CS-362L Operating System Lab 10

Assignment 10

Memory Management

Objectives: To understand memory management techniques and implement it.

What is memory management?
Memory Management is the process of controlling and coordinating computer memory, assigning portions known as blocks to various running processes to optimize the overall performance of the system. It helps processes to move back and forward between the main memory and disk. It helps OS to keep track of every memory location, irrespective of whether it is allocated to some process or it remains free.

What is contiguous memory allocation?
In contiguous memory allocation each process is contained in a single contiguous block of memory. Memory is divided into several fixed size partitions. Each partition contains exactly one process. When a partition is free, a process is selected from the input queue and loaded into it. The free blocks of memory are known as holes. The set of holes is searched to determine which hole is best to allocate.

Memory Allocation Methods
Memory allocation is a process by which computer programs are assigned memory or space. It is of three types.

Best Fit:

Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.

Processing steps:

1. Input the number of blocks, say **"m"**.
2. Input the size of each block.
3. Input the number of processes, say **"n"**.
4. Input the size of each process.
5. Initialize all memory blocks as free.
6. For each process, start by searching the **SMALLEST** possible block size that can be assigned to current process i.e., find min (blockSize[1], blockSize[2],.....blockSize[n]) > processSize[current], if found then assign it to the current process.
7. If not then leave that process and keep checking the further processes.

First Fit:

Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended. We can stop searching as soon as we find a free hole that is large enough.

Processing steps:

1. Input the number of blocks, say **"m"**.
2. Input the size of each block.
3. Input the number of processes, say **"n"**.
4. Input the size of each process.
5. Initialize all memory blocks as free.
6. For each process, find the **FIRST** block that can be assigned to current process. If size-of-process ≤ size of block then assign and check for next process.
7. If not then keep checking the further blocks.

Worst Fit:

Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.

1. Input the number of blocks, say **"m"**.
2. Input the size of each block.
3. Input the number of processes, say **"n"**.
4. Input the size of each process.
5. Initialize all memory blocks as free.
6. For each process, find the **LARGEST** block that can be assigned to current process. i.e., find max(bockSize[1], blockSize[2],. blockSize[n]) > processSize[current], if found then assign it to the current process.
7. If not then keep checking the further blocks.

Conclusion: At the end of this lab, student will be able to implement different memory management techniques.

LAB TASK

1. Implement the **BEST FIT** memory management by taking a number of processes,and block and their respective size and implement it. You need to print a table showing the process size, block size and block number which is assigned to current process.
2. Implement the **FIRST FIT** memory management by taking a number of processes,and block and their respective size and implement it. You need to print a table showing the process size, block size and block number which is assigned to current process.
3. Implement the **WORST FIT** memory management by taking a number of processes,and block and their respective size and implement it. You need to print a table showing the process size, block size and block number which is assigned to current process.

Note: You need to prepare a word document containing your code along with the screenshot of output for each program.