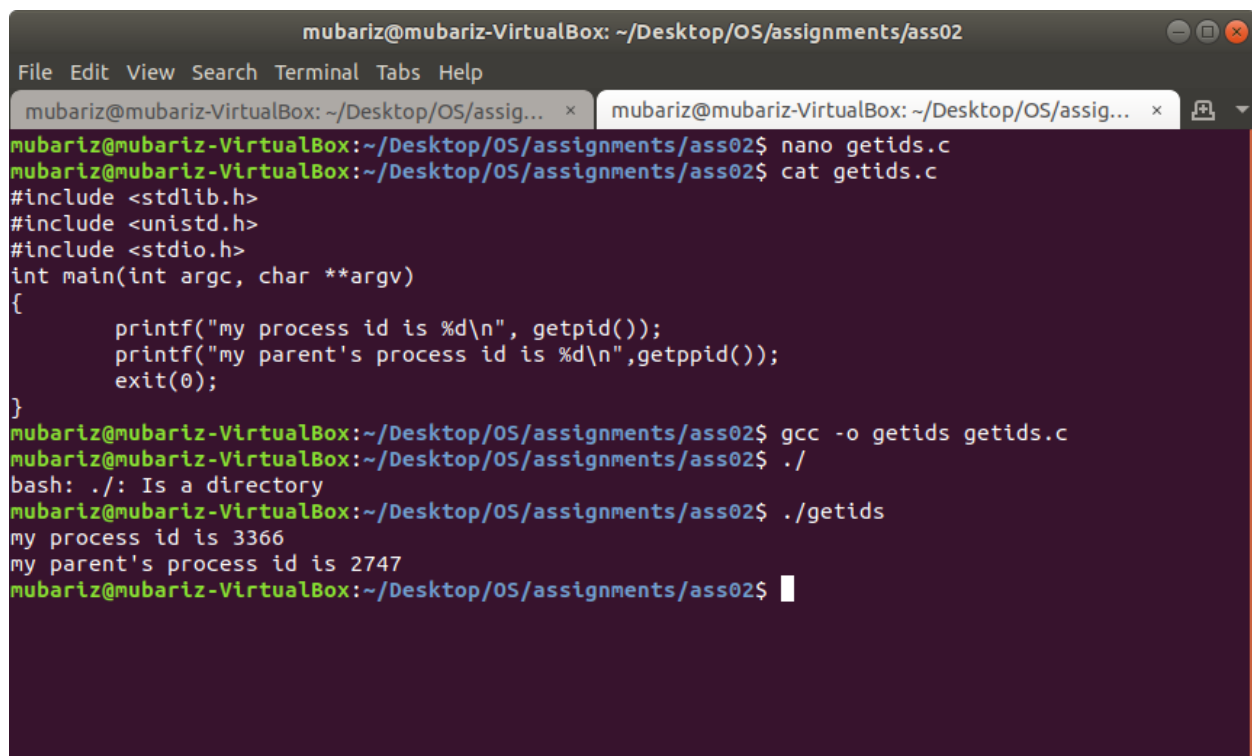


Mubariz Ahmed Khan P180010(B)

OS ASSIGNMENT #2

Q1:

A screenshot of a terminal window titled 'mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02'. The terminal shows the following commands and output:

```
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$ nano getids.c
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$ cat getids.c
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
int main(int argc, char **argv)
{
    printf("my process id is %d\n", getpid());
    printf("my parent's process id is %d\n",getppid());
    exit(0);
}
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ gcc -o getids getids.c
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ ./
bash: ./: Is a directory
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ ./getids
my process id is 3366
my parent's process id is 2747
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$
```

Q2:

Activities Screenshot 21:51

cs206-s20-a-02.pdf - Mozilla Firefox

Login Required - SLATE x cs206-s20-a-02.pdf x Lana Del Rey - Heroin - Y x New Tab x

slate.nu.edu.pk/access/content/attachment/PWRC5220SPRING2020CS/Assignment...

11 perror("Error");
12 }
13 sleep(1);
14 exit(0);
15 }

(b) Complete the two cases (p should print "In child" and both the parent's and the c

. Compile and run the following p

```
1 /* fork: create a new process */
2 #include <stdlib.h> /* needed to define
3 #include <unistd.h> /* needed for fork()
4 #include <stdio.h> /* needed for printf()
5 int main(int argc, char **argv) {
6
7     fork();
8     fork();
9     fork();
10
11     sleep(10000);
12     exit(0);
13 }
```

```
#include <sys/wait.h>
#include <stdio.h>
int main(int argc, char **argv){
    int pid;
    int status;
    pid = fork();

    if (pid == -1){
        perror("Error");
    }

    if (pid == 1)
    {
        printf("this is the child fork call %d\n", getpid());
    }

    else if(pid == 0){
        printf("this is the parent's process id %d\n", getppid());
        printf("this is child id? %d\n", getpid());
        wait(&status);
    }

    sleep(1);
    exit(0);
}
```

In another terminal, issue the command: `ps aux | grep forkexample`

Notice how many processes are currently running

Activities Terminal 21:53

cs206-s20-a-02.pdf - Mozilla Firefox

Login Required - SLATE x cs206-s20-a-02.pdf x A2ss2.png x

11 perror("Error");
12 }
13 sleep(1);
14 exit(0);
15 }

(b) Complete the two should print "In child" and both the parent's

. Compile and run the fo

```
1 /* fork: create a new process */
2 #include <stdlib.h> /* needed to define
3 #include <unistd.h> /* needed for fork()
4 #include <stdio.h> /* needed for printf()
5 int main(int argc, char **argv) {
6
7     fork();
8     fork();
9     fork();
10
11     sleep(10000);
12     exit(0);
13 }
```

```
    }

    if (pid == 1)
    {
        printf("this is the child fork call %d\n", getpid());
    }

    else if(pid == 0){
        printf("this is the parent's process id %d\n", getppid());
        printf("this is child id? %d\n", getpid());
        wait(&status);
    }

    sleep(1);
    exit(0);
}
```

```
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$ gcc -o f2 f2.c
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$ ./f2
this is the parent's process id 3937
this is child id? 3938
mubariz@mubariz-VirtualBox: ~/Desktop/OS/assignments/ass02$
```

In another terminal, issue the command: `ps aux | grep forkexample`

Notice how many processes are currently running

Q3:

Activities Screenshot 21:58 cs206-s20-a-02.pdf - Mozilla Firefox

mubariz@mubariz-VirtualBox: ~

```
File Edit View Search Terminal Help
mubariz@mubariz-VirtualBox:~$ ps aux | grep forkexample
mubariz 4002 0.0 0.0 4372 796 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4003 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4004 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4005 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4006 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4007 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4008 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4009 0.0 0.0 4372 72 pts/1 S+ 21:58 0:00 ./forkexample
mubariz 4030 0.0 0.0 21860 1004 pts/2 S+ 21:58 0:00 grep --color=au
to forkexample
mubariz@mubariz-VirtualBox:~$
```

```
7 fork();
8 fork();
9 fork();
10
11 sleep(10000);
12 exit(0);
13 }
```

this is the parent's process id 3937
this is child id? 3938

```
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ nano forkexample.c
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ gcc -o f2 f2.c
mubariz@mubariz-VirtualBox:~/Desktop/OS/assignments/ass02$ ./f2
```

child fork call %d\n", getpid());
process id %d\n", getppid();
n", getpid());

OS/assignments/ass02\$
OS/assignments/ass02\$ gcc -o f2 f2.c
OS/assignments/ass02\$./f2

In another terminal, issue the command: `ps aux | grep forkexample`
Notice how many processes are currently running