BLG 312E - Assignment 1

Student Name: Ahmet Furkan Kavraz Student ID: 150190024

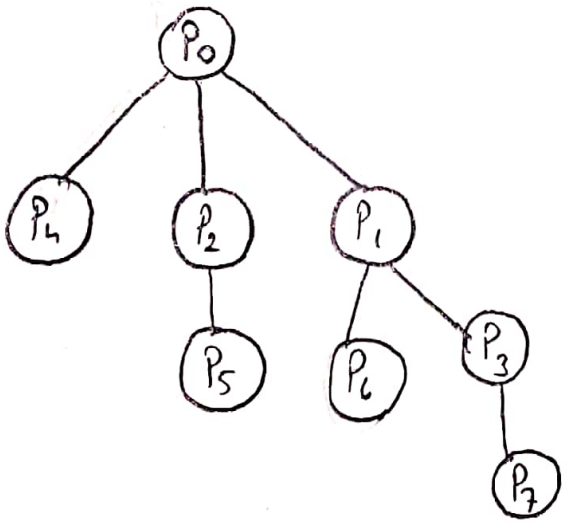
Q1)

a) 8 processes will be created by the code. We can identify like:

1 of them is only parent.
4 of them are only child.
3 of them are both parent and child.

Normally, we have only one parent process and all processes created by that process. But, as you see below, some process, which are not neither root nor leaf, created by fork and also, creates a child with fork. So, we can clustrify them both parent and child.

b)



P_0: Parent → main process (also root)
P_4, P_5, P_6, P_7 → child
P_1, P_2, P_3 → both child and parent

With first fork call; P_1 is created.
With second fork call; P_2, P_3 are created.
With third fork call; P_4, P_5, P_6, P_7 are created.

First fork call is invoked when i=1.
Second fork calls are invoked when i=2.
Third fork calls are invoked when i=3.

Student Name: Ahmet Furkan Kavraz Student ID: 150190024

c) → The edges of tree represents the fork system calls. There are 7 edges. So, there are 7 fork calls.

→ Printf invokes 14 times. There are:

3 printfs in both P_0 and P_1 ⟹ $3.2 = 6$
2 printfs in both P_2 and P_3 ⟹ $2.2 = 4$
1 printfs in all P_4, P_5, P_6, P_7 ⟹ $1.4 = 4$
$+$
14 printf

d) Output code look like:

Parent process (i=1)	x 1 times
Child process 1.	x 1 times
Parent process (i=2)	x 2 times
Child process 2.	x 2 times
Parent process (i=3)	x 4 times
Child process 3.	x 4 times

The numbers must be as inthe next. Also, we can limit that in one process printfs should invoke in order (i=1, i=2, i=3). There can be an output like i=3 output comes before i=2 but any i=3 output can not comes before all i=2 outputs.

Every process have an order in itsef. But we can not say child comes before / after parent. Since the processing times in CPU is not certain about which will process before parent / child, and in code, there is not any "wait" function; we can only say there is an order inside processes. So, it is possible to generate different outputs as when we run the code.