

Name: Shreerang Mhatre

Rollno: 29

Faculty: SY B.Tech Electrical & Computer Engg

PRN: 1032211745

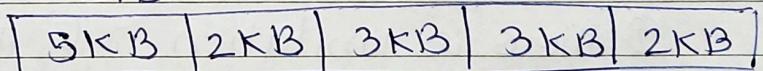
Course: COOS

Q 1) Explain contiguous memory allocations policies with suitable example.

→ In contiguous Memory Allocation, each process is contained in a single contiguous section of memory. In this memory allocation, all the available memory space remains together in one place which implies that the freely available memory partitions are not spread over here and there across the whole memory space.

• Eg -

An example of fixed size partitioning scheme, we will divide memory size of 15 kB into fixed size partitions. $\leftarrow 15 \text{ kB} \rightarrow$



It is important to note that these partitions are allocated to the processes as they arrive and the partition that is allocated to the arrived process basically depends on the algorithm followed.

Q 2) compare fixed and variable sized partitioning.

Fixed partitioning

- ① In multi-programming with fixed partitioning the main memory is divided into fixed size partitions.
- ② Only one process can be placed in a partition.
- ③ It does not utilize the main memory effectively.
- ④ There is presence of internal fragmentation and external fragmentation.
- ⑤ Degree of multi-programming is less.
- ⑥ It is more easier to implement.
- ⑦ There is limitation on size of process.

Variable partitioning

- ① In multi-programming with variable partitioning the main memory is not divided into fixed sized partitions.
- ② In variable partitioning, the process is allocated at chunk of free memory.
- ③ It utilizes the main memory effectively.
- ④ There is external fragmentation.
- ⑤ Degree of multi-programming is higher.
- ⑥ It is less easier to implement.
- ⑦ There is no limitation on size of process.

Q 3) Explain segmentation with suitable example.

→ * Segmentation -

Segmentation came into existence because of the problems in the paging technique. In the case of the paging technique, a function or piece of code is divided into pages without considering that the relative parts of code can also get divided. Hence, segmentation was introduced in which the code is divided into modules so that related code can be combined in one single block.

Example -

→ Let us assume we have five segments namely:
Segment-0, Segment-1, Segment-2, Segment 3... ↴

Initially, before the execution of the process, all the segments of the process are stored in the physical memory space. We have a segment table as well. The segment table contains the beginning entry address of each segment. The segment table also contains the length of each segment.

The base address of Seg-0 is 1400 & its length is 1000, the base address of Seg-1 is 6300 & length is 400, the base address of Seg-2 is 4300 and its length is 400.

Q 4) Explain paging with suitable example.

→ Paging -

In Operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages. The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.

• Example -

- ① Let us consider the main memory size 16 KB and Frame size is 1 KB therefore the main memory will be divided into the collection of 16 frames of 1 KB each.
- ② There are 4 processes in the system that is P₁, P₂, P₃ & P₄ of 4 KB each. Each process is divided into pages of 1 KB each so that one page can be stored in one frame.
- ③ Initially, all the frames are empty therefore pages of the processes will get stored in the contiguous way.
- ④ Let us consider that P₂ and P₄ are moved to waiting state after some time. Now 8 frames become empty and therefore other pages can be loaded in that empty place. The process P₅ of size 8 KB is waiting inside the ready queue.

(Q 5) Describe any four types of file organization.

→ The following are the different types of file organization:

① Sequential File Organization:

This is the most straight forward technique of file arrangement. Files are saved in this method in sequential order.

② Heap File Organization:

It is the most fundamental and basic type of organizational structure. It's based on data chunks. The records are inserted at the end of the file in the heap file organization. The ordering and sorting of records are not required when the entries are added.

③ Hash File Organization

The computation of the hash function on some of the fields of the records is used by Hash File Organization.

④ Cluster File Organization

Clusters are created when two or more records are saved in the same file. There will be two or more than two tables in the very same data block in these files, and key attributes are used to link these tables.

(Q6) Explain file system and methods to access file.

→ * File System Method -

A file is a collection of related information that is stored on secondary storage. Information stored in files must be persistent i.e. not affected by power failures & system reboots. Files may be free from such as text files or may be formatted rigidly.

- Methods to access files are:-

① Sequential access:

It is the simplest access method. Information in the file is processed in order i.e. one record after another. A process can read all the data in a file in order starting from beginning but can't skip & read arbitrarily from any location.

② Direct access:

A file is made up of fixed length-logical records that allow programs to read & write records rapidly in no particular order. This method can be used when we disk are used for storing files. This method is used in many applications eg. data base systems.

Q) what is I/O buffering? Explain types of buffers.

→ * I/O buffering -

Input/output buffering, or IO buffering is a type of buffering that occurs between a processor and a peripheral device in order to smooth out the difference in rates at which the devices can receive and process data.

- Types of buffers:

- ① Single buffer:

A buffer is provided by the operating system to the system portion of the main memory.

- ② Double buffer:

There are two buffers in the system. One buffer is used by the driver or controller to store data while waiting for it to be taken by higher level of the hierarchy.

- ③ Circular buffer:

When more than two buffers are used, the collection of buffers is itself referred to as a circular buffer. In this, the data do not directly pass from the producer to the consumer because the data would change due to overwriting of buffers before they had been consumed.

Q 8) Consider the following string

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

The number of page frames = 4

calculate page faults using

- (i) FIFO
- (ii) LRU
- (iii) Optimal

→ (i) FIFO

String 1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

F₁ 1 1 1 1 1 1 5 5 5 5 5 5 3 3 3 3 3 3 1 1 1

F₂ 2 2 2 2 2 2 2 6 6 6 6 6 7 7 7 7 7 7 3 3

F₃ 3 3 3 3 3 3 3 2 2 2 2 2 6 6 6 6 6 6 6

F₄ 4 4 4 4 4 4 4 1 1 1 1 1 2 2 2 2

* * * * Hit Hit * * * * Hit * * * Hit *

* * * * Hit Hit

No. of page faults = 14

$$\text{Page fault ratio} = \frac{14}{20} \times 100$$

$$= 70\%$$

② LRU (Last recently used)

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
 .
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6
 2
 3 3 3 5 5 5 5 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 4 4 4 4 6 6 6 6 6 7 7 7 7 1 1 1
 * * * * H H * * H H H * * * H H * H H H

$$\text{No. of page faults} = 10$$

$$\text{Page fault ratio} = \frac{10}{20} \times 100 = 50\%$$

③ Optimal

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 7 7 7 7 1 1 1
 2
 3
 4 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 * * * * H H * * H H H H * H H H H * H H H

$$\text{No. of page faults} = 8$$

$$\text{Page fault ratio} = \frac{8}{20} \times 100 = 40\%$$