

OPERATING SYSTEM



Assignment # 02

STUDENT NAME: UMAIR AZAD

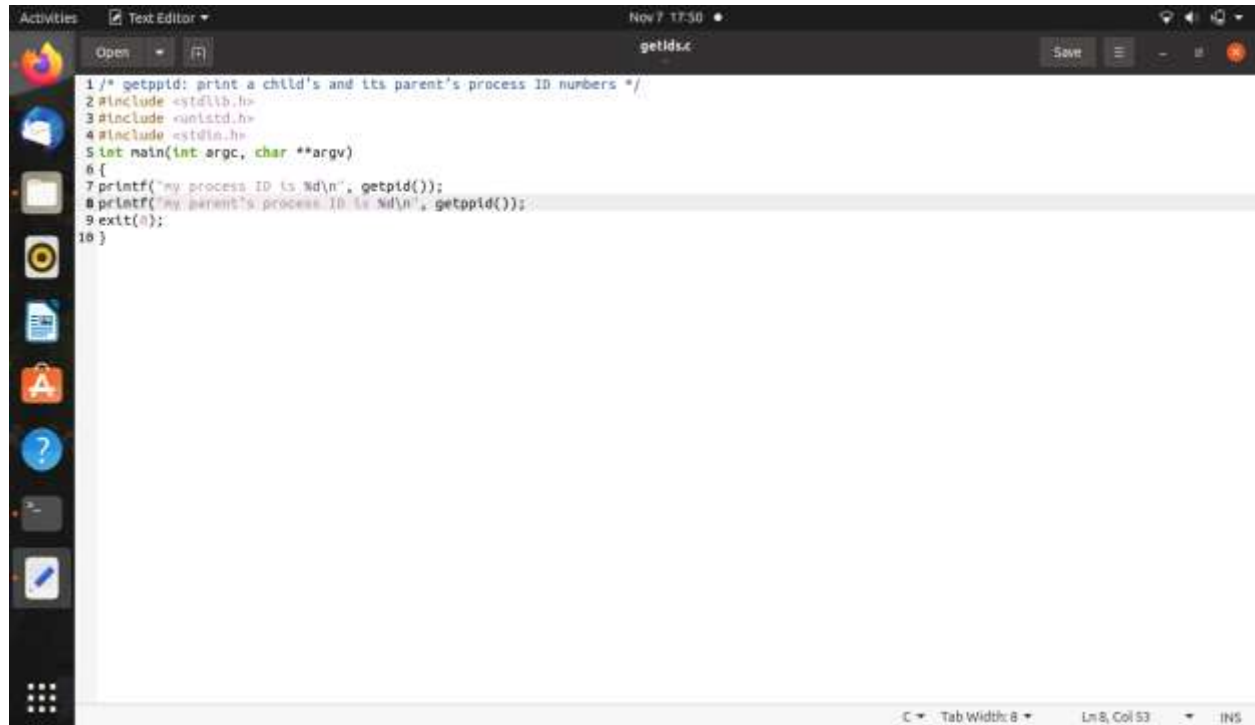
ROLL NO : 19P-0030

SECTION : 5A

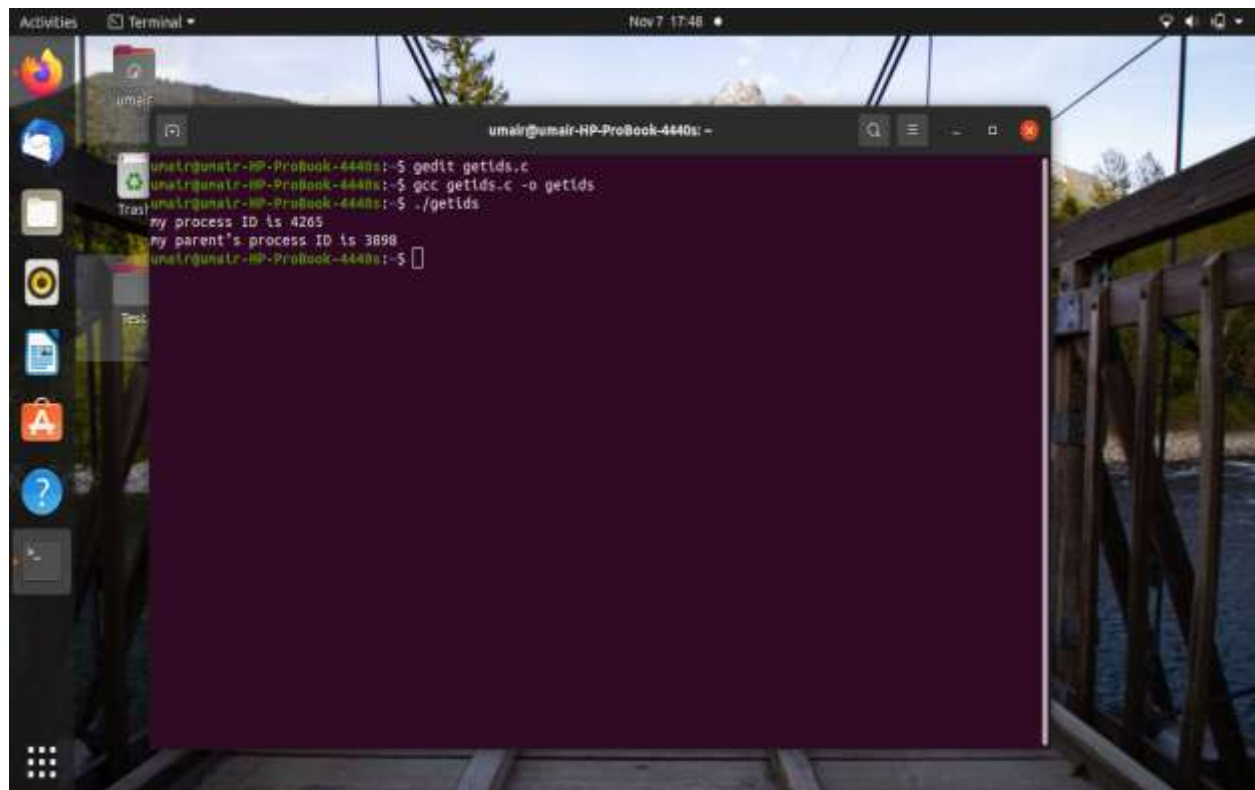
**Fast National University of Computer and Emerging
Sciences, Peshawar**

Department of Computer Science

Q#01:

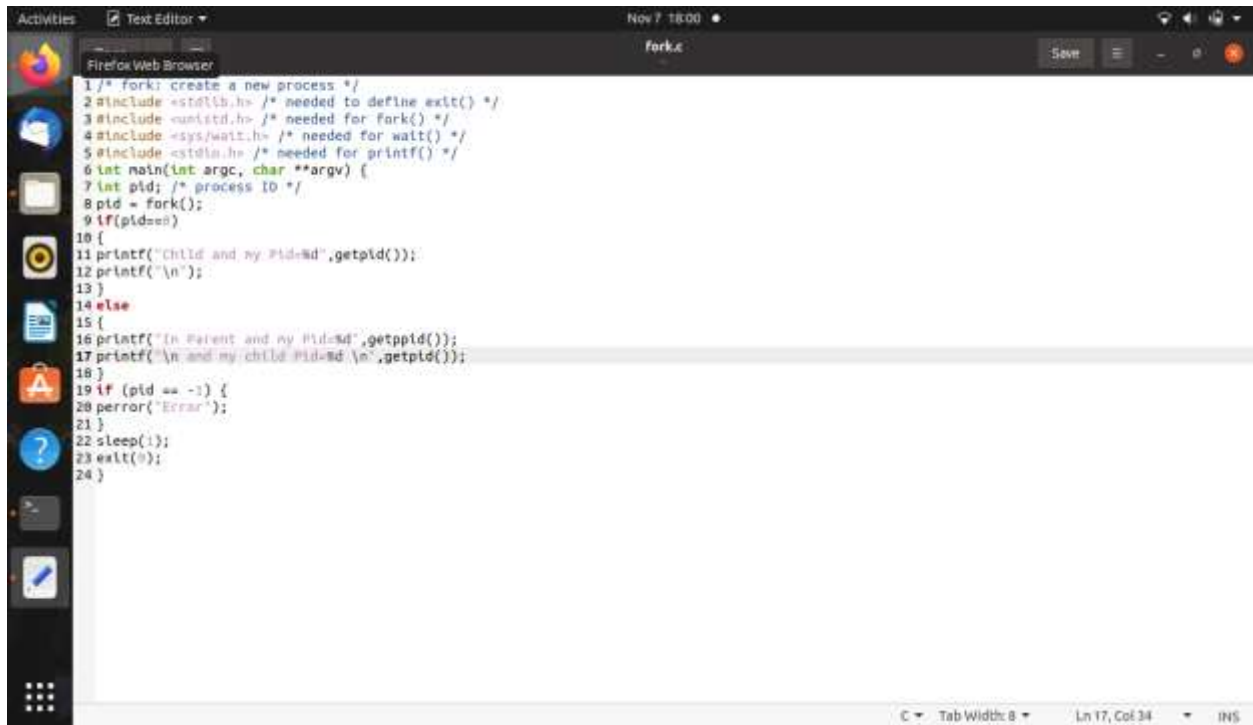


```
1 /* getpid: print a child's and its parent's process ID numbers */
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <stdio.h>
5 int main(int argc, char **argv)
6 {
7     printf("my process ID is %d\n", getpid());
8     printf("my parent's process ID is %d\n", getppid());
9     exit(0);
10 }
```



