System Programming – Assignment 2

Aims:

1. Recall process creation/termination and process management in Linux.
2. Understand and modify the task descriptor structure in the kernel
3. Learn how to add a new system call to the Linux kernel.
4. Learn how to compile a Linux kernel.

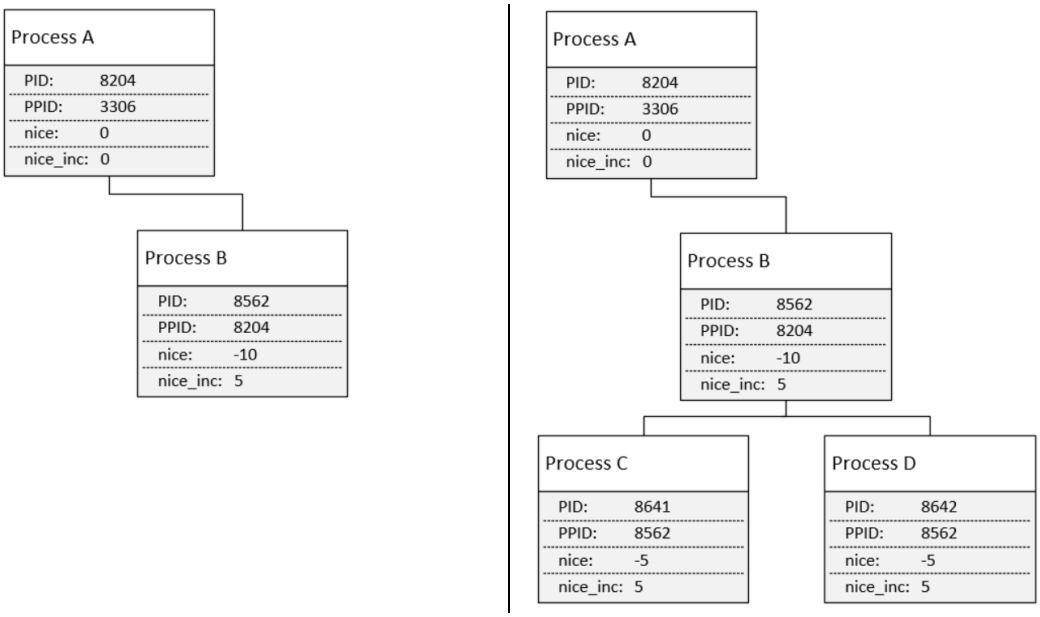
Please remember that this is an individual assignment. You should setup a GitHub account and create a project for this assignment. You should share the project with **uyaritu** and **secintiitu** github accounts. Your progress throughout the semester will be evaluated as well as your final submission.

Implementation:

In this assignment, you are asked to slightly alter the functions of `fork` and `exit` system calls, where `nice` value of the spawned process will be adjusted based on the `nice` value of the parent process and the `nice_increment` value, which will be stored in a field in the task descriptor of the parent process.

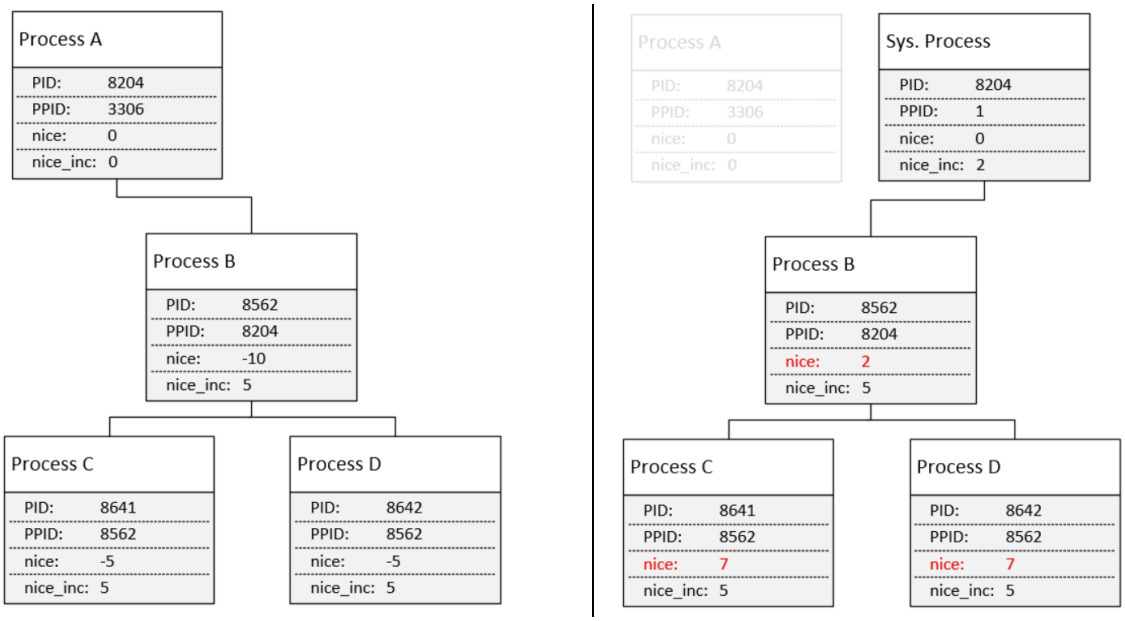
- Upon process generation, the nice value of the child process should be the sum of nice value and the `nice_increment` value of the parent, so the child processes should have an equal or less priority than the parent process, considering the `nice_increment` value should only have **non-negative integers**.
- When a parent process is terminated, nice values of the orphan processes should be re-adjusted following the same approach by using the values from their new parents.

You may find example scenarios below:



Step1. A creates B
Step2. `nice` and `nice_inc` values of B are set.
Step3. B creates two new processes.

Figure 1: Process Creation



Step1: A process tree with non-zero `nice_inc` values.
Step2: Scheduling termination of Process A.

Step3. Modifying nice values after re-parenting. .

Figure 2: Process Termination

You should complete three (3) particular tasks to successfully implement the mechanisms described above.

1. You are required to add a new field to the task descriptor. The name and type of the field is as follows:

   ```
   int nice_inc;
   ```

   Note: A created process should inherit its `nice_inc` field from its parent process.

2. You should write a system call, that sets the value of `nice_inc` field in the task descriptor of a process. The prototype of the system call is given below:

   ```
   long set_nice_inc(pid_t pid, int value);
   ```

   If the `nice_inc` value is set to zero, `fork` system call should work as default. `exit` system call may still has to adjust the `nice` value of the child processes based on the values of their new parent process when their old parent process terminates itself.

3. You should modify the code of both `fork` and `exit` system calls in kernel in order to implement the actions utilizing `nice_inc` value.

Lastly, you should design and implement a user level test program to test your system calls in C.

Beside the repositories that you will setup on GitHub, you should prepare a zipped folder for your final submission on Ninova. This folder should contain (i) source files you have modified in kernel, (ii) the related diff files and (iii) the files of your test program.

References:

Please read the chapters 3, 7 and 10 from the book "Understanding the Linux Kernel, 3rd Edition" by Daniel P. Bovet, Marco Cesati (Publisher: O'Reilly Pub, 2005) which is accessible from ITU Library.