CSCE 5640, Fall 2017
Project #2
Due: 12-4-2017

For this project you may be working in groups. Each member of your group **must** sign up on the corresponding project sheet in order to receive a grade. You task is to develop a memory manager and experiment with different memory management regimes.

Using a simple *process generator* generate a set of 50 processes with different memory and cycle (i.e. runtime) requirements. The memory requirements for processes are in the range of <10KB, 2MB> and the runtimes range between 200 and 2,500 cycles. You may assume that the processes arrive in the system every 50 cycles (e.g., at times 0, 50,100, 150, 200, …etc.). Upon arrival, the system will allocate the required memory for each process, however, a process can only start executing if the required memory is available and can be allocated. After the process has completed the number of required cycles, the system will *free* the memory that was allocated to this process.

1. Assuming that the **combined** memory requirement of all 50 processes is less than 100 MB, use the system calls *malloc()* and *free()* to dynamically allocate and de-allocate memory to your processes. Measure the total system time that is required to simulate the execution of your 50 processes.
2. Develop your own memory management system using **dynamic partitioning** with corresponding function calls *my_malloc()* and *my_free()* to manage a pre-defined block of memory of size 100 MB. The function *my_malloc()* will try to find the required chunk of memory within the pre-defined 100 MB block and allocate it to an arriving process. Upon completion, the memory partition is returned to the memory pool via the function *my_free(), to be used by other processes.* Compare the performance of your memory manager to the performance of *malloc()* and *free()*. Note that your memory manager only calls the system's *malloc()* once to request the initial 100 MB block, hence you are reducing the number of context switches.
3. Develop your own memory management system using **fixed (static) partitioning** with corresponding function calls *my_malloc()* and *my_free()* to manage a pre-defined block of memory of size 100 MB. You will partition the available memory into uniform 5 MB regions. Compare the performance of your dynamic memory management approach used above.

<u>Deliverables</u>: Write a (approx. 5-page) report that highlights your implementation of the memory manager and shows the results of the performance analysis. You must discuss all the limitations and assumptions. You must submit well-documented code.

Rubrics for this project are:

a. Implementation of my_malloc() and my_free() – 30pts
b. Design and Analysis of the simulator – 30pts
c. Report of problem analysis, implementation, experiments, and results – 20pts
d. Oral presentation (about 15 mins) during finals week - 20

Have FUN!!