CS342 Operating Systems – Fall 2017
Homework 2

```
Assigned: 05 Oct 2017
Due date: 12 Oct 2017, 23:55
```

Submit through Moodle. Make sure you start submitting one day before the deadline. You can overwrite your submission as many times as you wish. Late submissions will ***not be accepted*** *(no excuse – no email will be accepted).*

1) Write a C program called **a** in Linux (the source file will be a.c and the executable file will be **a**) that generates the following process tree consisting of the processes a, b, c, d, e, f, g. The process a is the parent of processes b, c, f, and process c is the parent of processes d and e, and process f is the parent of the process g. We don't care in which order the processes are created. Have each process do the following: in a while loop, the process will print its letter (like a or b), its integer process-id, and its parent's process-id to the screen 100 times and then will sleep 50 seconds using the sleep() call. Use the ps aux command to check the ids of the created processes. Also learn about the ps command by typing ps.

2) Write a simple C program **myexec** (source will be myexec.c) that will take a command (a program name) and its parameters and will execute that command with those parameters. For example: "myexec ls -al" will execute the ls command with -al options. Consider simple commands and programs with no redirections of input/output and with no pipes.

3) Write a simple producer program P (P.c) and consumer program C (C.c) . The producer process P will create 3 child processes P1, P2, P3 and a message queue, and each child process Pi will generate integers from 1 to 10000 and will pass them one by one to the consumer process C by using the same message queue. The consumer C will receive and print those integers to the screen. Hence there will be 3 processes sending integers to the same message queue, and one process receiving from the message queue. Learn message queues from Internet sources. There is also man page man mq_overview that will give you information about Linux message queues.

4) Write a simple program T (T.c) that will create N threads. N is given as an argument to T. The program will create N threads and each thread will write to the screen the integers 1 to 100. N can be between 1 and 100. You will use Ptreads. Learn Pthreads from Internet sources. Time (using the **time** command) your program. Report the total (ream)time, user time and system time. Explain what user time is, what system time is.

Put all your programs and the requested information into a PDF file. Submit the PDF file as your homework.