Ch-1

Introduction,

Process Management,

Memory management

➢ Operating system

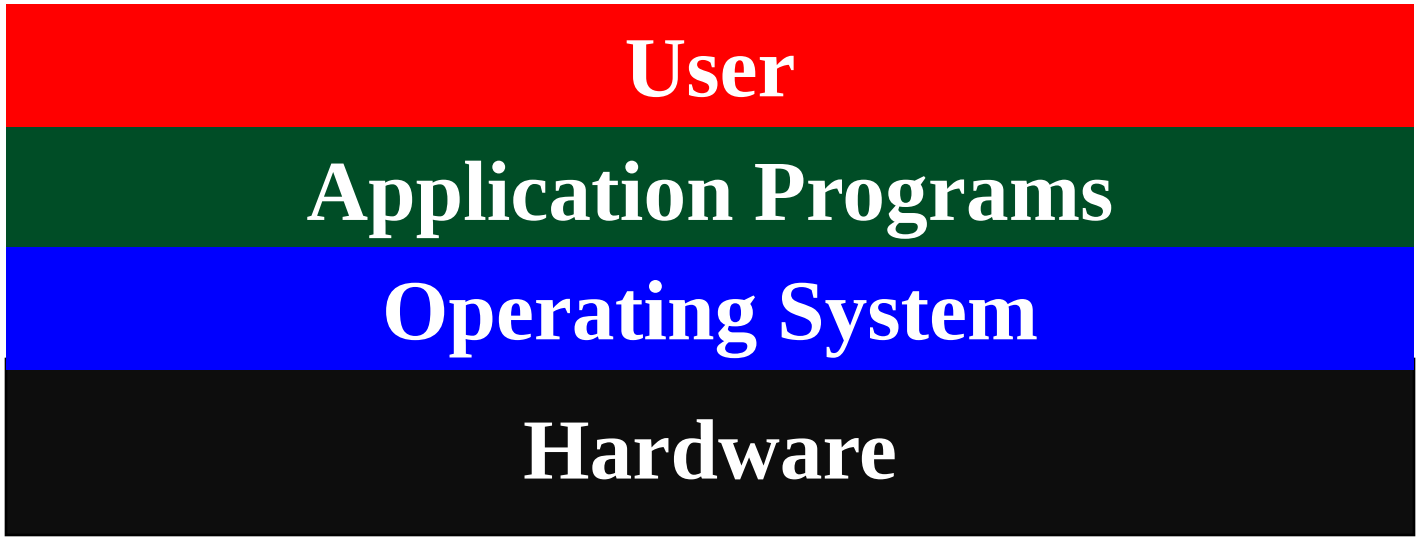
- Operating system is a program/software that

  1) Manages the computer hardware

  2) Provide user program with similar interface to the hardware.

- There are two main goals of the OS :
  - First, convenience for user
  - Second, efficient operation of the computer system.

Component of OS



Cont...

- The hardware comes at the lowest level. It contains physical device such as processor, memory, keyboard, etc.
- The next level is for the operating system .it manages all the underlying hardware and masks the complex from the user.
- Application program perform particular task.
- User are at the top , users interact with the system by using application program to perform particular tasks.

Function of OS :

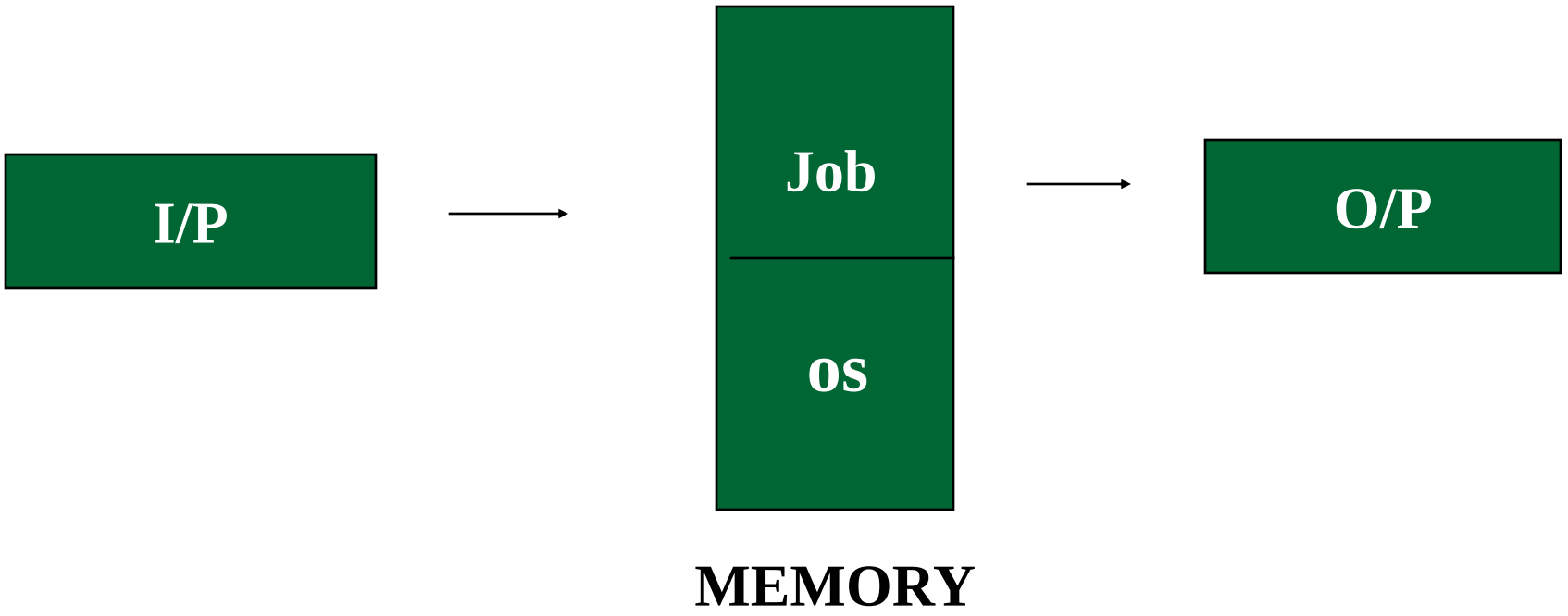
- Memory management
- Process management
- I/O management
- File management
- Scheduling
- Security management
- Processor management

Types of OS :

- **There are six types of OS :**
  1. **Batch OS**
  2. **Multiprogramming OS**
  3. **Multitasking / time sharing OS**
  4. **Network OS**
  5. **Distributed OS**
  6. **Multithreading OS**

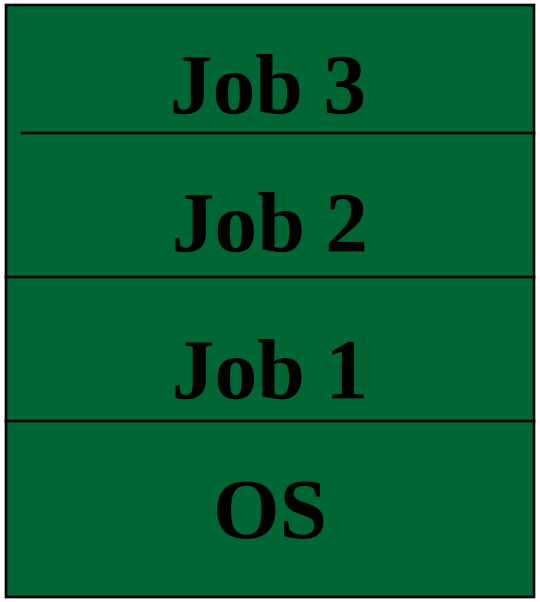
1. Batch OS :

- Batch OS is execution of a series of program on a computer without human interaction .



2 . Multiprogramming OS :

- A multiprogramming system permits multiple programs to be loaded into the main memory and executes them concurrently.



3 . Multitasking / time sharing os :

- Multitasking is a logical extension of the multiprogramming OS.
- CPU is multiplexed by time among several jobs that are kept in main memory and on disk.

4 . Network OS :

- A network os allows a set autonomous computer, interconnected by a computer network, to be used together in a convenient and cost-effective manner.

Cont...

5 . Distributed OS :

- Distributed OS distributes the computation among many processor.

6 . Multithreading OS :

- Multithreading OS support the concept of multiple thread within single process environment.

Introduction :

- A process can simply be defined as a program in execution.
- It can defined as a program currently use of the processor at any one time.
- A process is a tuple consists of process id, code, data, register values, pc value .

Process control :

1 Process creation :

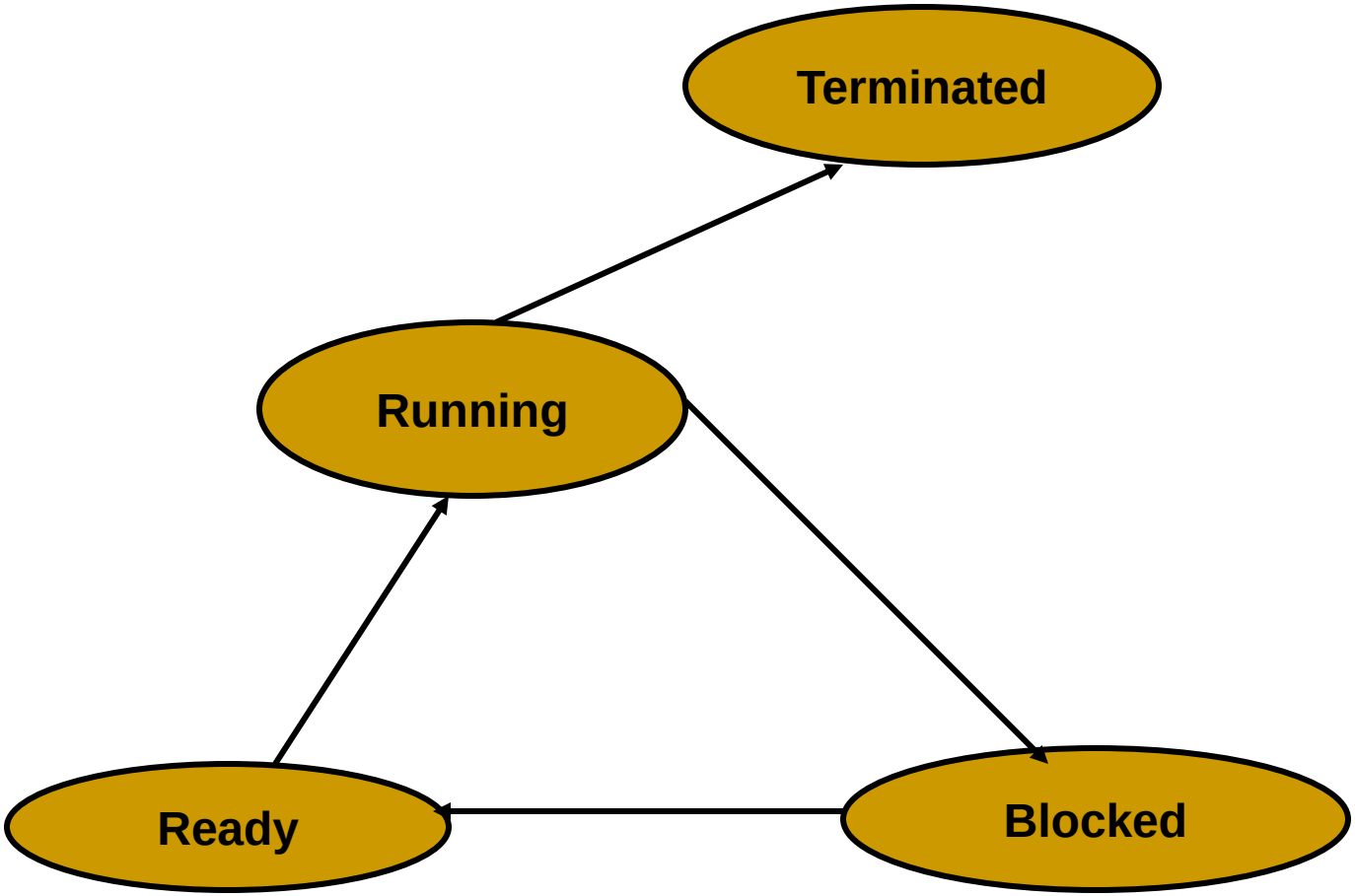
- OS perform the following actions when a new process is created :
  - Created a process control block (PCB) for the process.
  - Assign process id and priority.
  - Allocate memory and other resources to the process.
  - Set up the process environment.
  - Initialize resources accounting information for the process.

Process control :

2 Process state :

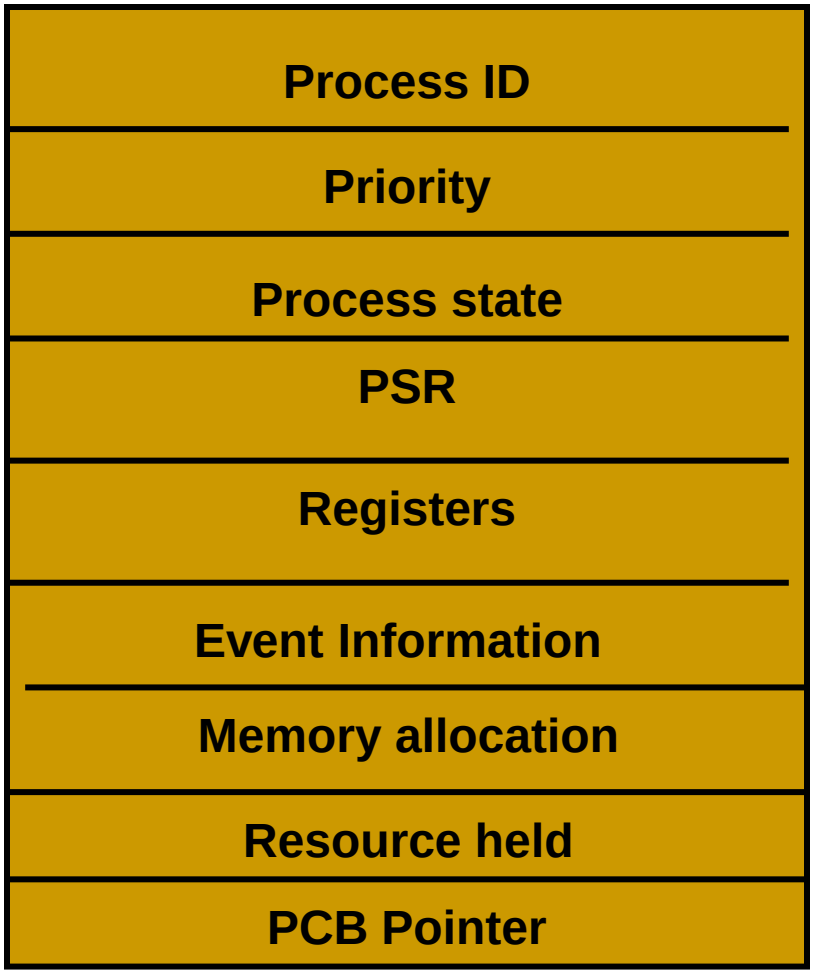
- The life time of a process can be divided into several stages, as state each with certain characteristics that describe the process.
  - Each process may be in one of the following state :
    - New : The process has been created .
    - Ready : The process is waiting to be allocated to a processor.
      - Running :
      - Waiting Block :
      - Ready :
      - Terminated :

Process State Transition Diagram :



Process control block :

- A data structure called the PCB is used by an OS to keep tracks of all information concerning a process.



