# School of Computer Science

## COMP-3330 Operating Systems Fundamentals - Winter 2020
## Assignment 2

<u>Deadline:</u>  **Tuesday, February 4, 2020 at 11:59pm**

**Rules:** This assignment must be submitted individually and will be evaluated individually.

**Objectives:** The aim of this assignment is to help students understand how to create processes and how to trace the status of a process. Students will obtain hands-on experience in process creation through the fork system-call and follow up of processes via other system-calls in UNIX or Linux.

> Note: If you wish, you can use also the Windows API or the Java API (with their own system-calls equivalent to the fork system-call and etc of UNIX) in this assignment. I do not recommend it though. Using a POSIX API is much simpler: you have seen the differences in process creation between POSIX and Windows API's.

**Tasks:**

1. Using either a UNIX or a Linux system, write a C program that forks a child process that ultimately becomes a zombie process. This zombie process must remain in the system for at least 10 seconds. Process states can be obtained from the command:

   ```
   ps -l
   ```

   The process states are shown below the S column; processes with a state of Z are zombies. The process identifier (pid) of the child process is listed in the PID column, and that of the parent is listed in the PPID column.

   The easiest way to determine that the child process is indeed a zombie is to run the program that you have written in the background (using the &) and then run the command `ps -l` to determine whether the child is a zombie process. Because you do not want too many zombie processes existing in the system, you will need to remove the one that you have created. The easiest way to do that is to terminate the parent process using the kill command. For example, if the process id of the parent is 4884, you would enter

   ```
   kill -9 4884
   ```

2. Write another C program that does #1 automatically. That is, you do not have to enter the commands `ps` and kill manually. But your program has to: (a) create zombie processes by running your C program of #1, (b) Obtain the state of each process and find if there is any process that is zombie, (c) Show a list of processes with their states, (d) kill the parent of any process that is zombie, and (e) show the updated list of processes with their states.


**Submission:**

1. You must submit: (i) a short report (in PDF or word), which contains the details of the implementation and the input/output of at least two runs of your program, (ii) the source code of your program. You should upload all these files to the Blackboard, as a **single** zip file called Assignment-2.zip.
2. In your report, you must provide details about the implementation, including the programs' input/output, as required. Marks will be deducted if explanations/outputs are missing.
3. Add the following note at the beginning of your report: "*I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any*

*unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work."* + Name + SID

4. Any submission after the deadline will receive a penalty of 10% for the first 24 hrs, and so on, for up to seven days. After seven days, the mark will be *zero*.

5. Unlimited resubmissions are allowed. But keep in mind that we will consider/mark the last submission. This means that if you resubmit after the deadline, a penalty will be applied, even if you submitted an earlier version before the deadline.