Operating System-Memory Management

by

Vishal Singh
(Vishalchd11@yahoo.com)

Why Memory management ?

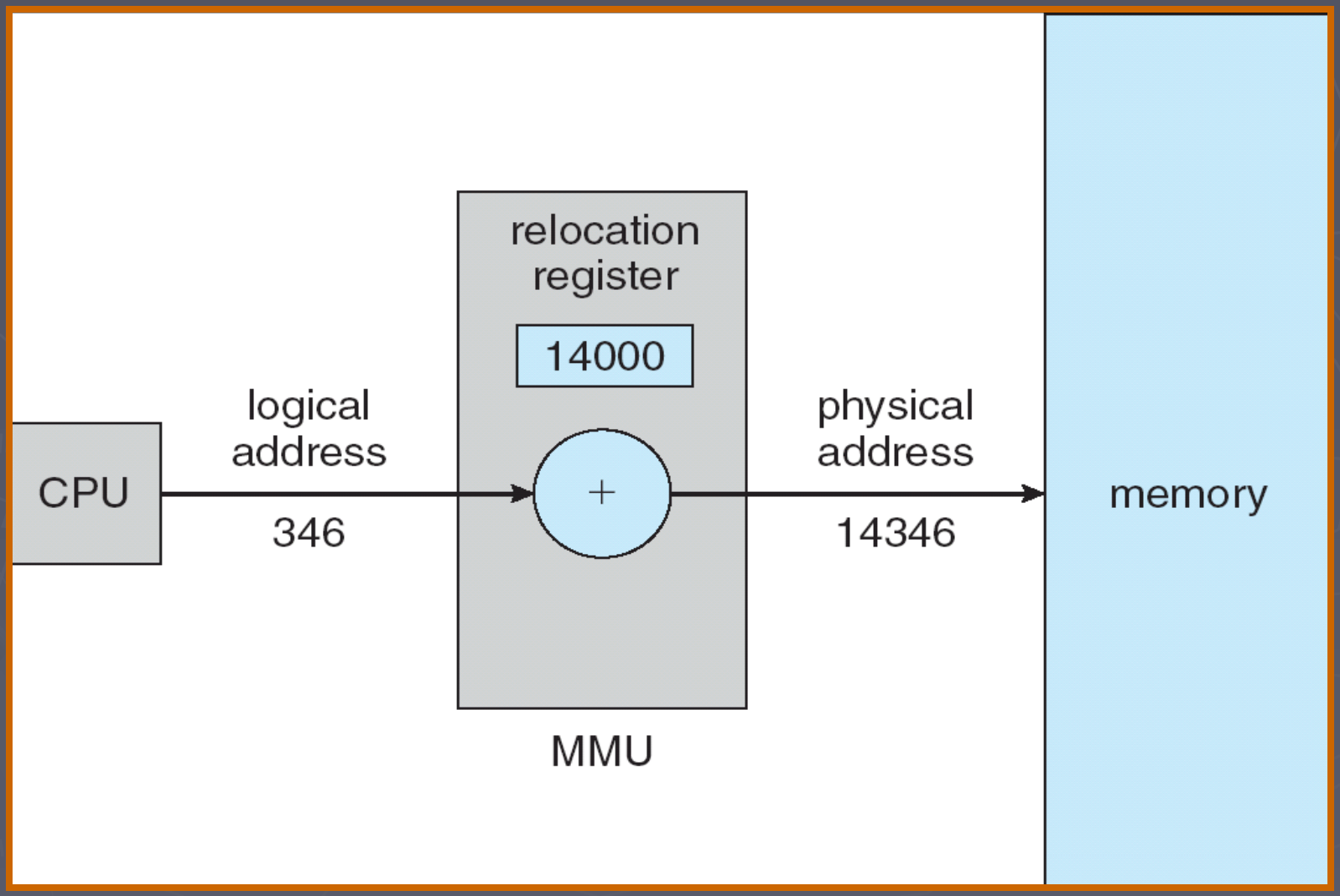
1. To ensure protection of different processes from each other (so that they do not interfere with each other's operations).
2. To place the programs in memory (such that memory is utilized to its fullest extent.

Logical vs. physical address space

- A logical address is the address of an instruction or data as used by a program.

- A physical address is the effective memory address of an instruction or data i.e. it is the address obtained after binding of logical addresses has been done.

Program relocation



Relocation register

- Relocation register is a special register in the CPU used for program relocation means mapping of logical addresses used by the program to physical addresses of the system's main memory

Storage Allocation and Management Techniques

The Storage allocation can be of two types:

(i) Contiguous storage allocation.

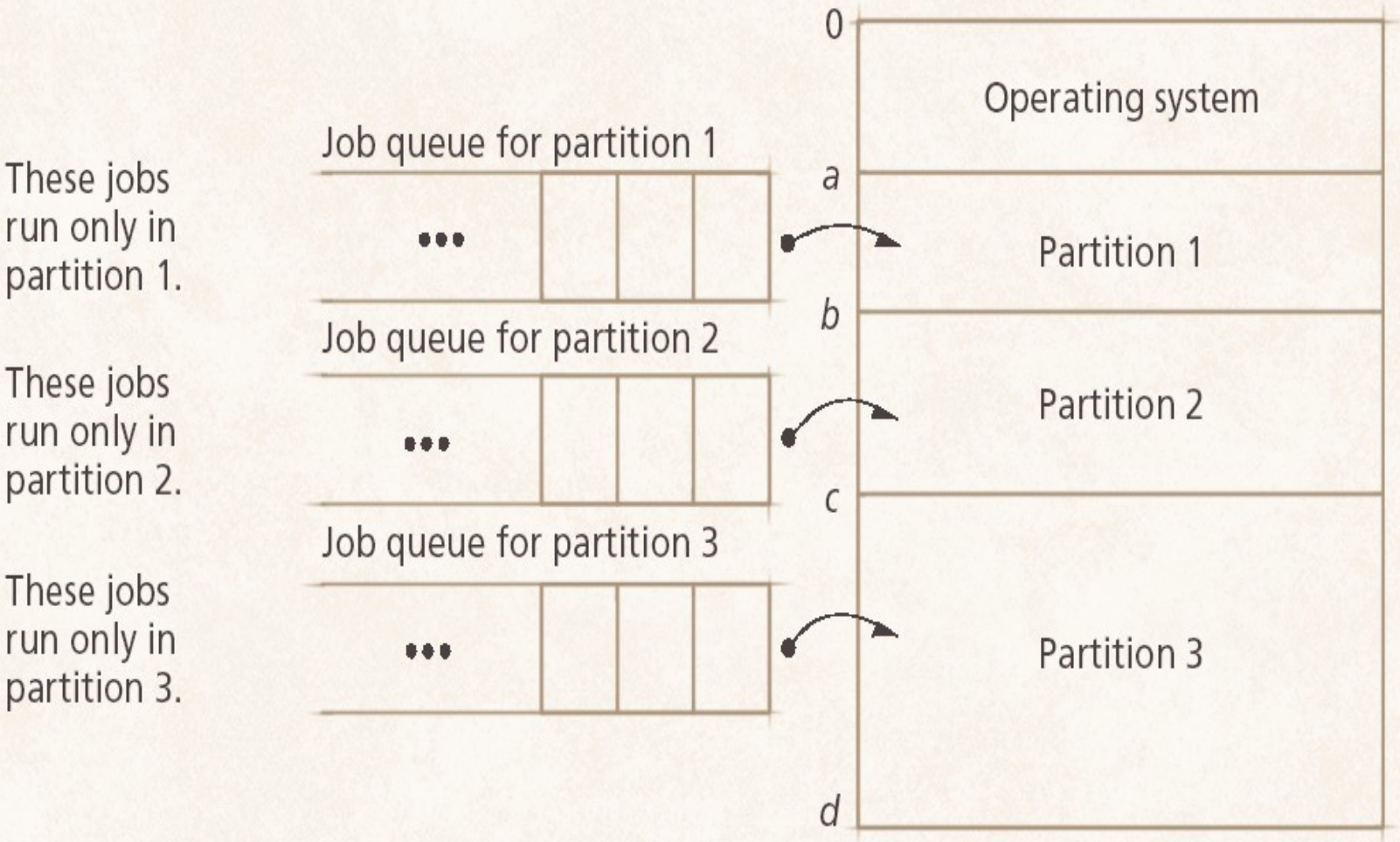
(ii) Non-contiguous storage allocation.

