**Gebze Technical University**

**Computer Engineering Departmant**

**Operating Systems(CSE 312)**

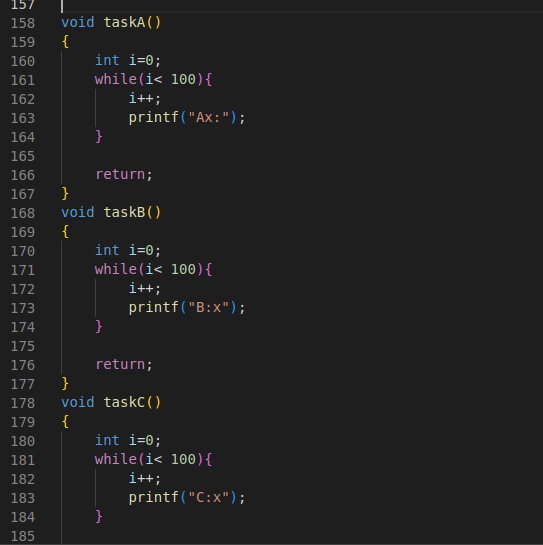
**Homework #1**

**Berkan AKIN 171044073**

A simple operating system development assignment has been given for the Operating Systems course. The development steps of the operating system are explained item by item. The requirements section describes the requirements that have been met for the assignment. How to initiate and run the assignment is explained in detail in the **readme.txt** file.

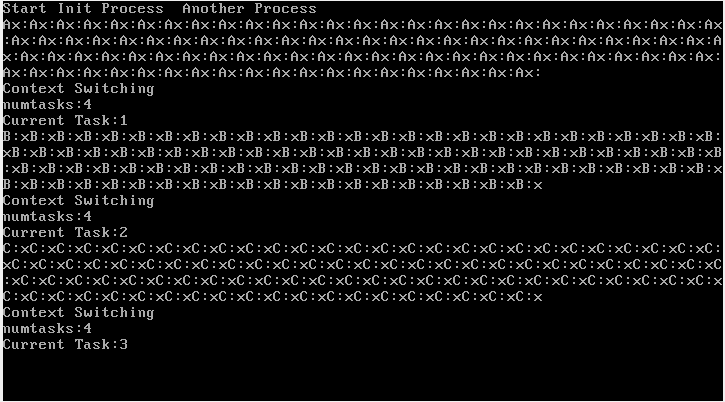
1. **REQUİREMENTS**
2. In the first Hw, we used only the timer interrupts. Instead, in this homework, we include keyboard and mouse click interrupts
3. Handle all those interrupts, which will help us to context switching for our tasks.
4. You have to create all of your processes by using fork syscall
5. By using implemented system calls such as fork, waitpid, execve, any other POSIX call that you need, you will give chance to the user to the process context switching.
6. To this end, instead of context switching for every timer interrupt, you , will ignore at least 10 timer interrupts. Instead, processes may be switching by keyboard and mouse click interrupts.
7. Loading multiple programs into memory: Kernel will be able to load multiple programs into memory
8. Handling multi-programming: you need to develop a Process Table that will hold the necessary information about the processes in the memory. You should study what Process Tables holds. You can read carefully throughout the chapter 2 of the course book or any other online resource).
9. Handling Interrupts: Our given source code can handle interrupts, and your kernel will handle and respond between multiple processes.
10. **Development Stages**
11. Starting other processes with init process and context switching

* The process whose task is finished is context switching with the next process. the latest process runs with an infinite while loop



