UNIT-4

MEMORY MANAGEMENT

MEMORY MANAGEMENT:- Memory management is a form of resource management applied to computer memory. The essential requirement of memory management is to provide the ways to dynamically Portions of memory to programs at their request, and free it for reuse when no longer needed.

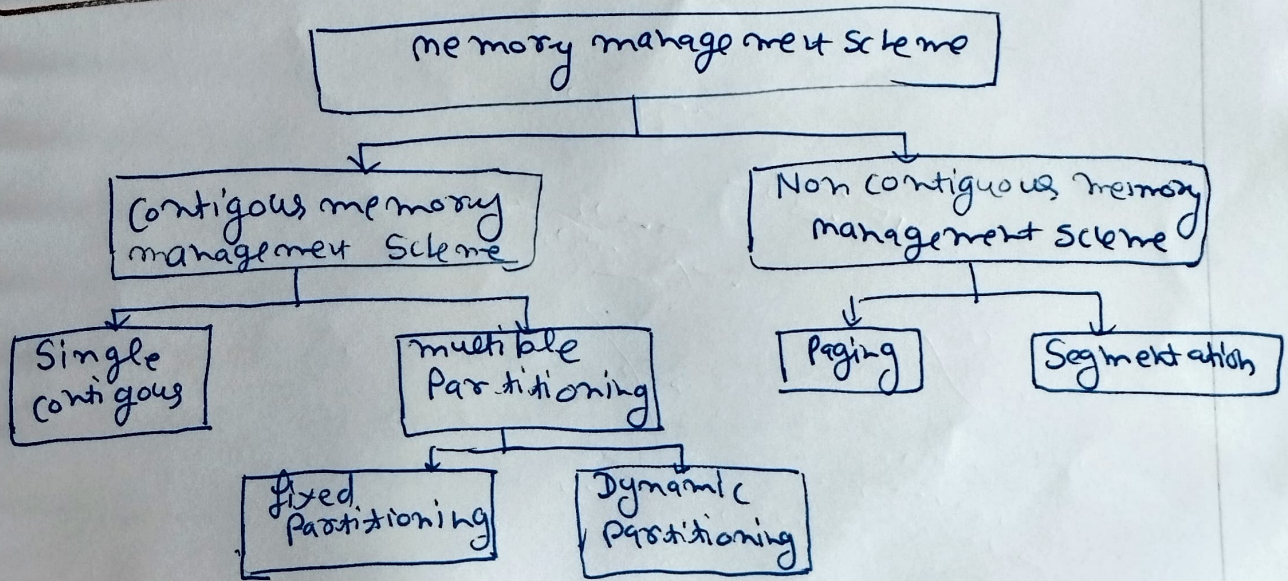
Why memory is Required:-

Allocate and deallocate memory before and after Process execution.

To keep track of used memory space by Processes.

to minimize fragmentation issues. to Proper utilization of main memory. To maintain data integrity while executing of Process.

memory management Techniques:-



CONTIGUOUS MEMORY ALLOCATION:-

In a contiguous memory management scheme, each program occupies a single block of storage location such as a set of memory locations with consecutive addresses.

