

CECS 326 Assignment 1 (20 points)

Due: February 11, 2020 by class time on BeachBoard

This assignment serves as a review of C/C++ programming that includes dynamic memory allocation, pointer, struct or class, queue, array, and random number generation. The program is to simulate much simplified operations for part of the process and memory management in operating systems.

When your program executes, it should first create a memory block table (MBT) data structure, which consists of an integer variable and a Boolean array that maintain the state of the memory. Assume the system has a total of 512 blocks and the first 32 blocks are reserved for the operating system use. Besides the MBT, the program should also create an empty ready queue (The ready queue should be implemented as a linked list. DO NOT use a class from the C++ STL.) The program should then display a menu with four choices and respond to each choice as follows:

1. Initiate a process:

Response:

- generate a random number in the range of 25 to 120 for required memory blocks (this represents the size of the address space of the process being initiated)
- check MBT for free blocks, if insufficient free blocks are available then display error message and return to menu
- allocate dynamic memory for a process control block (PCB)
- obtain a unique process ID (PID) and save it in the PCB
- allocate dynamic memory for a page table of size equal to the number of required memory blocks, save the table size and the pointer to page table in PCB
- allocate memory blocks (only a simulation by designating the required number of blocks in MBT as "allocated") and store block numbers of the allocated blocks in the page table (Note that the allocated blocks need not be consecutive.)
- insert PCB in ready queue
- return to menu

2. Print system state:

Response:

- output MBT
- for each process in ready queue, output its PID, number of blocks allocated to it, and all the blocks that are allocated to it
- return to menu

Note: All output must be properly labelled and formatted.

3. Terminate process with a specific PID:

Response:

- search through the ready queue for the given PID, output error message if not found
- deallocate memory blocks allocated for the process (only a simulation by designating those assigned blocks to this process as "free")
- delete page table for the process
- delete PCB for the process and return to menu

4. Exit:

Response:

- if ready queue is not empty, output PIDs in ready queue and ask user to confirm the request to exit
- delete all processes and deallocate all dynamically allocated memory if affirmative
- when ready queue is empty end program execution, otherwise return to menu

NOTE:

- The program may be implemented in C or C++, and you may use your preferred platform for its development.
- The program must be properly formatted and adequately commented to enhance readability and understanding.
- Prepare a cover page that provides: your name, your student ID, course # and section, assignment #, due date, submission date, and a clear program description. An assignment cover page template is available on BeachBoard.

For submission, do the following:

- Upload on BeachBoard, the cover page, and your program file(s). All uploaded files should have their proper extensions, e.g., .docx (for the cover page), .h, .c, .cpp, etc. DO NOT upload a zip or tar file.

2. Demonstrate your program and explain details of your implementation. For the program demo, bring a hardcopy of the cover page.