalterbach, Germany" ;;

    "audi") echo "Headquarters - Ingolstadt, Germany" ;;

    "bmw") echo "Headquarters - Chennai, Tamil Nadu, India: " ;;

esac

student@student-VirtualBox:~$
```

```
Activities Terminal Oct 16 11:39 student@student-VirtualBox: ~
student@student-VirtualBox:~$ vi regret.sh
student@student-VirtualBox:~$ bash regret.sh
enter your name
riys

wtmp begins Tue Aug 9 13:01:37 2022
student@student-VirtualBox:~$ vi fox.sh
student@student-VirtualBox:~$ bash fox.sh
enter a file
science.sh
enter starting line
2
enter ending line
3
a=10
b=20
student@student-VirtualBox:~$ vi facts.sh
student@student-VirtualBox:~$ bash facts.sh
facts.sh: line 3: [: missing `]'
Enter a word
ant
lines containing given word are deleted
student@student-VirtualBox:~$ vi greedy.sh
student@student-VirtualBox:~$ bash greedy.sh
enter file name
english.sh
enter word
cars
file before removing cars
#!/bin/sh
CARS="bmw"
case "$CARS" in

bdiya@ubuntu:~$ vi odd.sh
bdiya@ubuntu:~$ bash odd.sh
Enter number and I will check if it is even or odd 6
Your number is even.
bdiya@ubuntu:~$ vi fact.sh
bdiya@ubuntu:~$ bash fact.sh
Enter the number you want to get factorial for
5
120
bdiya@ubuntu:~$ vi concat.sh
bdiya@ubuntu:~$ bash concat.sh
the secret is..bash
bdiya@ubuntu:~$
```


EXPERIMENT NO: 7

OBJECTIVE: Write a shell script and C program to perform the following string operations:

- a) To extract a substring from a given string.
- b) To find the length of a given string.

Compare the running time of above shell script and c program using the time command.

```
student@student-VirtualBox: ~
student@student-VirtualBox:~$ vi length.sh
student@student-VirtualBox:~$ bash length.sh
enter a string
hello world
length of input string is : 11
student@student-VirtualBox:~$ gedit lengthc.c
student@student-VirtualBox:~$ gcc lengthc.c
student@student-VirtualBox:~$ ./a.out
enter the string
hello_world
length of a given string is 11
student@student-VirtualBox:~$ time ./length.sh
bash: ./length.sh: Permission denied

real    0m0.001s
user    0m0.001s
sys     0m0.000s
student@student-VirtualBox:~$ time ./lengthc.c
bash: ./lengthc.c: Permission denied

real    0m0.001s
user    0m0.001s
sys     0m0.000s
student@student-VirtualBox:~$
```

```
nikhil@Ubuntu:~/Desktop/ShellWith_C$ gcc cprog.c -o cpi
nikhil@Ubuntu:~/Desktop/ShellWith_C$ ./cpi
input a string:
nikhilgupta
enter start point and end point:
3
6

required substring is: khilgu
nikhil@Ubuntu:~/Desktop/ShellWith_C$ cat cprog.c
#include<stdio.h>
#include<string.h>
int main(){
    char string[1000],sub[1000];
    int start=0,end=0,c=0;
    printf("input a string: \n");
    scanf("%s",string);
    printf("enter start point and end point: \n");
    scanf("%d %d",&start,&end);
    while(c<end){
        sub[c]=string[start+c-1];
        c++;
    }
    sub[c]='\0';
    printf("\nrequired substring is: %s\n",sub);
    return 0;
}
```

```
nikhil@Ubuntu:~/Desktop/ShellWith_C$ ./shell.sh
total characters in a string: 23
extracting 1st 10 characters of string
w
ectracting from specific character onwards
you on linux
extract character between given range
you on
nikhil@Ubuntu:~/Desktop/ShellWith_C$ cat shell.sh
str="we welcome you on linux"
echo "total characters in a string: ${#str}"

echo "extracting 1st 10 characters of string"
substr="${str:0:1}"
echo "$substr"

echo "ectracting from specific character onwards"
substr="${str:11}"
echo "$substr"

echo "extract character between given range"
substr="${str:11:6}"
echo "$substr"
nikhil@Ubuntu:~/Desktop/ShellWith_C$
```

```
nikhil@Ubuntu:~/Desktop/ShellWith_C$ gcc cprog.c -o cpp
nikhil@Ubuntu:~/Desktop/ShellWith_C$ ls -l
total 28
-rwxrwxr-x 1 nikhil nikhil 16104 Oct 15 21:48 cpp
-rwxrwxr-x 1 nikhil nikhil 357 Oct 15 19:06 cprog.c
-rwxrwxr-x 1 nikhil nikhil 324 Oct 13 11:23 shell.sh
-rwxrwxr-x 1 nikhil nikhil 146 Oct 13 11:27 substring.sh
nikhil@Ubuntu:~/Desktop/ShellWith_C$ time ./cpp
input a string:
helloworld
enter start point and end point:
3
5

required substring is: llowo

real    0m4.558s
user    0m0.000s
sys     0m0.002s
nikhil@Ubuntu:~/Desktop/ShellWith_C$
```

EXPERIMENT NO: 8

OBJECTIVE: Write a C program that takes, as a command line argument, the number of megabytes of memory it will use and during execution it should consume that much memory. Observe memory usage during program execution using free command.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>

int main(int argc, char* argv[]){
    printf("Current Process ID = %d\n",getpid());
    long long int size = ((long long int)atoi(argv[1]))*1024*1024;
    int* buffer = (int*)malloc(size);

    time_t endwait, seconds, start;
    seconds=atoi(argv[2]);
    start=time(NULL);
    endwait = start+seconds;

    while(start<endwait){
        printf(".");
        fflush(stdout);
        for(long long int i=0; i<size/sizeof(int); i++){
            buffer[i] = i;
        }
        start = time(NULL);
    }
    printf("(done)\n");
    return 0;
}
```



```
--mega      show output in megabytes
--giga      show output in gigabytes
--tera      show output in terabytes
--peta      show output in petabytes
-k, --kibi  show output in kibibytes
-m, --mebi  show output in mebibytes
-g, --gibi  show output in gibibytes
--tebi      show output in tebibytes
--pebi      show output in pebibytes
-h, --human  show human-readable output
--si        use powers of 1000 not 1024
-l, --lohi  show detailed low and high memory statistics
-t, --total  show total for RAM + swap
-s N, --seconds N repeat printing every N seconds
-c N, --count N repeat printing N times, then exit
-w, --wide   wide output

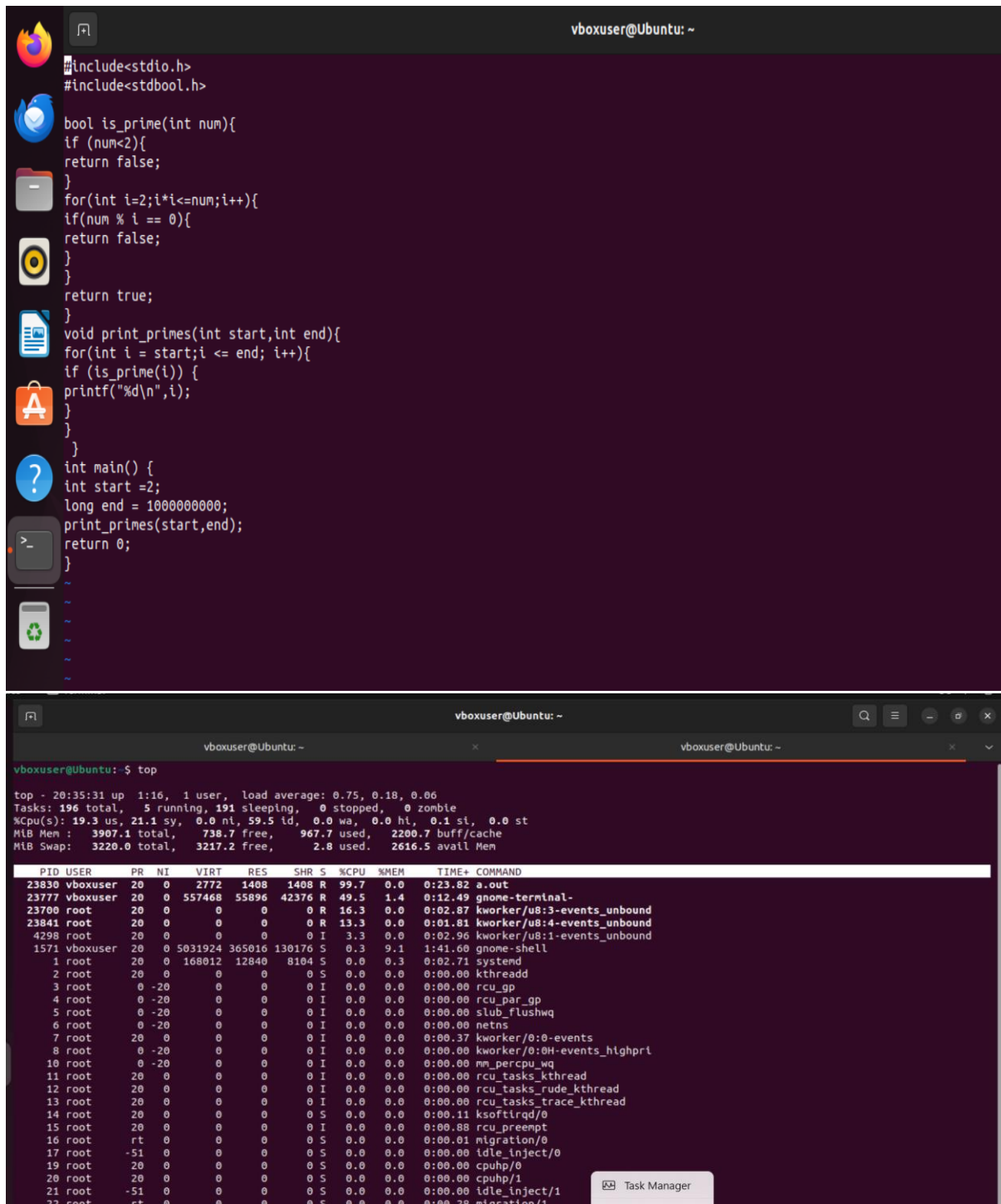
--help      display this help and exit
-V, --version output version information and exit

For more details see free(1).
student@student-VirtualBox:~$ free -m
              total        used        free      shared    buff/cache   available
Mem:          1977         686         1059          12          231        1128
Swap:          1873           84         1788
student@student-VirtualBox:~$
```

```
Segmentation fault (core dumped)
student@student-VirtualBox:~$ ./a.out 1000 20
Current Process ID = 2611
.....(done)
student@student-VirtualBox:~$
```

