SWAPPING, NONCONTINUOUS MEMORY LOCATION

Swapping:

A process must be in memory to be executed. A process, however can be swapped temporarily out of memory to a backing store and then brought back into memory for continued execution. For example assume a multiprogramming environment with a Round Robin CPU scheduling algorithms. When a quantum expires, the memory manger will start to swap out the process that just finished and to swap another process into the memory space that has been freed.

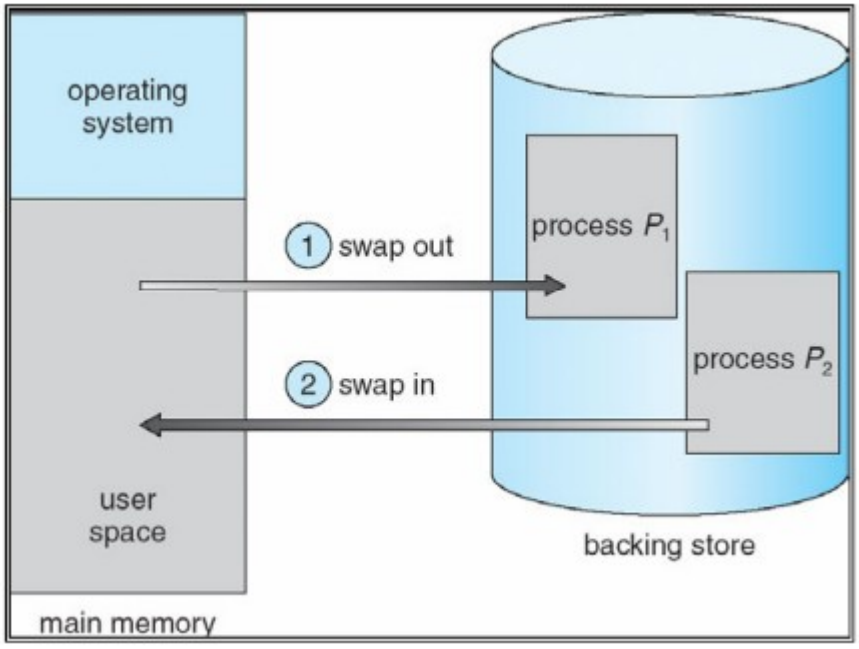

Fig: Swapping of two processes using a disk as a backing store

Logical Address VS Physical Address:

An address generated by the CPU is commonly referred to as a l**ogical address**, whereas address seen by the memory unit- that is the one loaded into the memory-address register of the memory- is commonly referred to as a **physical address**.

The compile time and load time address binding methods generate identical logical and physical addresses. However the execution time address binding scheme results in differing logical and physical address. In that case we usually refer to the logical address as Virtual address.

The run time mapping from virtual address to physical address is done by a hardware called **Memory management unit (MMU).**

Set of all logical address space generated by a program is known as l**ogical address space** and the set of all physical addresses which corresponds to these logical addresses is called **physical address space**.

Non-contiguous Memory allocation:

Fragmentation is a main problem is contiguous memory allocation. We have seen a method called compaction to resolve this problem. Since its an I/O operation system efficiency gets reduced. So, a better method to overcome the fragmentation problem is to make our logical address space non-contiguous.

Consider a system in which before applying compaction, there are holes of size 1K and 2K. If a new process of size 3K wants to be executed then its execution is not possible without compaction. An alternative approach is divide the size of new process P into two chunks of 1K and 2K to be able to load them into two holes at different places.

1. If the chunks have to be of same size for all processes ready for the execution then the memory management scheme is
called **PAGING.**
2. If the chunks have to be of different size in which process image is divided into logical segments of different sizes then this method is called **SEGMENTATION.**
3. If the method can work with only some chunks in the main memory and the remaining on the disk which can be brought into main memory only when its required, then the system is called **VIRTUAL MEORY MANAGEMENT SYSTEM.**

Virtual Memory:

The basic idea behind virtual memory is that the combined size of the program, data, and stack may exceed the amount of physical memory available for it. The operating system keeps those parts of the program currently in use in main memory, and the rest on the disk. For example, a 512-MB program can run on a 256-MB machine by carefully choosing which 256 MB to keep in memory at each instant, with pieces of the program being swapped between disk and memory as needed.
Virtual memory can also work in a multiprogramming system, with bits and pieces of many programs in memory at once. While a program is waiting for part of itself to be brought in, it is waiting for I/O and cannot run, so the CPU can be given to another process, the same way as in any other multiprogramming system.

Virtual memory systems separate the memory addresses used by a process from actual physical addresses, allowing separation of processes and increasing the effectively available amount of RAM using disk swapping.

Source : http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf