OS TERM PAPER

NAME – SIDDHARTHA ADDY

STREAM – CSE

YEAR – 2020

SEMESTER – 5TH

ROLL NUMBER – 33200118013

SUBJECT – OPERATING SYSTEM

Paging

Paging is where programs get allocated to physical memory when it is made available. In a typical windows operating systems environment, page tables and page directories are created for each of the running programs. In the windows environment, the physical address of a page table is entered into the page directory during the creation of a process (Bhat, 22).

These page table entries can either be valid or invalid. Depending on their status of validity, each page table may contain the physical address pages that are directly allocated to a running program. At this point, it is important to remember that processes do not identify physical addresses but only know logical addresses.

Therefore, as discussed elsewhere, it is the responsibility of the MMU and the processor to map logical addresses to physical addresses. In windows operating system, the page directory page address is the page directory where a process can be found in the physical memory of the system. It is important to note that the page directory of the page address is found located in the register of the CPU and is referred to as CR3 (Bhat, 22).

Typically, pages are created by dividing the available logical memory data structure into storage blocks while physical storage is partitioned into fixed memory chunks known as frames. Therefore page tables are set to translate logical addresses to physical addresses as discussed above. The operating system ensures that frames are kept track of as they are used during program execution. Therefore, a program acquires a number of pages that are similar to the number of frames to load into.

Ideally, virtual memory is partitioned into page sizes that are equal to page frames, a characteristic that allows the OS to fetch and move pages with flexibility on the disk to the page frames found in the physical memory. That flexibility of fetching and mapping pages to page frames is illustrated in the fig. 4 below.

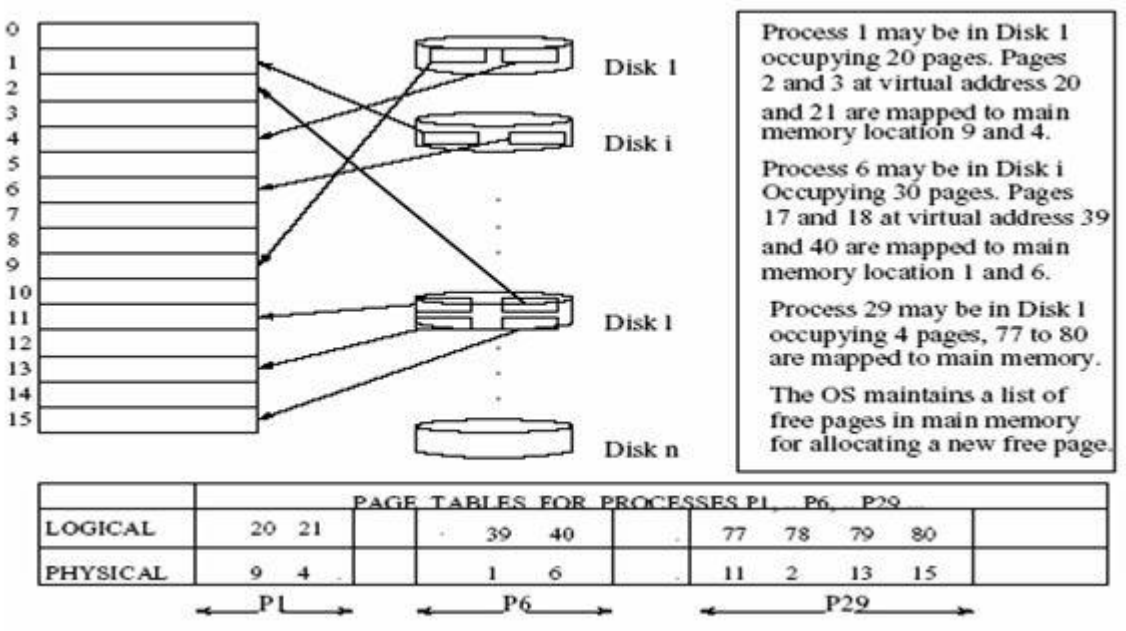

Fig. 4. Mapping of addresses (Silberschatz and Gagne)

In the above illustration, paging requires the support of the OS and the hardware as illustrated in the figure 4 above. As shown on the diagram, a number of pages have been created and made available in main memory. The pages created using this strategy are referred to as the resident set with a locality of reference being taken care of always.

It is important to note that the set of pages that are required by a running program are referred to as the working set. Both the resident set and the working set have to be the same despite the fact that the requirement is not always fulfilled. It is the duty of the OS to enforce the latter requirement. When the OS fails to achieve the latter objective, then it is referred to as a page fault. However, when a page fault occurs, the OS's responsibility is to identify the required page and fetch it and load it into its free page frame.

Then, the OS proceeds to make entries for the specific page into the designate page table. On the other hand, for optimal use of the CPU and main memory, the operating system swaps processes into and out of memory by deleting specific entries of the page tables for the programs that are swapped out of memory. It is important to see the link at this point between the OS and the memory manager. Sometimes the OS forces pages out of memory to allow other pages to be loaded into memory.

That situation arises when all the page frames in memory are in use, but another process wants to use a page frame. To achieve that objective, the OS uses a page

replacement policy. A page replacement policy is typically characterized by the way a process uses page frames. The OS ensures that it has a record of the way pages are used in a read and write operation.

Once a page has been written to or read from, it subjected to a modified bit. One reason is that the page has already been referenced. Implying that the page can be moved using the correct page replacement policy. That ensures that the system throughput is maintained to achieve good system efficiency (Bhat, 22).

A page replacement policy allows the OS to swap processes within the memory storage area following a specific algorithm. These policies include FIFO, LRU, and NFU policies. The FIFO policy is based on first in first out process swapping. Thus, processes are moved based on their arrival time. On the other hand, LRU is a least recently used policy that identifies and moves pages whose usage was further from the current time. The NFU is based on the frequency of usage of the pages based on program count

In a typical windows operating systems environment, the frame size is usually 1024 bytes. However, other page frames vary up to 4 k. It is important to note that paging is a very important concept in multiprogramming. Paging therefore supports an environment where several resident programs execute in the CPU at the same time

What is Paging Protection?

The paging process should be protected by using the concept of insertion of an additional bit called Valid/Invalid bit. Paging Memory protection in paging is achieved by associating protection bits with each page. These bits are associated with each page table entry and specify protection on the corresponding page.

Advantages of Paging

Here, are advantages of using Paging method:

- Easy to use memory management algorithm
- No need for external Fragmentation
- Swapping is easy between equal-sized pages and page frames.

Disadvantages of Paging

Here, are drawback/ cons of Paging:

- May cause Internal fragmentation
- Complex memory management algorithm
- Page tables consume additional memory.
- Multi-level paging may lead to memory reference overhead.

What is Segmentation?

Segmentation method works almost similarly to paging, only difference between the two is that segments are of variable-length whereas, in the paging method, pages are always of fixed size.

A program segment includes the program's main function, data structures, utility functions, etc. The OS maintains a segment map table for all the processes. It also includes a list of free memory blocks along with its size, segment numbers, and it's memory locations in the main memory or virtual memory.

Advantages of a Segmentation method

Here, are pros/benefits of Segmentation

- Offer protection within the segments
- You can achieve sharing by segments referencing multiple processes.
- Not offers internal fragmentation
- Segment tables use lesser memory than paging

Disadvantages of Segmentation

Here are cons/drawback of Segmentation

- In segmentation method, processes are loaded/ removed from the main memory. Therefore, the free memory space is separated into small pieces which may create a problem of external fragmentation
- Costly memory management algorithm

Summary:

- Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages.
- The paging process should be protected by using the concept of insertion of an additional bit called Valid/Invalid bit.
- The biggest advantage of paging is that it is easy to use memory management algorithm
- Paging may cause Internal fragmentation
- Segmentation method works almost similarly to paging, only difference between the two is that segments are of variable-length whereas, in the paging method, pages are always of fixed size.
- You can achieve sharing by segments referencing multiple processes.
- Segmentation is costly memory management algorithm