Chapter-6

Memory management

Memory management

- In a uni-programming system,the main memory is divided into two parts, one part for Operating system and the other part for the program currently running.
- In multi-programming system,the user part of the memory is subdivided among several processes.
- The task of sub division is done by the operating system and is known as memory management.

Memory management requirements

- Hardware
- Address binding
- Dynamic loading
- Dynamic linking
- Logical and physical address
- overlays

Basic hardware

- CPU can only access its registers and main memory.
- The data stored in the secondary memory must be transferred into the main memory before the CPU can work with it.
- Each process can have base register and limit register.
- Base register holds the starting address of the process and limit register holds the size of the process

Address binding

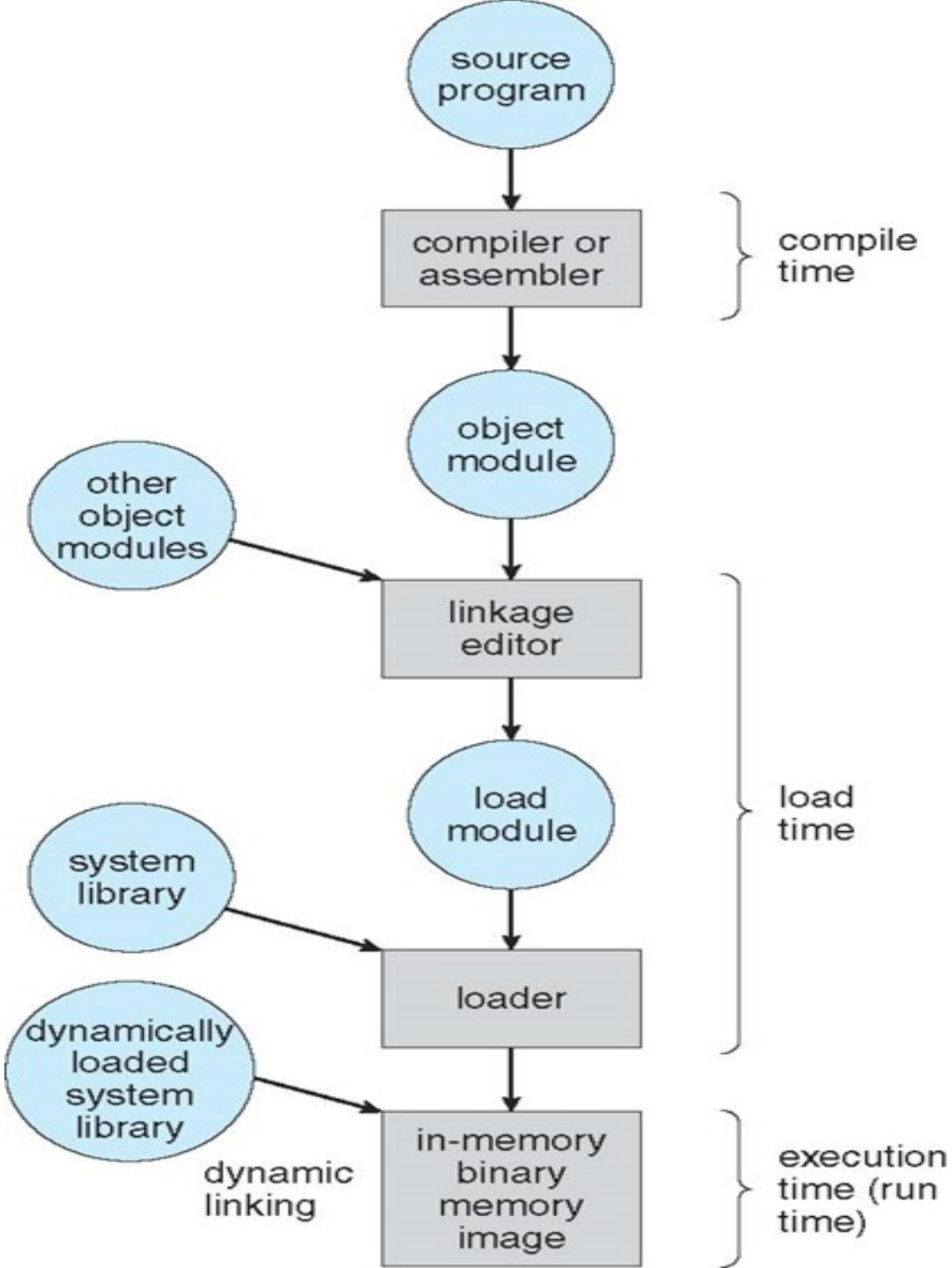
- Address binding of instructions and data to memory addresses can happen at three different stages
  - **Compile time**: If location of the process in the memory is known at compile time, **absolute code** can be generated;
  - **Load time**: If location of the process in the memory is not known at compile time, **relocatable code** can be generated;
  - **Execution time**: Binding delayed until run time if the process can be moved during its

Dynamic loading

- The main program is first loaded into the main memory and executed.
- Next when a routine calls another routine, the calling routine first checks to see whether that routine has been loaded.
- If not ,the relocatable loader ,loads the required routine into the memory.

Dynamic linking

- A dynamic linker is a special part of an operating system that loads external shared libraries into a running process and then binds those shared libraries dynamically to the running process.


source program
compiler or assembler
compile time
object module
other object modules
linkage editor
load module
load time
system library
loader
dynamically loaded system library
dynamic linking
in-memory binary memory image
execution time (run time)

- **Logical address or virtual address**

An address generated by CPU is called logical address.

- **Physical address**

An address generated by Main memory is called physical address.

- **Logical address space**

A set of all logical addresses generated by a program is known as logical address space.

- **Physical address space**

- Memory Management unit (MMU)

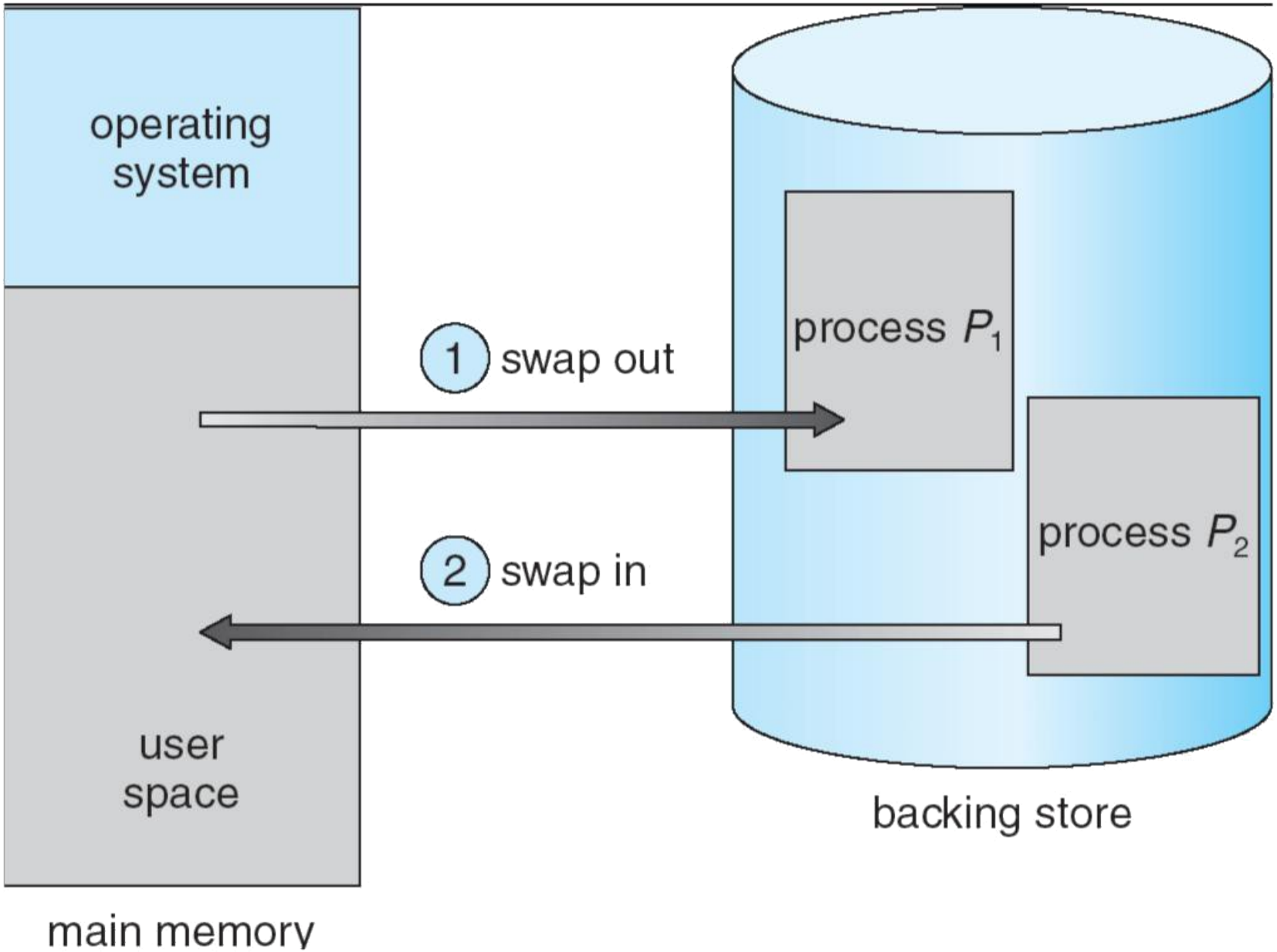
The run time mapping from a logical address to physical address is done by a hardware device called Memory management unit.

Overlays

- Overlays are used to enable a process to be larger than the amount of memory allocated to it.
- The main objective of the scheme is to keep only those instructions and data in memory, which are required at that time.
- Other instructions are loaded into the memory whenever they are required.

swapping

- Swapping is a mechanism in which a process can be moved temporarily from 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 secondary storage to main memory.


operating system
process P_1
1 swap out
process P_2
2 swap in
user space
backing store
main memory

Contiguous memory allocation

- **Contiguous memory allocation** is a classical **memory allocation** model that assigns a process in consecutive **memory** blocks (that is, **memory** blocks having consecutive addresses).

