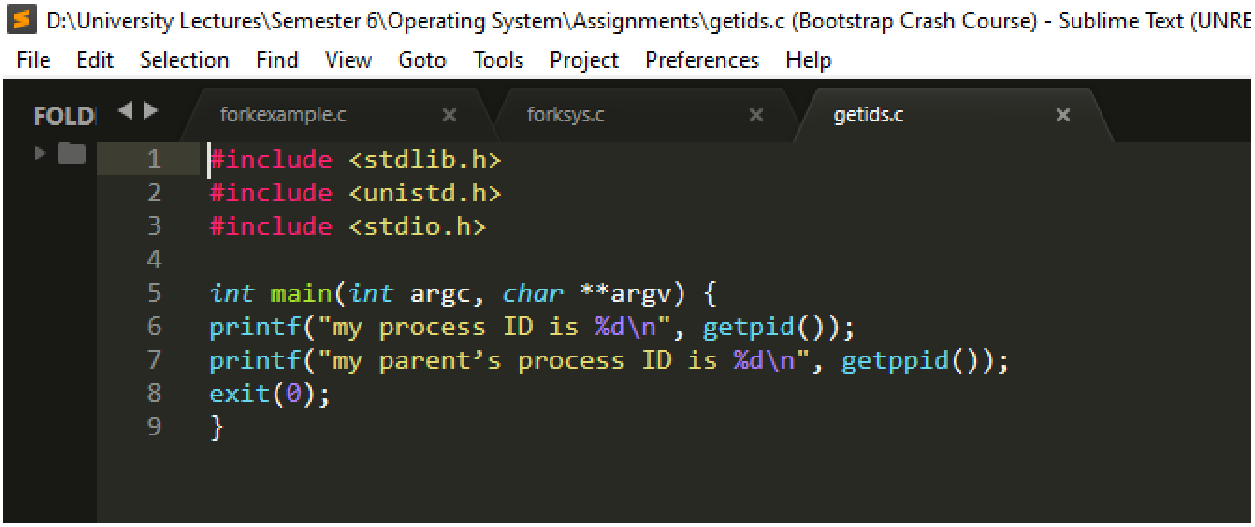
Name	**Tashfeen Latif**
Roll No	**17P-6035**
Section	**CS-A**
Course	**Operating System**
Professor	**Dr.Muhammad Nauman**

Assignment # 02

Task # 1:

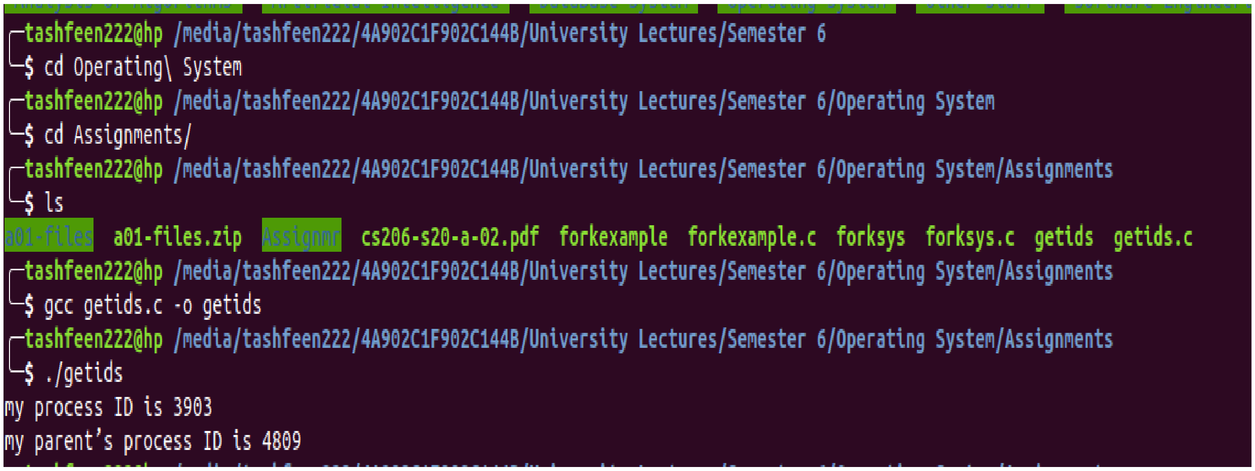
a) Write the following code in getids.c:


D:\University Lectures\Semester 6\Operating System\Assignments\getids.c (Bootstrap Crash Course) - Sublime Text (UNRE
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
FOLD
forkexample.c
forksys.c
getids.c

```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);
}
```

(b) Complete the code above by replacing the contents of square brackets. The getpid() function from unistd.h returns the process’s PID and getppid() returns that of the parent.