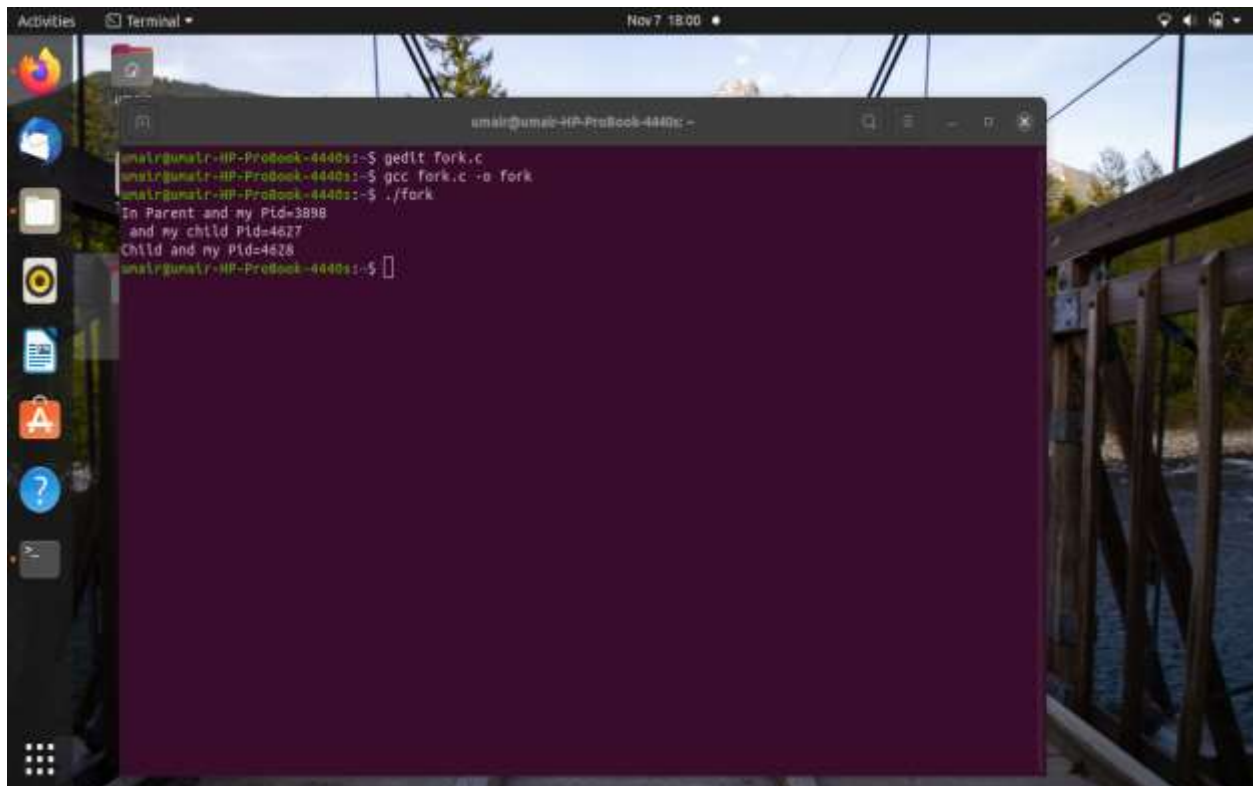
```
umair@umair-HP-ProBook-4440s: ~$ gedit getids.c
umair@umair-HP-ProBook-4440s: ~$ gcc getids.c -o getids
umair@umair-HP-ProBook-4440s: ~$ ./getids
my process ID is 4265
my parent's process ID is 3898
umair@umair-HP-ProBook-4440s: ~$
```

