Chapter 3: Processes

Muhammad Shafeen
22P-9278

Contents

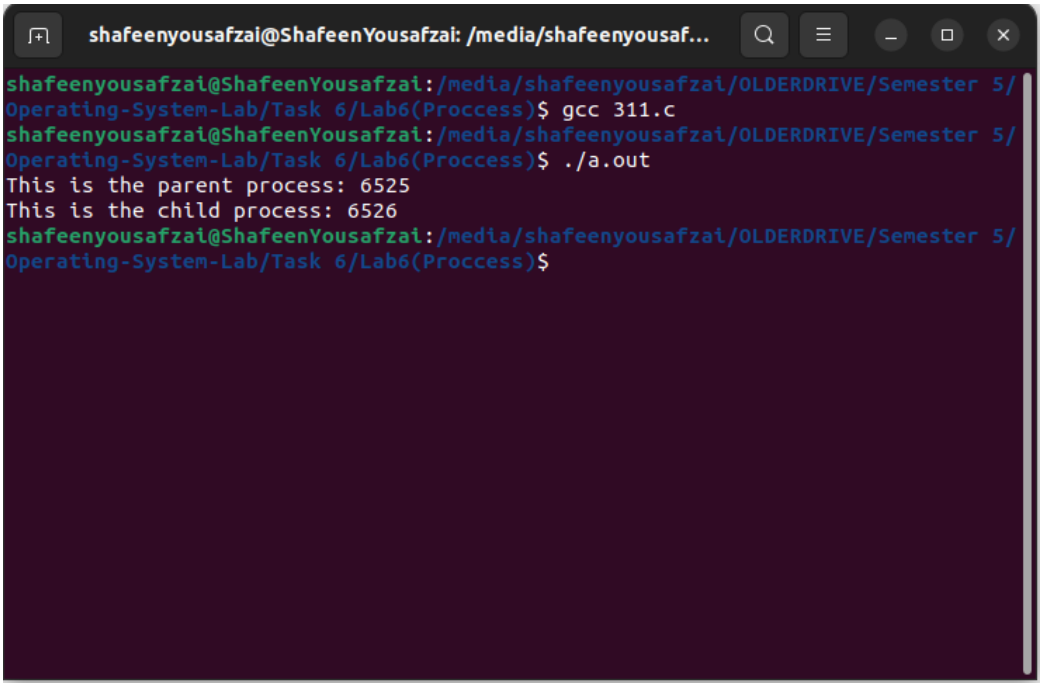
1 3.1.1 Exercise

Write a code that finds the following:

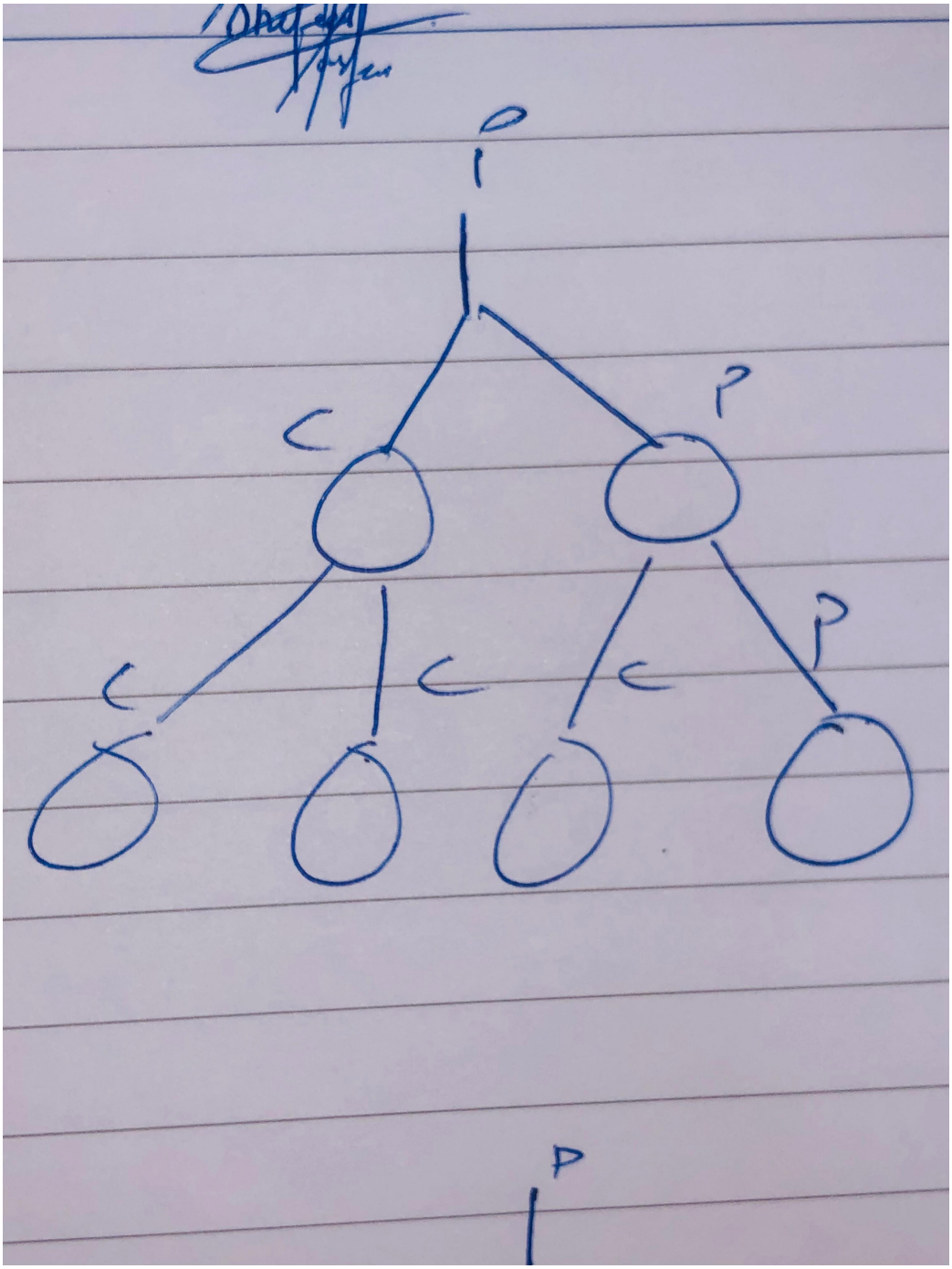
- PID value for `myfirst.c`
- PPID value for `myfirst.c`