Contiguous Storage Allocation

- Contiguous storage allocation implies that a program's data and instructions are assured to occupy a single contiguous memory area.
- It is further subdivided into Fixed-partition storage allocation strategy and variable-partition storage allocation strategy.

1. Fixed-partition contiguous storage allocation

- processes with small address space use small partitions and processes with large address space use large partitions. This is known as fixed partition contiguous storage allocation.


These jobs run only in partition 1.
These jobs run only in partition 2.
These jobs run only in partition 3.
Job queue for partition 1
Job queue for partition 2
Job queue for partition 3
0
a
b
c
d
Operating system
Partition 1
Partition 2
Partition 3

2. Variable - partition contiguous storage allocation

This notion is derived from parking of vehicles on the sides of streets where the one who manages to enter will get the space. Two vehicles can leave a space between them that cannot be used by any vehicle. This means that whenever a process needs memory, a search for the space needed by it, is done. If contiguous space is available to accommodate that process, then the process is loaded into memory.

External Fragmentation

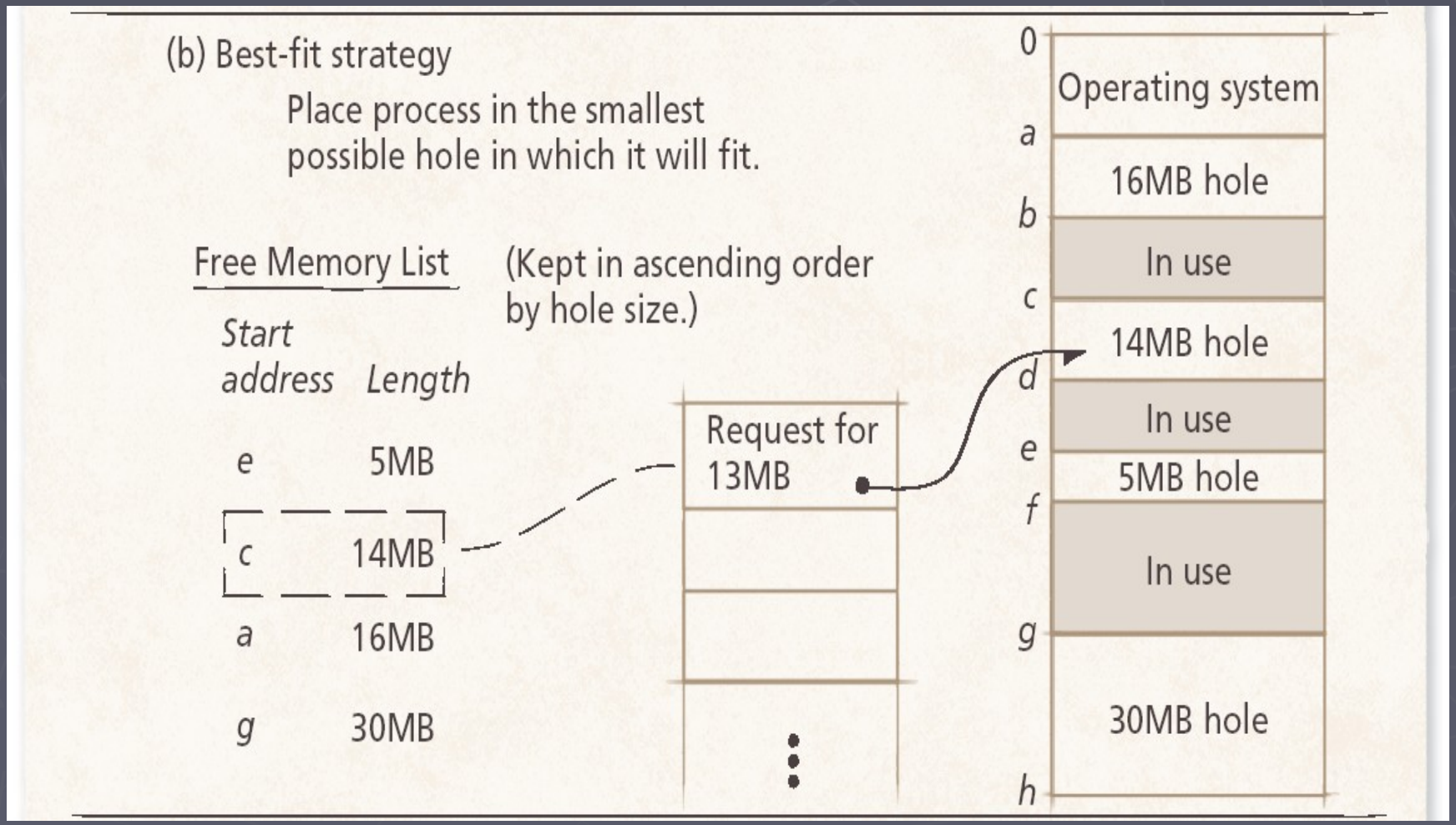
- This phenomenon of entering and leaving the memory can cause the formation of unusable memory holes (like the unused space between two vehicles). This is known as External Fragmentation.

Three strategies that can be used to allocate memory to Variable - partition

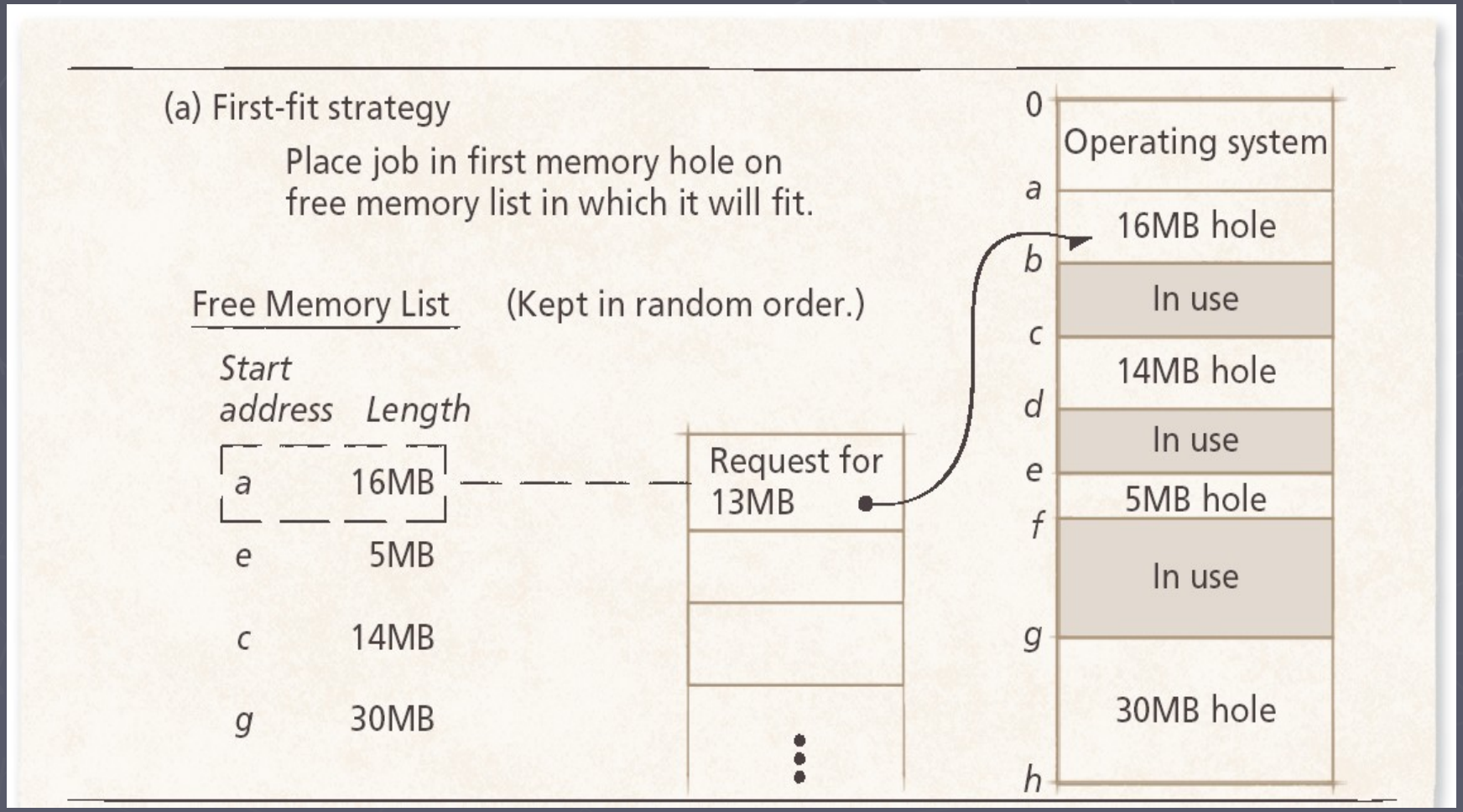
They are the following:

1. **Best - Fit**
2. **Worst - fit**
3. **First - fit**

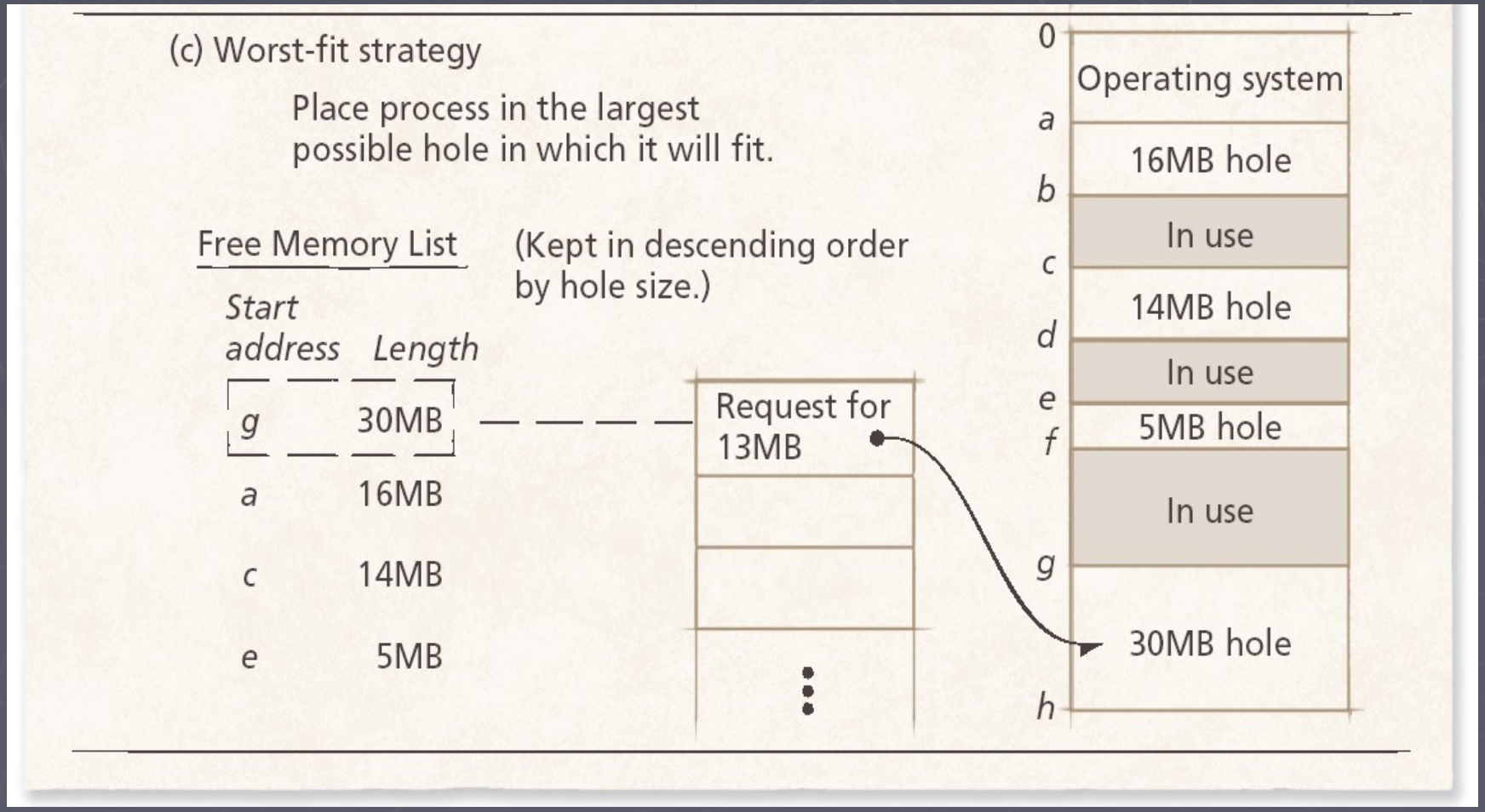
Best - Fit - chooses a partition that is smallest and whose size is greater than equal to k. It leaves small-sized unusable partitions or holes.



First - fit - chooses the first partition whose size is greater than equal to k.



Worst - fit - chooses the largest partition and allocates it to process p. It can leave bigger unusable partitions.



Compaction means to move the processes in the memory in such a way that scattered pieces of unused (free) memory can be placed together so that any other process demanding contiguous memory for it can use it.

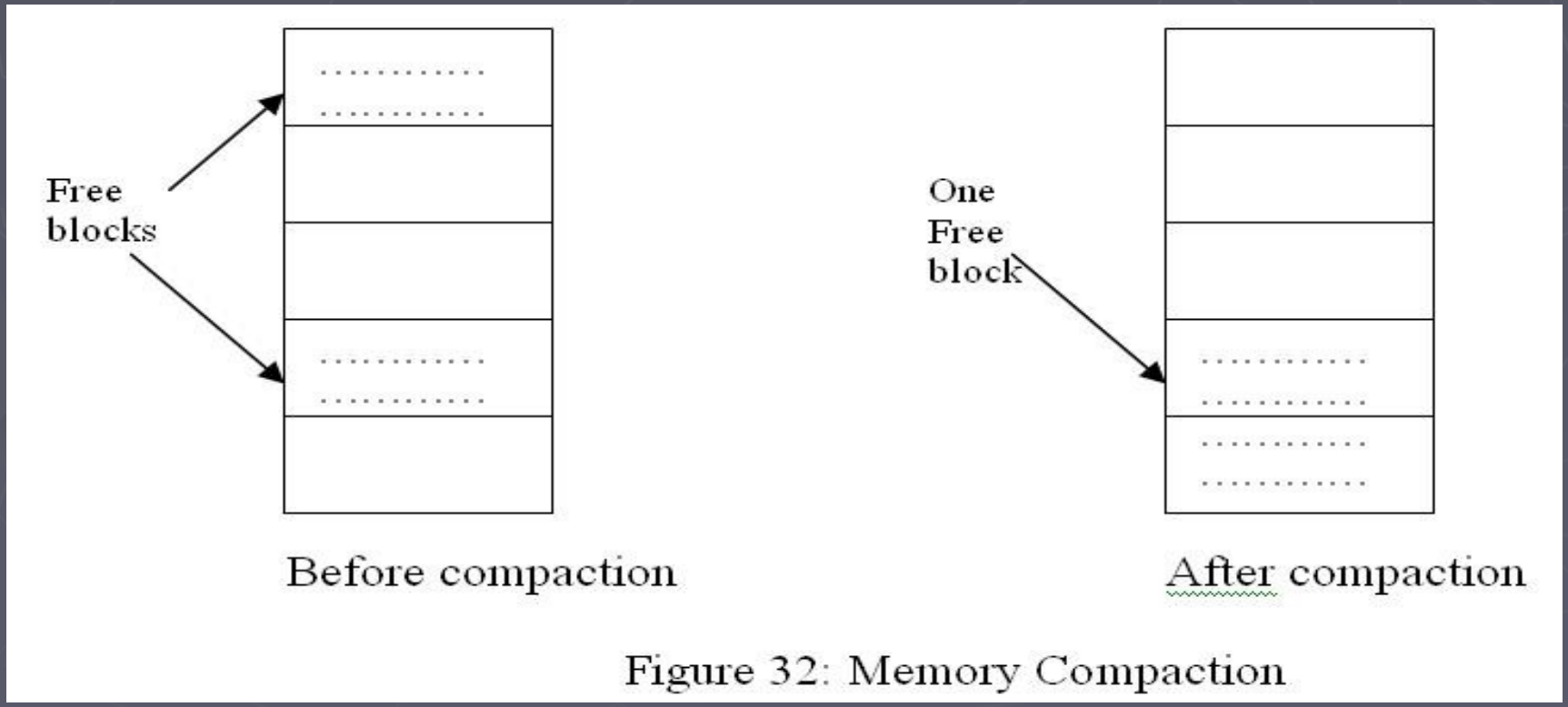


Figure 32: Memory Compaction

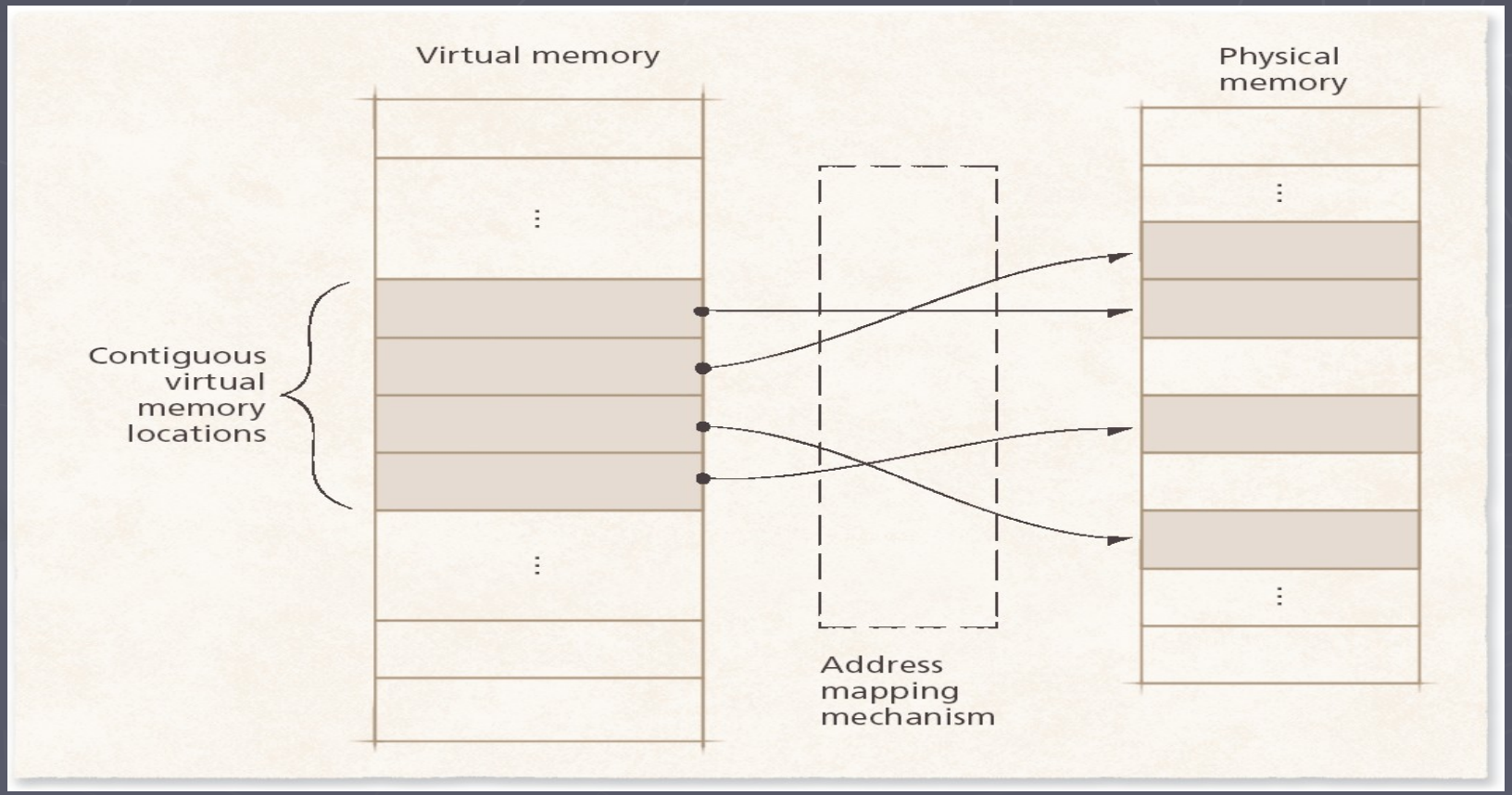
Non-contiguous Storage Allocation

- To resolve the **problem of external fragmentation** and to **enhance the degree of multiprogramming** to a greater extent, it was decided to sacrifice the simplicity of allocating contiguous memory to every process. It was decided to have a non-contiguous physical address space of a process so that a process could be allocated memory wherever it was available.

Non-contiguous Storage Allocation

There are 2 techniques for non-contiguous allocation:

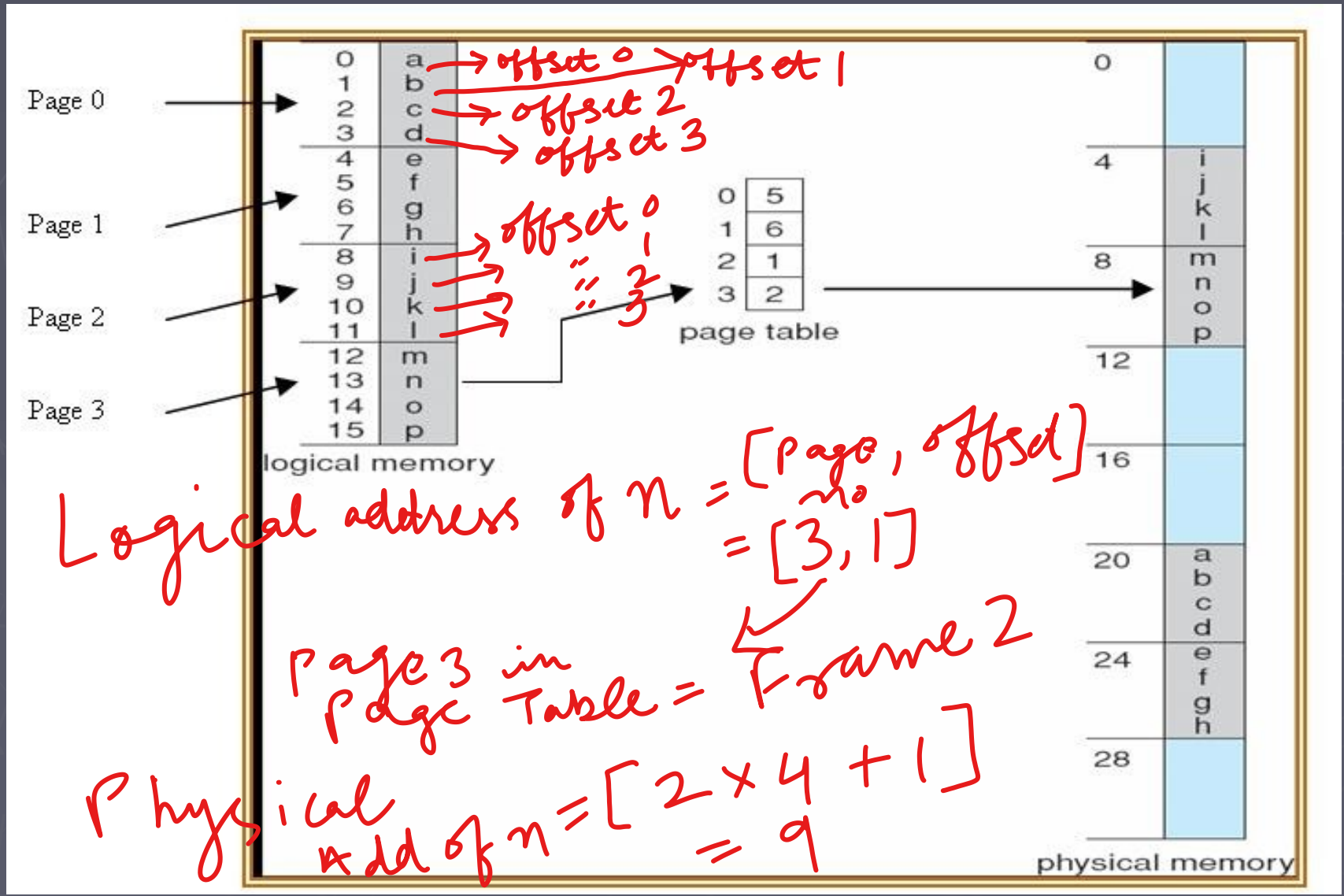
1. Paging
2. Segmentation



1. Paging

- In this approach, **physical memory is divided into fixed-size blocks called frames** and the **logical memory is divided into the fixed-sized blocks called pages.** The size of a page is same as that of frame. The key idea of this method is to place the pages of a process into the available frames of memory, whenever, this process is to be executed. The address mapping is done by Page table.

