CMPT 424N-111 - Fall 2019 - Dr. Labouseur

Lab Three

Eric Stenton

Eric.Stenton1@Marist.edu

September 22, 2019

1 Problem One

Question: Explain the difference between internal and external fragmentation.

Answer: The difference between internal and external fragmentation lies with the location of the 'holes' of free memory space that result from processes being loaded into main memory using particular strategies. The first-fit and best-fit algorithms are two methods of divvying out memory space that accrue areas of external fragmentation [1]. When available memory space is given to a process, the remaining space not used is fragmented from other areas of free space. On the other hand, if main memory is divided into blocks of predefined lengths to be given to processes, then fragmentation may still occur within a block that is being used. Internal fragmentation describes the areas of free memory space that result from processes being given blocks of memory that are too large and have an unused amount [1]. Thus, internal and external fragmentation both describe the existence of areas of free memory that cannot be properly utilized due to the method of allocating memory space for processes. The difference is whether the free memory space exists within predefined memory blocks or between processes that are simply given slices of memory that they require.

2 Problem Two

Question: Given five (5) memory partitions of 100KB, 500KB, 200KB, 300KB, and 600KB (in that order), how would optimal, first-fit, best-fit, and worst-fit algorithms place processes of 212KB, 417KB, 112KB, and 426KB (in that order)?

Answer:

First-fit: The first-fit algorithm allocates memory based on the first hole that it finds suitable for a process [1]. Thus, the algorithm will take the 212KB process, skip the 100KB memory partition since it is too small, and place the process in the 500KB partition resulting in a 288KB hole. Assuming searching starts at the beginning again, the algorithm will take the 417KB process, skip over the first 4 partitions of memory, and place it in the 600KB partition resulting in a 183KB hole. Next, the 112KB process will be placed in the second partition that has a 288KB hole resulting in a 176KB hole. The final 426KB process is unable to fit into any of the partitions assuming that all previous processes are still allocated. The current amount of free memory in each partition are as follows: 100KB, 176KB, 200KB, 300KB, 183KB.

Best-fit: The best-fit algorithm allocates memory based on the smallest hole that will accommodate a process [1]. Starting with the 212KB process, it will be placed in the 4th partition with 300KB of free memory resulting in a hole of 88KB. The 417KB process will be placed in the 2nd partition with 500KB resulting in a hole of 83KB. Next, the 112KB process will be placed into the 3rd partition with 200KB creating a hole of 88KB. Lastly, the 426KB program will be placed in the 5th and last partition with 600KB creating a hole of 174KB. All processes are able to fit into memory and the amount of free memory in each partition are as follows: 100KB, 83KB, 88KB, 88KB, and 174KB.

Worst-fit: The worst-fit algorithm allocates memory based on the largest hole that will accommodate a process [1]. The first 212KB process will be placed in the last partition with 600KB creating a hole of 388KB. The second 417KB process will be placed in the 2nd partition with 500KB resulting in a hole of 83KB. The third 112KB process will find its home in the last partition of 388KB resulting in a hole of 276KB. The fourth and last 426KB process is unable to fit in any of the remaining holes in the partitions assuming none of the previous processes were terminated and gave up their allotted memory, and that no form of compaction between the partitions occurred. The partitions and amount of free memory in each are as follows: 100KB, 83KB, 200KB, 300KB, and 276KB.

Optimal: In the above situation, the best-fit algorithm allocates memory to the desired processes while conforming to the organization of the partitions in the most optimal manner. It was able to allot memory for each process, specifically the last 426KB process in which both the first-fit and worst-fit algorithms were unable to accomplish.

References

[1] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts (9. ed.).* Wiley, 2013.