NAME: AHMED ALI
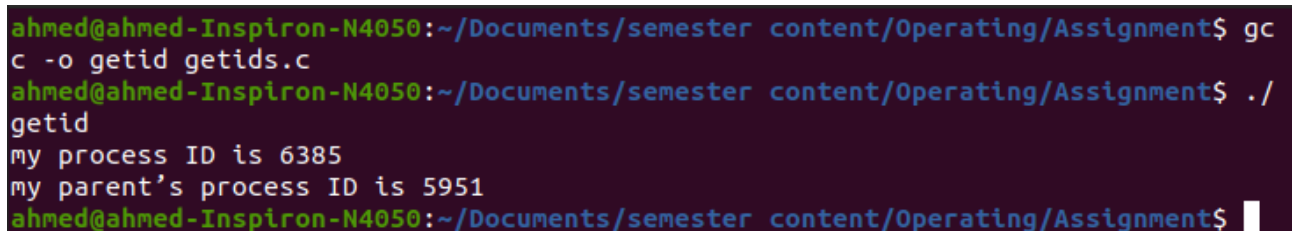# ROLL-NO: P19-0042
SECTION: 4B-CS

Answer1:

```
/* getppid: print a child's and its parent's process ID numbers */
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
int main(int argc, char **argv) {
        int p_id,P_pid;
        p_id=getpid();
        P_pid=getppid();
        printf("my process ID is %d\n", p_id);
        printf("my parent's process ID is %d\n", P_pid);
        exit(0);
}
```

SCREENSHOTS:

ANSWER 2:

```c
#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("Child Process %d",getpid());
        }
        else
        {
                printf("IN Parent\n" );
                printf("Parent Id  %d\n",getppid());
                printf("Child id  %d\n",getpid());
        }
        if (pid == -1) {
                perror("Error");
        }
        sleep(1);
        exit(0);
}
```
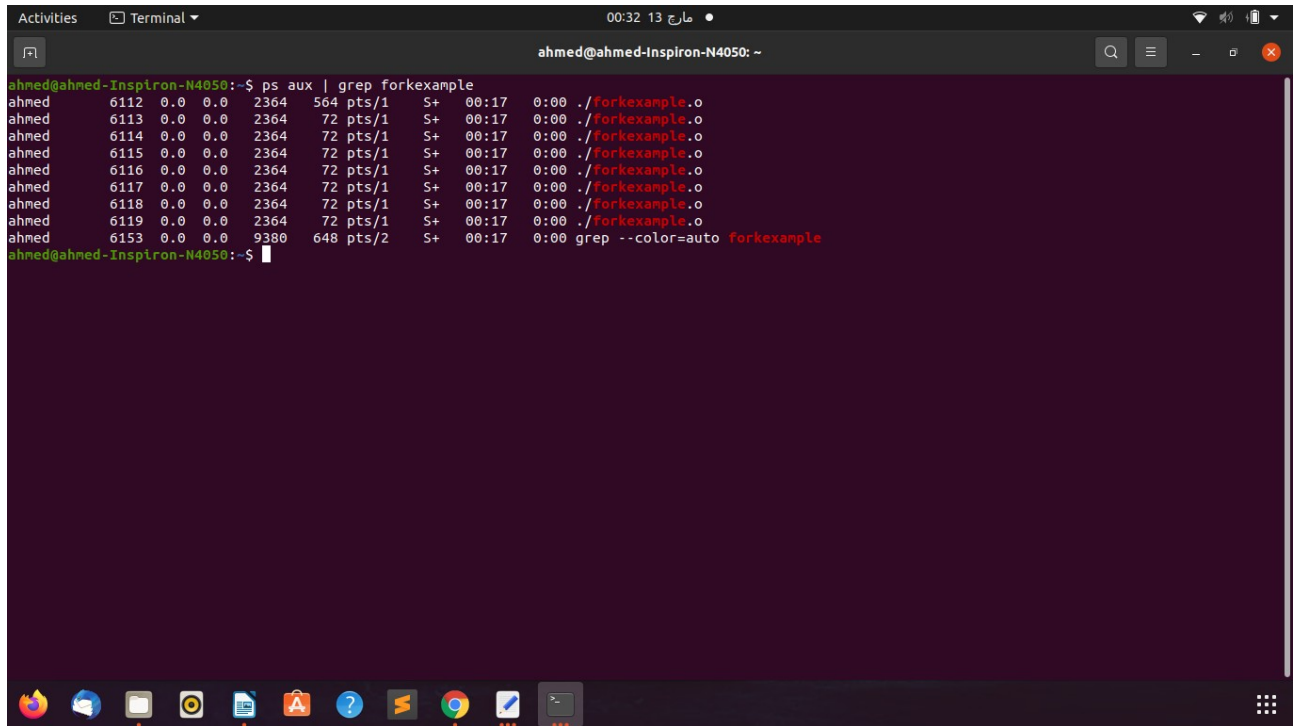
SCREENSHOT 2:


```
ahmed@ahmed-Inspiron-N4050:~/Documents/semester content/Operating/Assignment$ ./
fork.o
IN Parent
Parent Id  5951
Child id  6542
IN CHILD
Child Process 6543ahmed@ahmed-Inspiron-N4050:~/Documents/semester content/Operat
ing/Assignment$
```

ANSWER 3:

```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);
    }
```

SCREENSHOT3: