ISE302 – Operating Systems
Fall 2021
Assignment-1 Report

Aral ŞEN
150170217

1 (30 points) Please investigate the given code below. Compile and run the program, and answer the following questions accordingly.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    int returnValue = 0;

    int i;
    for (i = 0; i < 3; i++) {
        returnValue = fork();
        if (returnValue == -1) {
            exit(0);
        }
        else if (returnValue != 0) {
            wait(NULL);
        }

        printf("Current process id: %d\n",  getpid());
    }

    return 0;
}
```

a) (5 points) How many times will the system call *fork()* be called?

fork() will be called 7 times

b) (5 points) What will the program's output look like?

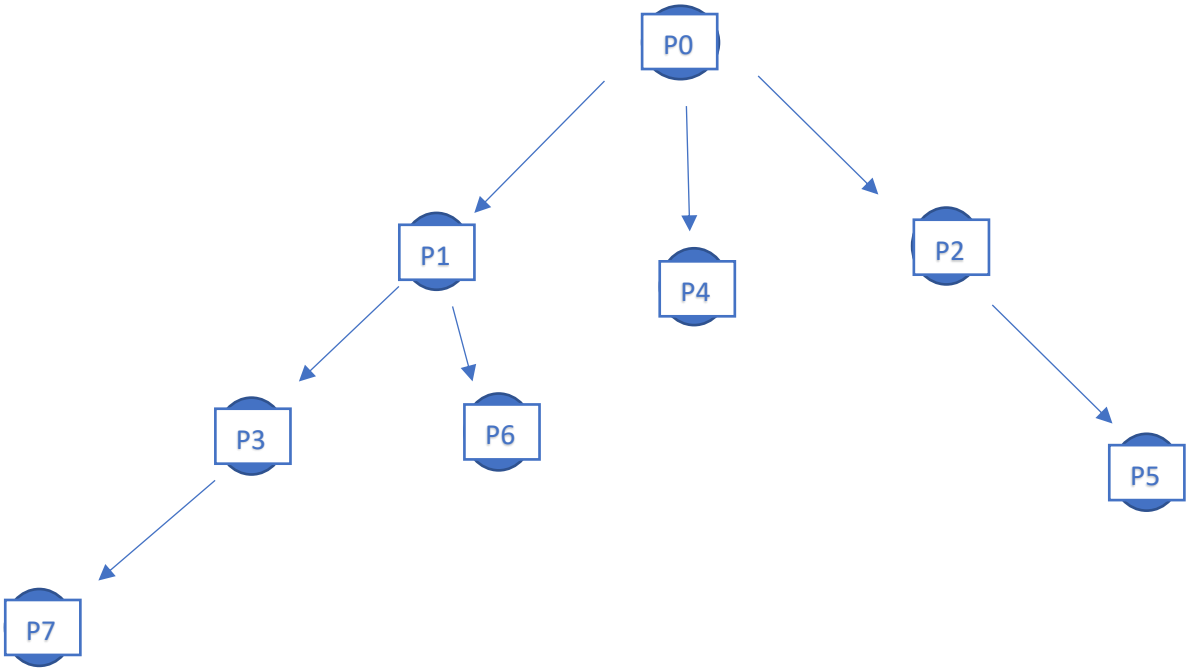
Current process id: 611
Current process id: 612
Current process id: 613
Current process id: 612
Current process id: 611
Current process id: 614
Current process id: 611
Current process id: 609
Current process id: 615
Current process id: 616
Current process id: 615
Current process id: 609
Current process id: 617
Current process id: 609

Processes id’s will be different in each program run.

c) (10 points) How many processes will the program end up with in total? How many of them can be identified as parent and as child processes?

8 processes will the program end up with in total. There will be 7 child processes, and 1 parent process.

d) (10 points) Draw a tree that represents the hierarchy of the created processes



2 (70 points) In this part of the homework, you are asked to write a program using multiple threads to determine the largest element in an integer array. In addition you need to evaluate the performance of your program in terms of time-complexity.

The following commands can be used to compile and run the program:

```
$ gcc hw1.c -o hw1 -pthread -w
$ time ./hw1
```

Thread Count (N)	The Actual Time Elapsed (CPU Seconds)	Time Spend by the user-mode (CPU Seconds)	Time Spend by the kernel (CPU Seconds)
1	0.001	0.001	0.000
10	0.018	0.011	0.006
100	0.021	0.019	0.007
1000	0.088	0.031	0.061
100000	2.068	0.461	2.038
200000	2.375	0.615	2.244

To find max element in an array, I initialize the maximum element to the first element and then traverse the array, comparing each element and update maximum whenever necessary. We know that time complexity of linear search is O(n). As we can see in the table, the more thread we create, our elapsed time increses. I think creating threads demand resources, and allocating more resources increase the elapsed time of our program.