Paging example



It works as:

Translate logical address 13 = 01101 in binary

Offset = 01 in binary or 1 in decimal

Page no. = 011 in binary or 3 in decimal

Physical address = pageframe*pagesize + offset i.e. 2*4+1 = 9 (which is the real address of page containing character “n” in the main memory)

Physical addr = [frame No. × frame size] + offset

Problem with paging

- It uses a page table per process for translating logical to physical address space. A basic issue associated with the use of page table is, that the look up in the page table should be very fast (which is generally slow), as it is done on every memory reference.
- So to make the look up time less, caching is used. In this case, the specific cache device is called a translation look aside buffer (TLB).

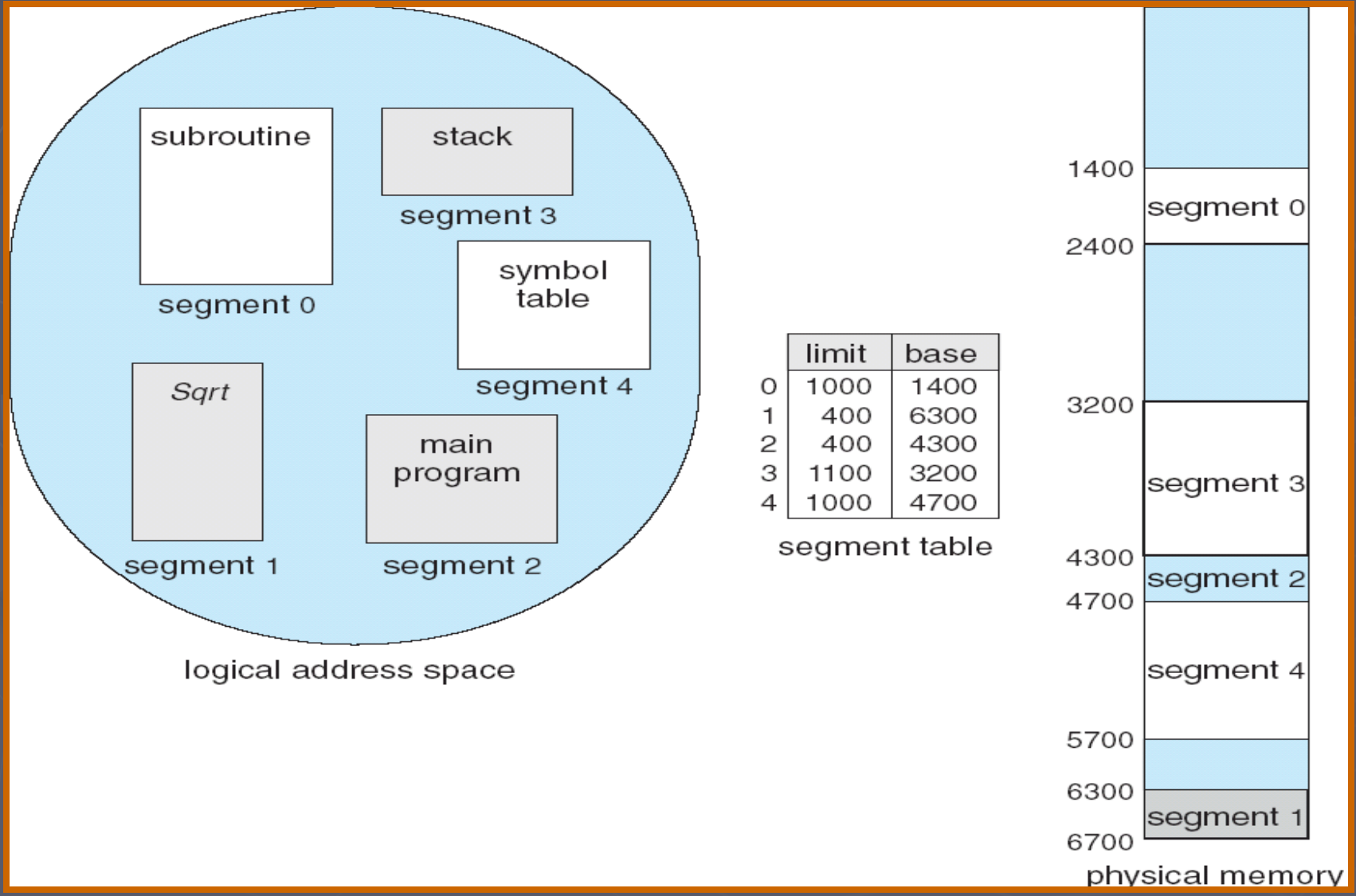
Translation look aside buffer (TLB).

- The TLB contains a set of entries, each of which contains a page number, the corresponding frame number, and the protection bits.
- So if we use TLB, then mapping is done by TLB and we generally find the entry in TLB but in case the entry is not found in TLB than we have a TLB miss and finally we go to page table.

Segmentation

- segmentation is another technique for the noncontiguous storage allocation. It is different from paging as it supports users' view of his program.
- For a programmer it might be more relevant to divide the logical address space of his program into variable sized segments (with respect to his view of main program, subroutines, data, etc.) than to divide it into fixed size pages. Such variable sized segments, which are a collection of logically related information, are the basis of segmentation technique.

