BLG312-ASSIGNMENT -1-

Emir Kaan Erdoğan- 150200706,

02.04.2023

1 Introduction

Within the scope of the first assignment we are asked to implement a process tree. In the first question the process tree has a depth of N right children and each right child and the root has merely **one** left child.
On the other hand, in the second question we are requested to extend the tree so that its left subtrees are parametric as well meaning that, a user can desire M left subtrees and N right subtrees.

2 Question 1

To implement the first question, I take the parameter N from the user and create a tree using recursive approach by updating the parameter N that is given by the user. In each iteration I create left child and right child by calling the fork function twice accordingly. In the function, if the height of the tree is not equal to zero, I call fork function for the first time. Then I check whether I am at the parent node or the left child. If it is at the parent node, I call the fork function for the second time. If the lastly created pid2 is at parent, I print it out. On the other hand, in case it is the **right child**, I call the recursive function by updating the height of the tree.
In the following, I give an example output of the function and the visual of the tree that is created.

```
Please enter the N value: 3
The root Node's pid: 1697
Parent Node with pid: 1697
Left Child with pid: 1698, created from the parent with pid: 1697
Rigth Child with pid: 1699 ,created from the parent with pid: 1697
Parent Node with pid: 1699
Left Child with pid: 1700, created from the parent with pid: 1699
Rigth Child with pid: 1701 ,created from the parent with pid: 1699
Left Child with pid: 1702, created from the parent with pid: 1701
Parent Node with pid: 1701
Rigth Child with pid: 1703 ,created from the parent with pid: 1701
The last Parent Node with the pid: 1703 ,and it is created from the parent with pid: 1701
Left Child Node with pid:1704, and it is created from the parent: 1703test@blg335e:~/hostvolume/cos_hw1$
```

Figure 1: Q1 OUTPUT

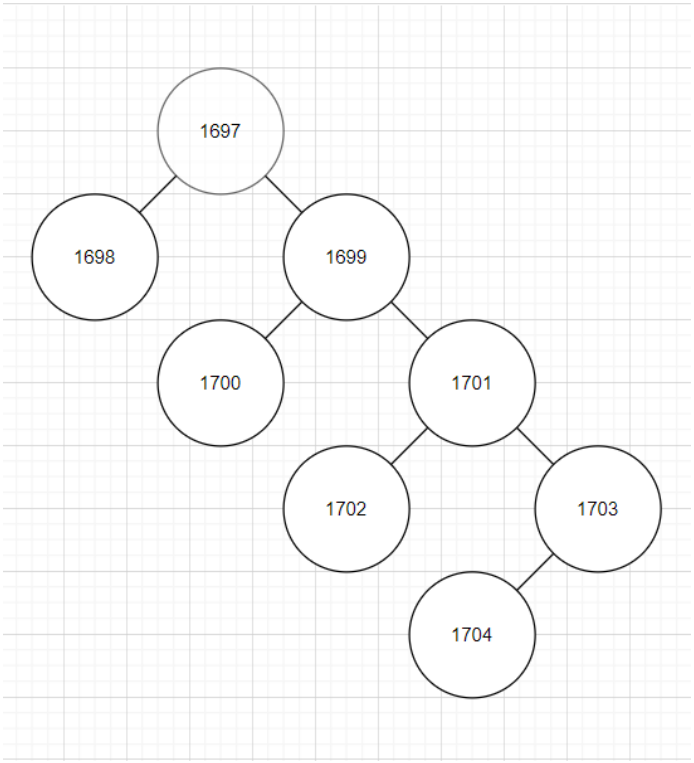

Figure 2: Q1 Tree

2.1 Q1.2

Assume f(x) is the function returning the number of parent nodes.
Base Case f(0)= 1 ; consider root as parent ;
f(1)= 3;
f(2)= 5;
f(3)= 7;

The number of parent processes in the process tree relies on N.

The base case f(0) = 1 represents the root process at depth 0. The root can be considered as a parent process.
In further depths, it is added up to the tree 2 nodes in each iteration. The nodes become parent processes at the further iteration, namely at depth d+1. Therefore, in each iteration we obtain 2 parents and since in the base case we have 1 parent we can calculate the number of parent processes as in the following:
f(x)= 2x+1 ; is the number of parent processes ;
f(x) = 2*(x+1) is the total number of processes ;

3 Question 2

In the following, I provide some outputs for example test cases. The first test case is such as in the first part, where N=3 and M=1:

```
test@blg335e:~/hostvolume/cos_hw1$ gcc -o q1 q1.c
test@blg335e:~/hostvolume/cos_hw1$ ./q1
Please enter the N value: 3
Please enter the M value:
1
The root Node's pid: 15322
Left Child with pid:15323, created from the parent with pid: 15322
Parent Node with pid: 15322
Rigth Child with pid: 15324 ,created from the parent with pid: 15322
Left Child with pid:15325, created from the parent with pid: 15324
Parent Node with pid: 15324
Rigth Child with pid: 15326 ,created from the parent with pid: 15324
Left Child with pid:15327, created from the parent with pid: 15326
Parent Node with pid: 15326
Rigth Child with pid: 15328 ,created from the parent with pid: 15326
Last Parent of the right subtree: 15328 ,and is created from the parent: 15326
Left Child with pid:15329, created from the parent with pid: 15328
```

