Advanced Operating Systems

ASSIGNMENT NO. 1

1. Write the output of below code and draw the process tree.

```
if (fork() && (!fork())) {
    if (fork() || fork()) {
        fork();
    }
}
printf("Hello");
```

2. Using exec system call list all the files and directories of home then store the result in "output1.txt". Run a second command "grep" and find the word "output1.txt" in all the text files of the home also place home path in "PATH" environment variable while doing exec for grep. This all will be done in one process.

3. Write a program that receives one command line argument: the name of a text file. Your program will work as follows:

- Start out by creating a child process.
- The child process calls exec to run cat command with command line argument that the parent received.

• The parent process calls wait so that it blocks until the child terminates and passes back its termination status.

• If the child process terminates without error, the parent spawns another child and, again, calls wait so that it can block until the child terminates.

• The new child calls exec again, but this time it runs **wc** command on the same argument that the parent received from the command line.

• Once the parent learns that the child has terminated, it goes on to terminate also. If the parent gets to this point, it's because all has gone well.

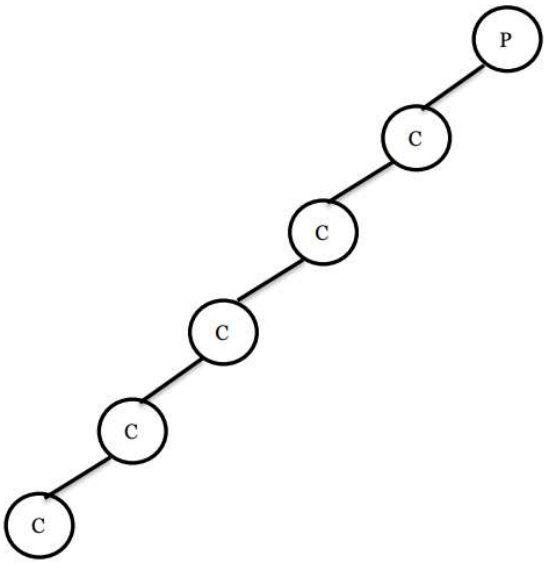
4. Write a program which take string from user and you have to do following tasks with the string:

1. Reverse

2. Find Length

3. Sort

You have to create child of child's as given below process tree for each task and each child exec with the image of program of particular task. Print the string after each operation.



5. Develop a stopwatch with lap counter and total time calculator. The program will create two processes and they will start calculating time. One process will be used to calculate total time and the other will calculate lap time. The total time calculator process will keep on calculating time. In the lap time, calculator process when it reaches lap time limit, the process will display lap number and set its counter to zero and start calculating again. Take input of lap time and number of laps from the user and start the program. At the end, display total time from total time calculator process.

6. How many number of processes will be created in the following program? Prove your answer by making a replica code that will count no of processes created.

```
1  int main()
2  {
3  fork()
4  fork()
5  fork()
6  fork()
7  }
```

Guidelines:

- A single violation of guideline will lead to **Zero mark** in your assignment.
- Deadlines should be kept in mind. **No extension** in assignment dates would be given.
- Submit **scanned document** of output and tree for **Question No 1**.
- Submit the ***.c*** or ***.cpp*** file and the output's screenshot in **Word** file on **Google classroom**.
- You are encouraged to take help from Internet and books.
- You will have maximum marks if you have done the entire task.
- This is an individual assignment. **PLAGARISM IS NOT ACCEPTABLE!**
- Follow the instructions as it is, otherwise your assignment would not be accepted at all.