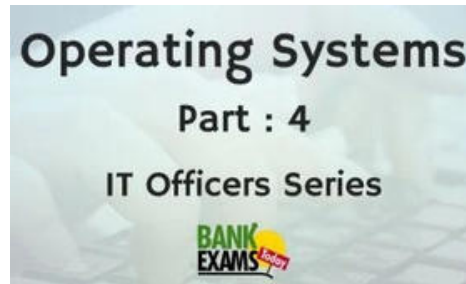


Operating System (Memory Management) - Part 6

Published on Friday, February 26, 2016

Hi Folks,



Today we'll cover **memory management** in operating system. For previous article on CPU scheduling, [click here](#). Use comments section for any queries.

This article is a part of [PK Series](#) for IT Officers exam.

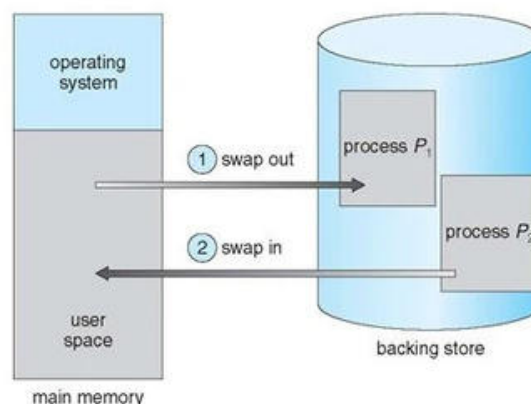
Dynamic Loading and Linking

In Dynamic Loading, some part of the program code which resides in some external library, is loaded **into memory** during execution, mostly **on demand**. Dynamic loading is not controlled by OS but rather by programmer. It encourages **better memory utilization** as unused routines are never loaded into memory.

In **Dynamic Linking**, system libraries and program code are combined by loader at **execution time**. When the libraries are combined at load time itself, it is called static linking.

Swapping

In this mechanism, a process can be **temporarily swapped out** of memory to a backing store (usually a hard disk drive) and then brought back into memory for continuing with execution. Whether or not, the swapped out process will be swapped back into same physical address, depends on address binding method used. Diagrammatic representation of swapping is as follows:

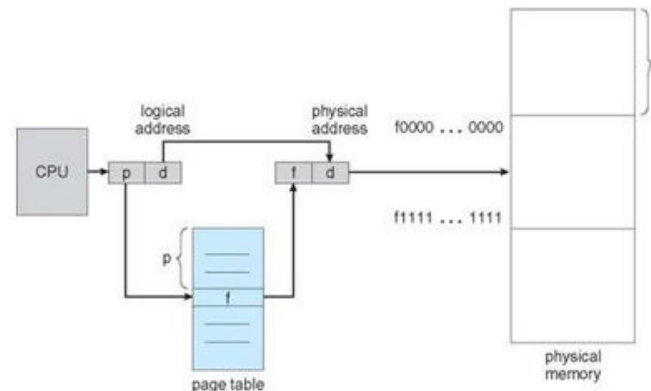


Paging

Paging mechanism divides physical memory into fixed size blocks called **frames**. And it divides the logical memory into blocks of same size called **pages**. Logical addresses are mapped to a real or physical memory addresses. Logical addresses

are also called virtual addresses and are generated by CPU.

Now whenever a process has to be executed, its corresponding pages are loaded in available memory frames. Operating system sets up a **page table to translate** logical to physical addresses. Logical addresses are divided into **page number** and **page offset**. Page number is used as an index to page table and page offset defines the physical address associated. Page tables are of two types further- **hashed and inverted**.



Virtual Memory

It allows **separation of logical memory** from physical memory. Large programs store themselves in form of pages here and at execution, not entire programs but required pages are loaded into main memory. It allows address space to be shared by several processes. Virtual memory can be implemented via demand paging and demand segmentation.

Demand Paging

In this concept, pages are brought into memory only when they are needed. It provides less I/O and more faster response. **How this works** - whenever a page is needed, a reference is made to it and brought into memory. If invalid reference, instruction is aborted and if required page is not in memory- it is brought into it and referenced again. When the page required is not available - an interrupt called trap is issued to operating system, this process is called **page fault**.

Page Replacement

As we discussed earlier that whenever a page fault occurs, referenced page is brought into memory. Now let's say memory has no free space and some other page needs to be swapped out for that. This process is called **page replacement**. This swapped out page when referenced later will have to be read from disk and this might involve waiting for I/O completion. This is what defines the quality of page replacement algorithm used. Various algorithms used are as follows:

- **RAND (Random)**- chose any page to replace.
- **OPT (Optimal)** - replace the page that will be reference furthest in future.
- **FIFO (Fist in, First out)** - replace the page that has been longest in the memory.
- **LRU (Least recently used)** - replace the page that was last reference long time ago.

Linux is present in highly critical applications across the world ranging from Japan's bullet trains, NYSE, CERN and various nuclear submarines.