Process Scheduling :

- Process scheduling consists of three sub components :

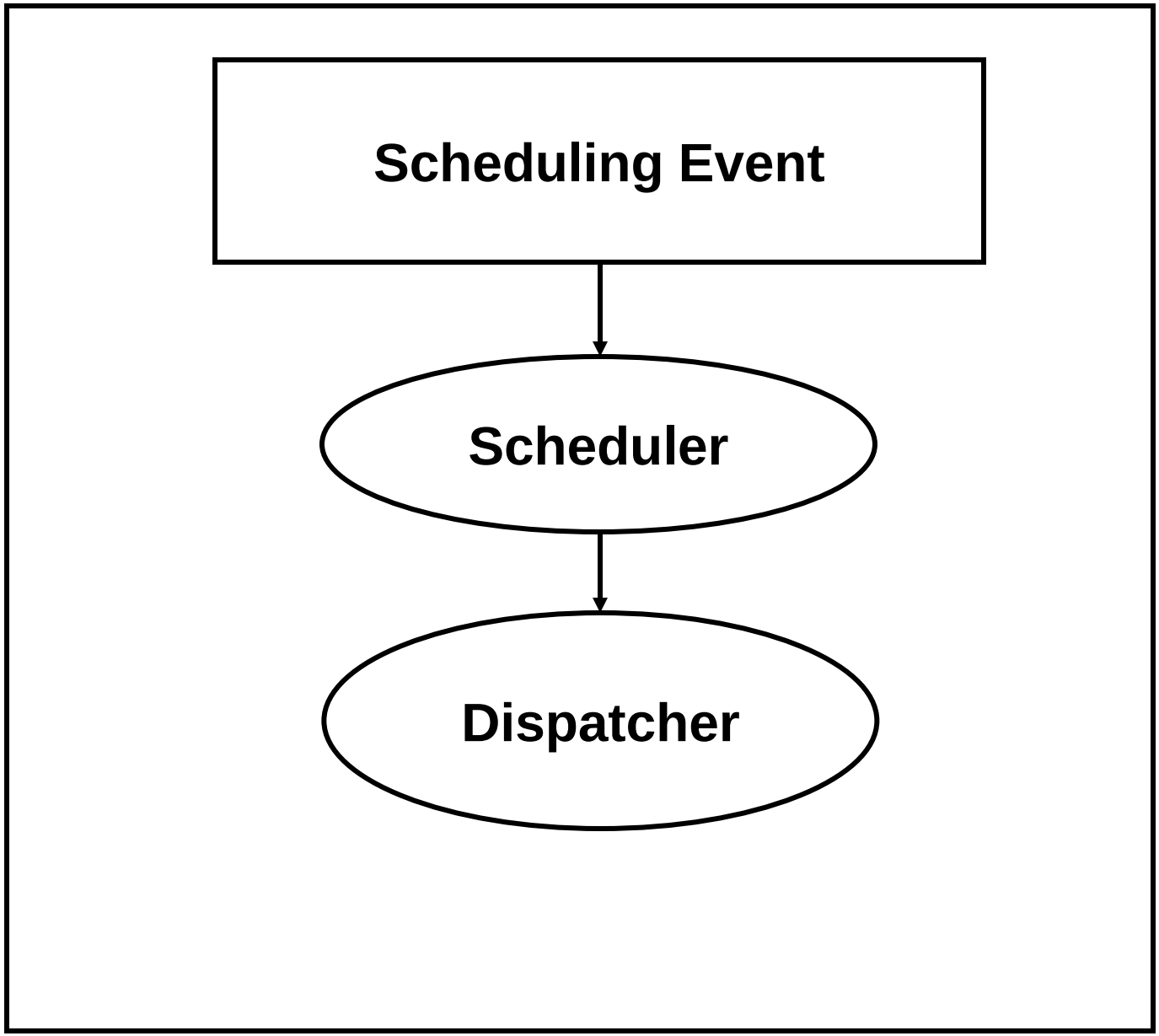
1 Event Handler :

Event Handler performs Following activity.

- Event Monitoring
- ECB

2 Process Scheduler :

- Process Scheduler schedules the process.


Scheduling Event
Scheduler
Dispatcher

3 Process Dispatcher :

- Dispatcher is an OS module which transfer control of CPU from one process to another.

- It is last step in scheduling.

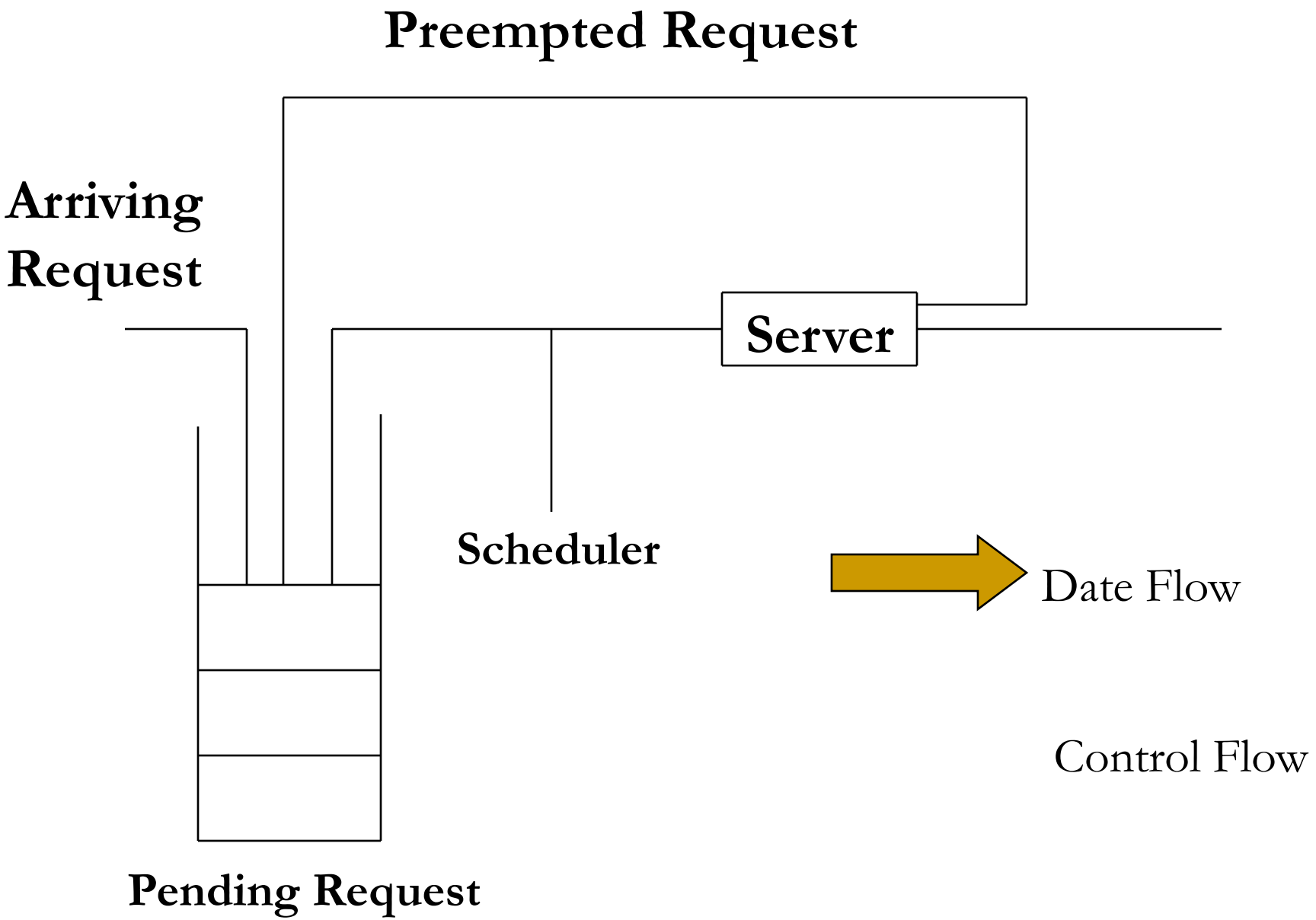
Scheduling :