Ex

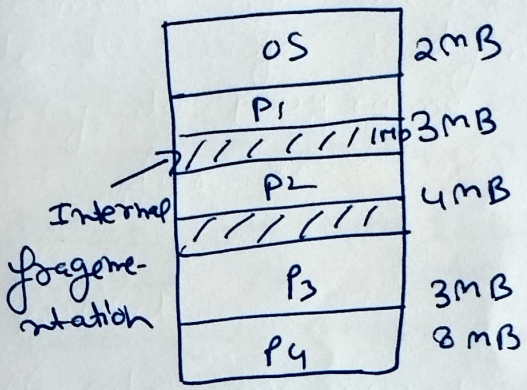
$P_1 \Rightarrow 1MB$

$P_2 \Rightarrow 2MB$

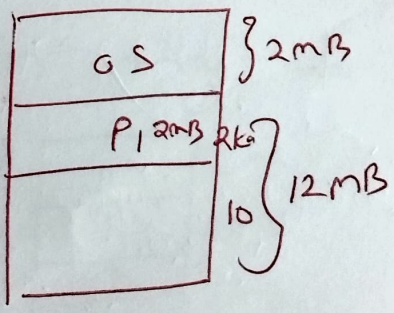
$P_3 \Rightarrow 3MB$

$P_4 \Rightarrow 12MB$

8MB 2MB



Single Contiguous memory Allocation:- In this scheme the main memory is divided into two contiguous areas or partitions. The operating system (OS) reside permanently in one partition, generally at the Low memory and the user process is loaded into the other partition.



Advantages: (1) Simple to implement

(2) Does not require expertise to understand and use such as a system.

Disadvantage of single contiguous memory:-

(1) Wastage of memory space.
(2) The CPU remain Idle.
(3) It can not be executed if the Program is too large.
(4) It does not support multiprogramming.

MULTIPLE PARTITIONING:-

In this operating system needs to divide the available main memory into multiple parts to load multible Processes into the main memory.

It is two types:-

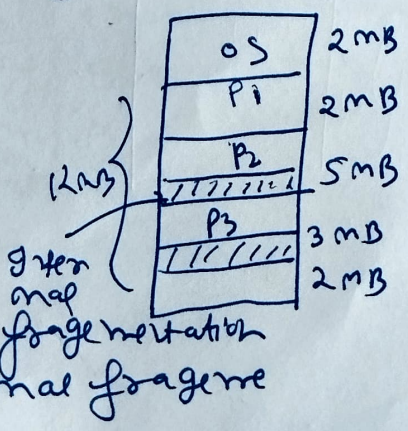
(1) fixed Partitioning
(2) variable Partitioning.

(1) fixed Partitioning:-

In this technique is also known as Static Partitioning. In this schem the system divides the memory into fixed size Partitions.

The partitions may or may not be the same size.

P1 → 2MB
P2 ⇒ 2MB
P3 ⇒ 1MB
P4 ⇒ 3MB



Fragmentation!- memory fragmentation can be of two types.

(1) Internal fragmentation.

(2) External fragmentation.

Internal fragmentation!- In Internal fragmentation there is wasted space internal to a Partition due to fact the block of data loaded is smaller than the partition.

External fragmentation!- The External fragmentation there is a hole of in multiple Partition allocation Scheme. Next Process request of memory. Actually of memory is free which satisfy the request but Hole is not contiguous.

Difference Between Internal and External fragmentation.

Internal	External
(1) memory allocated to a process may be slightly larger than the requested memory the difference between these two numbers is Internal fragmentation.	(1) External fragmentation exists when there is enough total memory space to satisfy a request but available spaces are not contiguous.

(2) first-fit and Best-fit memory allocation does not suffer from internal fragmentation

(3) Systems with fixed size allocation units such as the single partitions scheme and Paging suffer from internal fragmentation

(2) first-fit and Best fit memory allocation suffers from external fragmentation.

(3) Systems with variables sized allocation units such as the multiple partitions scheme and segmentation suffers from external fragmentation

UNIT-2

READERS - WRITERS PROBLEM!

A Database is to be shared among several Concurrent Process. Some of the Processes may want to update i.e read/write the database.

~~Hard~~

if a writer and some other thread access the database simultaneously choose may email. to ensure these difficulties do not arise all require. that writers have exclusive access to Shared database. This is known as Reader's writer Problem.

Solution!-

we will use 2 Semaphores and one integer variables

(1) mutex (m) → ~~and~~ initialised to 1

(2) wrt → initialised to 1

(3) readcount → initialised to 0

writer:

```
do
{
wait (wrt);
// performs write operation
signal (wrt);
}
while (TRUE);
```

Reader's Process:-

```
do
{
wait (mutex)
read count ++;
if (read count == 1)
wait (wrt);
signal (mutex);
//perform reading
wait (mutex);
read count --;
if (read count == 0)
signal (wrt);
signal (mutex);
```

```
while (TRUE);
```

BARE MACHINE:-

Bare machine is a simplest form of memory management. Bare machine is logical Hardware which is used to execute the Program in the Processor without using the operating system.