Figure 3: Q2 OUTPUT

The tree that is created with this parameters is visualized in the following:

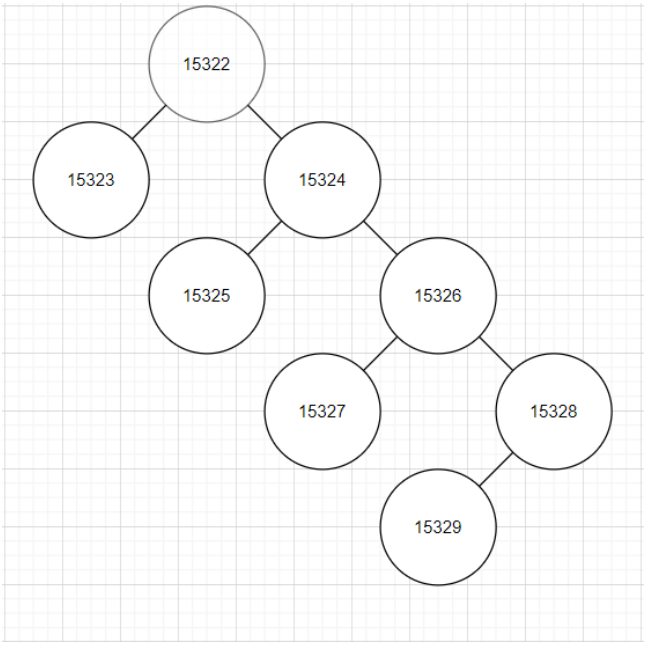

Figure 4: Tree Q2

Another test case is added up where N=3 and M=2. The output is checked and printed out. In the following, we can see a picture representing the output of the case with this parameters.

```
test@blg335e:~/hostvolume/cos_hw1$ gcc -o q1 q1.c
test@blg335e:~/hostvolume/cos_hw1$ ./q1
Please enter the N value: 3
Please enter the M value: 2
The root Node's pid: 15378
Parent Node with pid: 15378
Left Child with pid:15382, created from the parent with pid: 15378
Rigth Child with pid: 15383 ,created from the parent with pid: 15378
Left Child with pid:15382, created from the parent with pid: 15378
Left Child with pid:15384, created from the parent: 15382
Left Child with pid:15385, created from the parent with pid: 15383
Parent Node with pid: 15383
Rigth Child with pid: 15386 ,created from the parent with pid: 15383
Left Child with pid:15385, created from the parent with pid: 15383
Left Child with pid:15387, created from the parent: 15385
Left Child with pid:15388, created from the parent with pid: 15386
Parent Node with pid: 15386
Rigth Child with pid: 15389 ,created from the parent with pid: 15386
Left Child with pid:15388, created from the parent with pid: 15386
Left Child with pid:15390, created from the parent: 15388
Last Parent of the right subtree: 15389 ,and is created from the parent: 15386
Left Child with pid:15391, created from the parent with pid: 15389
The parent Node with the pid: 15391, and it is created from the parent with pid: 15389
Left Child with pid: 15392, created from the parent with pid: 15391
```

Figure 5: Q2 OUTPUT -2-

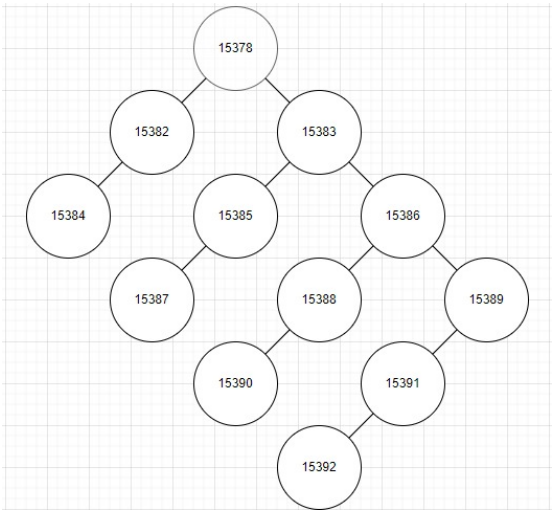

Figure 6: Tree Q2

3.1 Q2.2

We have M parents in each level, and the number of levels is N when we do not count the base case. Considering the base case where the number of parents is equal to 1, we obtain the equation:
f(x)=(N+1)*M