Q#02:



The screenshot shows a text editor window titled 'fork.c' with the following C code:

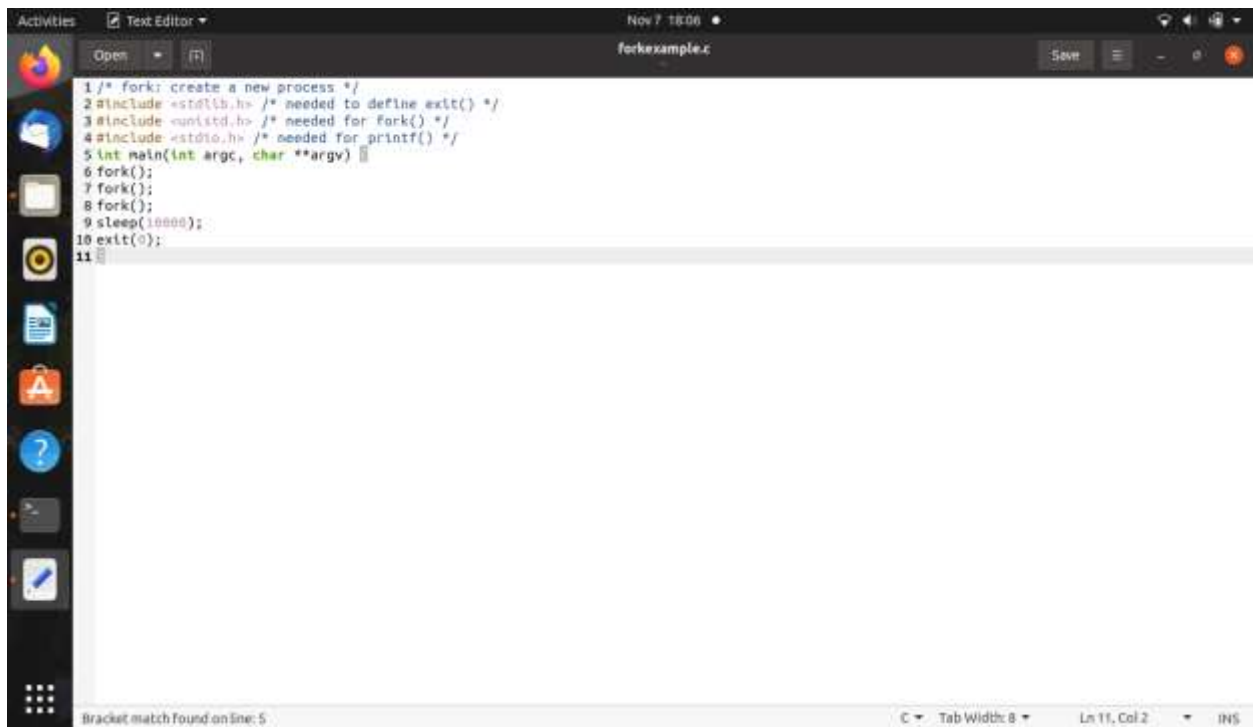
```
1 /* fork: create a new process */
2 #include <stdlib.h> /* needed to define exit() */
3 #include <unistd.h> /* needed for fork() */
4 #include <sys/wait.h> /* needed for wait() */
5 #include <stdio.h> /* needed for printf() */
6 int main(int argc, char **argv) {
7     int pid; /* process ID */
8     pid = fork();
9     if(pid==0)
10 {
11     printf("Child and my Pid=%d",getpid());
12     printf("\n");
13 }
14 else
15 {
16     printf("\n Parent and my Pid=%d",getppid());
17     printf("\n and my child Pid=%d",getpid());
18 }
19 if (pid == -1) {
20     perror("Error");
21 }
22 sleep(1);
23 exit(0);
24 }
```



The screenshot shows a terminal window with the following commands and output:

```
umair@umair-HP-ProBook-4440s:~$ gedit fork.c
umair@umair-HP-ProBook-4440s:~$ gcc fork.c -o fork
umair@umair-HP-ProBook-4440s:~$ ./fork
In Parent and my Pid=3898
and my child Pid=4627
Child and my Pid=4628
umair@umair-HP-ProBook-4440s:~$
```

