UNIT 3

2 marks

1. What is a File Management System?
2. List any two functions of a File Management System.
3. What are the key components of a File System Architecture?
4. Explain the term **File Organization** in operating systems.
5. How does File Access differ from File Organization?
6. Define **File Sharing** and mention one advantage of it.
7. Explain **File Allocation** in the context of a file system.
8. Name any two **File Allocation Methods**.

5/6 marks

1. Explain the **File Management System** and discuss its key functions.
2. Describe the **File System Architecture** with a neat diagram.
3. Compare different **File Organization methods** with examples.
4. Explain the **File System Layout** and its importance in an operating system.
5. Discuss the different types of **File Directories** and their advantages.

6. Describe **File Allocation Methods** (Contiguous, Linked, Indexed) with examples.

UNIT4

2 marks

1. Define **memory management** in an operating system.
2. What is **relocation** in memory management?
3. Differentiate between **fixed and dynamic partitioning**.
4. What is **swapping** in memory management?
5. Explain the concept of **fragmentation**.(Internal and external)

5 marks

1. Explain **Contiguous Memory Allocation** and its advantages & disadvantages.
2. Describe **Fixed Partitioning** and **Dynamic Partitioning** with examples.
3. What is **internal and external fragmentation**? How does it affect memory allocation?
4. Compare **First-Fit, Best-Fit, and Worst-Fit** memory allocation strategies.
5. explain **paging** and its **structure of page table**.
6. Describe **segmentation** with an example.
7. What are **page replacement algorithms**? Compare **FIFO and LRU**.

8.Given Memory Partitions:

Sizes of available memory blocks:
100K, 500K, 200K, 300K, 600K

Processes to be allocated:

Sizes:
212K, 417K, 112K, 426K

Processes are allocated by Memory Allocation Using First Fit, Best Fit, and Worst Fit

Solution:

<mark>**First-Fit Algorithm**</mark>

Steps:

1. **212K** → Placed in **500K** block (remaining: 500 - 212 = 288K)

2. **417K** → Placed in **600K** block (remaining: 600 - 417 = 183K)

3. **112K** → Placed in **288K** block (remaining: 288 - 112 = 176K)

4. **426K** → Cannot be allocated (must wait)

<mark>**Best-Fit Algorithm**</mark>

Steps:

1. **212K** → Placed in **300K** block (remaining: 300 - 212 = 88K)

2. **417K** → Placed in **500K** block (remaining: 500 - 417 = 83K)

3. **112K** → Placed in **200K** block (remaining: 200 - 112 = 88K)

4. **426K** → Placed in **600K** block (remaining: 600 - 426 = 174K)

<mark>**Worst-Fit Algorithm**</mark>

Steps:

1. **212K** → Placed in **600K** block (remaining: 600 - 212 = 388K)

2. **417K** → Placed in **500K** block (remaining: 500 - 417 = 83K)

3. **112K** → Placed in **388K** block (remaining: 388 - 112 = 276K)

4. **426K** → Cannot be allocated (must wait)

📌 **Final Allocation (Worst Fit):**

9.Memory Partitions Available:

150K, 350K, 600K, 175K, 400K

Processes to be Allocated:

Process 1 → 200K

Process 2 → 375K

Process 3 → 100K

Process 4 → 275K

Process 5 → 50K

Processes are allocated by Memory Allocation Using First Fit, Best Fit, and Worst Fit

Solution:

First-Fit Allocation

Steps:

1. **200K** → Allocated in **350K** (Remaining: 350 - 200 = 150K)

2. **375K** → Allocated in **600K** (Remaining: 600 - 375 = 225K)

3. **100K** → Allocated in **150K** (Remaining: 150 - 100 = 50K)

4. **275K** → Allocated in **400K** (Remaining: 400 - 275 = 125K)

5. **50K** → Allocated in **remaining 50K (from step 3)** (Fully utilized)

Best-Fit Allocation

Steps:

1. **200K** → Allocated in **exactly 200K** (but no such block exists) → Allocated in **350K** (Remaining: 350 - 200 = 150K)

2. **375K** → Allocated in **400K** (Remaining: 400 - 375 = 25K)

3. **100K** → Allocated in **150K** (Remaining: 150 - 100 = 50K)

4. **275K** → Allocated in **600K** (Remaining: 600 - 275 = 325K)

5. **50K** → Allocated in **remaining 50K (from step 3)** (Fully utilized)

Worst-Fit Allocation

Steps:

1. **200K** → Allocated in **600K** (Remaining: 600 - 200 = 400K)

2. **375K** → Allocated in **remaining 400K** (Remaining: 400 - 375 = 25K)

3. **100K** → Allocated in **350K** (Remaining: 350 - 100 = 250K)

4. **275K** → Cannot be allocated (no large enough block)

5. **50K** → Allocated in **remaining 250K (from step 3)** (Remaining: 250 - 50 = 200K)