- Scheduling is the activity of determining which service request should be handled next by a server. Scheduling policy specifies criteria for service request.

- The figure shows movement of request in the system. All requests waiting to be received are kept in a list of pending request.

Scheduling :

- The preempted request is put back into the list of pending request.
- In either case scheduler performs scheduling to select the next request to be serviced.


Preempted Request
Arriving Request
Server
Scheduler
Date Flow
Control Flow
Pending Request

➢ The Following are the police of scheduling:

- FCFS (First Come First Served) Scheduling.
- SJN (Shortest Job Next) Scheduling.
- Round Robin Scheduling.
- Priority Based Scheduling.
- Preemptive Scheduling.
- Non Preemptive Scheduling.
- Deadline Scheduling.

FCFS (First Come First Served) Scheduling :

- Selection Criteria :
  - The process that request first is served first. It means that process are served in the exact order' as they come.
- Decision Mode :
  - Non-preemptive : Once a process is selected, it runs until it blocks for I/O or some event or it terminates.

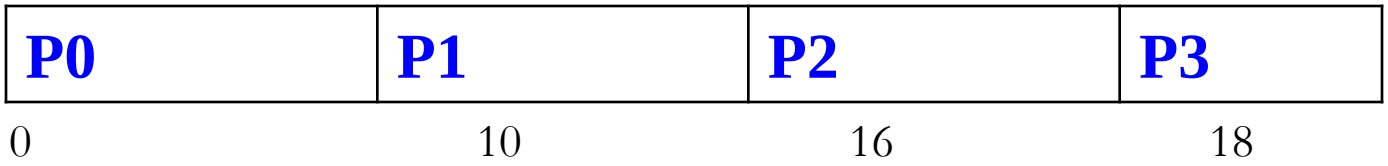
FCFS (First Come First Served) Scheduling :

- Implementation :
  - This strategy can be easily implemented by using FIFO queue. When CPU becomes free, a process from the first position in a queue is selected to run.

➢ Example :

Process	Arrival Time (T0)	Time Required For Completion (ΔT)
P0	0	10
P1	1	6
P2	3	2
P3	5	4

➢ Gantt Chart :



Initially only process P0 is present and it is allowed to run. But, when P0 completes, all other processes are present. So, next (P1) from ready queue is selected and allowed to run till all processes complete their execution.

➢ Statistics :

Process	Arrival Time (T0)	Completion Time(ΔT)	Finish Time (T1)	Turnaround Time (TAT=T1-T0)	Waiting Time (TAT-ΔT)
P0	0	10	10	10	0
P1	1	6	16	15	9
P2	3	2	18	15	13
P3	5	4	22	17	13

Average Turn-around Time : (10+15+15+17)/4 = 57/4 = 14.25 ms

Average Waiting Time : (0+9+13+13)/4 = 35/4 = 8.75 ms

. SJN (Shortest Job Next) Scheduling :

- **Selection criteria :**

The process, that requires shortest time to complete execution, is served first.

- **Decision Mode :**

Non-preemptive : Once a process is selected, it runs until it blocks for an I/O or some event, or it terminates.

Cont...

- **Implementation :**

  This strategy can be implemented by using sorted FIFO queue. When CPU becomes free, a process from the first position in a queue is selected to run.

➢ Example :

Process	Arrival Time (T0)	Time Required For Completion (ΔT)
P0	0	10
P1	1	6
P2	3	2
P3	5	4

➢ Round Robin Scheduling :

➢ **Selection criteria :**

The process that request first. it means that processes are served in the exact order as they come.

➢ **Decision Mode :**

Preemptive : each selected process is assigned a time interval called time quantum or time slice .process is allowed to run only for this time interval.

➢ Round Robin Scheduling :

➢Implementation :

This strategy can be implemented by using sorted FIFO queue. When CPU becomes free, a process from the first position in a queue is selected to run.

Round Robin Scheduling :

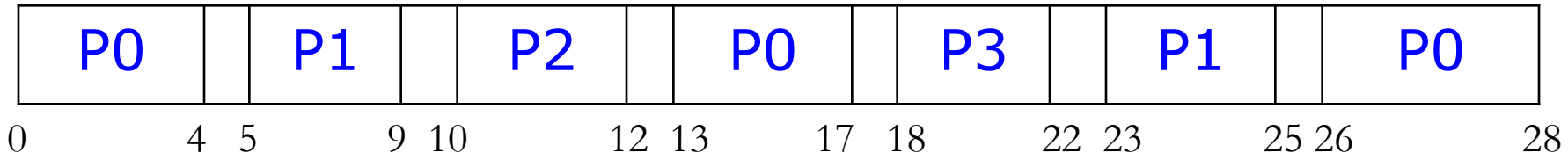
- Example :

consider that time quantum is of 4ms ,and context switch overhead is of 1ms.

Process	Arrival Time (T0)	Time Required For Completion (ΔT)
P0	0	10
P1	1	6
P2	3	2
P3	5	4

Round Robin Scheduling :

Gantt Chart :



➢ Statistics :

Process	Arrival Time (T0)	Completion Time (ΔT)	Finish Time (T1)	Turnaround Time (TAT=T1-T0)	Waiting Time (TAT-ΔT)
P0	0	10	28	28	18
P1	1	6	25	24	18
P2	3	2	12	9	7
P3	5	4	22	17	13

Average Turn-around Time : (28+24+9+17)/4 = 78/4 = 19.5 ms

Average Waiting Time : (18+18+7+13)/4 = 56/4 = 14.00 ms

Priority Based Scheduling :

Selection criteria :

The process that has highest priority, served first .

Decision Mode :

Preemptive : When a new process arrives its priority is compared to the current process' priority. .

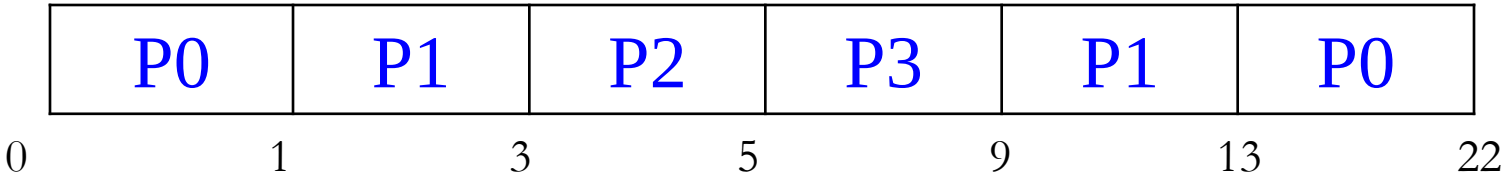
Implementation :

This strategy can be implemented by using sorted FIFO queue. All process in a queue are sorted based on their priority with highest priority process at front end.

➢ **Example :** consider all time values in ms and small value for priority means higher priority of a process

Process	Arrival Time (T0)	Time required For completion	Priority
P0	0	10	**5**
P1	1	6	**4**
P2	3	2	**2**
P3	5	4	**0**

Gantt Chart :



Statistics :

Process	Arrival Time (T0)	Completion Time (ΔT)	Finish Time (T1)	Turnaround Time (TAT=T1-T0)	Waiting Time (TAT-ΔT)
P0	0	10	22	22	12
P1	1	6	**13**	12	6
P2	3	2	5	**2**	0
P3	5	4	**9**	4	**0**

Average Turn-around Time : (22+12+2+4)/4 = 40/4 = 10.00 ms

Average Waiting Time : (12+6+0+0)/4 = 18/4 = 4.5 ms

Preemptive Scheduling :

- In this type of scheduling, each process is allocated a priority level. It is depending on the priority level, the OS can run a process of high priority .
- Scheduler is invoked whenever an event that changes the state of process is detected.
- The preemptive scheduling a running process may be replaced by a higher priority process at any time.

Non Preemptive Scheduling :

- This type of scheduling each process continues to run till terminates or requires an I/O or some synchronizations signal.

- In non-preemptive scheduling, once scheduled , a selected job runs till it is completed.

- However when the running process becomes suspended as a result of it own action, say by waiting for I/O completion, another ready process may be scheduled.

Introduction :

- The memory system of a computer is constructed as a hierarchy of layer containing registers, cache memory, main memory, hard disks, and magnetic tapes.
- Two kind of memory management are performed during the operation of an Operating System
- This Involve Allocation an deallocation of memory...
  - To program or process
  - Within a program or process

Memory Management

- Memory Management is the act of managing computer memory.
- In its simpler forms this Involve providing ways to allocate portion of memory to program at their request , and freeing it for reuse when no longer need.
- Garbage collection is the automated allocation and deal location of computer memory resource for a program.

Goals

- The principals goals of the operating systems memory management are :
- To provide memory space to enable several processes to be executed at the same time
- To share memory space between processes
- To protect each programs resources

Features

- Memory management system on multi tasking operating system usually deal with the following issues.

Reallocation

- In system with virtual memory program in memory must be able to reside in different parts of the memory different time
- Memory management in the operating system should there for be able to relocate program in memory and handle memory references in the code of the program so that they always in point to the right location in memory.

Protection

- Processes should not be able to reference the memory for another processes without permission .

Sharing

- Even though the memory for different processes is protected from each other different processes should be able to share information an their for access the same part of memory

Logical organization

- Programs are often organized in modules.
- Some, of this modules should be shared between different program some are read only some contain data that can be modified.

Physical organization

- Memory is usually divided into fast primary storage and slow secondary storage.

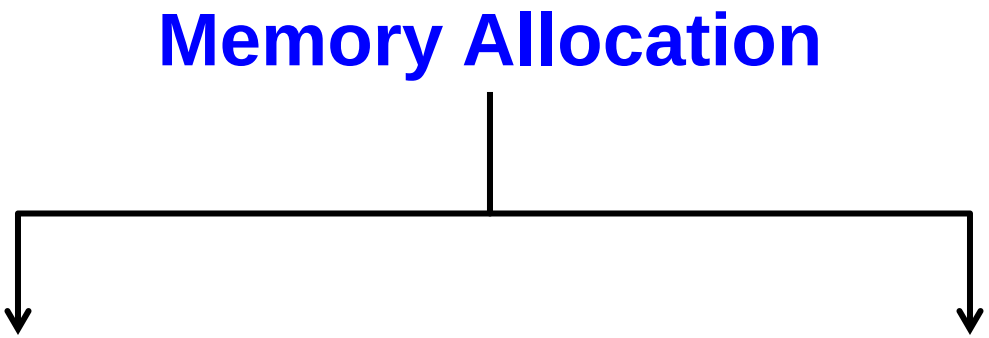
Memory Compaction

- The technique of relocating all occupied area of memory to one and of the memory so as to get one large block of free memory space is called compaction.

Memory Allocation :

- As described earlier, user programs are stored in main memory before execution.
- Memory should be allocated to various processes and data as per requirements.
- Memory has a limited capacity compared to disk.
- Ti is possible that more than one process may be executing simultaneously.

➢ Memory Allocation :



Contiguous Memory Allocation.

- **Single process Monitor.**
- **Multi-programs, with Fixed Partitioning**
- **Multi-programs, with Dynamic Partitioning**

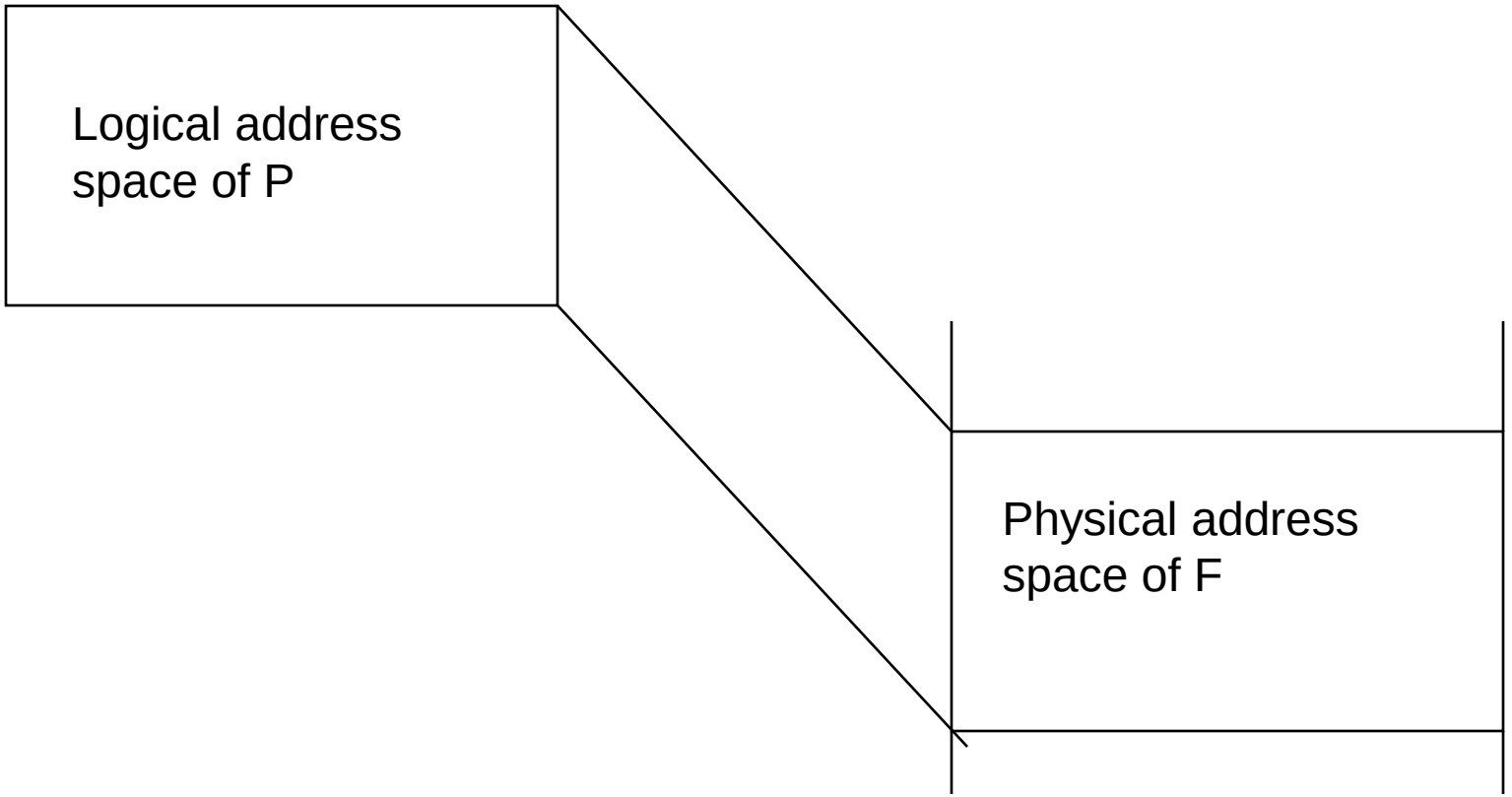
Non –Contiguous mem,Allocation.

- **Paging**
- **Segmentation**

➢ Contiguous Memory Allocation :

- This is simple and old method allocation. it is not used in modern operating system.
- If a contiguous a memory space of the required site is not available in the main memory the processes is made to wait until contiguous space of the required size is available.
- Logical address space not divided into any partitions. Also, Physical address space will be contiguous without any gaps.

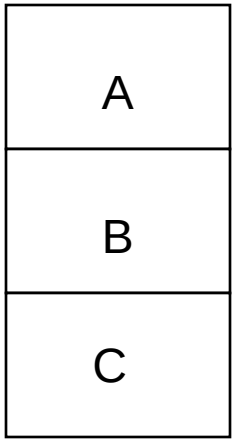
Contiguous Memory Allocation



➢ Non-Contiguous Memory Allocation :

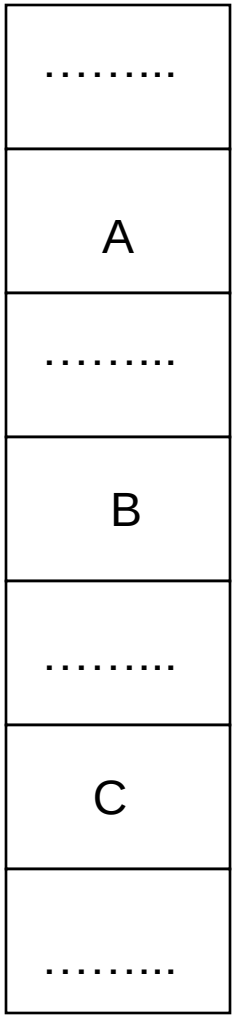
- This method is use by most modern operating system.
- Logical address space of process is divided into partition. And for each partition contiguous chunk or free memory is allocated and physical address space will not be contiguous now.

Non-Contiguous Memory Allocation



Local Address Space

Memory



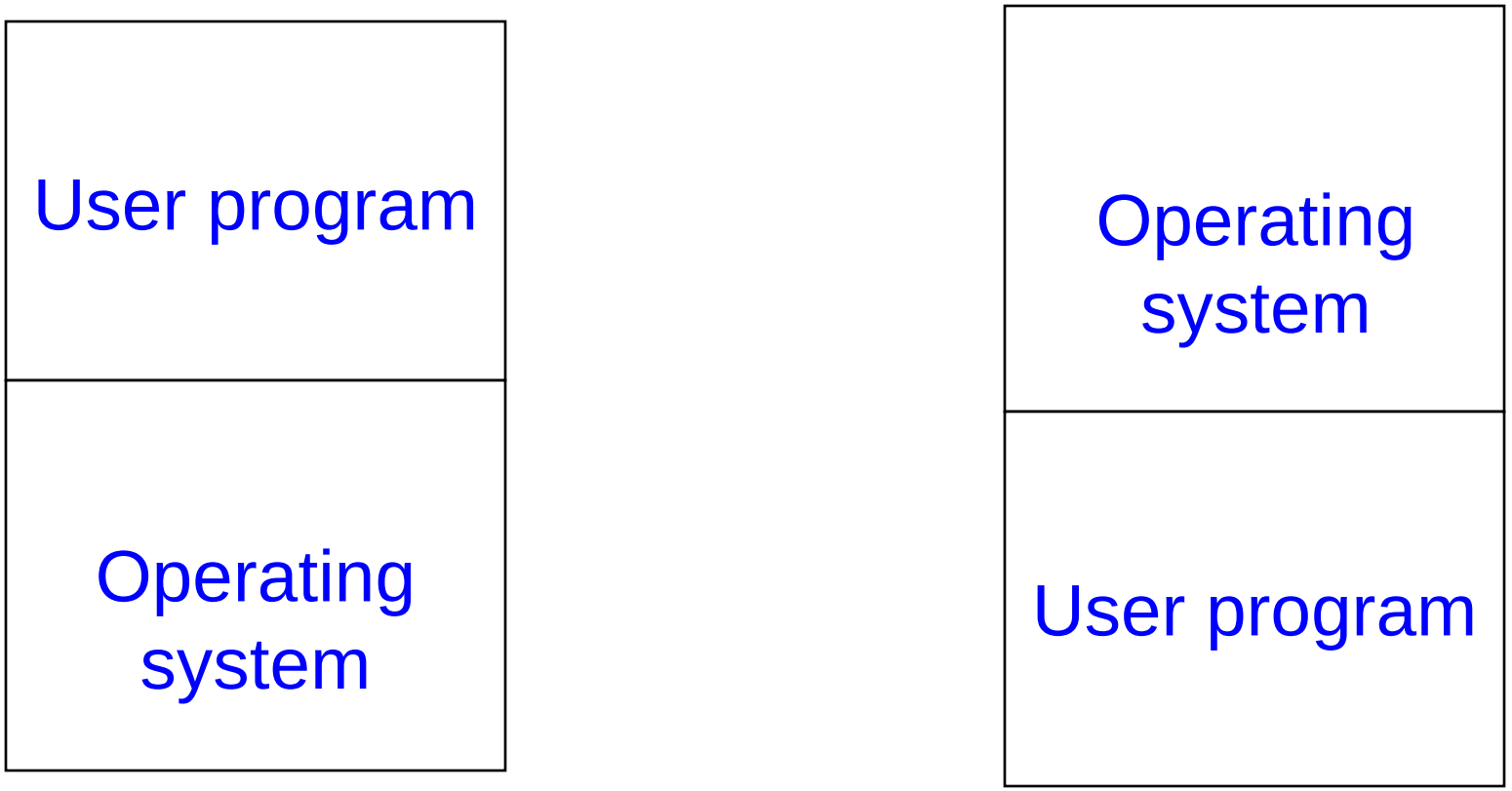
➢ Single process Monitor

- This is the simplest possible memory management scheme.
- Here, only single process is allowed to run and Memory is shared between the process and the operating system.
- The operating system keep track of only the lowest and highest addresses available for the user programs.

➢ Advantage/Disadvantage:

- Simple and easy to implement .
- Lowest utilization of CPU
- Memory wastage
- Dose not support Multiple - programming.

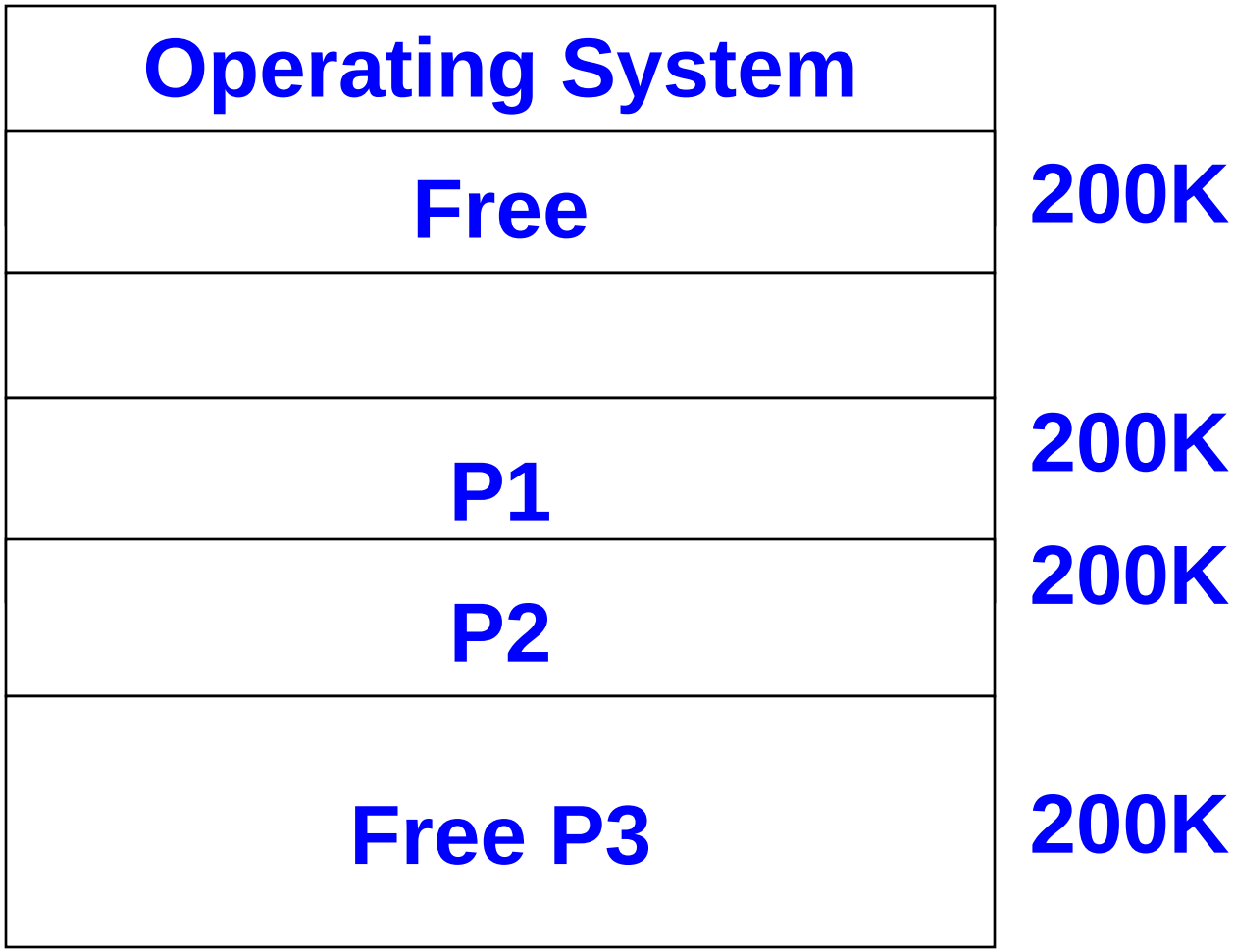
Single process Monitor



➢ Multiprogramming with Fixed Partition

- This method allows multiple process to execute simultaneously.
- Hear, memory is divided into fixed size partition. Size can be equal or unequal for different partition.
- Each partition can accommodate exactly one process, means only single process can be placed in one partition.
- If there is no free partition available of required size, than that process need to wait, such process will be put in queue.

Multiprogramming with Fixed Partition



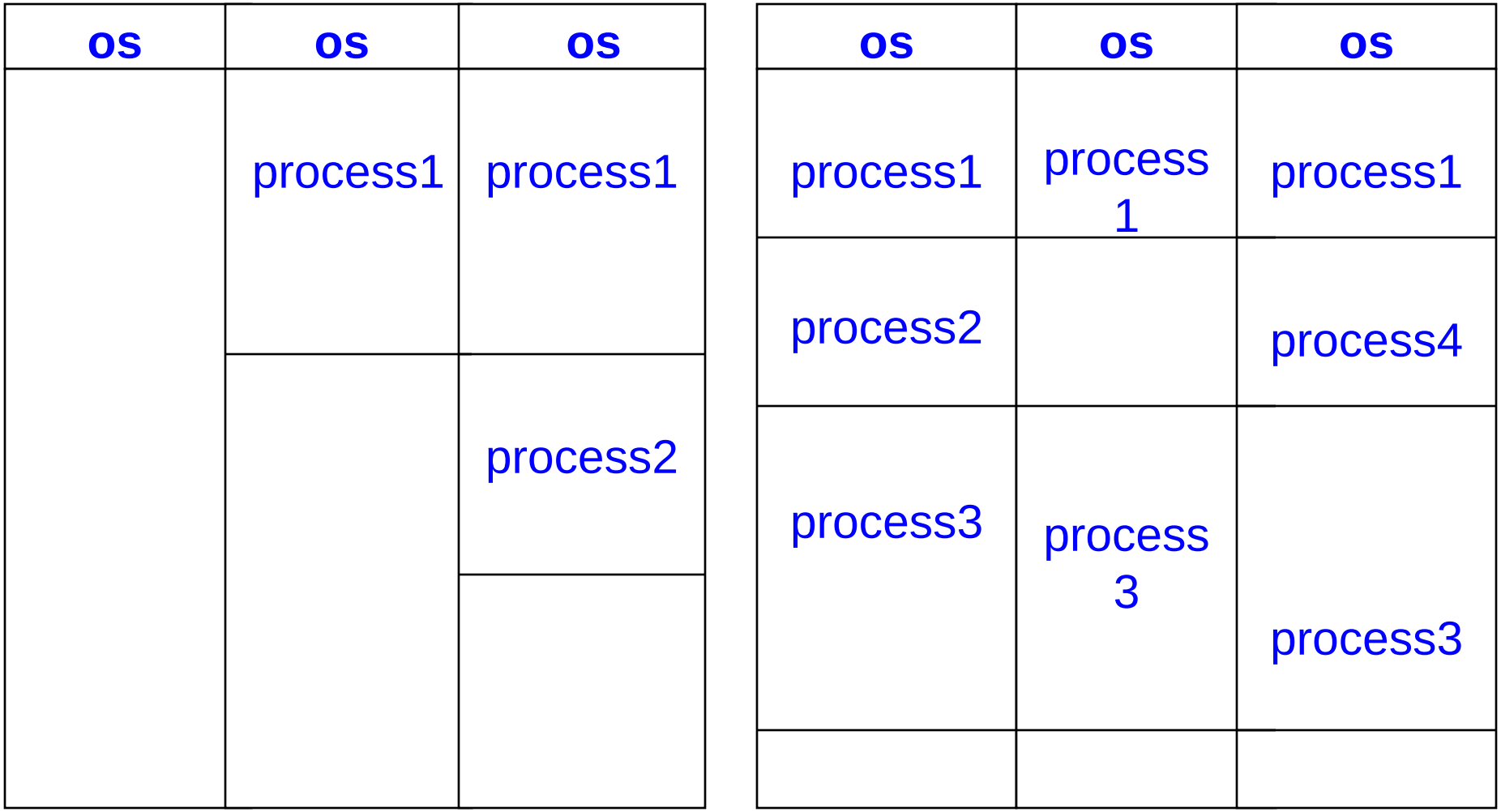
➢ Multiprogramming with Fixed Partition

- **Advantages :=>**
  - Simple and easy to implement.
  - Less additional hardware required.
  - Algorithm used is not very complicated .
- **Disadvantage:=>**
  - Internal fragmentation of memory.
  - Degree of multiprogramming depends on the number of partitions.

Multiprogramming with Dynamic Partition

- Variable partitioning (MVT-Multiple Variable tasks), primary used in a batch environment, implies the division of memory of into fragments in a dynamic manner.
- The size and number of programs will be decided during the run-time.
- When a program is allocated space it is loaded into the memory and then it can complete for CPU.
- When the program terminates it, releases the memory that can be given to another program.

Multiprogramming with Dynamic Partition



(a) Initial state (b) Pro1 enters (c) Pro2 enters (d) Pro3 enters (e) Pro2 terminates (f) Pro4 enters.

Advantages/Disadvantages

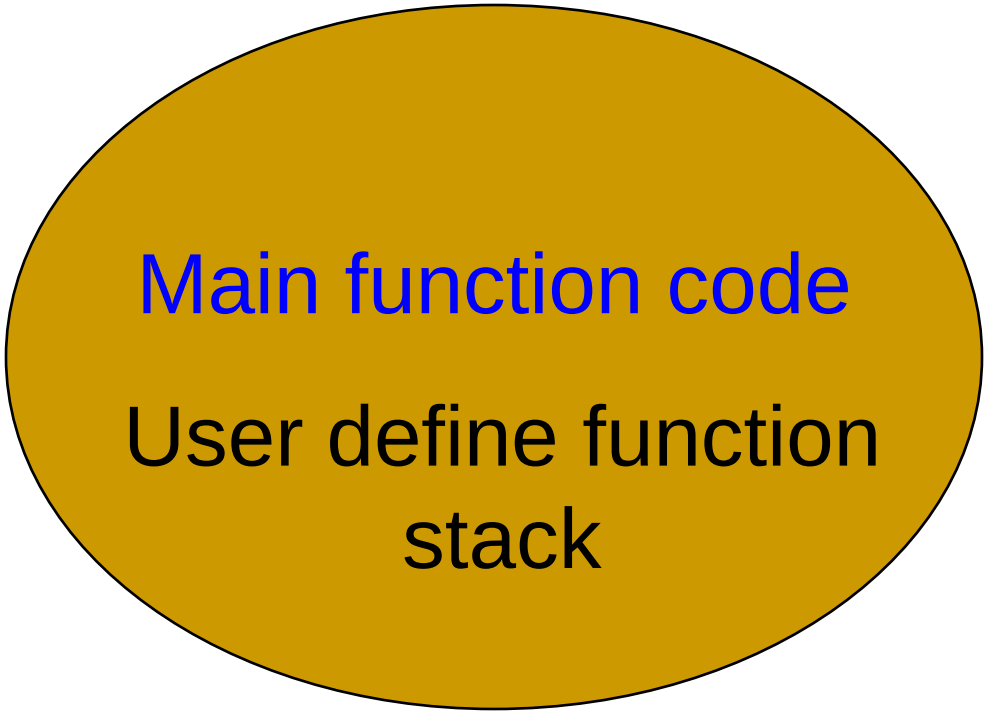
- Advantages:
  - Better utilization of memory.
  - Degree of multi programming is not fixed here.
- Disadvantage:
  - External fragmentation

➢ Non-contiguous memory allocation

- **Paging :** The logical address space of process is divided into block of fixed size called pages.
- Also the physical memory is divided into block of fixed size called frames.
- The pages and frames will be of the same size. It various between 512 bytes to a few MB.

Segmentation

- The logical address space of a process is divided into blocks of varying size is called segmentations.
- Each segment contains a logical unit of a process
- Each segment can be consider as a completely independent address space.



Logical add space

Physical memory

➢ Advantage / Disadvantage

- **Advantage :**
  - All segments are independent from each other.
  - Sharing of procedures and data among various process is simple.
  - There is no internal fragmentation.
- **Disadvantage :**
  - External fragmentation is possible, which require memory de-fragmentation.

Virtual Memory

- Virtual memory is computer system technique which given an application program impression that it has contiguous working memory.
- A computer system is said to be use virtual memory if a memory address used by an instruction in likely to be different from the effective address of memory location accessed during its execution.

Types of virtual memory

1. Address translation:

- A address translation can be performed in a table based manner where table contains start address of various components of program.
- In an address comp is used to indeed this table to obtain its start address in memory.

2. Logical and physical address:

- A logical address is address of an instruction or data word as used by a program.
- A physical address is effective memory address of an instruction or data word.

Cont...

3. Paged virtual memory :

- All implementations of virtual memory divide virtual address space of an application program into pages a page is a block of contiguous virtual memory address.

4. Segmented virtual memory :

- Some system such as the Burroughs large systems do not use paging to implement virtual memory.
- Instead they are segmentation so that an application virtual address space is divided into length segments.

Virtual memory using paging technique

- Paging is process of saving inactive virtual memory pages to disk and restoring them to real memory when required.
- A page is block of contiguous virtual memory addresses.
- Pages are usually at least 4k bytes in size and system or large amount of read memory generally use large page sizes.

Cont...

- **Page table:**
  - If there is only one different application which are running at the same time share single virtual address space.
  - They are different part of single range of virtual address space.
  - Page table collect logical address, physical address and indicator and these of description use at runtime.

Cont...

- Page supervisor:
  - If dynamic address translation hardware realest a page fault exception paging supervisor searches page file for the page contain the required virtual address reads in into real physical memory.
  - Update the page table to reflect the new location of the virtual address and finally dynamic address translation in mechanism to start the search again.

➢ Virtual memory using segmentation

- Segmentation is a memory management scheme which supports programmers view of memory.
- Formation of these segments varies from one compiler to another a pascal compiler might create separate segments for
  - Code of each procedure.
  - Global data.
  - Local data or variable.
  - Stake for calling procedure and storing its parameter.
- Array might be formed as separate segments.

Segmented virtual memory

- Some system such as Burroughs system do not use paging to implement virtual memory.
- They are segmentation so that an application virtual address space is divided into variable length segments.
- To obtain it the processor looks up segment number is a segment table to find a segment is present in main memory.

➢ Demand paging

The needed to implement demand paging:

- Page faults:
  - A field PMT entry page pi indicates whether page currently exists in memory.
  - If pi does not exit in memory when referenced ATU raise a missing page interrupt also called page fault.
- Page in and page out operation:
  - When a page fault occurs during a reference to page pi VM handle finds a free page block in memory and load pi in it this called page in operation for a pi.