The execution of an instruction is done by directly on Hardware without using any interfering Hardware.

Bare machine accepting the instruction in only machine language due to this those Person who has sufficient knowledge about computer field are able to operate a Computer.

Resident monitor:- The Resident monitor

is a code that run on Bare machine.

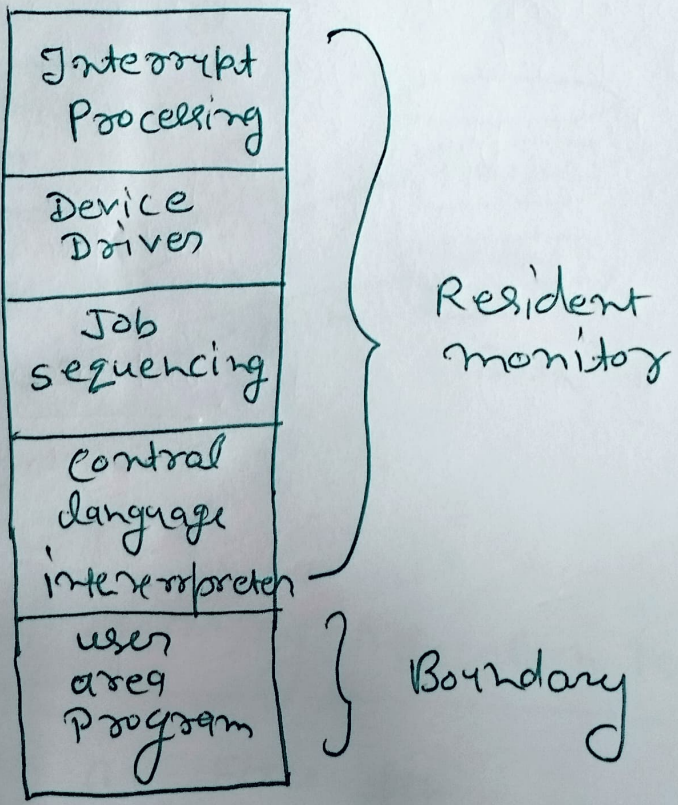
It also work like an operating System that controls the instructions and performs all necessary function.

It also work like Job sequencer because it also Sequences the Job and Send them to the Processor.

After Scheduling the Job Resident monitors loads the Program one by one into the main memory according to their sequence. when the Program executing occured there is no gap between the Program execution and the Processing is going to be faster

It is divided into 4 parts as:

(1) control Language Interepreter
(2) Loader
(3) Device Driver
(4) Interrupt Processing

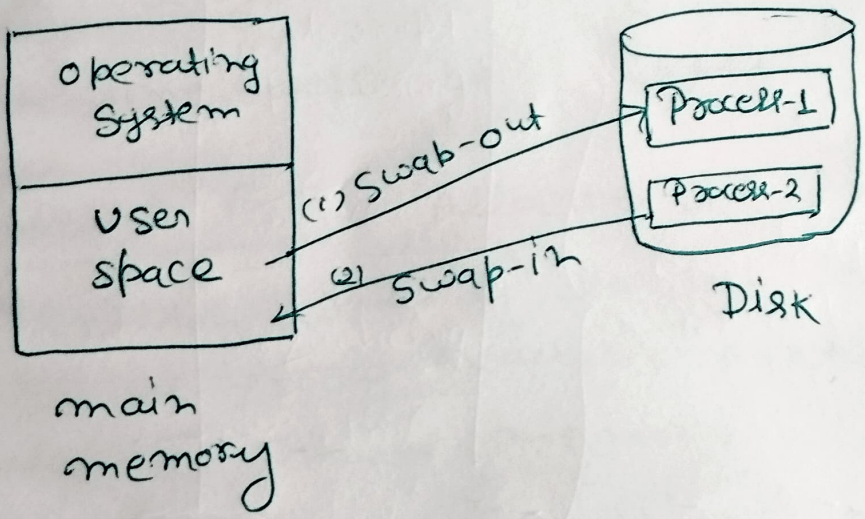


SWAPPING :- Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage (disk) and make that memory available to other processes.

At some later time the system swaps back the process from the second storage (disk) to main memory.

(1) Swap-out ⇒ Take out the current data from main memory

(2) Swap-in ⇒ Bring the data process of new user in to main memory.



Contiguous memory Allocation Techniques :-

Algorithms :-

(1) First fit
(2) Best-fit
(3) Worst-fit

(1) FIRST FIT :- In the first fit partition is allocated which is first sufficient from the top of main memory.

A → search time is small

D → memory loose an account of fragementation is likely to be High.

(2) Best-FIT :- Allocate the process to the Partition which is first smallest sufficient partition among the free available Partition.

memory loss on account of fragementation will be large in case of first fit.

search time will be large.

3) Worst-FIT :- Allocate the process to the partition which is largest sufficient among the freely available partition available in the main memory.

Ex Given five memory Partition of 100 Kb, 500 Kb, 200 Kb, 300 Kb, and 600 Kb in same order. How would each of the first-fit best fit and worst fit algorithms Place Processes of 212 Kb, 417 Kb, 112 Kb, and 426 Kb. Calculate which algorithm makes the most efficient use of memory.

Soln

memory partition
100 kb, 500 kb, 200kb, 300kb, and 600kb
in same order. How would each of 4 fi

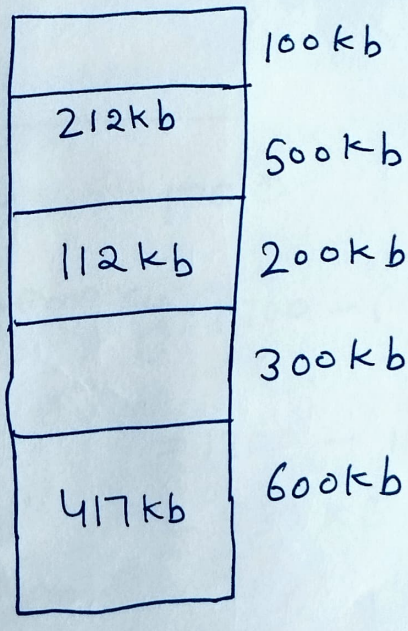
Process $P_1 \Rightarrow$ 212 kB

$P_2 \Rightarrow$ 417 kb

$P_3 \Rightarrow$ 112 kb

$P_4 \Rightarrow$ 426 kb

first-fit Algorithms :-



426 Kb can not get memory

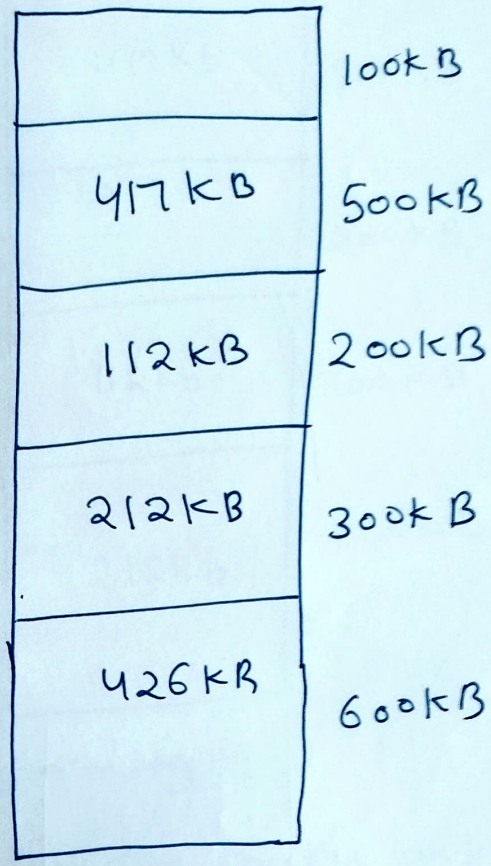
total memory = 1700 kB

memory wastage = 1700 − (212 + 112 + 417)