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

1.1 Screenshot of Exercise Code 1

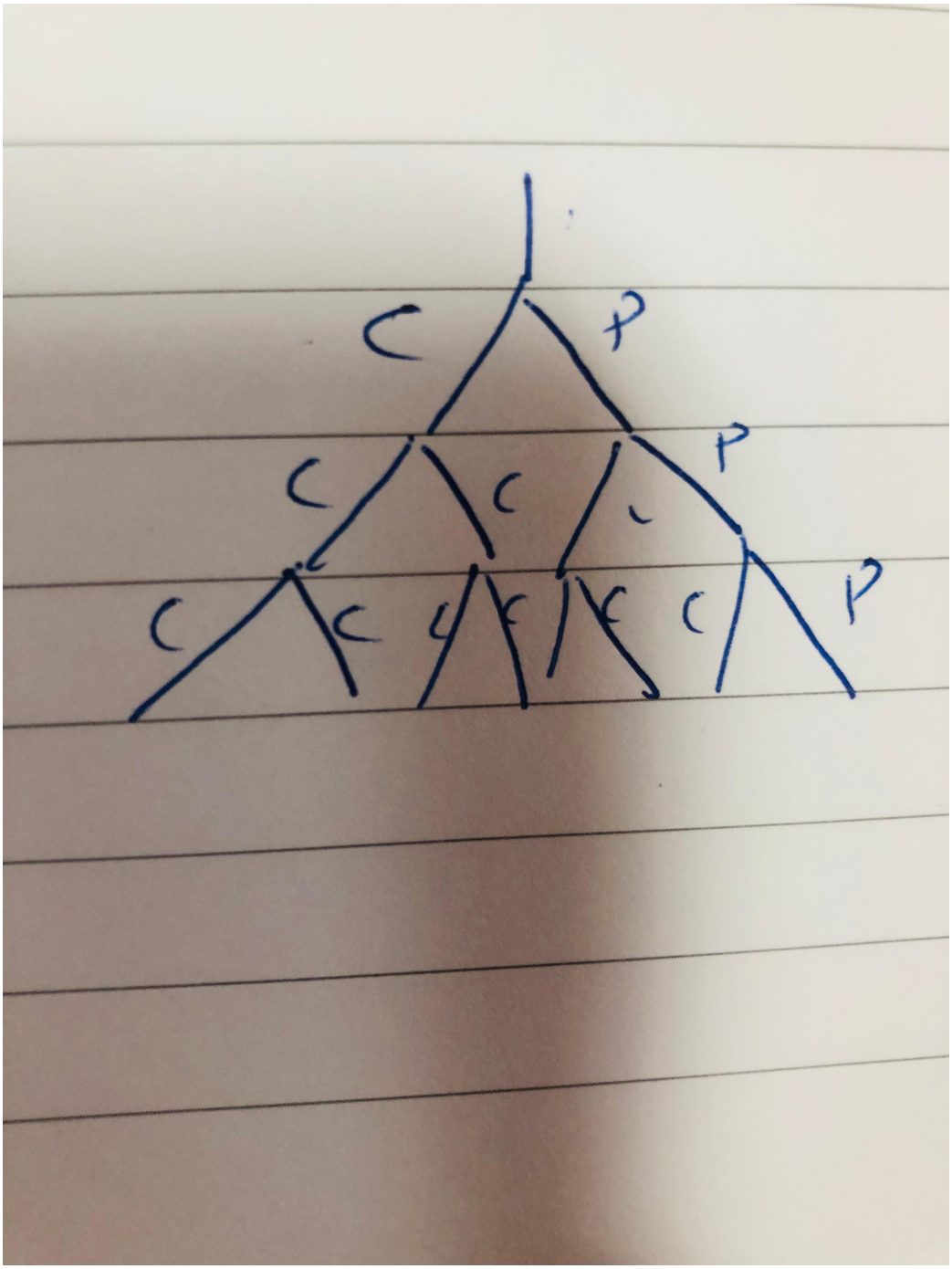


1.2 Tree of Code 1

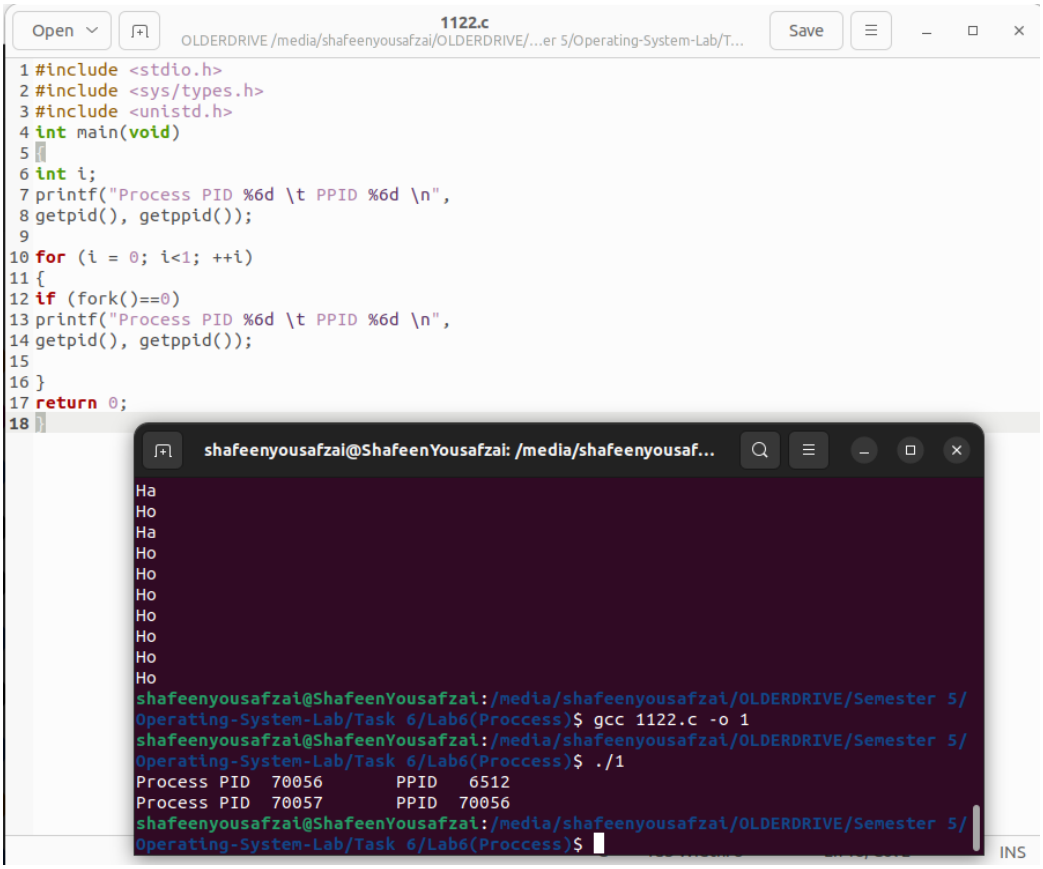


1.3 Screenshot of Exercise Code 2

1.4 Tree of Code 2



1.5 Q1 Answer



1.6 loop from i¡1 to i¡2

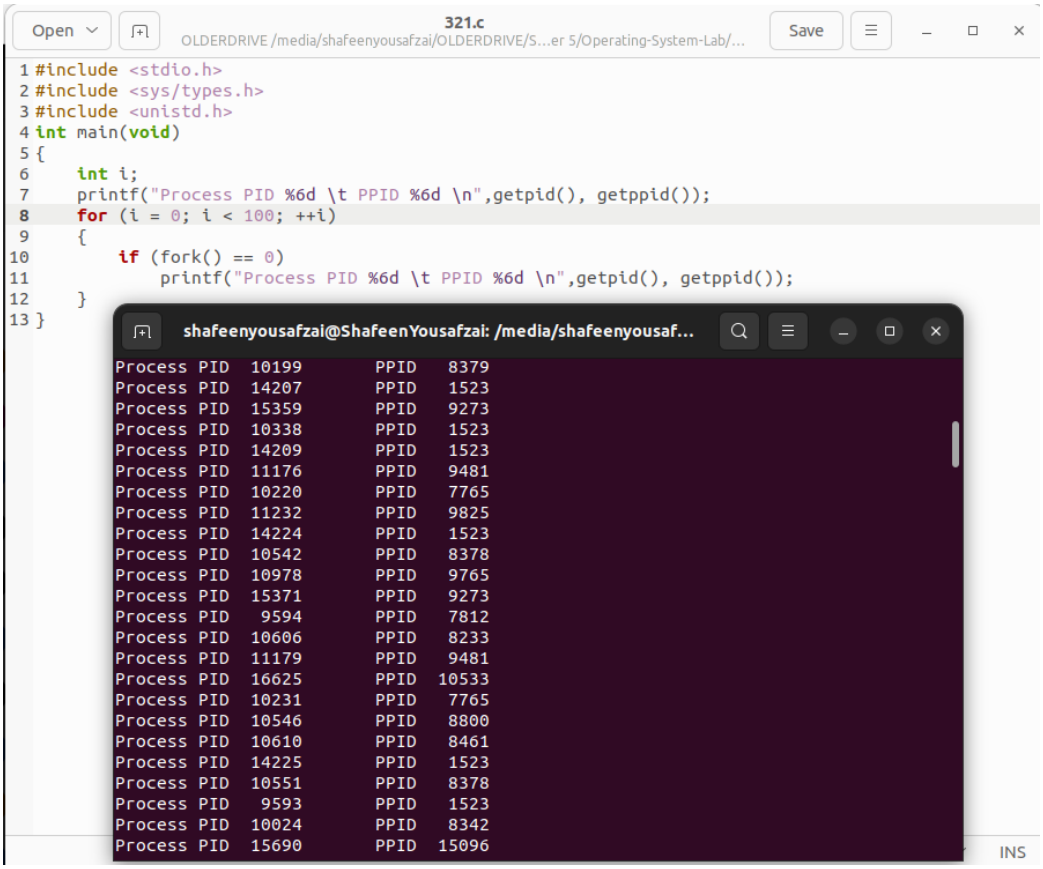
1.7 to i¡3

Open | 321.c | OLDERDRIVE /media/shafeenyousafzai/OLDERDRIVE/S...er 5/Operating-System-Lab/... | Save

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    int i;
    printf("Process PID %6d \t PPID %6d \n",getpid(), getppid());
    for (i = 0; i < 3; ++i)
    {
        if (fork() == 0)
            printf("Process PID %6d \t PPID %6d \n",getpid(), getppid());
    }
}
```

```
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



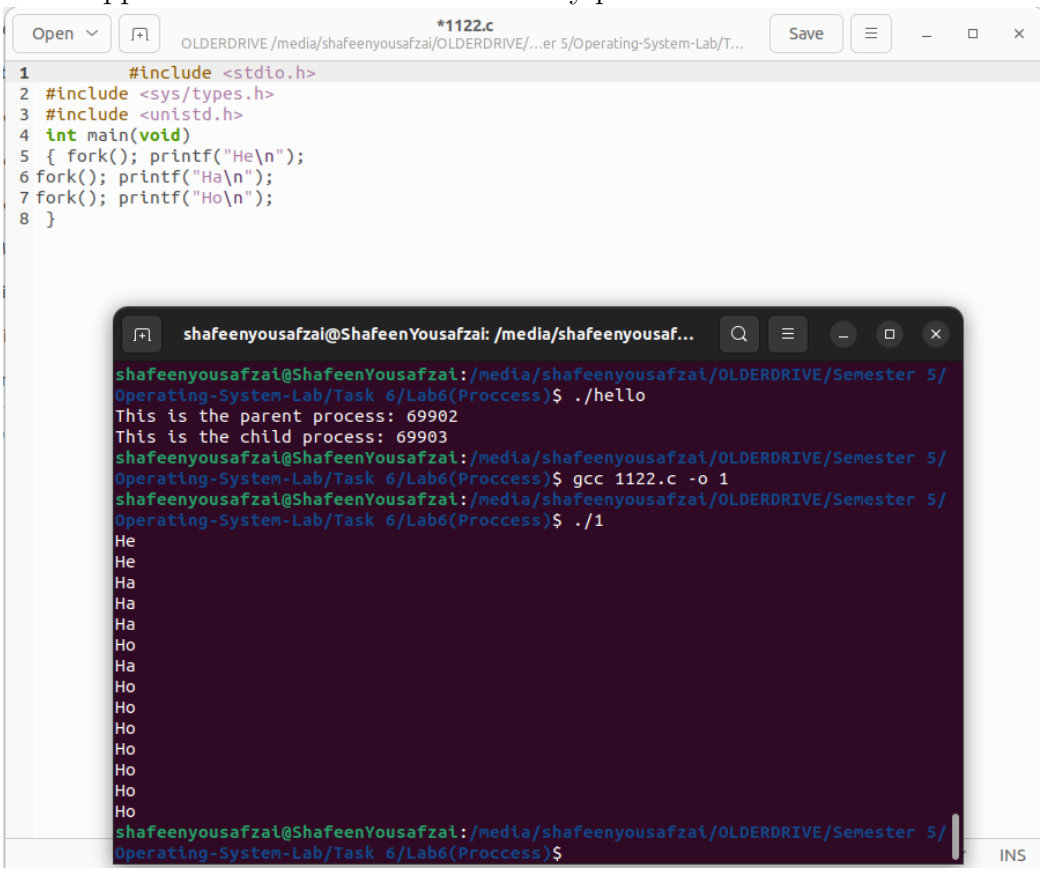
2 Fork Example

```
#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-



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