Due: Midnight, January 23, 2020

1. Lists and Structures

A simple operating system uses a ***double*** linked list of processes. For each process the following information is maintained:

- Process id
- The process context (Instruction pointer, stack pointer, two registers, AX, BX)
- Process status (initializing, running, suspended, ready, terminated)

Write a "C" structure definition for the process information.

2. C Pointers and References

Write a simple "C" program to:

- Allocate a block of memory whose size will be passed in as a parameter to the program
- Set the block to all zeros (hint: look up the memset function)
- Print the address of the memory block
- Print the contents of the memory block in hexadecimal.
- Free the memory

Submit a listing of the program along with a sample output. I would recommend cutting and pasting the listing and sample output into a document that has the answers to the other questions.

3. Process Transitions

For a process in each of the states below, list each other state the process could transition into and give at least one reason why it might transition into that state:

- Ready
- Suspended
- Running

4. Scheduling Policies

Over the years Linux has used a number of different processing scheduling algorithms. Two current ones are the CFS and BFS schedulers. For each of these algorithms describe, in a paragraph, the main features of the algorithm (are they priority based, time sliced, etc.)