= 1700 − 741

= 959 kB

Best - fit Algorithm:-



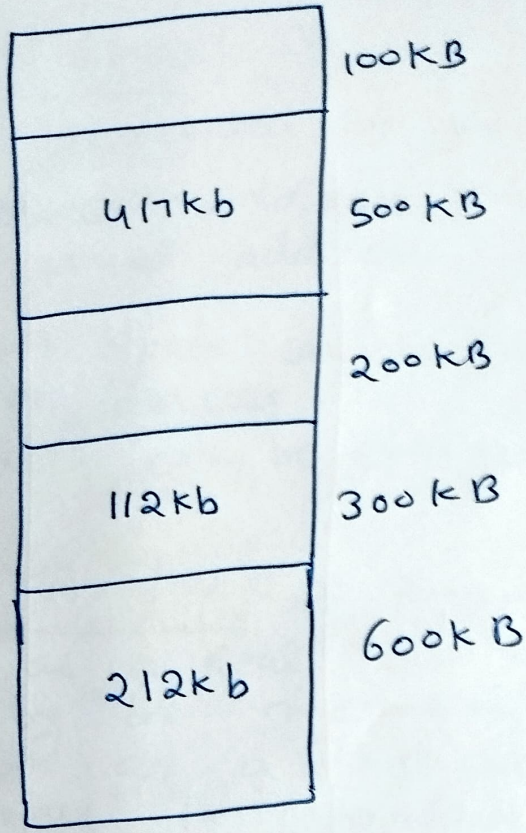
total memory = 1700

wastage of memory = 1700 − (417 + 112 + 212 + 426)

= 1700 − 1167

= 533 kB

worst - fit Algorithm:-



426Kb next is waiting

Wastage of memory in this algorithm

$= 1700 - (417 + 112 + 212)$

$= 1700 - (529 + 212)$

$= 1700 - (741)$

$= 959$

Best-fit algorithm makes the most efficient use of memory.

Logical Address Space:-

An address generated by the CPU is known as Logical Address. It is also known as virtual address.

Logical Address space can be defined as the size of the Process.
A Logical address can be changed.

Physical Address Space:- A Physical Address is also known as a Real address. an address seen by the memory unit is comming known as a Physical Address. A Physical Address is computed by (MMU) main memory unit.

STATIC AND DYNAMIC LOADING:- To load a Process into the main memory is done by a loader there are two types of loading.

Static Loading:- In Static Loading the entire Program into a fixed address. It requires more memory space.

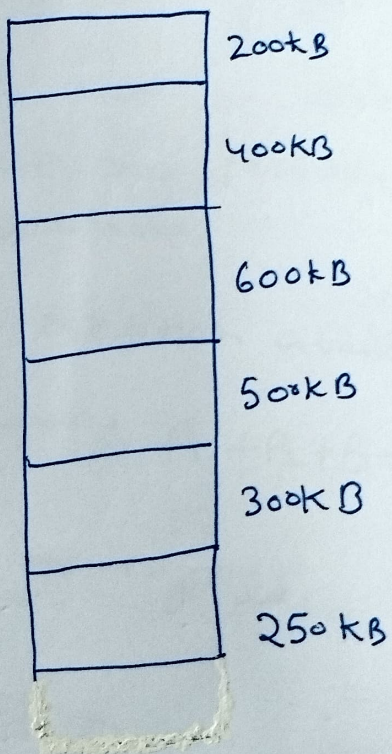
Dynamic Loading:- All the Program are loaded in the main memory for execution. Sometimes complex Program is loaded into the memory, by some

Practice Problem for memory allocation:-

Q Given memory partitions of 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB. These partitions need to be allocated to four processes of size 357 KB, 210 KB, 465 KB and 491 KB in that order using first-fit, Best fit and worst-fit algorithm.