EXPERIMENT NO: 10

OBJECTIVE: Write a program in c that checks whether it is CPU or I/O bound.



The image shows a C program in a terminal window and its execution results. The program, named `a.out`, checks if a number is prime and prints primes between 2 and 1000000000. It is then run using `top` to show system resource usage.

```
#include<stdio.h>
#include<stdbool.h>

bool is_prime(int num){
    if (num<2){
        return false;
    }
    for(int i=2;i*i<=num;i++){
        if(num % i == 0){
            return false;
        }
    }
    return true;
}

void print_primes(int start,int end){
    for(int i = start;i <= end; i++){
        if (is_prime(i)) {
            printf("%d\n",i);
        }
    }
}

int main() {
    int start =2;
    long end = 1000000000;
    print_primes(start,end);
    return 0;
}
```

Execution results from the `top` command:

```
top - 20:35:31 up 1:16, 1 user, load average: 0.75, 0.18, 0.06
Tasks: 196 total, 5 running, 191 sleeping, 0 stopped, 0 zombie
%Cpu(s): 19.3 us, 21.1 sy, 0.0 ni, 59.5 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 3907.1 total, 738.7 free, 967.7 used, 2200.7 buff/cache
MiB Swap: 3220.0 total, 3217.2 free, 2.8 used, 2616.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
23830	vboxuser	20	0	2772	1408	1408	R	99.7	0.0	0:23.82	a.out
23777	vboxuser	20	0	557468	55896	42376	R	49.5	1.4	0:12.49	gnome-terminal-
23700	root	20	0	0	0	0	R	16.3	0.0	0:02.87	kworker/u8:3-events_unbound
23841	root	20	0	0	0	0	R	13.3	0.0	0:01.81	kworker/u8:4-events_unbound
4298	root	20	0	0	0	0	I	3.3	0.0	0:02.96	kworker/u8:1-events_unbound
1571	vboxuser	20	0	5031924	365016	130176	S	0.3	9.1	1:41.60	gnome-shell
1	root	20	0	168012	12840	8104	S	0.0	0.3	0:02.71	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
7	root	20	0	0	0	0	I	0.0	0.0	0:00.37	kworker/0:0-events
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:00.11	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:00.88	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.01	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
21	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.28	migration/1