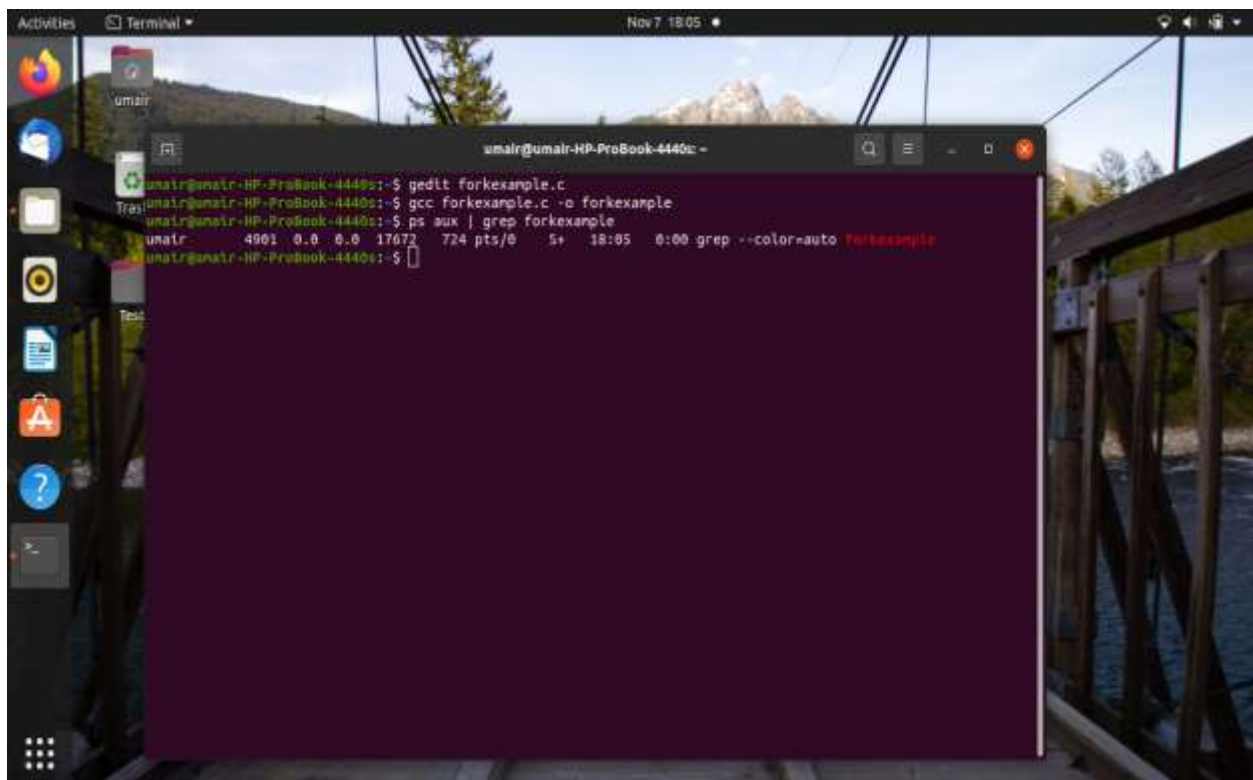
Q#03:



The screenshot shows a text editor window titled 'forkexample.c' with the following code:

```
1 /* fork: create a new process */
2 #include <stdlib.h> /* needed to define exit() */
3 #include <unistd.h> /* needed for fork() */
4 #include <stdio.h> /* needed for printf() */
5 int main(int argc, char **argv) {
6     fork();
7     fork();
8     fork();
9     sleep(10000);
10    exit(0);
11 }
```

The status bar at the bottom indicates 'Bracket match found on line: 5' and 'Ln 11, Col 2'.



The screenshot shows a terminal window with the following commands and output:

```
umair@umair-HP-ProBook-4440s:~$ gcc forkexample.c -o forkexample
umair@umair-HP-ProBook-4440s:~$ ps aux | grep forkexample
umair  4901  0.0  0.0 17672  724 pts/0    S+   18:05   0:00 grep --color=auto forkexample
umair@umair-HP-ProBook-4440s:~$
```

The terminal output shows the process list for 'forkexample' and the 'grep' command output.