Assignment 1

---------------------------

Batch: Sep21

Subject: OPERATING SYSTEM

Submission Date: 10/10/2021

**9. Explain a shell script?**

Ans : A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter. The different operations performed by shell scripts are program execution, file manipulation and text printing.

Types of Shells

Bourne Shell

This is default shell for version 7 Unix. The character $ is the default prompt for the bourne shell. The different subcategories in this shell are Korn shell, Bourne Again shell, POSIX shell etc.

### C Shell

This is a Unix shell and a command processor that is run in a text window. The character % is the default prompt for the C shell. File commands can also be read easily by the C shell, which is known as a script.

The capabilities of shell script are:-

Batch jobs

Several commands entered in command line can be executed automatically using shell scripting.

Programming

The various methods, arrays, variables, comments

Generalisation , shortcuts

**10. Explain a page fault?**

Ans : 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 secondary memory 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 generally saved 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 other volatile information to prevent the OS from destroying it.

Handling the Page Fault

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

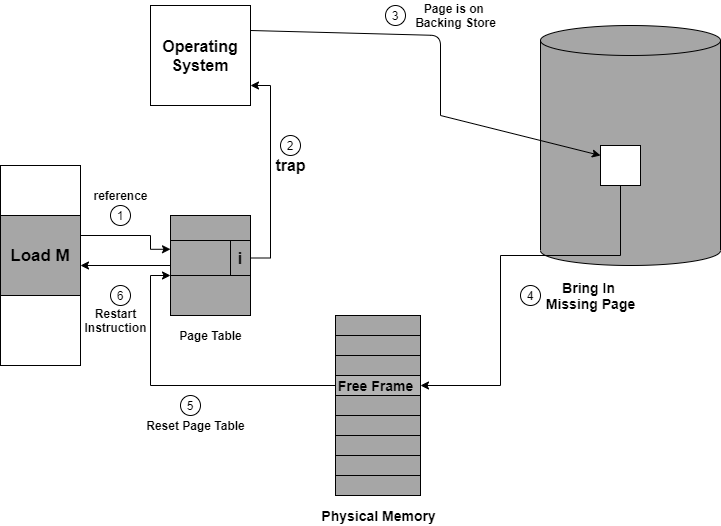


Figure: Steps to Handle the Page fault

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 we have not bought in that page so now we just page it in.
3. Then we locate the free frame list in order to find the free frame.
4. Now a disk operation is scheduled in order to read the desired page into the newly allocated frame.
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 can access the page as though it had always been in memory.

**11. Explain a deadlock?**

Ans: Every process needs some resources to complete its execution. However, the resource is granted in a sequential order.

1. The process requests for some resource.
2. OS grant the resource if it is available otherwise let the process waits.
3. The process uses it and release on the completion.

A Deadlock is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.

Let us assume that there are three processes P1, P2 and P3. There are three different resources R1, R2 and R3. R1 is assigned to P1, R2 is assigned to P2 and R3 is assigned to P3.

After some time, P1 demands for R1 which is being used by P2. P1 halts its execution since it can't complete without R2. P2 also demands for R3 which is being used by P3. P2 also stops its execution because it can't continue without R3. P3 also demands for R1 which is being used by P1 therefore P3 also stops its execution.

In this scenario, a cycle is being formed among the three processes. None of the process is progressing and they are all waiting. The computer becomes unresponsive since all the processes got blocked.

## os Deadlock

**12. Define the necessary conditions for deadlock?**

## Ans: Necessary conditions for Deadlocks

1. Mutual Exclusion

A resource can only be shared in mutually exclusive manner. It implies, if two process cannot use the same resource at the same time.

1. Hold and Wait

A process waits for some resources while holding another resource at the same time.

1. No pre-emption

The process which once scheduled will be executed till the completion. No other process can be scheduled by the scheduler meanwhile.

1. Circular Wait

All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.