```
vboxuser@Ubuntu: ~  
#include<stdio.h>  
void main(){  
int i,j,res,n;  
  
printf("enter number of which you want to find table : ");  
  
scanf("%d",&n);  
  
printf("enter the number to stop the table : ");  
scanf("%d",&j);  
  
for(i=0;i<=j;i++){  
res=n*i;  
printf("%d * %d = %d\n",i,j,res);  
}  
}  
~  
~  
~  
~  
~  
~
```

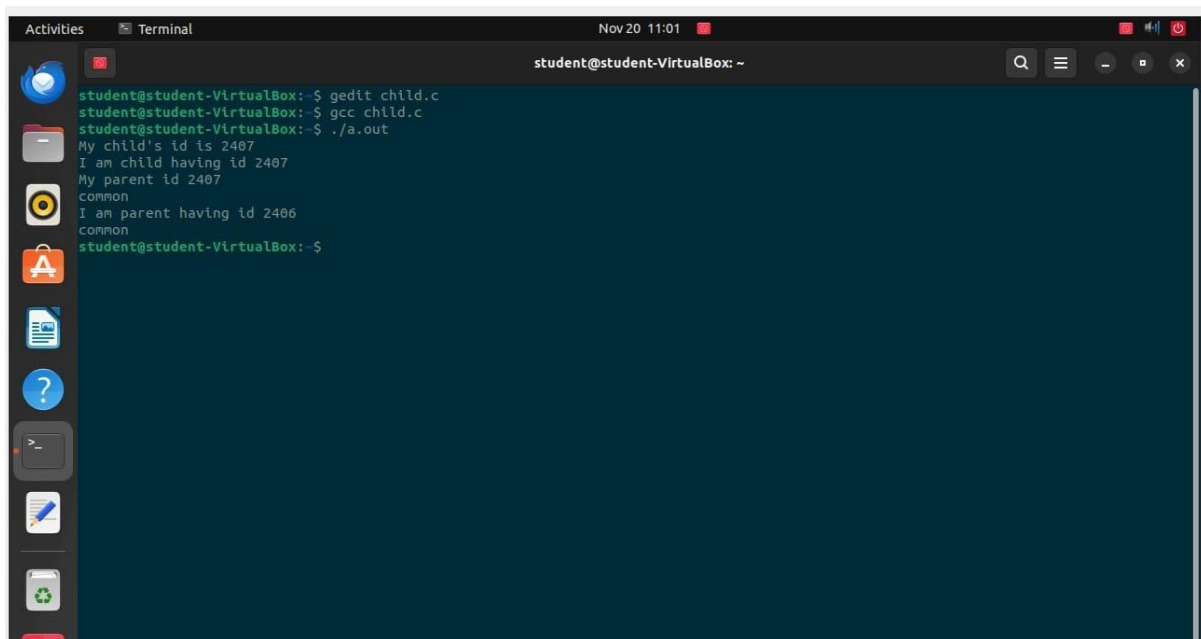
```
vboxuser@Ubuntu: ~  
vboxuser@Ubuntu: $ vi table.c  
vboxuser@Ubuntu: $ gcc table.c  
vboxuser@Ubuntu: $ ./a.out  
enter number of which you want to find table : 78  
enter the number to stop the table : 1000000000
```