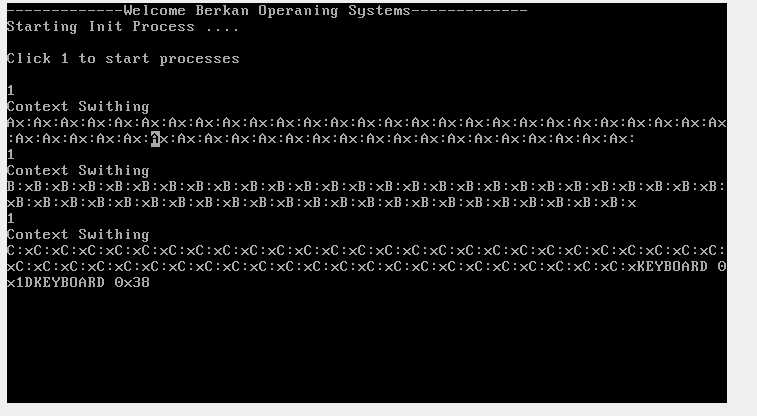


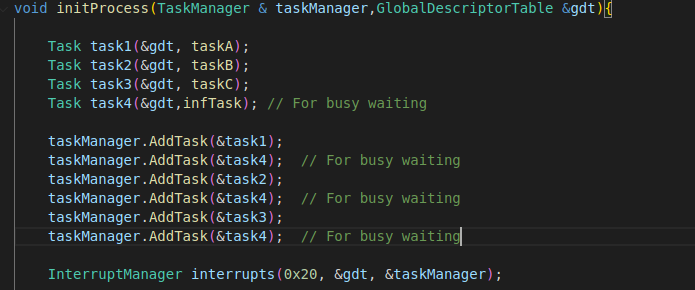
* Result



1. Changing processes with keyboard key

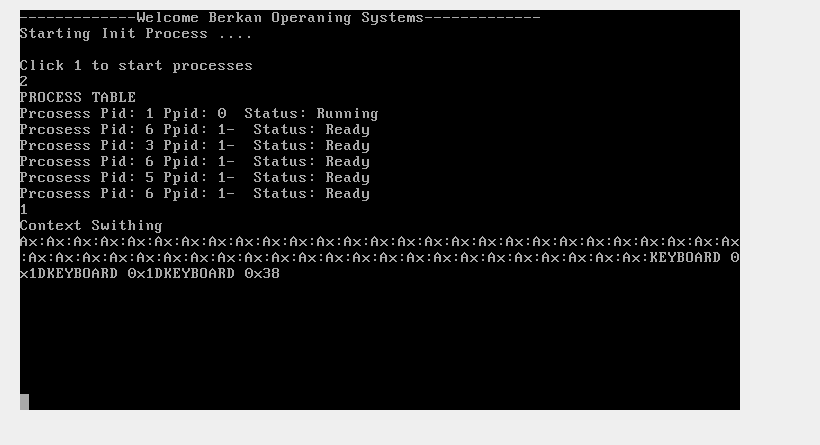
* To start and change processes, click the 1 button to change. Busy waitng is done when it is waiting for the other process.



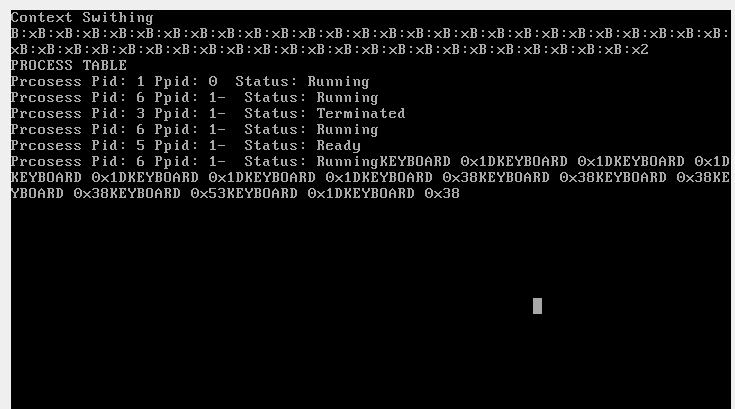


**3**.Process Table Print

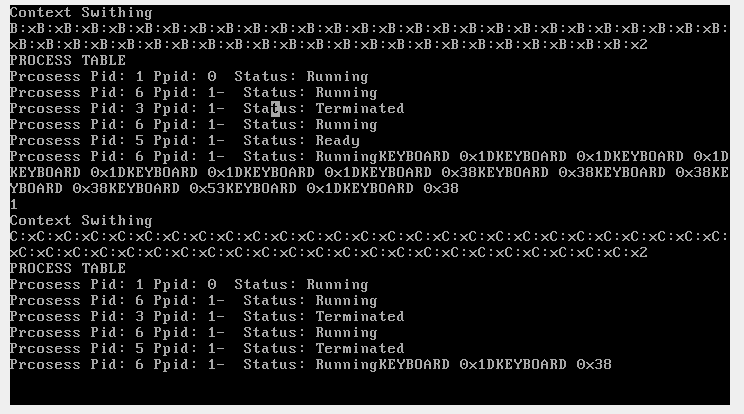
* The process table has been created. When you press the “**2**” key, the process table appears on the screen. The statuses of the processes are printed on the screen. The processes of the processes are put in the terminated state once. There are 3 process states. “Ready = 0 Running = 1, Terminated = 2” these states appear in the process table.



* Approved for finishing the 3rd process. Status terminated



* 5. After finishing the process process, it entered the terminated state



**Process Class**

* Task class is equal to process class (Task = Process)

