**Page Fault-**

* Paging is a non-contiguous memory allocation technique.
* In a paging scheme, a process is divided into several pages.
* The pages are then stored in different frames of the main memory.

**Page Fault-**

* When a page referenced by the CPU is not found in the main memory, it is called as a **page fault**.
* When a page fault occurs, the required page has to be fetched from the secondary memory into the main memory.

# Performance of Demand Paging

* A page fault occurs when the referenced page is not found in the main memory.
* Page fault handling routine is executed on the occurrence of page fault.
* The time taken to service the page fault is called as **page fault service time (**Page fault overhead + swap page in +swap page out + restart overhead**)**.

**Effective Access time-**

**Page Fault rate 0<=P<=1.0**

**If P=0, no page faults**

**If P=1, every reference is a page fault**

Effective memory access time (EAT)=(1-P) \* Memory access time+

P\*(Page fault overhead + swap page in + swap page out + restart overhead)

Let the page fault service time be 10 ms in a computer with average memory access time being 20 ns. If one page fault is generated for every 106 memory accesses, what is the effective access time for the memory?

Effective memory access time= (1/106 )\*10ms+(1-(1/106 ))\*20ns

=Approximately 30ns (29.99998ns)

**Translating Logical Address into Physical Address-**

In a paging scheme using TLB with possibility of page fault,

The logical address generated by the CPU is translated into the physical address using the following steps-

## ****Step-01:****

CPU generates a logical address consisting of two parts-

1. Page Number
2. Page Offset

## ****Step-02:****

* TLB is checked to see if it contains an entry for the referenced page number.
* The referenced page number is compared with the TLB entries all at once.

Now, two cases are possible-

### ****Case-01: If there is a TLB hit-****

* If TLB contains an entry for the referenced page number, a TLB hit occurs.
* In this case, TLB entry is used to get the frame number for the referenced page number.

### ****Case-02: If there is a TLB miss-****

* If TLB does not contain an entry for the referenced page number, a TLB miss occurs.
* In this case, page table is used to get the frame number for the referenced page number.
* The valid / invalid bit of the page table entry indicates whether the referenced page is present in the main memory or not.

Now, two cases are possible-

### ****Case-01: If Valid / Invalid Bit is Set to 1-****

* If valid / invalid bit is set to 1, it indicates that the page is present in the main memory.
* Then, page table is used to get the frame number for the referenced page number.
* Then, TLB is updated with the page number and its frame number for future references.

### ****Case-02: If Valid / Invalid Bit is Set to 0-****

* If valid / invalid bit is set to 0, it indicates that the page is not present in the main memory.
* A page fault occurs.
* The occurrence of page fault calls the page fault interrupt which executes the page fault handling routine.

### ****Page Fault Handling Routine-****

The following sequence of events take place-

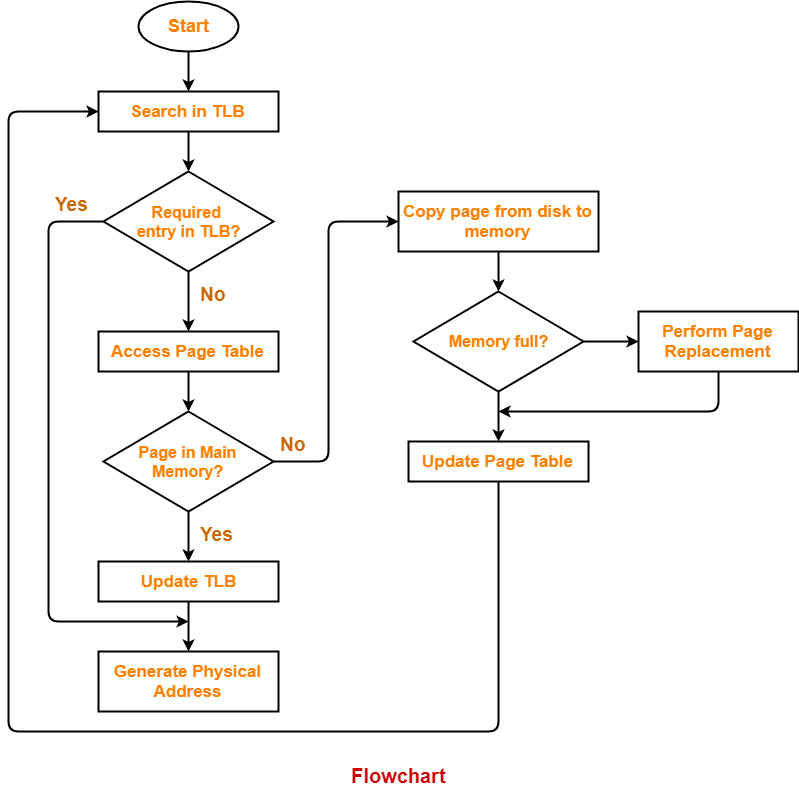
* The currently running process is stopped and context switching occurs.
* The referenced page is copied from the secondary memory to the main memory.
* If the main memory is already full, a page is replaced to create a room for the referenced page.
* After copying the referenced page successfully in the main memory, the page table is updated.
* When the execution of process is resumed, step-02 repeats.

**Step-03:**

* After the frame number is obtained, it is combined with the page offset to generate the physical address.
* Then, physical address is used to read the required word from the main memory.

**Flowchart-**

The following flowchart illustrates the above steps of translating logical address into physical address-

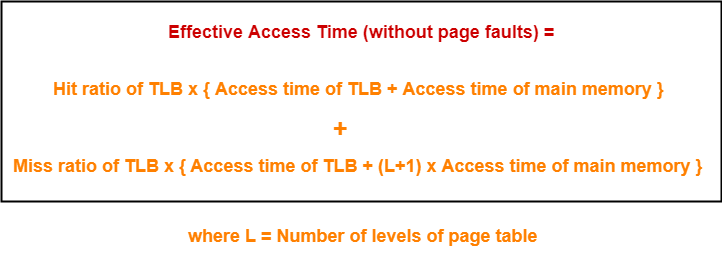


**Page Fault Service Time-**

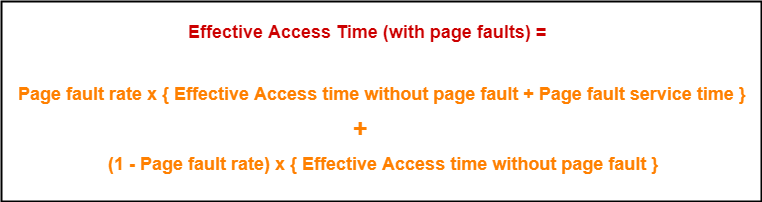
* The time taken by the page fault handling routine to service the page fault is called as page fault service time.
* Page fault service time is much greater than main memory access time.
* It increases the effective access time.

**Effective Access time-**

In a multilevel paging scheme using TLB without any possibility of page fault, effective access time is given by-



In a multilevel paging scheme using TLB with a possibility of page fault, effective access time is given by-



## ****PRACTICE PROBLEMS BASED ON PAGE FAULTS IN OS-****

## ****Problem-01:****

Let the page fault service time be 10 ms in a computer with average memory access time being 20 ns. If one page fault is generated for every 106 memory accesses, what is the effective access time for the memory?

1. 21 ns
2. 30 ns
3. 23 ns
4. 35 ns

## ****Solution-****

Given-

* Page fault service time = 10 ms
* Average memory access time = 20 ns
* One page fault occurs for every 106 memory accesses

### ****Page Fault Rate-****

It is given that one page fault occurs for every 106 memory accesses.

Thus,

Page fault rate

= 1 / 106

= 10-6

### ****Effective Access Time With Page Fault-****

It is given that effective memory access time without page fault = 20 ns.

Now, substituting values in the above formula, we get-

Effective access time with page fault

= 10-6 x { 20 ns + 10 ms } + ( 1 – 10-6 ) x { 20 ns }

= 10-6 x 10 ms + 20 ns

= 10-5 ms + 20 ns

= 10 ns + 20 ns

= 30 ns

Thus, Option (B) is correct.

## ****Problem-02:****

Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then, a 99.99% hit ratio results in average memory access time of-

1. 1.9999 milliseconds
2. 1 millisecond
3. 9.999 microseconds
4. 1.9999 microseconds
5. None of these

## ****Solution-****

Given-

* Page fault service time = 10 msec
* Average memory access time = 1 μsec
* Hit ratio = 99.99% = 0.9999

### ****Page Fault Rate-****

Page fault rate

= 1 – Hit ratio

= 1 – 0.9999

= 0.0001

### ****Effective Access Time With Page Fault-****

It is given that effective memory access time without page fault = 1 μsec.

Substituting values in the above formula, we get-

Effective access time with page fault

= 0.0001 x { 1 μsec + 10 msec } + 0.99999 x 1 μsec

= 0.0001 μsec + 0.001 msec + 0.9999 μsec

= 1 μsec + 0.001 msec

= 1 μsec + 1 μsec

= 2 μsec or 0.002 msec

Thus, Option (E) is correct.

## ****Problem-03:****

If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instruction is-

1. i + j / k
2. i + j x k
3. (i + j) / k
4. (i + j) x k

## ****Solution-****

Given-

* Page fault service time = j μsec
* Average memory access time = i μsec
* One page fault occurs every k instruction

### ****Page Fault Rate-****

It is given that one page fault occurs every k instruction.

Thus,

Page fault rate

= 1 / k

### ****Effective Access Time With Page Fault-****

It is given that effective memory access time without page fault = i μsec

Now, substituting values in the above formula, we get-

Effective access time with page fault

= (1 / k) x { i μsec + j μsec } + ( 1 – 1 / k) x { i μsec }

= j / k μsec + i μsec

= i + j / k μsec

Thus, Option (A) is correct.

## ****Problem-04:****

Consider a system with a two-level paging scheme in which a regular memory access takes 150 nanoseconds and servicing a page fault takes 8 milliseconds. An average instruction takes 100 nanoseconds of CPU time and two memory accesses. The TLB hit ratio is 90% and the page fault rate is one in every 10,000 instructions. What is the effective average instruction execution time?

1. 645 nanoseconds
2. 1050 nanoseconds
3. 1215 nanoseconds
4. 1230 nanoseconds
5. None of these

## ****Solution-****

Given-

* Number of levels of page table = 2
* Main memory access time = 150 ns
* Page fault service time = 8 msec
* Average instruction takes 100 ns of CPU time and 2 memory accesses
* TLB Hit ratio = 90% = 0.9
* Page fault rate = 1 / 104 = 10-4

Assume TLB access time = 0 since it is not given in the question.

Also, TLB access time is much less as compared to the memory access time.

### ****Effective Access Time Without Page Fault-****

Substituting values in the above formula, we get-

Effective memory access time without page fault

= 0.9 x { 0 + 150 ns } + 0.1 x { 0 + (2+1) x 150 ns }

= { 0.9 x 150 ns } + { 0.1 x 450 ns }

= 135 ns + 45 ns

= 180 ns

### ****Effective Access Time With Page Fault-****

Substituting values in the above formula, we get-

Effective access time with page fault

= 10-4 x { 180 ns + 8 msec } + (1 – 10-4) x 180 ns

= 8 x 10-4 msec + 180 ns

= 8 x 10-7 sec + 180 ns

= 800 ns + 180 ns

= 980 ns

### ****Effective Average Instruction Execution Time-****

Effective Average Instruction Execution Time

= 100 ns + 2 x Effective memory access time with page fault

= 100 ns + 2 x 980 ns

= 100 ns + 1960 ns

= 2060 ns

Thus, Option (E) is correct.

## ****Problem-05:****

A demand paging system takes 100 time units to service a page fault and 300 time units to replace a dirty page. Memory access time is 1 time unit. The probability of a page fault is p. In case of a page fault, the probability of page being dirty is also p. It is observed that the average access time is 3 time units. Then the value of p is-

1. 0.194
2. 0.233
3. 0.514
4. 0.981
5. None of these

## ****Solution-****

Given-

* Page fault service time = 100 time units
* Time taken to replace dirty page = 300 time units
* Average memory access time = 1 time unit
* Page fault rate = p
* Probability of page being dirty = p
* Effective access time = 3 time units

Now, According to question-

3 time units = p x { 1 time unit + p x { 300 time units } + (1 – p) x { 100 time units } } + (1 – p) x { 1 time unit }

3 = p x { 1 + 300p + 100 – 100p } + (1 – p)

3 = p x { 101 + 200p } + (1 – p)

3 = 101p + 200p2 + 1 – p

3 = 100p + 200p2 + 1

200p2 + 100p – 2 = 0

On solving this quadratic equation, we get p = 0.019258

Thus, Option (E) is correct.