Sol

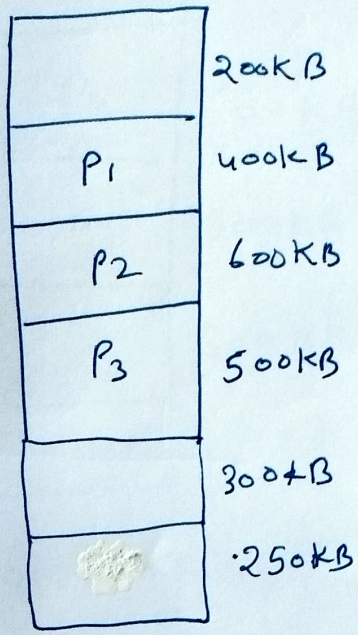
The main memory has been divided into fixed Partitions as:



given Process are

Process P_1 = 357 KB
process P_2 = 210 KB
Process P_3 = 468 KB
process P_4 = 431 KB

Allocation using first fit Algorithm :-



Process P_4 can not be allocated the memory because no partition of size of process P_4 is available.

Total memory Partition available = 2250 KB

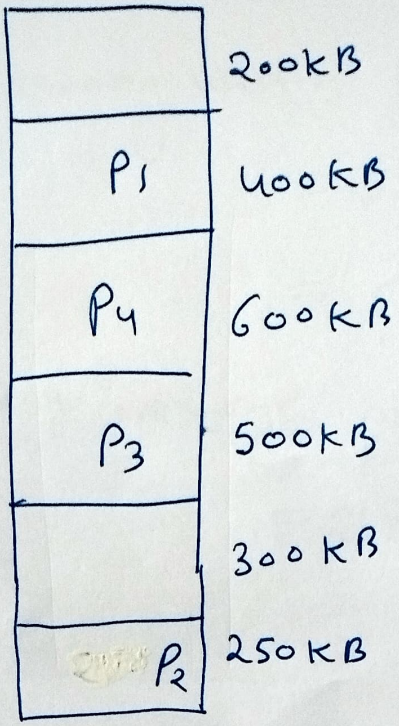
Total used memory $P_1 + P_2 + P_3 \Rightarrow 325 + 210 + 468$

= 1035 KB

total used memory loss = 2250 KB − 1035 KB

= 1215

Sol ⇒ Best-fit-Algorithm :-



total available memory Partition = 2250KB

total used memory = P1 + P2 + P3 + P4

= 357 + 210 + 468 + 491

= 1526 KB

total memory loss = 2250 - 1526

= 724 KB

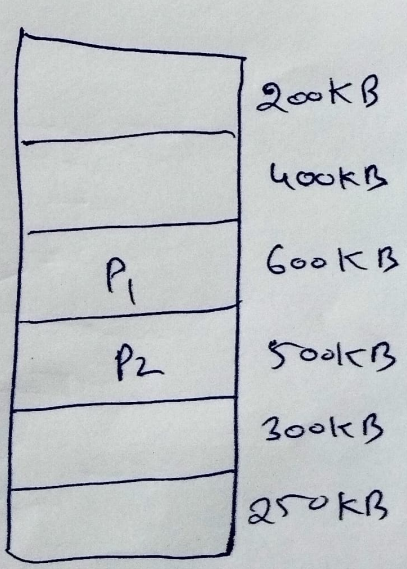
worst fit Algorithm:-

P_1 = 357 KB

P_2 = 210 KB

P_3 = 468 KB

P_4 = 431 KB



total memory used = $P_1 + P_2$

$= 357\,KB + 210\,KB$

$= 567\,KB$

total wastage memory = $2250 - 567$

$= 2183\,KB$