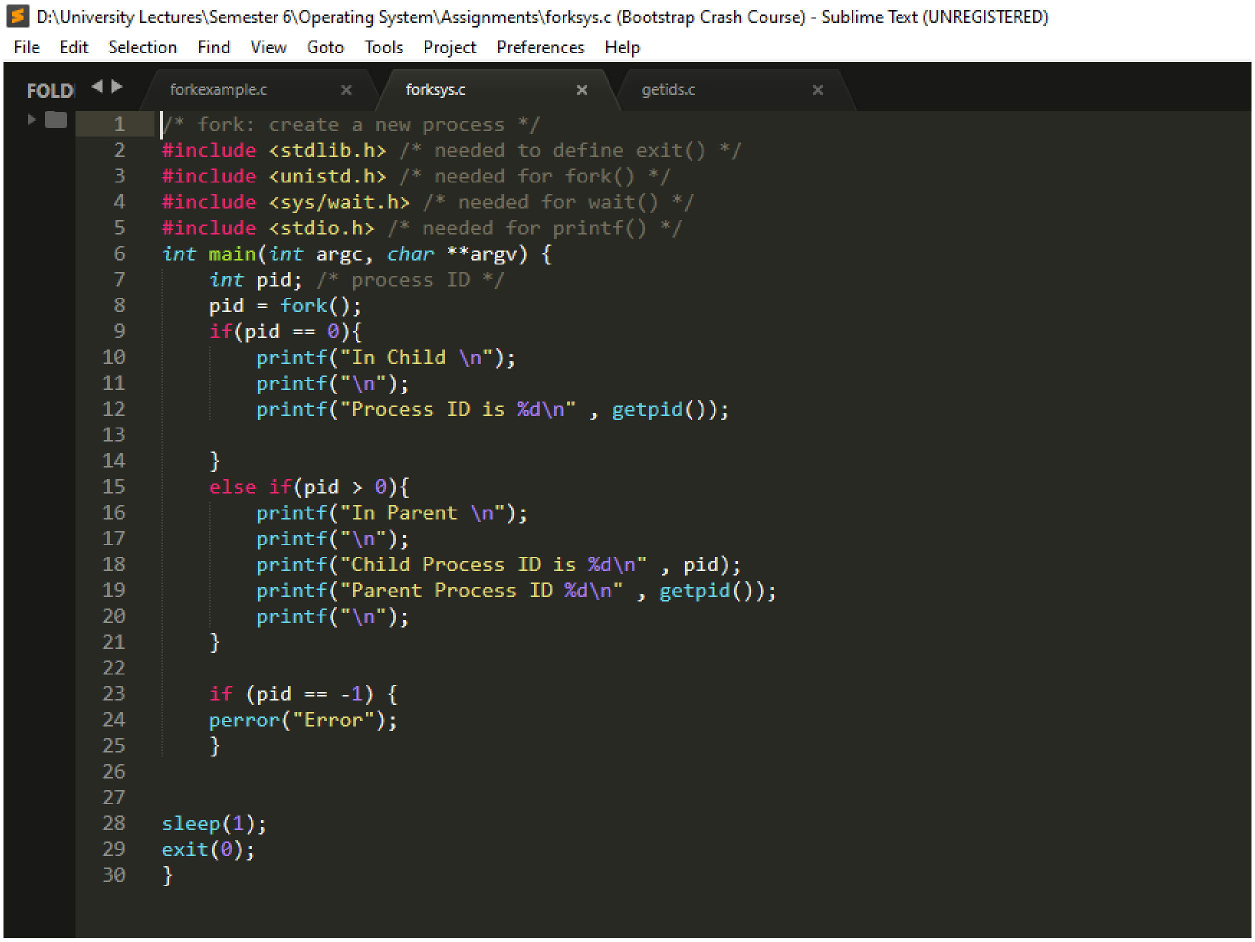
(c) Compile the program using the command:
gcc -o getids getids.c



```
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6
└─$ cd Operating\ System
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System
└─$ cd Assignments/
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ ls
a01-files  a01-files.zip  Assignmr  cs206-s20-a-02.pdf  forkexample  forkexample.c  forksys  forksys.c  getids  getids.c
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ gcc getids.c -o getids
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ ./getids
my process ID is 3903
my parent’s process ID is 4809
```

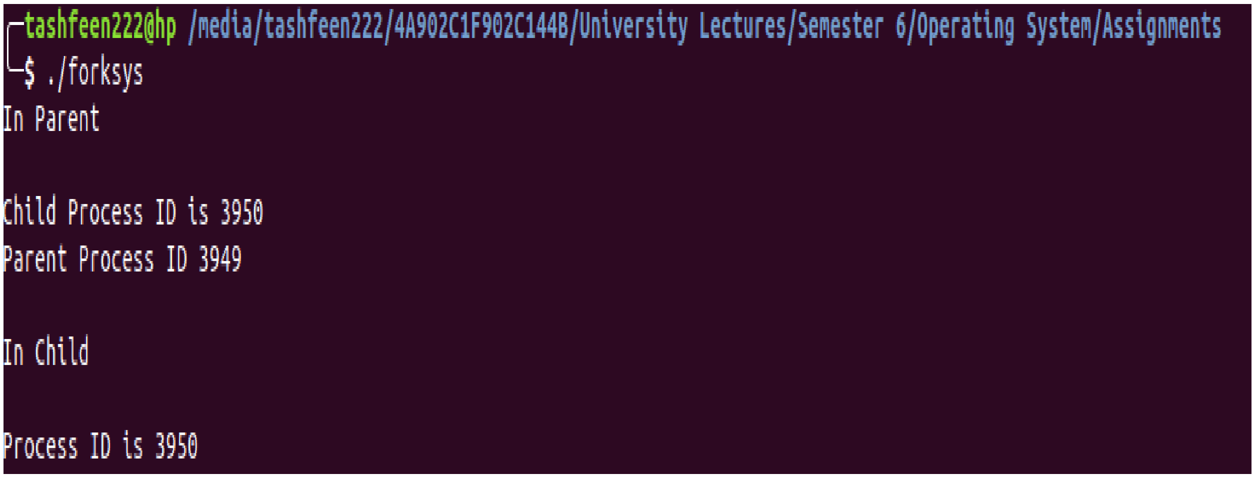
Task # 2:

(a) Write another program to issue the fork system call. Complete the following code for the provided requirements.


D:\University Lectures\Semester 6\Operating System\Assignments\forksys.c (Bootstrap Crash Course) - Sublime Text (UNREGISTERED)
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
FOLD
forkexample.c
forksys.c
getids.c

```c
/* fork: create a new process */
#include <stdlib.h> /* needed to define exit() */
#include <unistd.h> /* needed for fork() */
#include <sys/wait.h> /* needed for wait() */
#include <stdio.h> /* needed for printf() */
int main(int argc, char **argv) {
    int pid; /* process ID */
    pid = fork();
    if(pid == 0){
        printf("In Child \n");
        printf("\n");
        printf("Process ID is %d\n" , getpid());

    }
    else if(pid > 0){
        printf("In Parent \n");
        printf("\n");
        printf("Child Process ID is %d\n" , pid);
        printf("Parent Process ID %d\n" , getpid());
        printf("\n");
    }

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


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

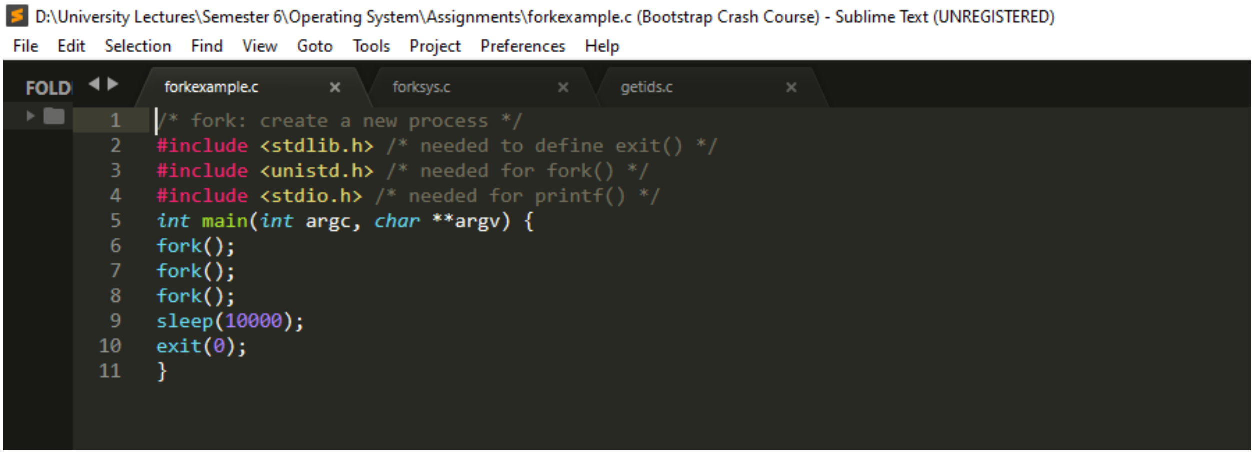
(b) Complete the two cases (parent/child) in the above code. One should execute only in child process and should print "In child" and the child's process ID. The second should run only if we are in parent and print both the parent's and the child's PID.



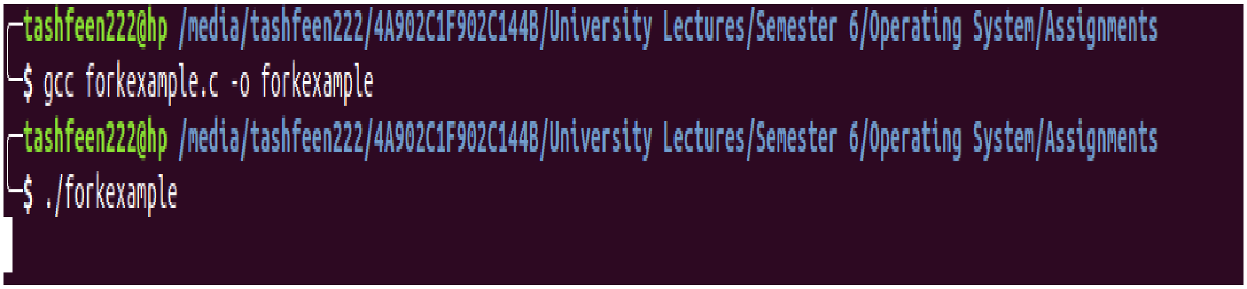
```
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ ./forksys
In Parent

Child Process ID is 3950
Parent Process ID 3949

In Child

Process ID is 3950
```

Task # 3:

Compile and run the following program in forkexample.c:

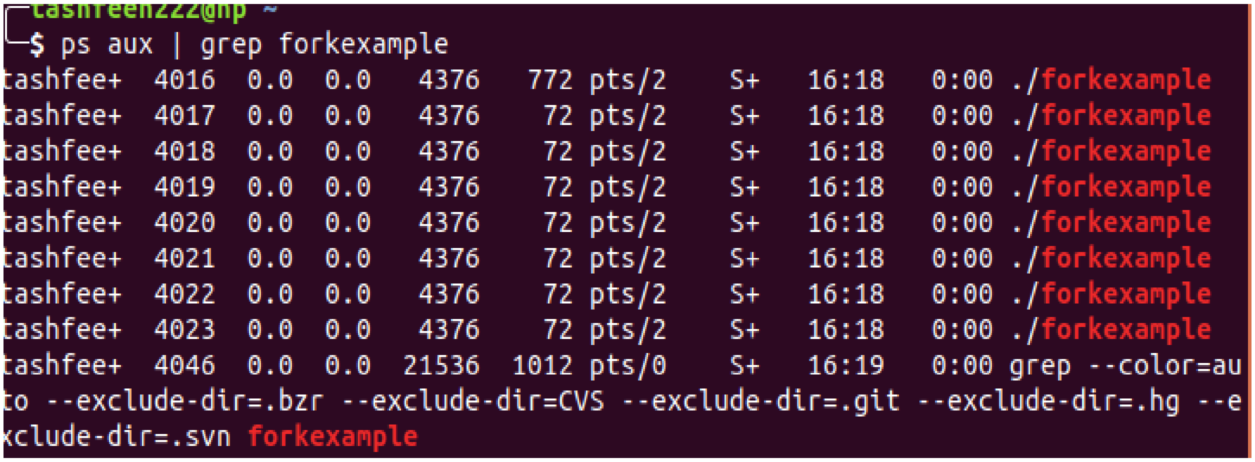

D:\University Lectures\Semester 6\Operating System\Assignments\forkexample.c (Bootstrap Crash Course) - Sublime Text (UNREGISTERED)
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
FOLD
forkexample.c
forksys.c
getids.c

```c
/* fork: create a new process */
#include <stdlib.h> /* needed to define exit() */
#include <unistd.h> /* needed for fork() */
#include <stdio.h> /* needed for printf() */
int main(int argc, char **argv) {
fork();
fork();
fork();
sleep(10000);
exit(0);
}
```



```
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ gcc forkexample.c -o forkexample
┌─tashfeen222@hp /media/tashfeen222/4A902C1F902C144B/University Lectures/Semester 6/Operating System/Assignments
└─$ ./forkexample
```

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



```
┌─tashfeen222@hp ~
└─$ ps aux | grep forkexample
tashfee+  4016  0.0  0.0   4376   772 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4017  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4018  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4019  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4020  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4021  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4022  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4023  0.0  0.0   4376    72 pts/2    S+   16:18   0:00 ./forkexample
tashfee+  4046  0.0  0.0  21536  1012 pts/0    S+   16:19   0:00 grep --color=au
to --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg --e
xclude-dir=.svn forkexample
```