SOFE 3950: Lab4 Host Dispatcher Shell Project

Group Member:
Umar Ehasn 100634240
Diba Shojaeigoradel 100621768
Yin Zhou 100314426
Thomas Jansz 100652111

Introduction:

The Hypothetical Operating System Testbed (HOST) utilizes four-level priority process dispatchers. The dispatchers is required to maintain Real time and user priority queue. The real time queue has the highest priority hence required to be emptied before another operation to be activated. While the normal operation runs in order from highest priority to lowest priority - feedback queue, the dispatcher can accept lower priority job and insert the job into appropriate queue creating a round-robin like structure. The real time queue ultizies 1 process queue and normal scheduling queue ulitizes 3 process queues. The memory used for this lab is 1024 MB which used among 6 resources (2 printers, 1 scanner, 1 modem, 2 CD drives)

Memory Allocation Algorithms

The memory allocation in host uses contiguous block that must be assigned and allocated before using. In a particular case, 64 Mbytes is assigned and used for real time job and 960 Mbytes is shared among all user jobs. There are various allocation schemes and be used and described below.

1. First Fit: uses first available block that is large enough to be found from top of main memory. It's the fastest search.
2. Next Fit: modification of first fit uses first available block from last assigned location
3. Best Fit: choose block that large enough from the nearest location. It requires full scan of memory to determine the block without sorting
4. Worst Fit: run through the entire memory and select largest block. Worst runtime

The program will utilize First Fit memory allocation scheme as it fastest search and easy to implement. However, the draw back from first fit algorithm is that there will be a small overhead at the beginning for accumulating small size blocks. Nevertheless, it is still the fastest search comparing to Best Fit and Worst Fit search. Although, First Fit and Next Fit are similar, Next Fit requires a lot of memory compaction.

Overall Structure and function used

Queueing:

The queue for process control block and memory structure is done using linked list that implements FIFO (First In First Out) queue.

The function used for queue are the following:

Push: creates a new node and add process the node, the node is added to the tail of the list

Pop: remove the first item added from the list and return the process that is popped

Process control block (PCB):

The PBC is used by the dispatcher which includes process information such as process ID, arrival time, CPU time, priority, memory and resource used and current status. It used linked list to allow queue of the process

The function used are the following:

Load_dispatch: load the process from the input file into the dispatcher using a linked list
Load_jobs: load each into the queue base on arriving time of the process
Terminate_dispatcher: end the dispatcher when the queue is empty
Run_job: run the job when jobs are loaded and there is available resource

Memory structure (MEM):
The Mem is used for memory allocation which contain the process required to be queued. The size of MEM is 1024MB which contains 960MB of shared memory and 64MB allocated for real time job.

The memory function used are the following:

Alloc_mem : allocation memory to max memory of 1024 with 960 for shared memory and 64 for dedicated memory for real time process

Free_mem: free the allocated memory when process is done so additional process can be later added into the memory

Resource Manager:

The resource manager is used to check, allocate, free resources that the process is required to run. The resources available for this particular project is two printers, one scanner, one modem, 2 CD driver. The allocation of the resource will be incorporated when process is loaded into the queue to reduce amount of the functions for the program. The same is applied for resource freeing. The resource become available again when process is terminated.

The function used for resource manager is the following:

Resource_avaliable: check if there is enough available resource for resource allocation

Conclusion:

The lab project is to build a HOST that allows dispatcher to insert process into the queue. Nevertheless, there is always shortcomings to the dispatcher. The main concern would be that as real time process is always have the highest priority, so user process may aging that is staying in the process queue for very long time as resources and memory are allocated by the real time process. The aging of the process can be reduced by increasing priority of the aging process at particular unit time. In addition, as mentioned, first first algorithm suffers from memory fragmentation at the beginning, based on the amount of resources available and memory mapping, next fit could be used for comparison analysis to see if any improved could be observed.