

1) Memory Management

a. What is a memory management?

Memory management is the part of the operating system which manages memory allocations. We can identify two types of partitioning strategy:

- Fixed memory
- Dynamic memory

In the fixed sized memory partitioning strategy, the memory is partitioned into equal sized portions. Each portion is able to load only one process at a time. However, we can meet an internal fragmentation. This situation is due to the fact that we cannot re-used free hole areas (spaces of the memory which are not used by a process). Moreover, there is no solution for this issue.

In the dynamic sized memory each process is coming in the queue and allocated in the memory the space it needs.

In the case that there is not enough space to swap in a process, we have an external fragmentation. The dynamic memory allowed us to merge all the free spaces in the memory to have bigger free space and thus swap in the process. This solution is called compaction and is the best solution for the external fragmentation.

b. How does it work?

We will take this information's table for the following examples:

Process	Size (in Kb)	CPU Burst time (in ms)
P1	600	6
P2	1000	10
P3	300	3
P4	700	7
P5	500	5

1. The fixed sized memory portioning strategy:

We will consider the following fixed memory:

OS area	0kb
	400kb
250kb	
250kb	650kb
250kb	900kb
250kb	1150kb
250kb	1400kb
250kb	1650kb
250kb	1900kb
250kb	2150kb
////////	2260kb

OS area	0kb
P1	400kb
P1	650kb
P1	900kb
////////	1150kb
250kb	1250kb
250kb	1500kb
250kb	1750kb
250kb	2000kb
250kb	2250kb
////////	2260kb

Now, if we want to add the first process, we are going to have P1 divided into 3 parts (250+250+100) as follow:

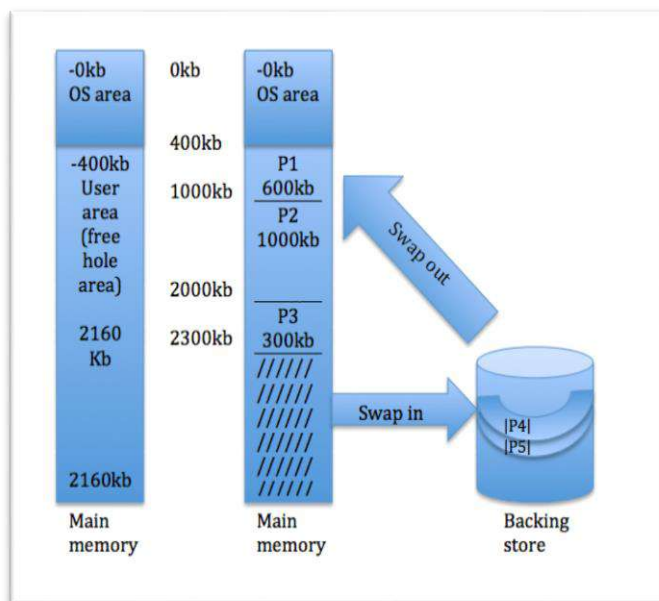
Thus, the third partition of P1 is not full. The remaining part cannot be re-used by another process until P1 is not swapped out.

Then we swap in P2:

OS area	0kb
P1	400kb
P1	650kb
P1	900kb
////////	1150kb
P2	1250kb
P2	1500kb
P2	1750kb
P2	2000kb
P2	2250kb
////////	2260kb

We now, have filled the memory. We must wait the end of a process to purchase.

2. The dynamic memory:



In this case, the memory allocated partitions with the size that the process needs. As soon as there are enough places for the next process, this one is swapped in the main memory. If a process in the RAM is not executed, because of fault memory or whatever, it will be swapped out.

3. Compaction:

This technic merges all free hole area to then permit the allocation of the next process.

OS area	0kb
P1	400kb
P2	1000kb
P3	2000kb
//////////	2300kb
	2560kb

Thus, we have the following allocation at first with FIFO.

OS area	0kb
//////////	400kb
P2	1000kb
P3	2000kb
//////////	2300kb
	2560kb

Then P1 ended, so its place in the memory is freed (before compaction). We have not enough space to load P4 (600kb 260kb) we are in external fragmentation.

Here is the compaction which gives us a 860kb of free hole area:

OS area	0kb
//////////	400kb
//////////	
P2	1260kb
P3	2260kb
	2560kb

Now we have enough free hole area to allocated P4 with its 700kb. We succeeded to solve the external fragmentation.

c. Page-replacement algorithms

FIFO page replacement algorithm is the most simplest. In this algorithm the time of arrival of each page is recorded and added to tail. If a page required by the CPU is not found in the main memory, we have a page fault event. The FIFO-page replacement algorithm is not efficient because it causes more number of page faults.

Then; the page which came first into RAM will be choose for swap out or replacement.

2) Conclusion

The memory management is very important in the operating system because it reduces time of execution of processes. By this way, each process is pre-loaded to avoid the CPU to look in the backing store and thus lost a lot of time for each process. This method allows identifying errors on the processes or deadlock situations before the CPU is using them.

3) Sources

<http://www.memorymanagement.org/articles/begin.html>

<http://tldp.org/LDP/tlk/mm/memory.html>

<http://www.microsoft.com/whdc/system/kernel/wmm.msp>