Prashant Yadav

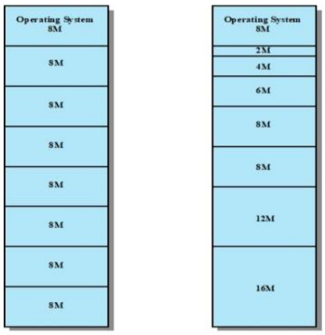
CS 4348.002 – Program 3 Report

Problem statement –

The assigned task was to implement memory allocation by below three configurations.

- Static memory allocation with equal sized block. (i.e. 8MBs each)
- Static memory allocation with unequal sized block as shown in Fig. below.
- Dynamic memory allocation for each process.

Run 1000 simulations of memory allocation for 1000 process for each configuration above and come up with a matrix to compare each configuration.



(a) Equal-size partitions

(b) Unequal-size partitions

Approach to solution –

I went through various algorithms for memory allocation from the lecture notes and book I did referred to few of the online resources like www.geeksforgeeks.org. I implemented first fit algorithm for memory allocation for each of the configuration. In the first fit, the partition is allocated which is first sufficient from the top of Main Memory. While allocating memory in static configuration I keep a check on block size so that when we are allocating memory, we fill entire that block. While in case of dynamic memory allocation block size could be anything based on the memory requirement.

I created an array of 56 length to signify memory location of 56MBs. When a process arrives to allocate program checks if there is a memory available for that process. If there is memory available, we fill that memory location with time required by the process. For each iteration we reduce all non-zero locations in memory array by 1. Here we are assuming all the process in memory will be executed parallelly.

Solution description –

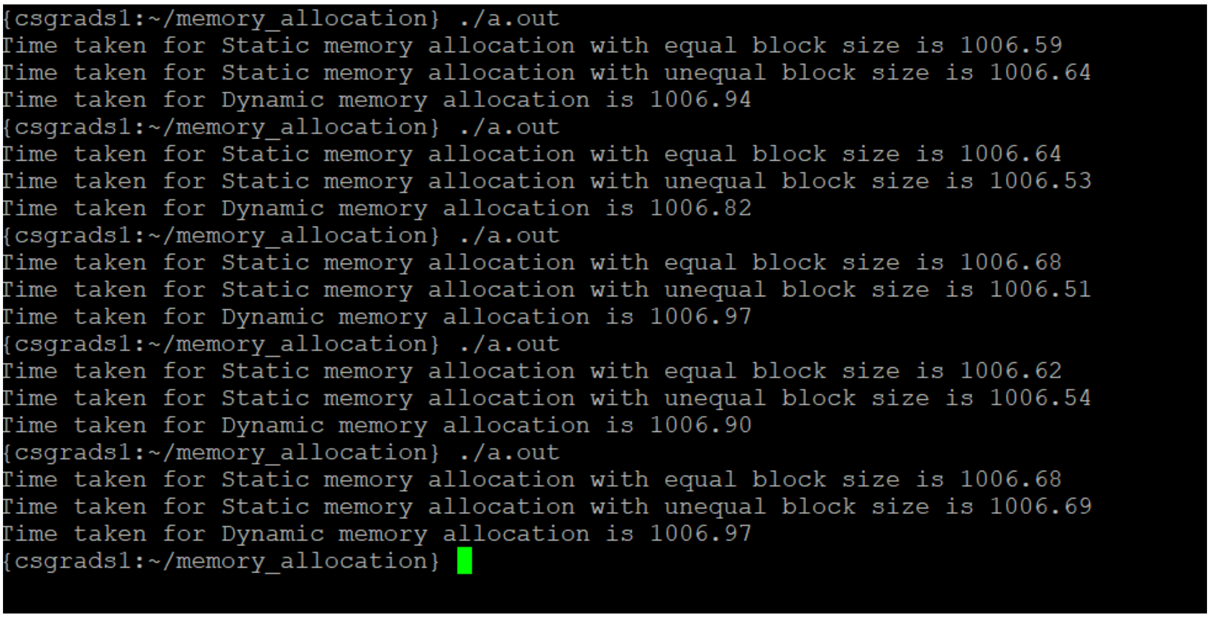
I implemented above mentioned strategy in C language. It does 1000 simulations for each of the configuration. In each simulation program does memory allocation for 1000 processes.

Here are the results for 5 simulation for each configuration.

Simulation #	Static with equal block size	Static with unequal block size	Dynamic memory allocation
1	1006.59	1006.64	1006.94
2	1006.64	1006.53	1006.82
3	1006.68	1006.51	1006.97
4	1006.62	1006.54	1006.90
5	1006.68	1006.69	1006.97

From above readings it shows that for the processes with memory sizes varying between (0,16) static configuration works better as compare to dynamic memory allocation.

Here is the screenshot of execution-



```
{csgrads1:~/memory_allocation} ./a.out
Time taken for Static memory allocation with equal block size is 1006.59
Time taken for Static memory allocation with unequal block size is 1006.64
Time taken for Dynamic memory allocation is 1006.94
{csgrads1:~/memory_allocation} ./a.out
Time taken for Static memory allocation with equal block size is 1006.64
Time taken for Static memory allocation with unequal block size is 1006.53
Time taken for Dynamic memory allocation is 1006.82
{csgrads1:~/memory_allocation} ./a.out
Time taken for Static memory allocation with equal block size is 1006.68
Time taken for Static memory allocation with unequal block size is 1006.51
Time taken for Dynamic memory allocation is 1006.97
{csgrads1:~/memory_allocation} ./a.out
Time taken for Static memory allocation with equal block size is 1006.62
Time taken for Static memory allocation with unequal block size is 1006.54
Time taken for Dynamic memory allocation is 1006.90
{csgrads1:~/memory_allocation} ./a.out
Time taken for Static memory allocation with equal block size is 1006.68
Time taken for Static memory allocation with unequal block size is 1006.69
Time taken for Dynamic memory allocation is 1006.97
{csgrads1:~/memory_allocation}
```