

# Chapter 3: Processes

Muhammad Shafeen  
22P-9278

## Contents

<b>1</b>	<b>3.1.1 Exercise</b>	<b>1</b>
1.1	Screenshot of Exercise Code 1 . . . . .	2
1.2	Tree of Code 1 . . . . .	3
1.3	Screenshot of Exercise Code 2 . . . . .	4
1.4	Tree of Code 2 . . . . .	5
1.5	Q1 Answer . . . . .	6
1.6	loop from i <sub>j</sub> 1 to i <sub>j</sub> 2 . . . . .	6
1.7	to i <sub>j</sub> 3 . . . . .	7
1.8	i <sub>j</sub> 100 . . . . .	8
<b>2</b>	<b>Fork Example</b>	<b>8</b>
2.1	Q5 Answer . . . . .	9
2.2	Q1 Answer . . . . .	9
2.3	Q2 Answer . . . . .	9
2.4	Q3 Answer . . . . .	10
2.5	Q4 Answer . . . . .	10
<b>3</b>	<b>3.2.1.1 Exercise 1</b>	<b>10</b>

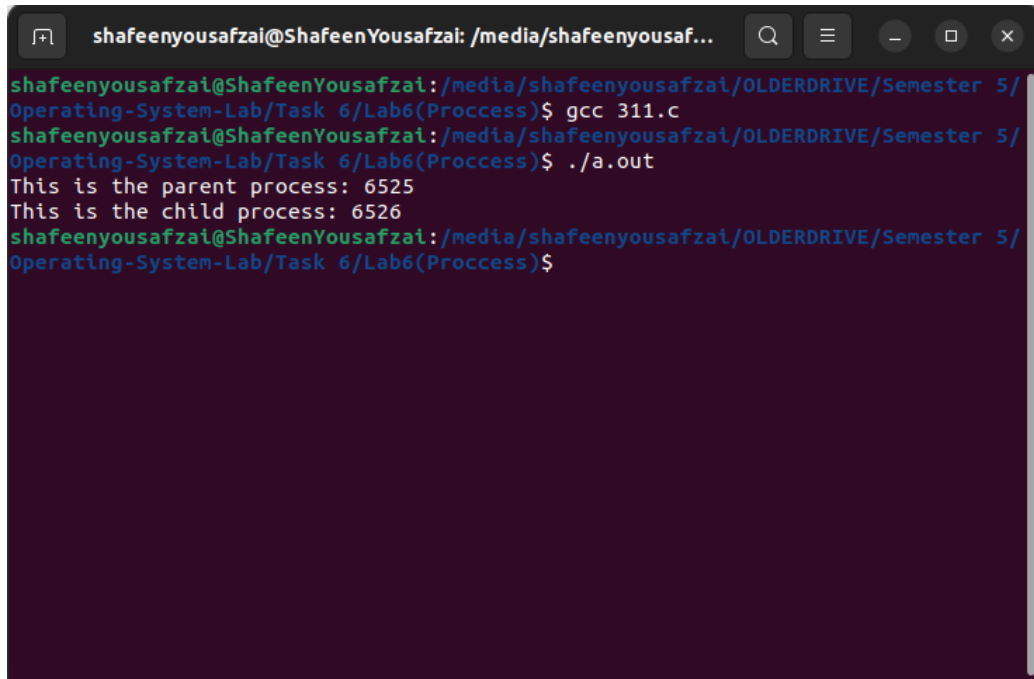
## 1 3.1.1 Exercise

Write a code that finds the following:

- PID value for `myfirst.c`
- PPID value for `myfirst.c`

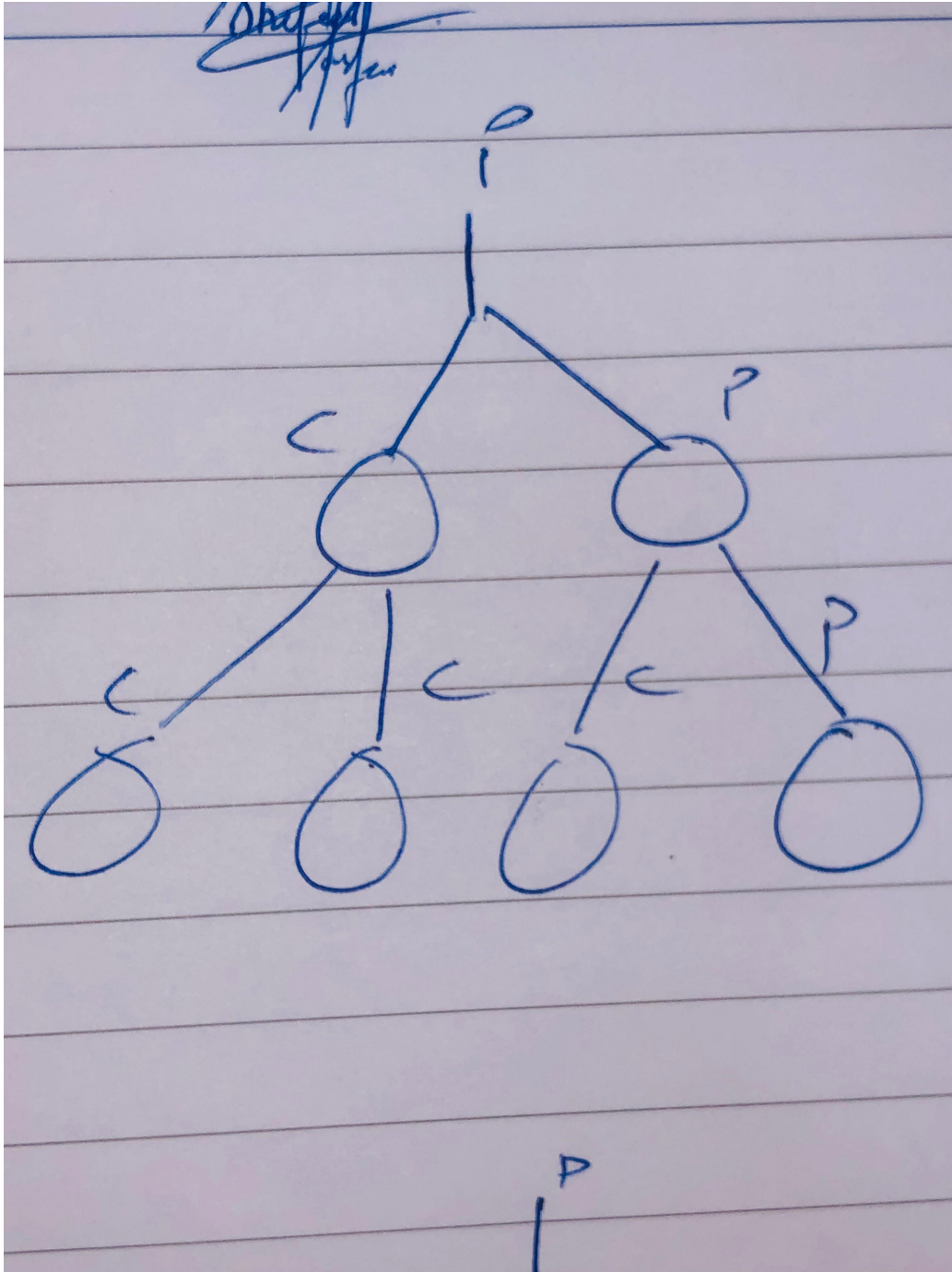
- Process name from the PPID value (using `pstree -p | grep <PPID>`).

## 1.1 Screenshot of Exercise Code 1

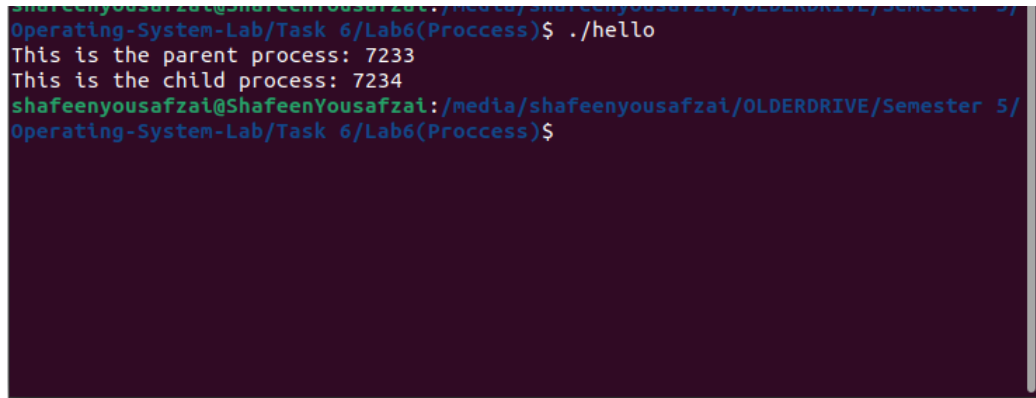


```
shafeenyousafzai@ShafeenYousafzai: /media/shafeenyousaf...  
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/  
Operating-System-Lab/Task 6/Lab6(Proccess)$ gcc 311.c  
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/  
Operating-System-Lab/Task 6/Lab6(Proccess)$ ./a.out  
This is the parent process: 6525  
This is the child process: 6526  
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/  
Operating-System-Lab/Task 6/Lab6(Proccess)$
```

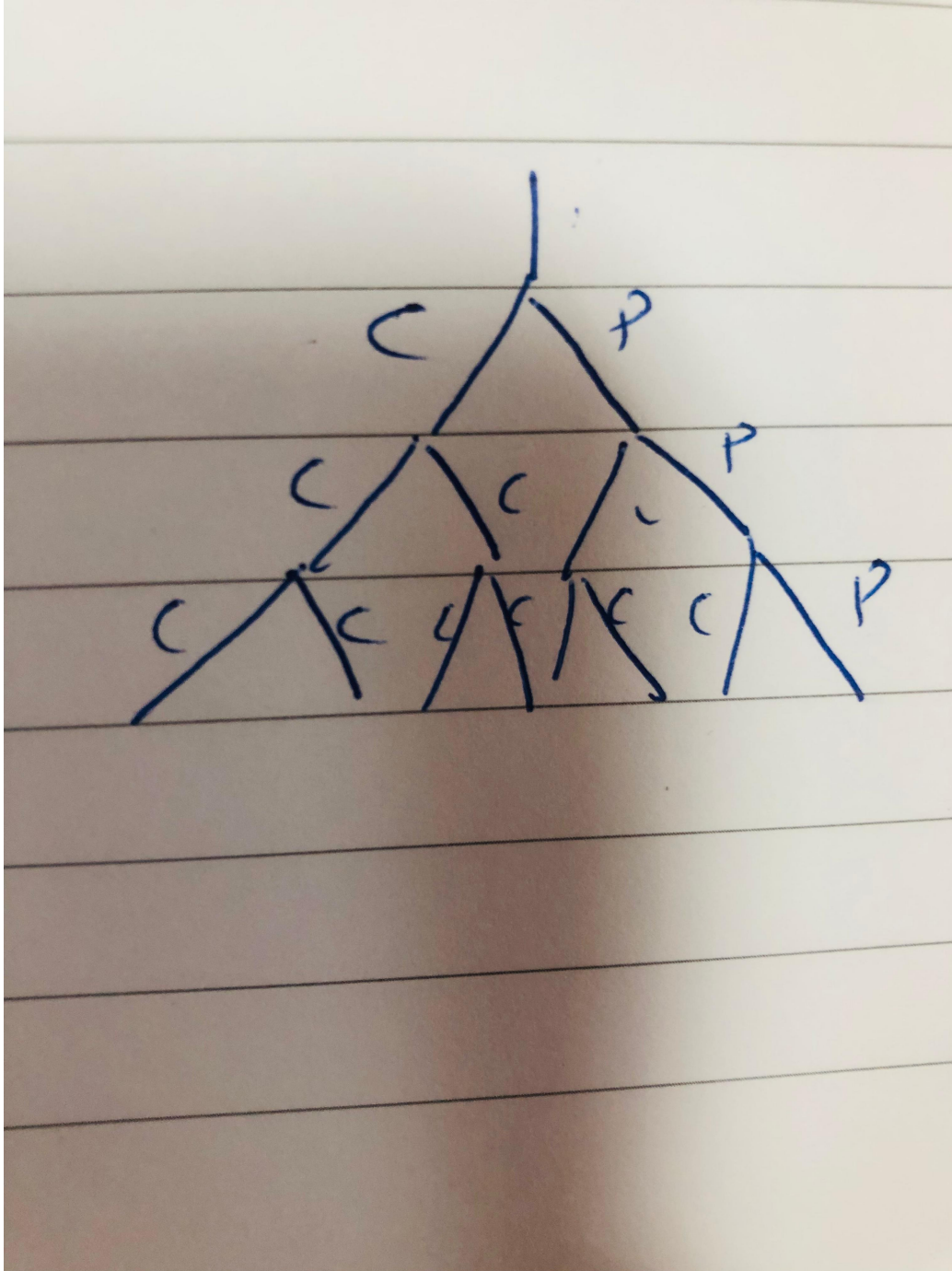
## 1.2 Tree of Code 1



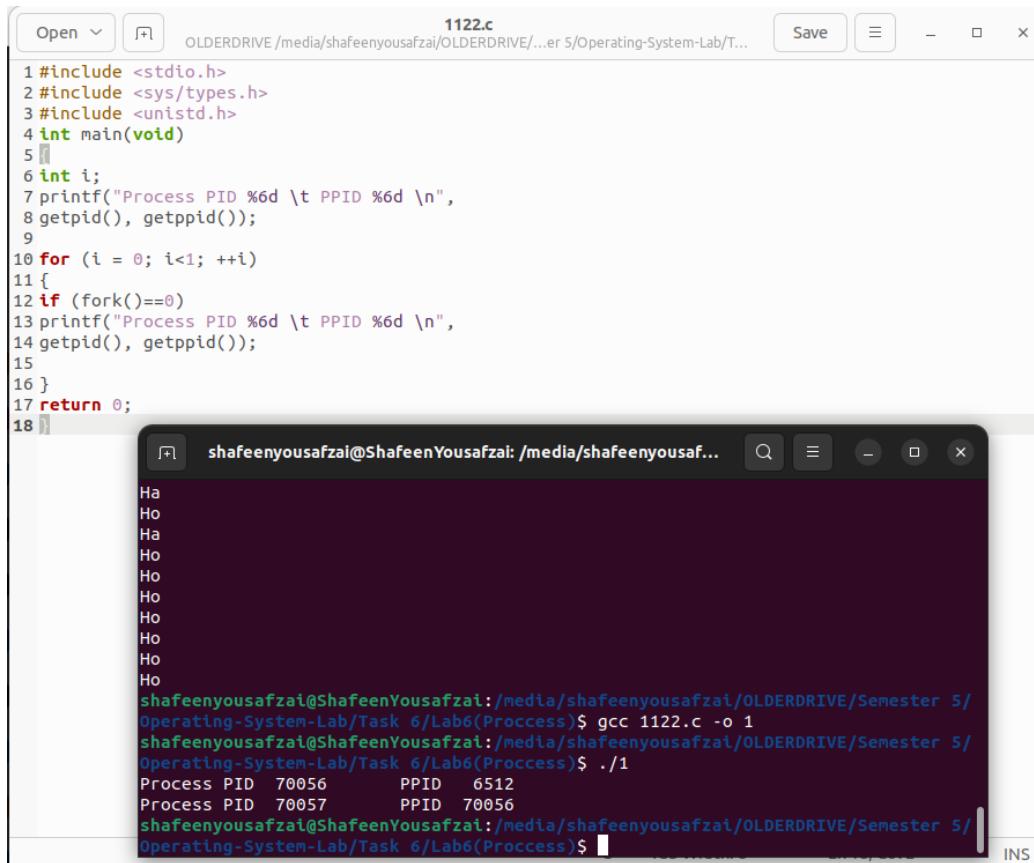
### 1.3 Screenshot of Exercise Code 2

A screenshot of a terminal window with a dark purple background and light blue text. The terminal shows the execution of a program named 'hello'. The prompt is 'shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/Operating-System-Lab/Task 6/Lab6(Proccess)\$'. The user enters './hello'. The output is 'This is the parent process: 7233' followed by 'This is the child process: 7234'. The prompt then changes to 'shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/Operating-System-Lab/Task 6/Lab6(Proccess)\$'.

#### 1.4 Tree of Code 2



## 1.5 Q1 Answer



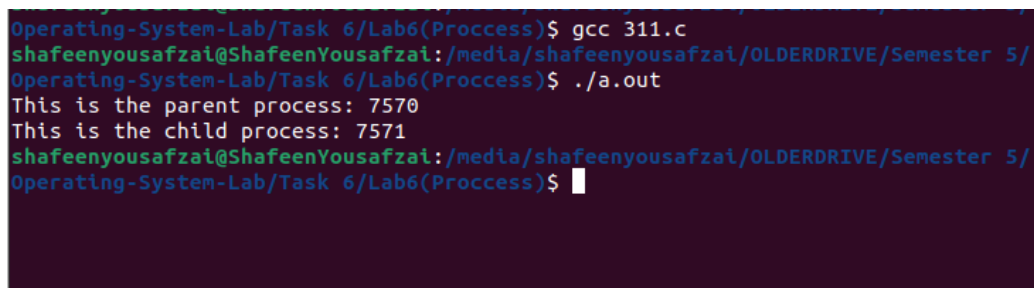
The screenshot shows a code editor window titled "1122.c" with the following C code:

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main(void)
5 {
6     int i;
7     printf("Process PID %d \t PPID %d \n",
8           getpid(), getppid());
9
10    for (i = 0; i<1; ++i)
11    {
12        if (fork()==0)
13        printf("Process PID %d \t PPID %d \n",
14              getpid(), getppid());
15    }
16 }
17 return 0;
18 }
```

Below the code editor is a terminal window showing the execution of the program. The terminal output is as follows:

```
shafeenyousafzal@ShafeenYousafzal: /media/shafeenyousaf...
Ha
Ho
Ha
Ho
Ho
Ho
Ho
Ho
Ho
Ho
Ho
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Process)$ gcc 1122.c -o 1
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Process)$ ./1
Process PID 70056      PPID  6512
Process PID 70057      PPID  70056
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Process)$
```

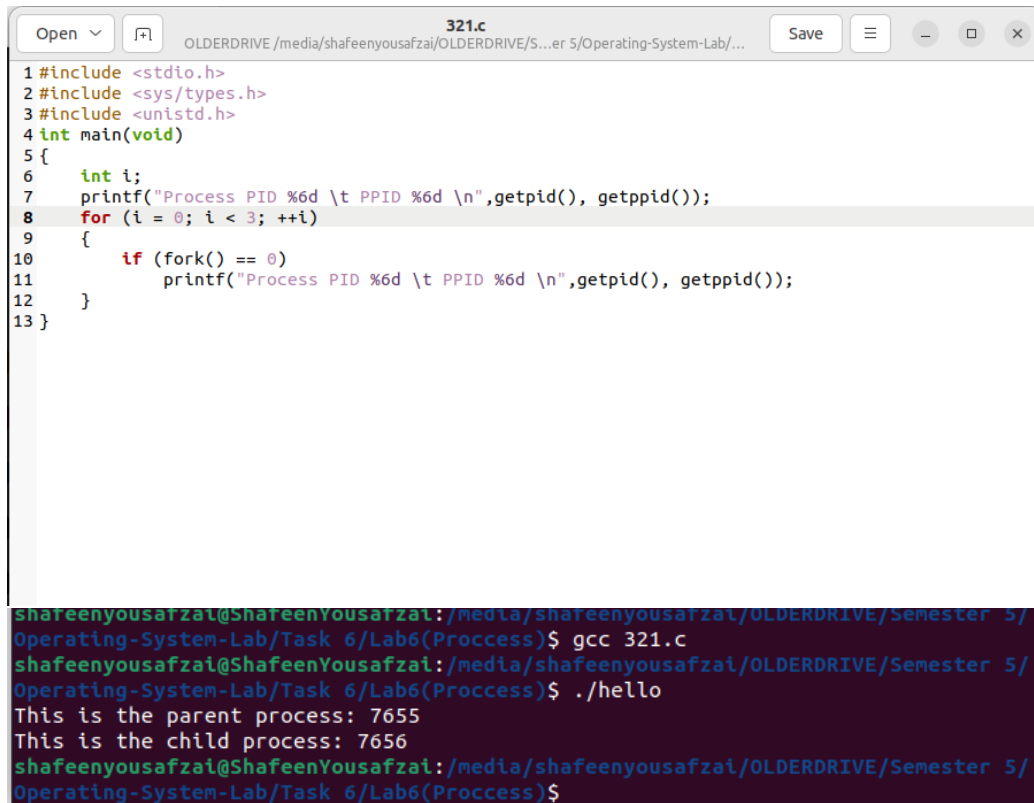
## 1.6 loop from i;1 to i;2



The screenshot shows a terminal window with the following output:

```
Operating-System-Lab/Task 6/Lab6(Process)$ gcc 311.c
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Process)$ ./a.out
This is the parent process: 7570
This is the child process: 7571
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Process)$
```

## 1.7 to i;3



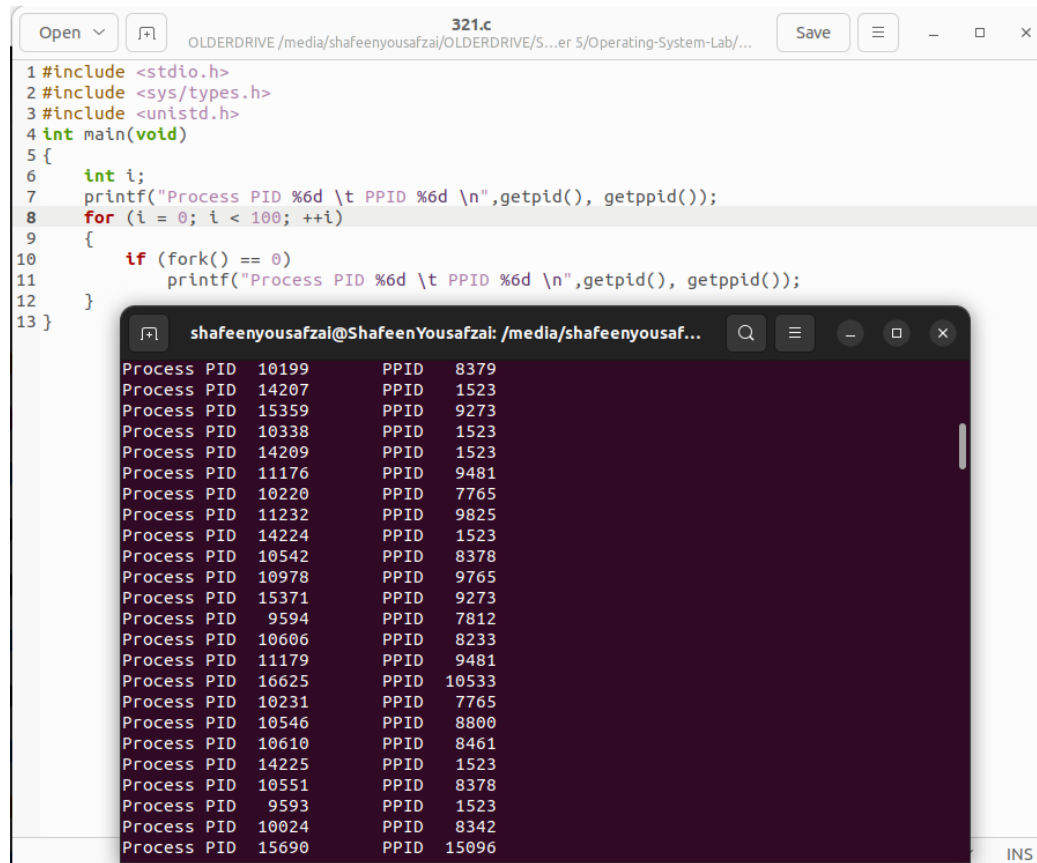
The image shows a code editor window titled "321.c" with the following C code:

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main(void)
5 {
6     int i;
7     printf("Process PID %d \t PPID %d \n", getpid(), getppid());
8     for (i = 0; i < 3; ++i)
9     {
10         if (fork() == 0)
11             printf("Process PID %d \t PPID %d \n", getpid(), getppid());
12     }
13 }
```

Below the code editor is a terminal window showing the execution of the program:

```
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ gcc 321.c
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ ./hello
This is the parent process: 7655
This is the child process: 7656
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$
```

## 1.8 i;100



The screenshot shows a code editor window titled '321.c' with the following C code:

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main(void)
5 {
6     int i;
7     printf("Process PID %d \t PPID %d \n", getpid(), getppid());
8     for (i = 0; i < 100; ++i)
9     {
10        if (fork() == 0)
11            printf("Process PID %d \t PPID %d \n", getpid(), getppid());
12    }
13 }
```

Below the code editor is a terminal window showing the output of the program. The output consists of 101 lines, each displaying the Process PID and PPID of a process. The first line shows the parent process (PID 10199, PPID 8379). The subsequent lines show the children of the parent process, with their PIDs and PPIDs. The output is as follows:

Process PID	PPID
10199	8379
14207	1523
15359	9273
10338	1523
14209	1523
11176	9481
10220	7765
11232	9825
14224	1523
10542	8378
10978	9765
15371	9273
9594	7812
10606	8233
11179	9481
16625	10533
10231	7765
10546	8800
10610	8461
14225	1523
10551	8378
9593	1523
10024	8342
15690	15096

## 2 Fork Example

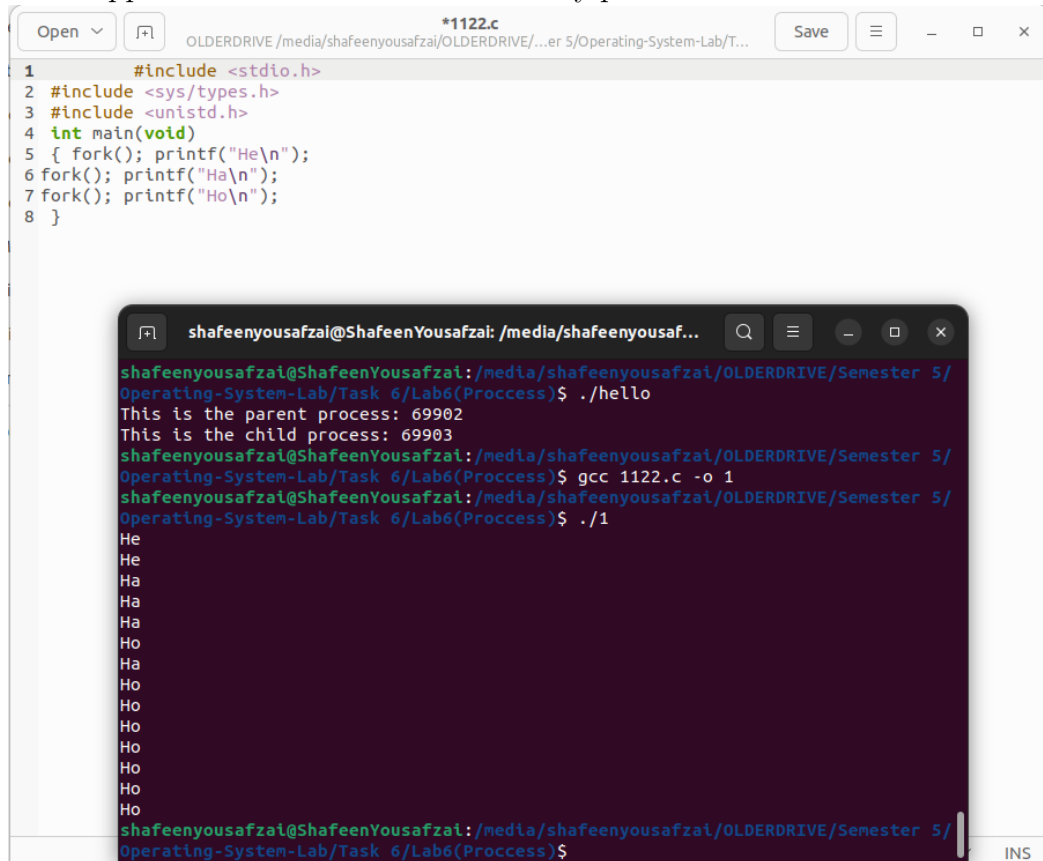
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(void) {
    fork(); printf("He\n");
    fork(); printf("Ha\n");
    fork(); printf("Ho\n");
}
```



## 2.1 Q5 Answer

Yes, a "Ho" can appear before a "He" due to the way processes execute con-



The image shows a code editor window with a C program named `*1122.c`. The program uses `fork()` to create two child processes. The first child prints "He" and the second child prints "Ho". The terminal output shows the execution of the program, where the parent process prints "This is the parent process: 69902" and "This is the child process: 69903". The child processes then print "He" and "Ho" respectively. The output shows that the child processes can execute out of order, resulting in "Ho" appearing before "He".

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main(void)
5 { fork(); printf("He\n");
6 fork(); printf("Ha\n");
7 fork(); printf("Ho\n");
8 }
```

```
shafeenyousafzai@ShafeenYousafzai: /media/shafeenyousaf...
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ ./hello
This is the parent process: 69902
This is the child process: 69903
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ gcc 1122.c -o 1
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ ./1
He
He
Ha
Ha
Ha
Ho
Ha
Ho
Ho
Ho
Ho
Ho
Ho
Ho
Ho
shafeenyousafzai@ShafeenYousafzai:/media/shafeenyousafzai/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$
```

currently.

## 2.2 Q1 Answer

We use `p = fork()` to distinguish between the parent (non-zero) and child (0) processes.

## 2.3 Q2 Answer

The `printf()` function is part of the `stdio.h` library.

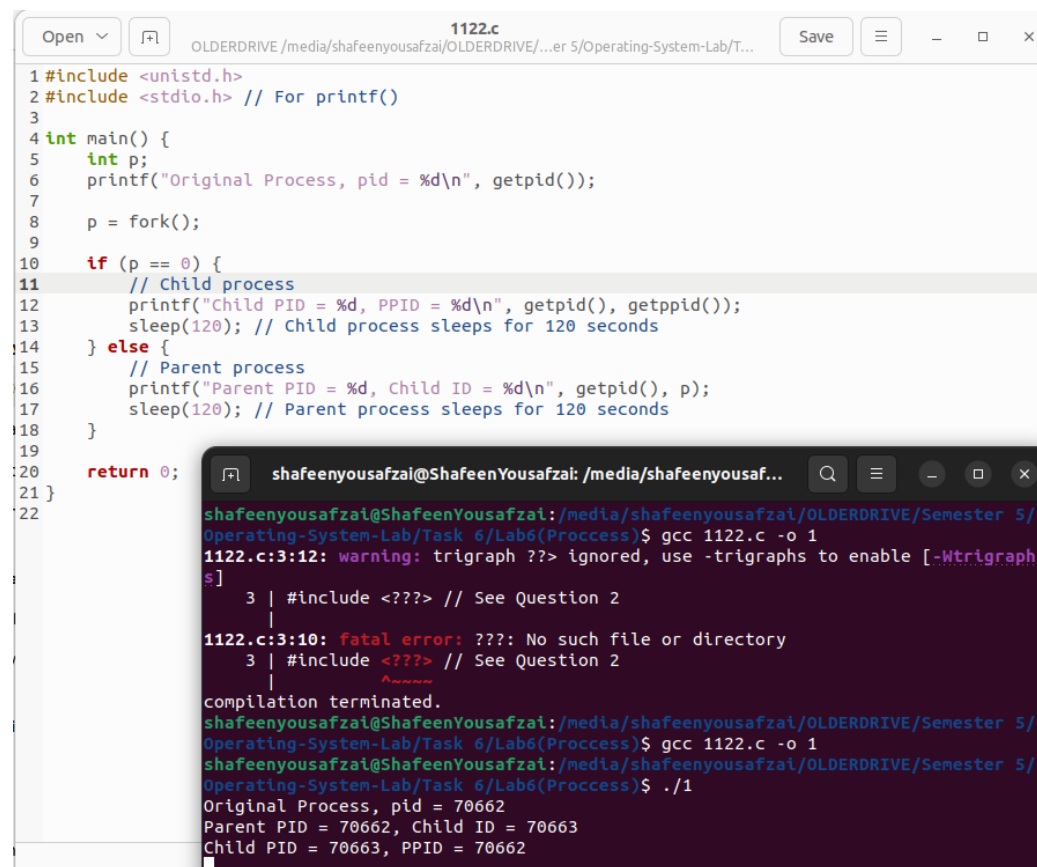
## 2.4 Q3 Answer

"Job Done" is printed twice because both the parent and child processes execute the printf() after the fork().

## 2.5 Q4 Answer

The output will show the child PID in the parent process and 0 in the child process.

## 3 3.2.1.1 Exercise 1



The image shows a code editor window titled "1122.c" with the following C code:

```
1 #include <unistd.h>
2 #include <stdio.h> // For printf()
3
4 int main() {
5     int p;
6     printf("Original Process, pid = %d\n", getpid());
7
8     p = fork();
9
10    if (p == 0) {
11        // Child process
12        printf("Child PID = %d, PPID = %d\n", getpid(), getppid());
13        sleep(120); // Child process sleeps for 120 seconds
14    } else {
15        // Parent process
16        printf("Parent PID = %d, Child ID = %d\n", getpid(), p);
17        sleep(120); // Parent process sleeps for 120 seconds
18    }
19
20    return 0;
21 }
22
```

Below the code editor is a terminal window showing the compilation and execution of the program:

```
shafeenyousafzal@ShafeenYousafzal: /media/shafeenyousaf...
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ gcc 1122.c -o 1
1122.c:3:12: warning: trigraph ??? ignored, use -trigraphs to enable [-Wtrigraph
s]
3 | #include <???> // See Question 2
  |
1122.c:3:10: fatal error: ??? : No such file or directory
3 | #include <???> // See Question 2
  |
compilation terminated.
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ gcc 1122.c -o 1
shafeenyousafzal@ShafeenYousafzal:/media/shafeenyousafzal/OLDERDRIVE/Semester 5/
Operating-System-Lab/Task 6/Lab6(Proccess)$ ./1
Original Process, pid = 70662
Parent PID = 70662, Child ID = 70663
Child PID = 70663, PPID = 70662
```