Segmentation example

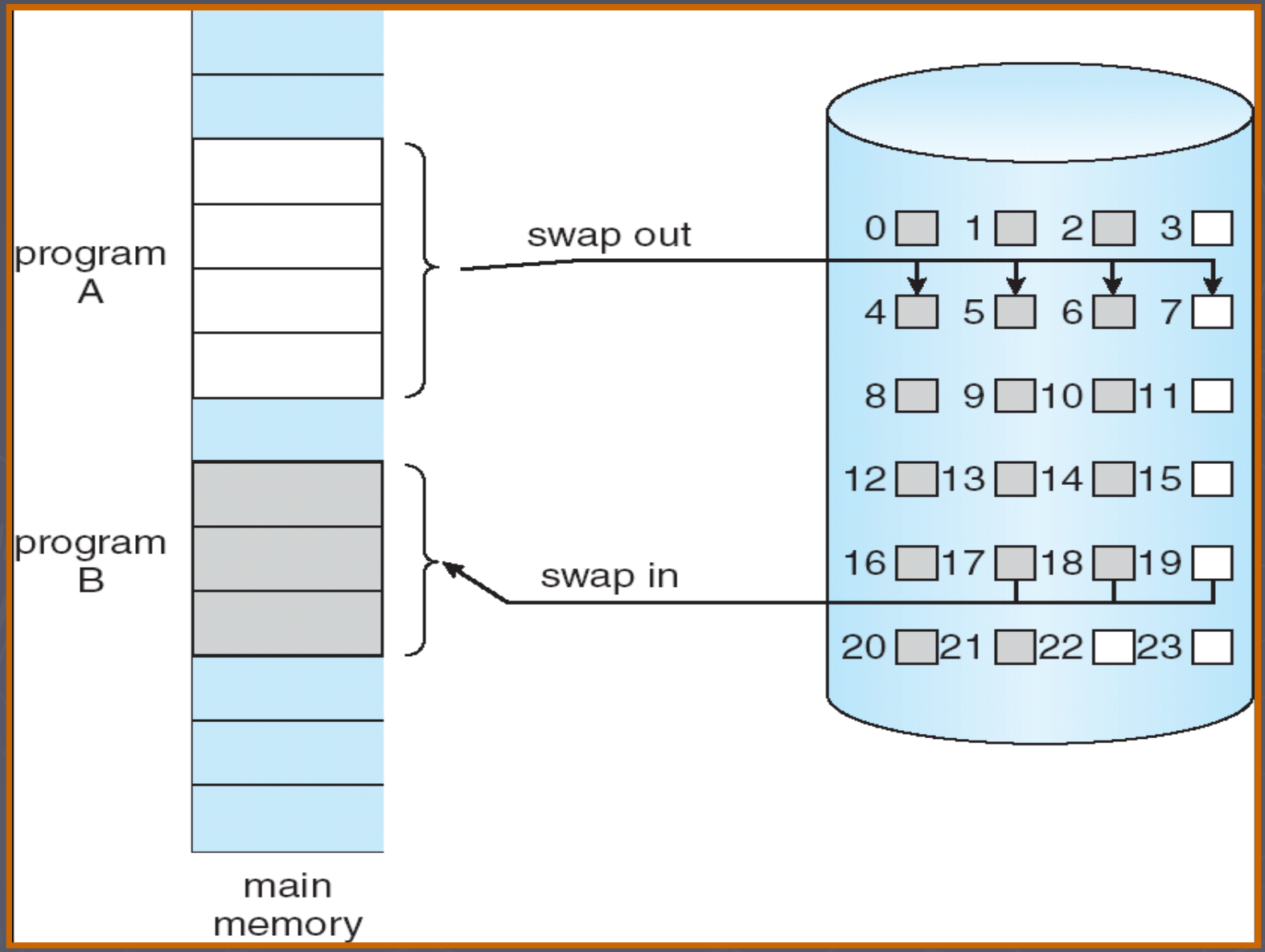


Virtual Memory

- Virtual memory is a technique of executing program instructions that may not fit entirely in system memory. This is done by calling instructions as and when the application requires it. Virtual memory is implemented by using secondary storage to augment the main memory. Data is transferred from secondary to main storage as and when necessary and the data modified is written back to the secondary storage according to a predetermined algorithm.

Swapping

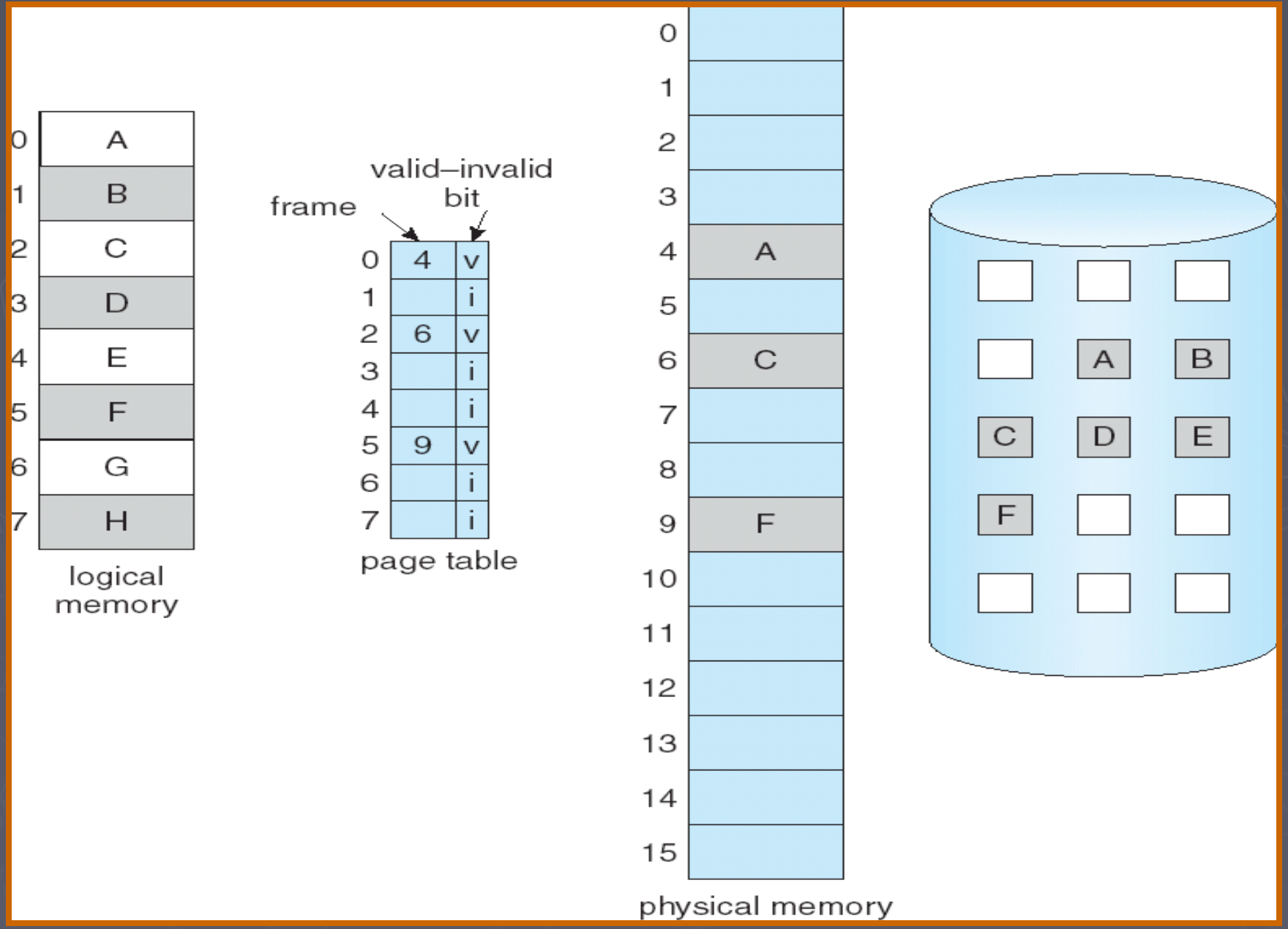
- Swapping is the technique of temporarily removing inactive programs from the main memory of a computer system. An inactive program is one that is neither executing nor performing an I/O operation.
- This may also happen when it is desired to place a higher-priority process in the memory. A lower priority process may be swapped out so that higher-priority process may be loaded and executed.

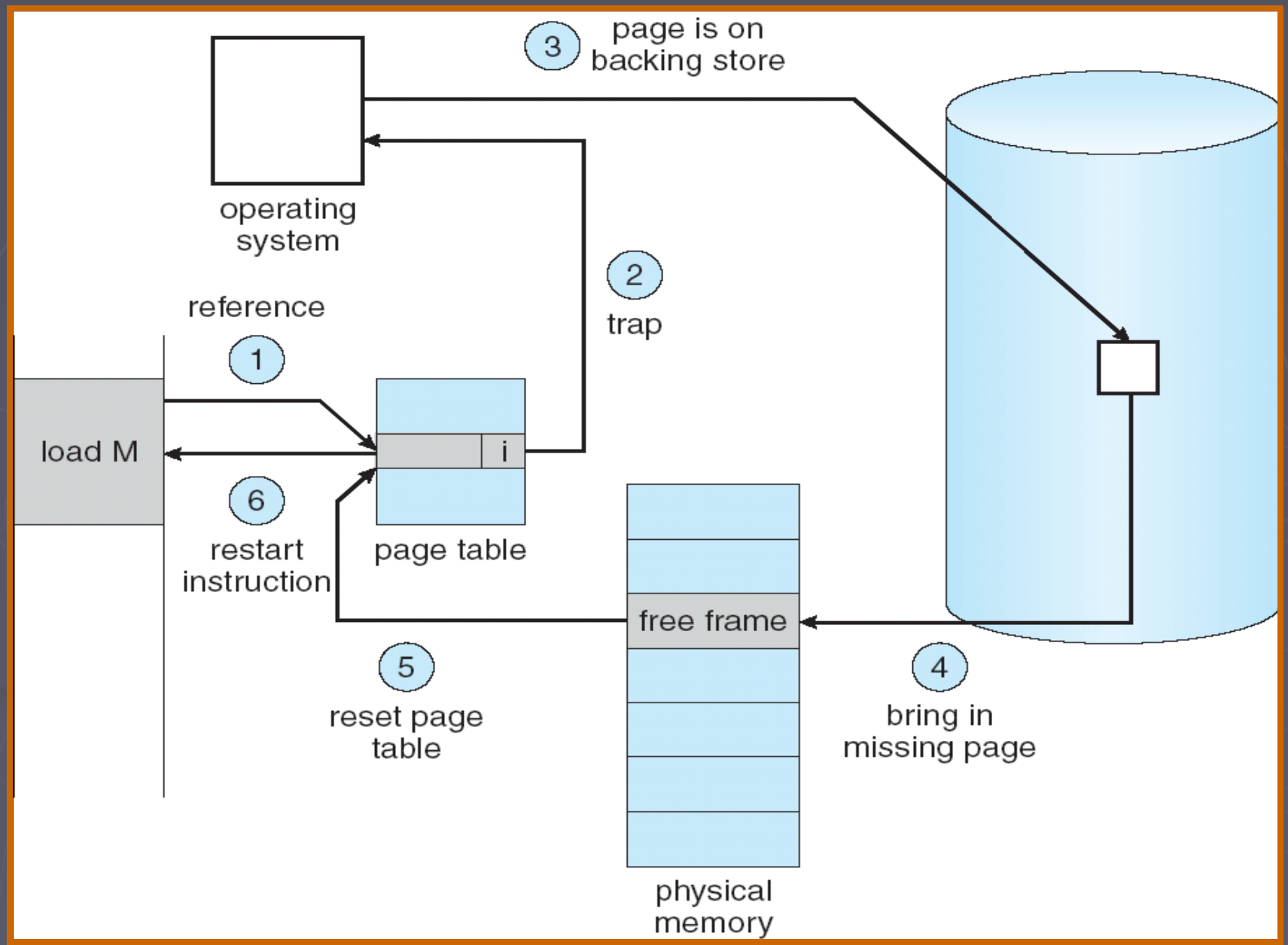

program A
program B
swap out
swap in
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
16 17 18 19
20 21 22 23
main memory

Demand Paging

- In virtual memory system, demand paging is a type of swapping in which pages of programs are not copied from disk to main memory until they are needed for execution.
- In demand paging, virtual address (logical address) is accessed by CPU, the corresponding page number is looked up in the page table and if it shows that currently this page is not in main memory, then this page must be brought into the main-memory.

- A **page fault** occurs when an invalid page is addressed. Page fault must be followed by swapping-in the page (demanded just now by the CPU) from disk to main-memory or a trap should be generated to the operating system if the page being demanded is not within the logical address space of the process. To determine whether the reference to the requested page is within the logical address space or not, an internal table may be consulted


0
A
1
B
2
C
3
D
4
E
5
F
6
G
7
H
logical memory
valid–invalid bit
frame
0 4 v
1 i
2 6 v
3 i
4 i
5 9 v
6 i
7 i
page table
0
1
2
3
4
A
5
6
C
7
8
9
F
10
11
12
13
14
15
physical memory
A
B
C
D
E
F


3
page is on
backing store
operating
system
2
trap
reference
1
load M
i
6
restart
instruction
page table
free frame
5
reset page
table
4
bring in
missing page
physical
memory

Pure Demand Paging

- Pure demand paging is the form of demand paging in which **not even a single page is loaded into memory, initially.** Thus, the very first instruction causes a page fault in this case. This kind of demand paging may significantly decrease the performance of a computer system by generally increasing the effective access time of memory.

Page Replacement

- Once the main memory fills up, a page must be swapped out to make room for any pages to be swapped in. This is known as page replacement.
- We have page replacement algorithms for page replacement purpose, which are as follows:

1. The optimal page replacement algorithm
2. The first-in, first-out (FIFO) page replacement algorithm
3. The second chance page replacement algorithm
4. The least recently used (LRU) page replacement algorithm
5. Simulating LRU in Software (Not frequently used algorithm)
6. The Working Set Page Replacement Algorithm