V COPY-ON-WRITE IN OPERATING SYSTEM:

- In this tutorial, we will be covering the Copy on Write which is one of the resource management techniques.
- **Copy-on-Write(CoW)** is mainly a resource management technique that allows the **parent and child process** to share the same pages of the memory initially. If any process either parent or child modifies the shared page, only then the page is copied.
- **The CoW** is basically a technique of efficiently copying the data resources in the computer system. In this case, if aunit of data is copied but is not modified then **"copy"** can mainly exist as a **reference to the original data.**
- But when the **copied data is modified**, then at that time its copy is created(where new bytes are actually written)assuggested from the name of the technique.
- The main use of this technique is in the implementation of the fork system call in which it shares the virtualmemory/pages of the Operating system.
- Recall in the **UNIX(OS)**, the fork() system call is used to create a **duplicate process of the parent process** which isknown as the child process.

The **CoW** technique is used by several Operating systems like Linux, Solaris, and Windows XP.

The **CoW** technique is an efficient process creation technique as only the pages that are modified are copied.

Free pages in this technique are allocated from a pool of **zeroed-out pages.**

The Copy on Write(CoW) Technique

The main intention behind the **CoW** technique is that whenever a parent process creates a child process both parent andchild process initially will share the same pages in the memory.

- These shared pages between parent and child process will be marked as copy-on-write which means that if the parent or child process will attempt to modify the shared pages then a copy of these pages will be created and the modifications will be done only on the copy of pages by that process and it will not affect other processes.

Now its time to take a look at the basic example of this technique:

Let us take an example where Process A creates a new process that is Process B, initially both these processes will sharethe same pages of the memory.

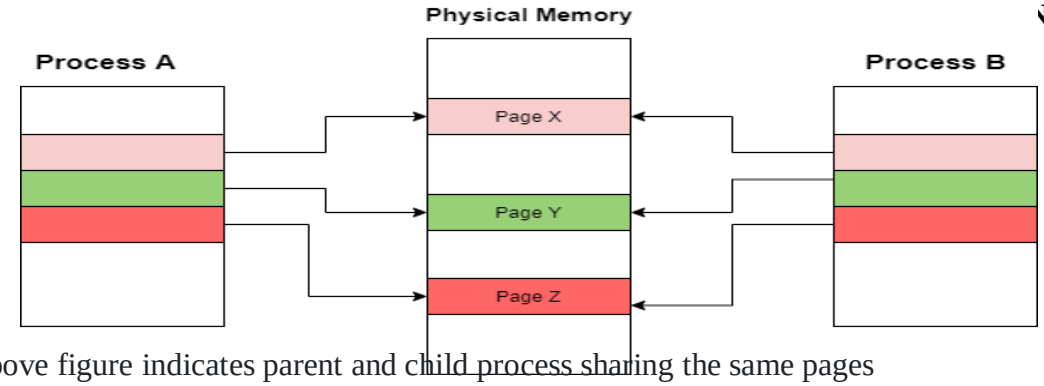


figure: Above figure indicates parent and child process sharing the same pages

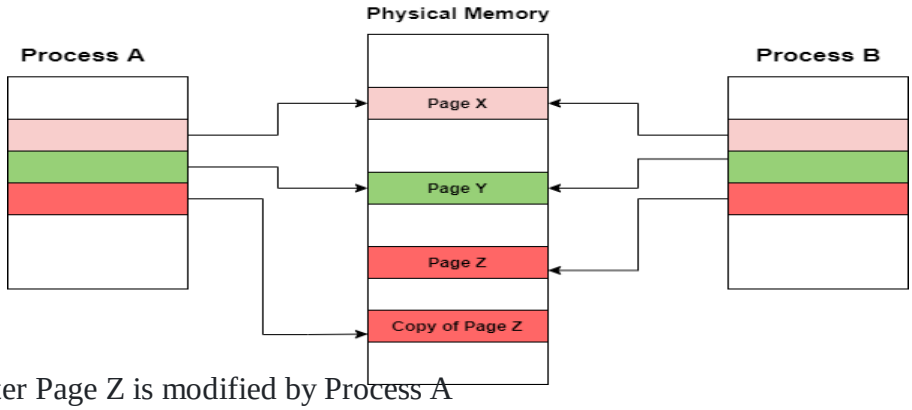


Figure: After Page Z is modified by Process A

Now, let us assume that process A wants to modify a page in the memory. When the **Copy-on-write(CoW)** technique isused, only those pages that are modified by either process are copied; all the unmodified pages can be easily shared by the parent and child process.

Whenever it is determined that a page is going to be duplicated using the copy-on-write technique, then it is important tonote the location from where the free pages will be allocated. There is a pool of free pages for such requests; provided by many operating systems.

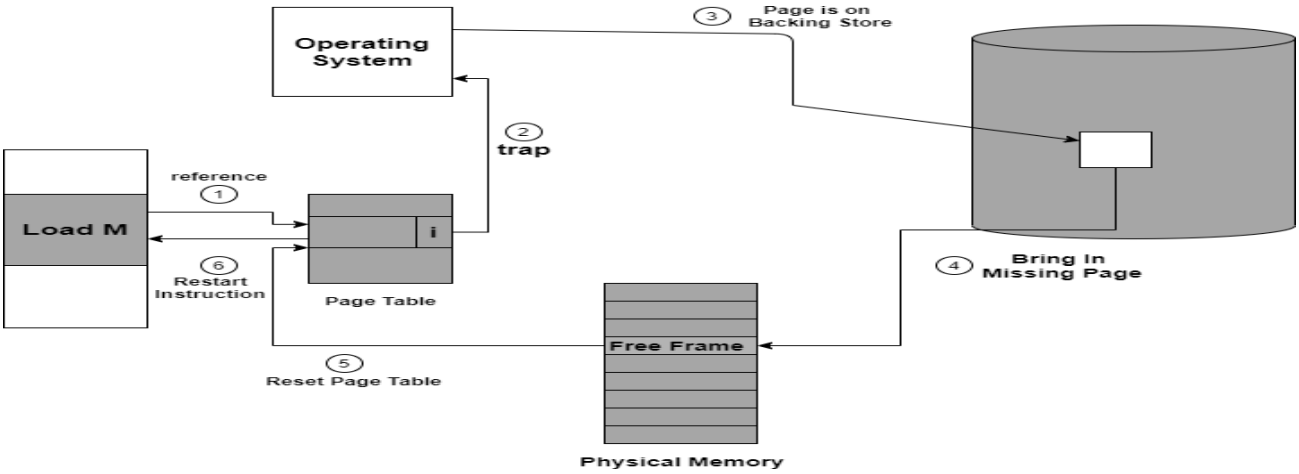
And these free pages are allocated typically when the stack/heap for a process must expand or when there are copy-on-write pages to manage.

These pages are typically allocated using the technique that is known as **Zero-fill-on-demand**. And the **Zero-fill-on-demand** pages are zeroed-out before being allocated and thus erasing the previous content.

Page Fault in Operating System

In this tutorial, we will be covering the concept of page fault and how to handle page fault in the

Operating System.



VI PAGE FAULT IN OPERATING SYSTEM

Page fault dominates more like an error. It mainly occurs when any program tries to access the data or the code that is in the address space of the program, but that data is not currently located in the RAM of the system.

- So basically when the page referenced by the CPU is not found in the main memory then the situation is termed as Page Fault.
- Whenever any page fault occurs, then the required page has to be fetched from the secondarymemory into the main memory.

  In case if the required page is not loaded into the memory, then a page fault trap arises

  The page fault mainly generates an exception, which is used to notify the operating system that it must have to retrieve the "pages" from the virtual memory in order to continue the execution. Once all the data is moved into the physical memory the program continues its execution normally. The Page fault process takes place in the background and thus goes unnoticed by the user.
- The hardware of the computer tracks to the kernel and the program counter (PC) is generallysaved on the stack.CPU registers store the information of the current state of instruction.
- An assembly program is started that usually saves the general registers and also saves the othervolatile information to prevent the OS from destroying it.

Handling the Page Fault

Given below is the simple procedure to handle the page fault:

Figure: Steps to Handle the Page fault

If you will access a page that is marked as invalid then it also causes a Page Fault. Then the Paging hardware during translating the address through the page table will notice that the invalid bit is set that will cause a trap to the Operating system.

This trap is mainly the result of the failure of the Operating system in order to bring the desired page into memory.

Let us understand the procedure to handle the page fault as shown with the help of the above diagram:

1. First of all, internal table(that is usually the process control block) for this process in order to determine whether the reference was valid or invalid memory access.
2. If the reference is invalid, then we will terminate the process. If the reference is valid, but wehave not bought in that page so now we just page it in.
3. Then we **locate the free frame lis**t in order to find the free frame.
4. Now a disk operation is scheduled in order to read the **desired page into the newly allocatedfram**e.
5. When the disk is completely read, then the **internal table is modified** that is kept with the process, and the page table that mainly indicates the page is now in memory.
6. Now we will restart the instruction that was interrupted due to the trap. Now the process canaccess the page as though it had always been in memory.

Page Replacement Algorithms in Operating System

In this tutorial, we will be covering the concept of Page replacement and its algorithms in theOperating system.

As studied in Demand Paging, only certain pages of a process are loaded initially into the memory. This allows us to get more processes into memory at the same time. but what happens when a processrequests for more pages and no free memory is available to bring them in. Following steps can be taken to deal with this problem :

1. Put the process in the wait queue, until any other process finishes its execution thereby freeingframes.
2. Or, remove some other process completely from the memory to free frames.
3. Or, find some pages that are not being used right now, move them to the disk to get freeframes. This technique is called **Page replacement** and is most commonly used.

In this case, if a process requests a new page and supposes there are no free frames, then the Operating system needs to decide which page to replace. The operating system must use any page replacement

algorithm in order to select the victim frame. The Operating system must then write the victim frame to the disk then read the desired page into the frame and then update the page tables. And all these require double the disk access time.

- Page replacement prevents the over-allocation of the memory by modifying the page-faultservice routine.
- To reduce the overhead of page replacement a **modify bit (dirty bit)** is used in order toindicate whether each page is modified.
- This technique provides complete separation between logical memory and physical memory.

Page Replacement in OS

In Virtual Memory Management, Page Replacement Algorithms play an important role. The main objective of all the **Page replacement policies** is to decrease the maximum number of **page faults**.

Page Fault – It is basically a memory error, and it occurs when the current programs attempt to access the memory page for mapping into virtual address space, but it is unable to load into the physical memory then this is referred to as Page fault.

Basic Page Replacement Algorithm in OS

Page Replacement technique uses the following approach. If there is no free frame, then we will find the one that is not currently being used and then free it. A-frame can be freed by writing its content toswap space and then change the page table in order to indicate that the page is no longer in the memory.

1. First of all, find the location of the desired page on the disk.
2. Find a free Frame: a) If there is a free frame, then use it. b) If there is no free frame then make use of the page-replacement algorithm in order to select the victim frame. c) Then after that write the victim frame to the disk and then make the changes in the page table and frame table accordingly.
3. After that read the desired page into the newly freed frame and then change the page and frame tables.
4. Restart the process.

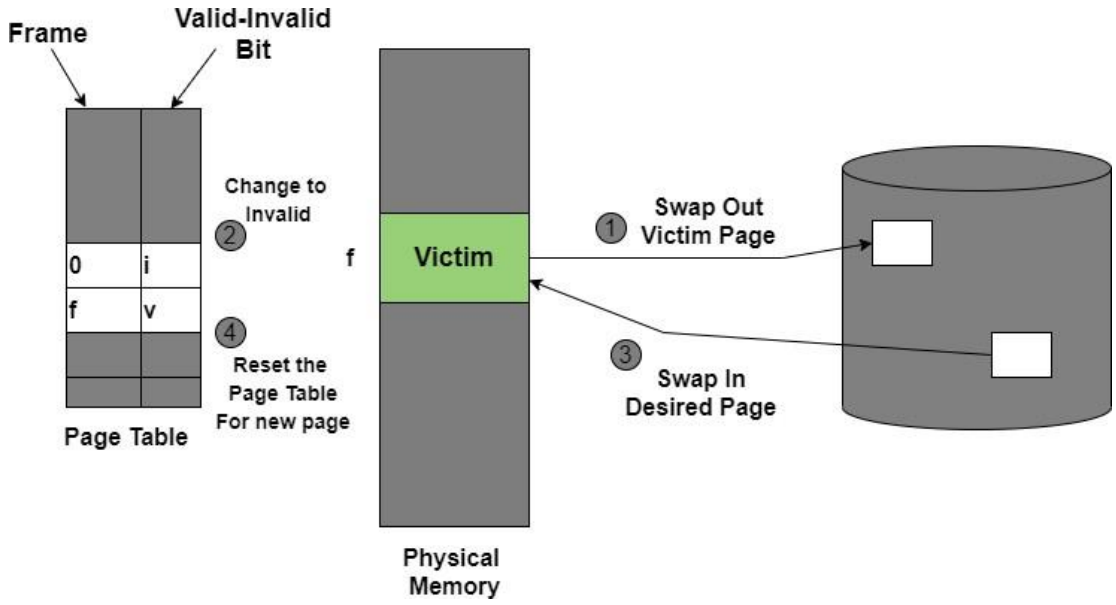
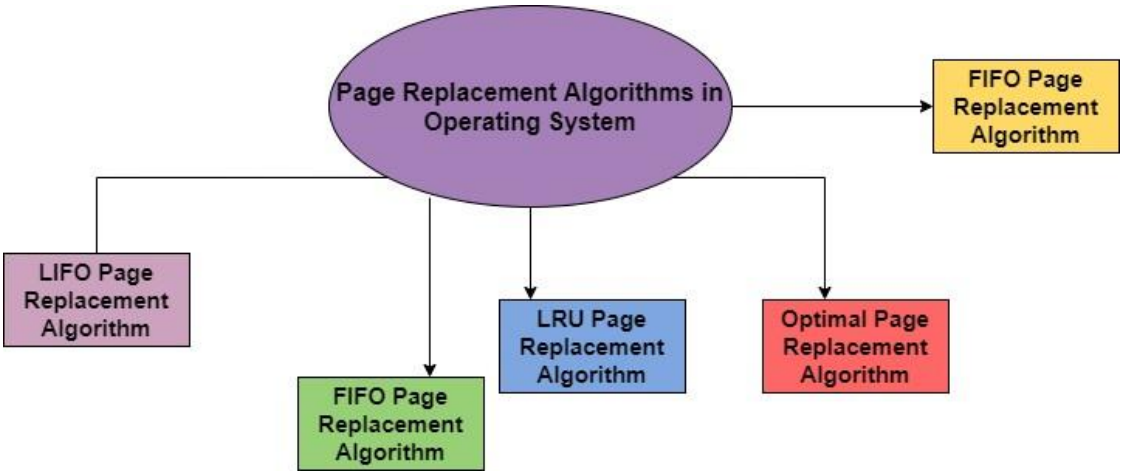


Figure: Page Replacement

Page Replacement Algorithms in OS

This algorithm helps to decide which pages must be swapped out from the main memory in order to create a room for the incoming page. This Algorithm wants the lowest page-fault rate.

Various Page Replacement algorithms used in the Operating system are as follows;



Let us discuss all algorithms one by one in the upcoming sections:

1. *FIFO Page Replacement Algorithm*

It is a very simple way of Page replacement and is referred to as First in First Out. This algorithm mainlyreplaces the oldest page that has been present in the main memory for the longest time.

- This algorithm is implemented by keeping the track of all the pages in the queue.

- As new pages are requested and are swapped in, they are added to the tail of a queue and thepage which is at the head becomes the victim.
- This is not an effective way of page replacement but it can be used for small systems.

Advantages

- This algorithm is simple and easy to use.
- FIFO does not cause more overhead.

Disadvantages

- This algorithm does not make the use of the frequency of **last used time rather** it just replacesthe Oldest Page.
- There is an increase in **page faults** as page frames increases.
- The performance of this algorithm is the worst.

2. *LIFO Page Replacement Algorithm*

This Page Replacement algorithm stands for "Last In First Out".This algorithm works in a similar way tothe LIFO principle.

- In this, the newest page is replaced which is arrived at last in the primary memory
- This algorithm makes use of the stack for monitoring all the pages.

3. *LRU Page Replacement Algorithm in OS*

This algorithm stands for "Least recent used" and this algorithm helps the Operating system to search those pages that are used over a short duration of time frame.

- The page that has not been used for the longest time in the main memory will be selected for replacement.
- This algorithm is easy to implement.
- This algorithm makes use of the counter along with the even-page.

Advantages of LRU

- It is an efficient technique.
- With this algorithm, it becomes easy to identify the faulty pages that are not needed for a longtime.
- It helps in Full analysis.

Disadvantages of LRU

- It is expensive and has more complexity.
- There is a need for an additional data structure.

4. *Optimal Page Replacement Algorithm*

This algorithm mainly replaces the page that will not be used for the longest time in the future. The practical implementation of this algorithm is not possible.

Practical implementation is not possible because we cannot predict in advance those pages thatwill not be used for the longest time in the future.

- This algorithm leads to less number of page faults and thus is the best-known algorithmAlso, this algorithm can be used to measure the performance of other algorithms.

Advantages of OPR

- This algorithm is easy to use.
- This algorithm provides excellent efficiency and is less complex.
- For the best result, the implementation of data structures is very easy

Disadvantages of OPR

- In this algorithm future awareness of the program is needed.
- Practical Implementation is not possible because the operating system is unable to track thefuture request

5. *Random Page Replacement Algorithm*

As indicated from the name this algorithm replaces the page randomly. This Algorithm can work likeany other page replacement algorithm that is LIFO, FIFO, Optimal, and LRU.