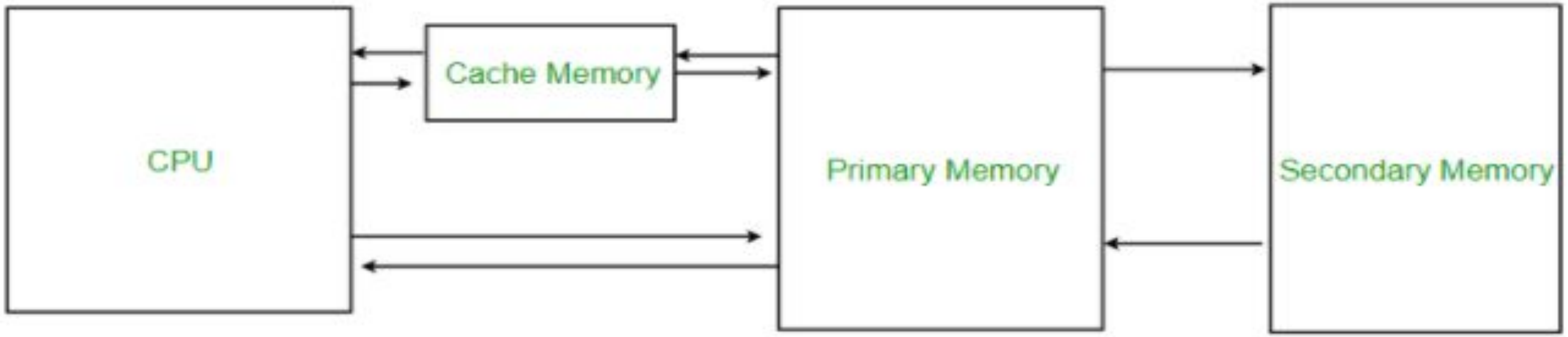
Cache Memory

Cache Memory is a special very high-speed memory. It is used to speed up and synchronizing with high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU registers. Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.



Levels of memory:

- **Level 1 or Register –** It is a type of memory in which data is stored and accepted that are immediately stored in CPU. Most commonly used register is accumulator, Program counter, address register etc.
- **Level 2 or Cache memory –** It is the fastest memory which has faster access time where data is temporarily stored for faster access.
- **Level 3 or Main Memory –** It is memory on which computer works currently. It is small in size and once power is off data no longer stays in this memory.
- **Level 4 or Secondary Memory –**
  It is external memory which is not as fast as main memory but data stays permanently in this memory.

Cache Performance:

When the processor needs to read or write a location in main memory, it first checks for a corresponding entry in the cache.

- If the processor finds that the memory location is in the cache, a cache hit has occurred and data is read from cache
- If the processor does not find the memory location in the cache, a cache miss has occurred. For a cache miss, the cache allocates a new entry and copies in data from main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called Hit ratio.

Hit ratio = hit / (hit + miss) = no. of hits/total accesses

Data writing policies

Data can be written to memory using a variety of techniques, but the two main ones involving cache memory are:

- **Write-through.** Data is written to both the cache and main memory at the same time.
- **Write-back.** Data is only written to the cache initially. Data may then be written to main memory, but this does not need to happen and does not inhibit the interaction from taking place.

Cache Mapping:

There are three different types of mapping used for the purpose of cache memory which are as follows: Direct mapping, Associative mapping, and Set-Associative mapping.

Page Replacement Algorithms

1. First-In-First-Out (FIFO)

- Simplest Algorithm
- This algorithm associates with each page the time when that page was brought into the memory
- When a page must be replaced, the oldest page is chosen.
- It is easy to understand and program
- Its performance is not always good.

2. Optimal Algorithm

- This has the lowest page fault of all algorithms, and will never suffer from Balady’s anomaly
- Replace the page that will not be used for the longest period of time.

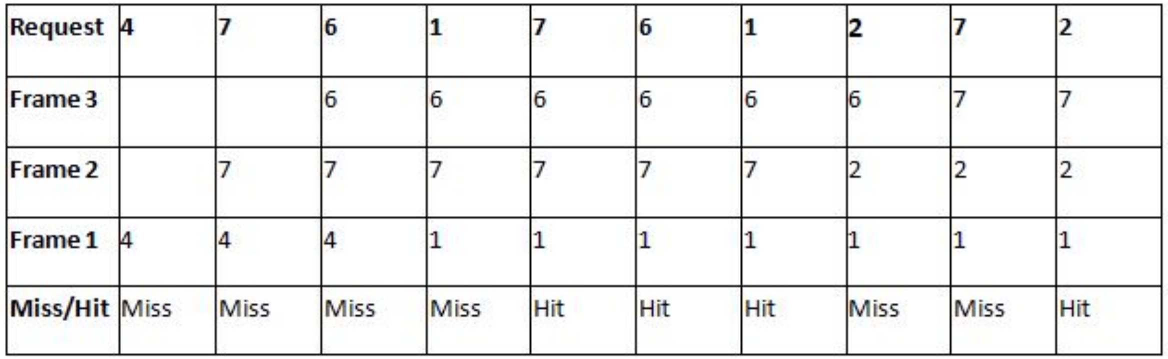
3. Least Recently Used (LRU) Algorithm

- Replace the page that has not been used for the longest period of time.

Reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2

3 frames (3 pages can be in memory at a time per process)

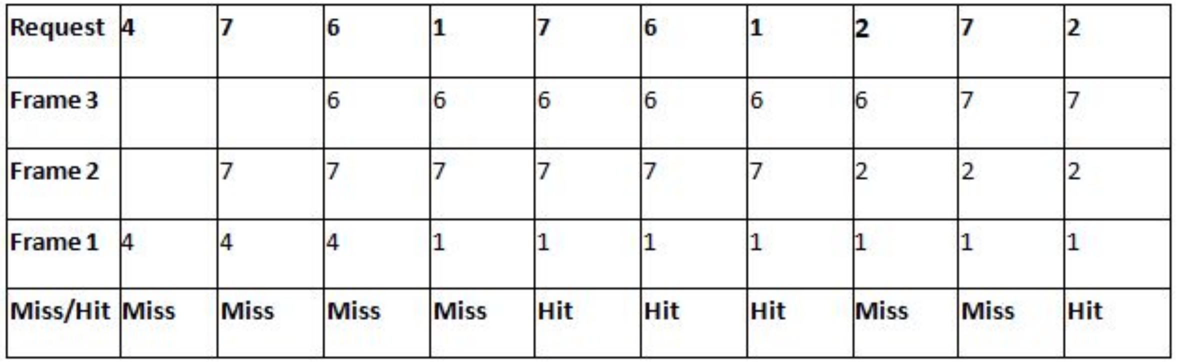
FIFO Page Replacement Algorithm



Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	6	7	7
Frame 2		7	7	7	7	7	7	2	2	2
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Miss	Hit

Number of Page Faults in FIFO = 6

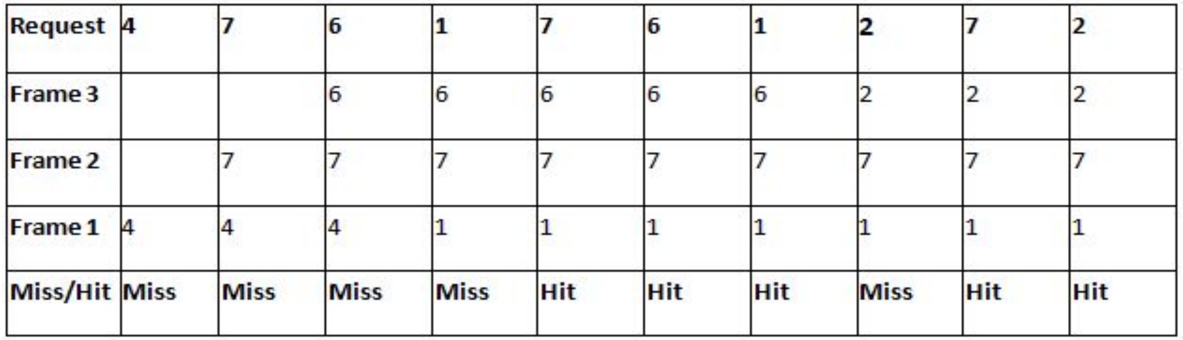
LRU Page Replacement Algorithm



Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	6	7	7
Frame 2		7	7	7	7	7	7	2	2	2
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Miss	Hit

