OPERATING SYSTEMS DESIGN

PROJECT REPORT

In order to implement the project, three cases are considered.

Case 1 : Designing a memory management system using system calls malloc() and free().

Case 2 : Designing a memory management system using custom functions for allocating and deallocating memory like my_malloc() and my_free().

Case 3(a) : System has only 50 % of the memory required by the set of 50 processes.

Case 3(b) : System has only 10 % of the memory required by the set of 50 processes.

Brief Overview:

Initially, user will be giving the number of processes to be created as input (process_count). Then we create a process structure that contains details of pid, memory required(assigned randomly), and number of cycles for execution(assigned randomly). Processes arrive in the system for every 50 cycles. And then based on the memory requested and constraints on the total memory available, required memory is assigned to the process using malloc function. We start the thread with process's pid and run to execute requested number of cycles by the process to perform the respective task. Once the assigned number of cycles is completed, the thread will free the memory using free function.

Assumptions:

- Processes can execute concurrently, and once a process has started, it will run to completion(i.e., there is no scheduling/preemption).
- A process can only start executing if the required memory is available and can be allocated.
- Total number of processes considered to be generated for analysis : 50
- Memory block of size : 10MB
- Combined memory requirement of all 50 processes is less than 10MB.
- A struct data structure used to stimulate the 50 processes with pid for the process id, mem for memory required by a process, cyc for number of cycles for process execution, and a memory_pointer to point to the initial address of the process's address space.

- Using the rand() function to generate random memory requirement and cycles for execution.
- Processes are assumed to be executed by performing no operation in a for loop until the process cycles are completed.

Functions:

1. Start_process: Creates a thread which is a simulation of a process.
2. Process: It is a callback function of a thread. It controls process's execution(it is set to wait until it completes its cycles using the for loop and release of memory associated with it.

Case 1 :

We generate the process using the random functions defined for the memory required and the cycles for processes.

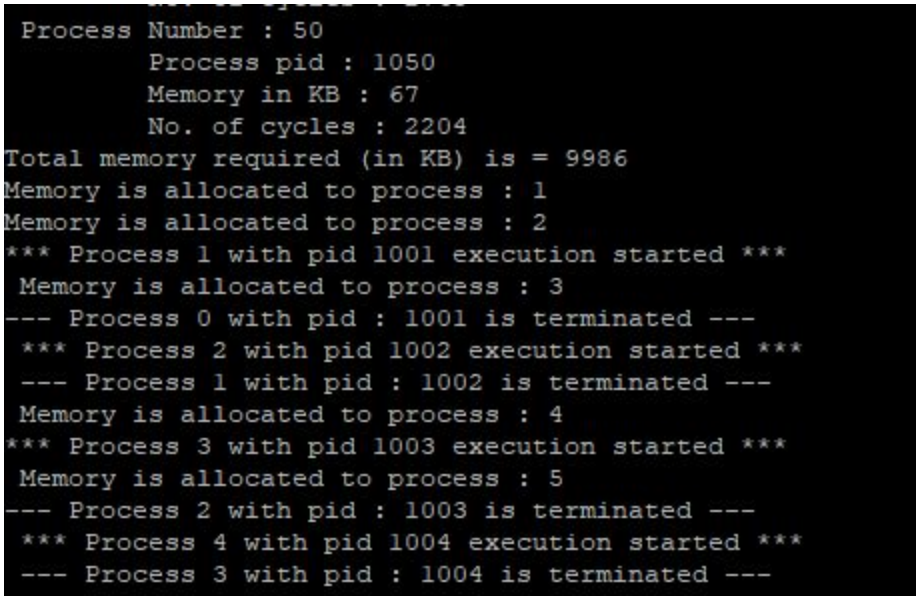
Memory is allocated and deallocated for a process using system calls malloc() and free().