```
vboxuser@Ubuntu: ~  
top - 20:48:39 up 1:29, 1 user, load average: 0.26, 0.10, 0.04  
Tasks: 196 total, 2 running, 194 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 11.2 us, 28.7 sy, 0.0 ni, 59.9 id, 0.1 wa, 0.0 hi, 0.1 st, 0.0 sr  
Mem: 3907.1 total, 730.5 free, 974.6 used, 2202.1 buff/cache  
Mem Swap: 3220.0 total, 3217.2 free, 2.8 used, 2609.7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
23942	vboxuser	20	0	558572	56652	42392	R	64.8	1.4	0:04.84	gnome-terminal-
23996	vboxuser	20	0	2772	1408	1408	S	59.8	0.0	0:03.98	a.out
23889	root	20	0	0	0	0	I	14.6	0.0	0:00.93	kworker/u8:2-events_unbound
23700	root	20	0	0	0	0	I	11.0	0.0	0:05.07	kworker/u8:3-events_unbound
23873	root	20	0	0	0	0	I	10.6	0.0	0:00.78	kworker/u8:0-events_unbound
4298	root	20	0	0	0	0	I	9.0	0.0	0:04.87	kworker/u8:1-ext4-rsv-conversion
1571	vboxuser	20	0	5036096	368968	130048	S	1.3	9.2	2:14.61	gnome-shell
15	root	20	0	0	0	0	I	0.3	0.0	0:01.00	rcu_preempt
179	root	20	0	0	0	0	S	0.3	0.0	0:00.34	jbd2/sda3-8
1652	vboxuser	20	0	338396	14592	12672	S	0.3	0.4	0:00.16	goa-identity-se
22943	root	20	0	0	0	0	I	0.3	0.0	0:00.53	kworker/1:1-events
23997	vboxuser	20	0	13100	4096	3328	R	0.3	0.1	0:00.01	top
1	root	20	0	168012	12840	8104	S	0.0	0.3	0:02.71	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:00.11	ksoftirqd/0
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.01	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_1
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1

EXPERIMENT NO: 10

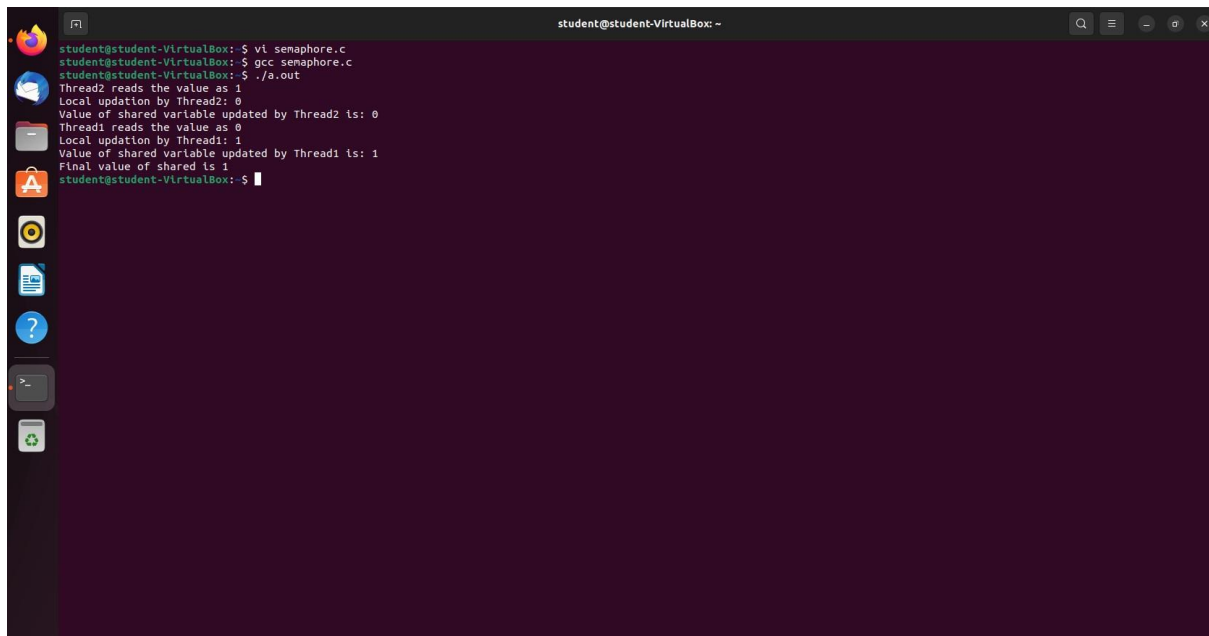
OBJECTIVE: Write a program in C that creates a child process, waits for the termination of the child and lists its PID.



```
student@student-VirtualBox: ~  
student@student-VirtualBox: $ gedit child.c  
student@student-VirtualBox: $ gcc child.c  
student@student-VirtualBox: $ ./a.out  
My child's id is 2407  
I am child having id 2407  
My parent id 2407  
common  
I am parent having id 2406  
common  
student@student-VirtualBox: $
```