Number of Page Faults in LRU = 6

Optimal Page Replacement Algorithm



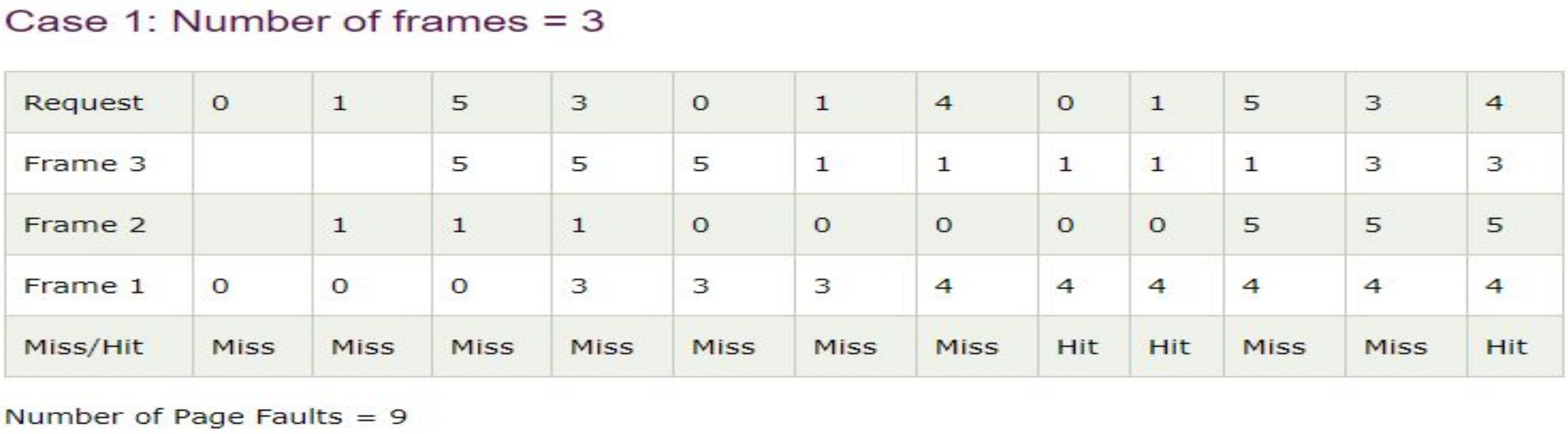
Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	2	2	2
Frame 2		7	7	7	7	7	7	7	7	7
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Hit	Hit

Number of Page Faults in Optimal = 5

Belady'sAnomaly

In the case of LRU and optimal page replacement algorithms, it is seen that the number of page faults will be reduced if we increase the number of frames. However, Balady found that, In FIFO page replacement algorithm, the number of page faults will get increased with the increment in number of frames . This is the strange behavior shown by FIFO algorithm in some of the cases . This is an Anomaly called as Belady’s Anomaly.

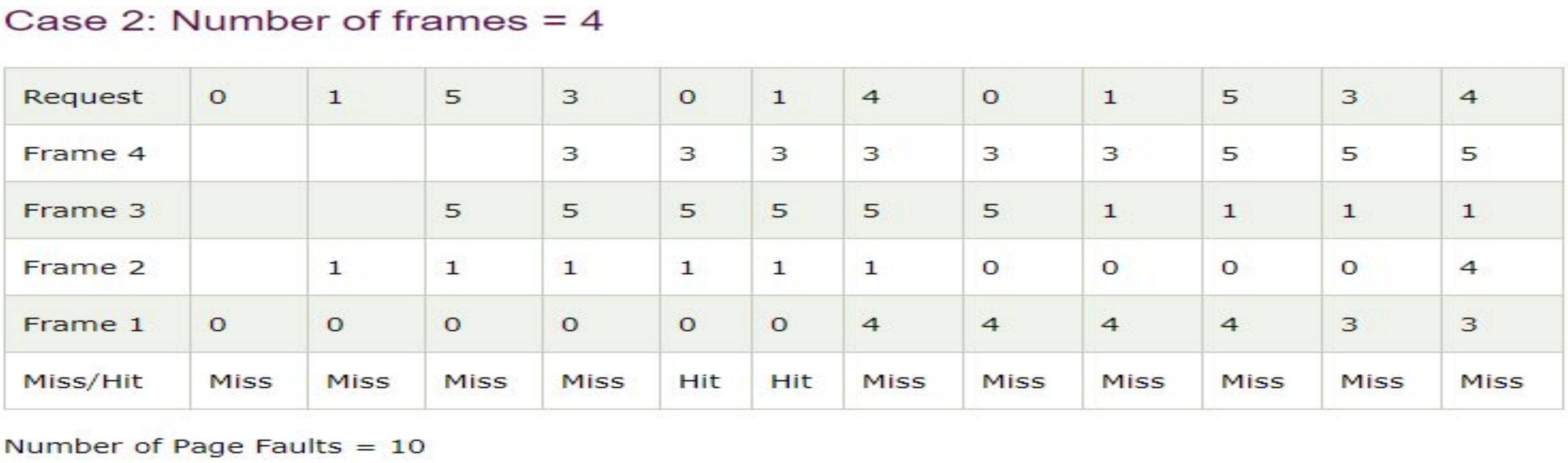
The reference String is given as 0 1 5 3 0 1 4 0 1 5 3 4. Let's analyze the behavior of FIFO algorithm in two cases.



Case 1: Number of frames = 3

Request	0	1	5	3	0	1	4	0	1	5	3	4
Frame 3			5	5	5	1	1	1	1	1	3	3
Frame 2		1	1	1	0	0	0	0	0	5	5	5
Frame 1	0	0	0	3	3	3	4	4	4	4	4	4
Miss/Hit	Miss	Miss	Miss	Miss	Miss	Miss	Miss	Hit	Hit	Miss	Miss	Hit

Number of Page Faults = 9



Case 2: Number of frames = 4

Request	0	1	5	3	0	1	4	0	1	5	3	4
Frame 4				3	3	3	3	3	3	5	5	5
Frame 3			5	5	5	5	5	5	1	1	1	1
Frame 2		1	1	1	1	1	1	0	0	0	0	4
Frame 1	0	0	0	0	0	0	4	4	4	4	3	3
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Miss	Miss	Miss	Miss	Miss	Miss

Number of Page Faults = 10

Reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2

4 frames (4 pages can be in memory at a time per process)

Exercise Operating System Concepts Apply FIFO page replacement, optimal page replacement, Least Recently Used (LRU) on following reference string (Assume 3 frames) 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

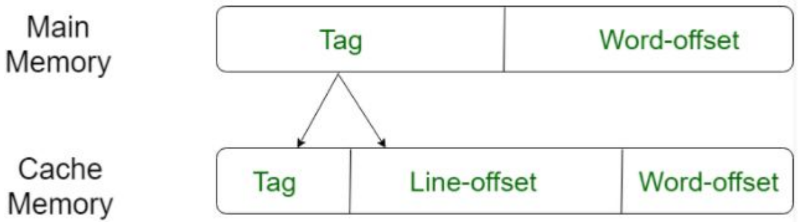
The transformation of data from main memory to cache memory is referred to as a ' **Mapping**' process. There are 3 types of mapping procedures are there for cache memory:

1. Associative Mapping
2. Direct Mapping
3. Set-Associative Mapping

Direct Mapping

In Direct mapping, assign each memory block to a specific line in the cache. If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory.

i = j modulo m
where
i=cache line number
j= main memory block number
m=number of lines in the cache



Associative Mapping

In this type of mapping, the associative memory is used to store content and addresses of the memory word. Any block can go into any line of the cache.

Set-associative Mapping

This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed. Set associative addresses the problem of possible thrashing in the direct mapping method. It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a *set*. Then a block in memory can map to any one of the lines of a specific set..Set-associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address. Set associative cache mapping combines the best of direct and associative cache mapping techniques.