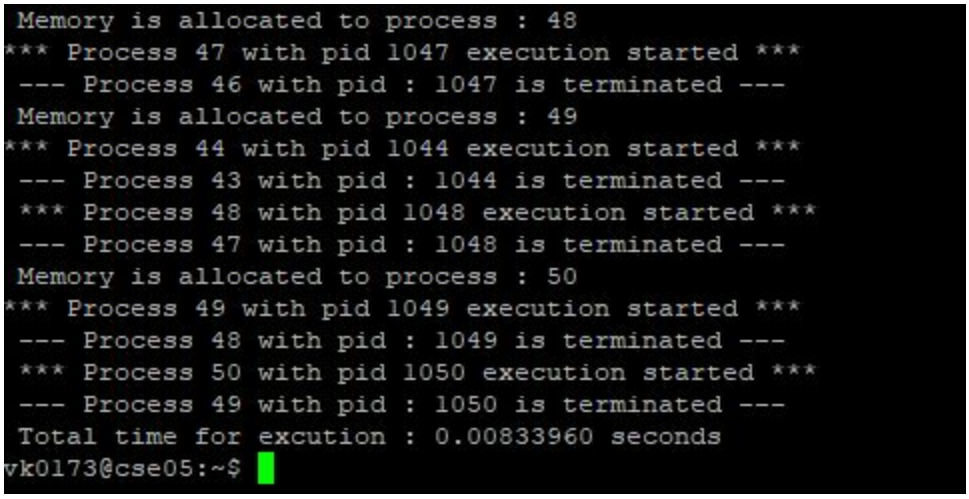
Malloc() - This system call will dynamically allocate memory to the input processes. Once if this system call is made, all the input process will be loaded into the table (inputprocess) and it will be allocated to the different processors available in the system through First Come First Serve basis. Irrespective of the process's cycles and memory required, it will allocate the processes to the corresponding available processor in the FCFS basis.

Free() – Once if this system call will dynamically deallocate the memory . When this is called, all the allocated memory in the p_assign table will be freed and if there is any input process, the system will process from the start.

```
Enter the number of processes to be generated
50

 Process Number : 1
        Process pid : 1001
        Memory in KB : 56
        No. of cycles : 2315
 Process Number : 2
        Process pid : 1002
        Memory in KB : 10
        No. of cycles : 1724
 Process Number : 3
        Process pid : 1003
        Memory in KB : 222
        No. of cycles : 2584
 Process Number : 4
        Process pid : 1004
        Memory in KB : 344
        No. of cycles : 1409
 Process Number : 5
        Process pid : 1005
        Memory in KB : 115
        No. of cycles : 678
 Process Number : 6
        Process pid : 1006
        Memory in KB : 78
        No. of cycles : 772
 Process Number : 7
        Process pid : 1007
        Memory in KB : 150
        No. of cycles : 1428
 Process Number : 8
        Process pid : 1008
        Memory in KB : 141
        No. of cycles : 3431
```





Case 2 :

We generate the process using the random functions defined for the memory required and the cycles for processes.

Memory is allocated and deallocated for a process using system calls malloc() and free().

my_malloc() -

It takes one parameter, the memory size of the process.

Every time we use my_malloc function by passing size argument, it adds the memory used to memory_block and assigns as address to the process. And updates the memory used by incrementing it with the size (memory requested by the process). Finally returns the base address of the process.

my_free() –

This method is used to free the memory associated to the processes that completes the execution. We subtract the process memory from the total memory used.

```
Enter the number of processes to be generated
50

 Process Number : 1
         Process pid : 1001
         Memory in KB : 56
         No. of cycles : 2315
 Process Number : 2
         Process pid : 1002
         Memory in KB : 10
         No. of cycles : 1724
 Process Number : 3
         Process pid : 1003
         Memory in KB : 222
         No. of cycles : 2584
 Process Number : 4
         Process pid : 1004
         Memory in KB : 344
         No. of cycles : 1409
 Process Number : 5
         Process pid : 1005
         Memory in KB : 115
         No. of cycles : 678
 Process Number : 6
         Process pid : 1006
         Memory in KB : 78
         No. of cycles : 772
 Process Number : 7
         Process pid : 1007
         Memory in KB : 150
         No. of cycles : 1428
```

```
        No. of cycles : 2743
 Process Number : 50
        Process pid : 1050
        Memory in KB : 67
        No. of cycles : 2204
Total memory required (in KB) is = 9986
Memory is allocated to process : 1
Memory used : 56
Memory is allocated to process : 2
Memory used : 66
*** Process 1 with pid 1001 execution started ***
 Freeing 57344 KB of memory
*** Process 2 with pid 1002 execution started ***
 Freeing 10240 KB of memory
--- Process 1 with pid : 1002 is terminated ---
```

```
*** Process 48 with pid 1048 execution started ***
 Freeing 370688 KB of memory
--- Process 47 with pid : 1048 is terminated ---
 *** Process 49 with pid 1049 execution started ***
 Freeing 146432 KB of memory
--- Process 48 with pid : 1049 is terminated ---
 *** Process 50 with pid 1050 execution started ***
 Freeing 68608 KB of memory
--- Process 49 with pid : 1050 is terminated ---
 *** Process 46 with pid 1046 execution started ***
 Freeing 191488 KB of memory
--- Process 45 with pid : 1046 is terminated ---
 Total time for excution : 0.01064550 seconds
yk0173@cse05:~$
```

CASE 3:

In here, we will have an addition of free pool concept which contains the freed memory blocks

my_malloc() -

Updated the my_malloc function to accommodate the case when the requested memory is greater than the available memory. It allocates the memory from free memory pool to the process via Best fit algorithm

my_free() –

This method is used to free the memory associated to processes that complete the execution(When the assigned number of cycles is completed). We subtract the process memory from the total memory and assigns the process in free pool with the address of the completed process.

Case 3(a) :

```
Enter the number of processes to be generated
50

 Process Number : 1
         Process pid : 1001
         Memory in KB : 56
         No. of cycles : 2315
 Process Number : 2
         Process pid : 1002
         Memory in KB : 10
         No. of cycles : 1724
 Process Number : 3
         Process pid : 1003
         Memory in KB : 222
         No. of cycles : 2584
```

```
Total memory required (in KB) is = 9986
Allocated 0
Memory used : 56
Allocated 1
Memory used : 66
Allocated 2
Memory used : 288
Process 0 with pid : 1001 is executing
 PROCESS RELEASED FREEING MEMORY
Process 0 with pid : 1001 is terminating
```

```
Process 44 with pid : 1045 is terminating
 ----------WAITING for memory to be released for process 48
----------WAITING for memory to be released for process 48
----------WAITING for memory to be released for process 48
----------WAITING for memory to be released for process 48
----------WAITING for memory to be released for process 48
----------WAITING for memory to be released for process 48
Process 45 with pid : 1046 is executing
 Process 42 with pid : 1043 is executing
 ----------WAITING for memory to be released for process 48
Process 46 with pid : 1047 is executing
 PROCESS RELEASED FREEING MEMORY
```

```
Memory used : 210
Process 48 with pid : 1049 is executing
 PROCESS RELEASED FREEING MEMORY
Process 48 with pid : 1049 is terminating
 Process 49 with pid : 1050 is executing
 PROCESS RELEASED FREEING MEMORY
Process 49 with pid : 1050 is terminating
 Total time for excution : 0.01085862 seconds
vk0173@cse05:~$
```

Case 3(b) :

```
Enter the number of processes to be generated
50

 Process Number : 1
         Process pid : 1001
         Memory in KB : 56
         No. of cycles : 2315
 Process Number : 2
         Process pid : 1002
         Memory in KB : 10
         No. of cycles : 1724
 Process Number : 3
         Process pid : 1003
         Memory in KB : 222
         No. of cycles : 2584
```

```
Total memory required (in KB) is = 9986
Allocated 0
Memory used : 56
Allocated 1
Memory used : 66
Allocated 2
Memory used : 288
Process 1 with pid : 1002 is executing
 PROCESS RELEASED FREEING MEMORY
Process 1 with pid : 1002 is terminating
 Allocated 3
```

```
PROCESS RELEASED FREEING MEMORY
Process 13 with pid : 1014 is terminating
 ----------WAITING for memory to be released for process 17
----------WAITING for memory to be released for process 17
----------WAITING for memory to be released for process 17
----------WAITING for memory to be released for process 17
----------WAITING for memory to be released for process 17
----------WAITING for memory to be released for process 17
Process 14 with pid : 1015 is executing
 PROCESS RELEASED FREEING MEMORY
Process 14 with pid : 1015 is terminating
 Process 15 with pid : 1016 is executing
 PROCESS RELEASED FREEING MEMORY
Process 15 with pid : 1016 is terminating
 ----------WAITING for memory to be released for process 17
```

```
Process 48 with pid : 1049 is executing
 PROCESS RELEASED FREEING MEMORY
Process 48 with pid : 1049 is terminating
 Process 49 with pid : 1050 is executing
 PROCESS RELEASED FREEING MEMORY
Process 49 with pid : 1050 is terminating
 Process 47 with pid : 1048 is executing
 PROCESS RELEASED FREEING MEMORY
Process 47 with pid : 1048 is terminating
 Total time for excution : 0.01318988 seconds
vk0173@cse05:~$
```

Observed output values:

System malloc()	Dynamic malloc()
0.00796048	0.00325984
0.01437671	0.01366081
0.01113906	0.00891962
0.00369263	0.00327898
0.01726007	0.00830352
0.01651644	0.01207539
0.01295337	0.00976252
0.01274815	0.00901993

0.01156480	**0.01057702**
0.01176737	**0.01003514**
0.01230430	**0.01154257**
0.01671355	**0.01399164**
0.01097077	**0.01007500**
0.01031491	**0.01018163**

Average Result obtained	**0.01209161**	**0.00962025**