EXPERIMENT NO: 11

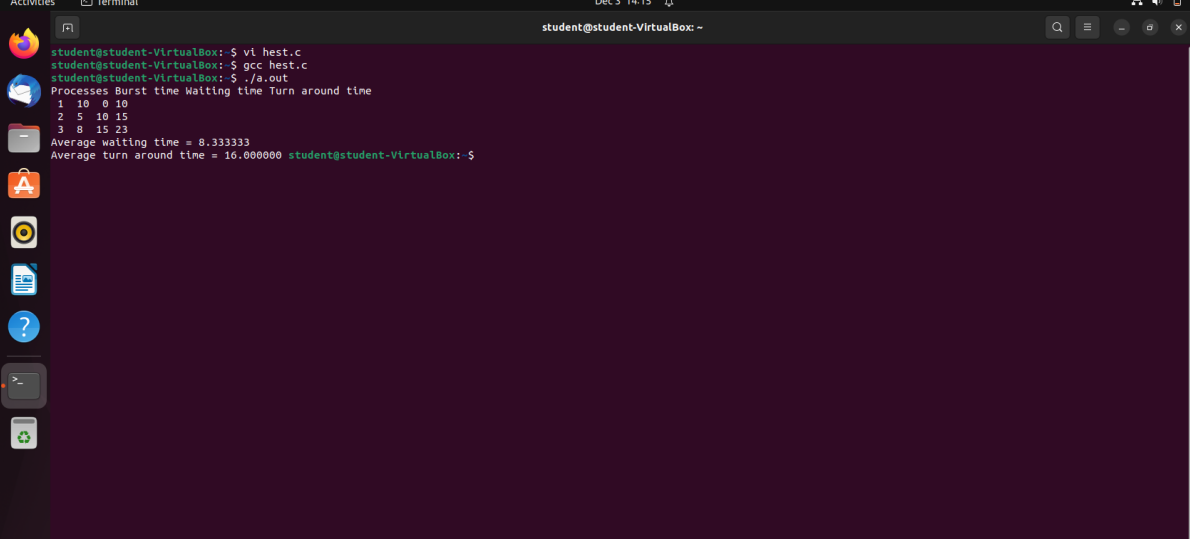
OBJECTIVE: Write a C program to demonstrate wait and signal operations in semaphores.



```
student@student-VirtualBox: ~  
student@student-VirtualBox: $ vi semaphore.c  
student@student-VirtualBox: $ gcc semaphore.c  
student@student-VirtualBox: $ ./a.out  
Thread2 reads the value as 1  
Local updation by Thread2: 0  
Value of shared variable updated by Thread2 is: 0  
Thread1 reads the value as 0  
Local updation by Thread1: 1  
Value of shared variable updated by Thread1 is: 1  
Final value of shared is 1  
student@student-VirtualBox: ~
```

EXPERIMENT NO: 12

OBJECTIVE: Write a C program to demonstrate program for FCFS CPU Scheduling.



The screenshot shows a terminal window titled "Terminal" with the date and time "Dec 3 14:15". The user is logged in as "student" on a machine named "student-VirtualBox". The terminal displays the following commands and output:

```
student@student-VirtualBox:~$ vi hest.c
student@student-VirtualBox:~$ gcc hest.c
student@student-VirtualBox:~$ ./a.out
Processes Burst time Waiting time Turn around time
1 10 0 10
2 5 10 15
3 8 15 23
Average waiting time = 8.333333
Average turn around time = 16.000000 student@student-VirtualBox:~$
```

The output shows the execution of a C program that implements FCFS CPU scheduling. The program takes three processes as input, each with a burst time and a waiting time. The output shows the burst time, waiting time, and turn around time for each process, along with the average waiting time and average turn around time.