Types of partitioning in contiguous memory allocation

- **Fixed partitioning**

Main memory is divided into several partitions of fixed size. Each partition can accommodate only one process for execution.

- **Dynamic partitioning**

Partitions are created dynamically so that each process is loaded into the partition of exactly the same size as that of process.

Three strategies in contiguous memory allocation

- **First-fit**

First-fit allocates the first free partition which is large enough to accommodate the process.

- **Best-fit**

Best-fit allocates the smallest free partition that is large enough for that process.

- **Worst-fit**

First-fit	Best-fit	Worst-fit
Allocates first free partition which is big enough	Allocates the smallest free partition	Allocates the largest partition
Execution fastest	Slower execution	Slower execution
Lesser efficient utilization of memory	Most efficient utilization of memory	Lesser efficient utilization of memory

Fragmentation

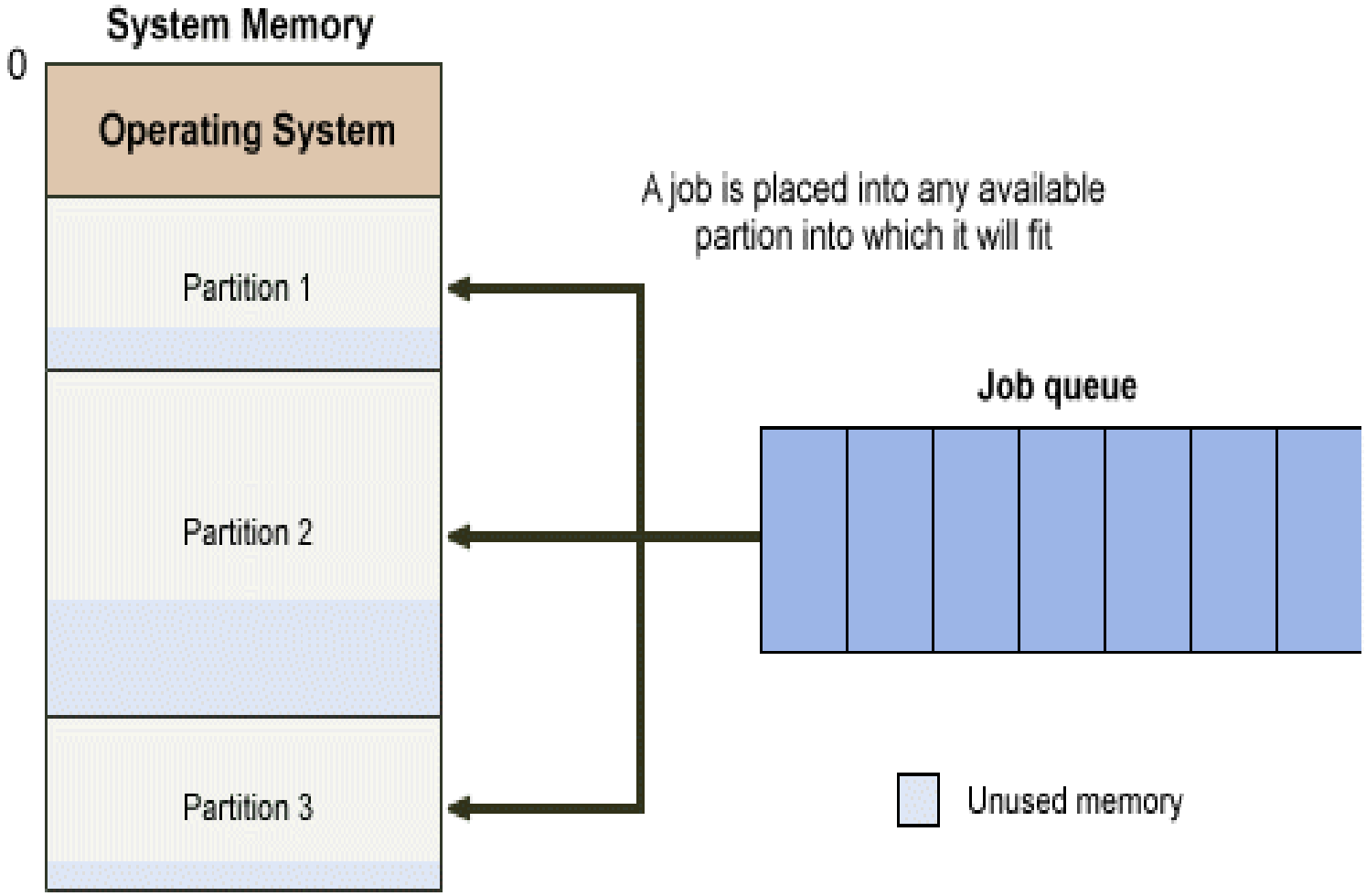
- Fragmentation means wastage of memory.
- **Internal fragmentation**

Internal fragmentation is the wasted space within each allocated block.

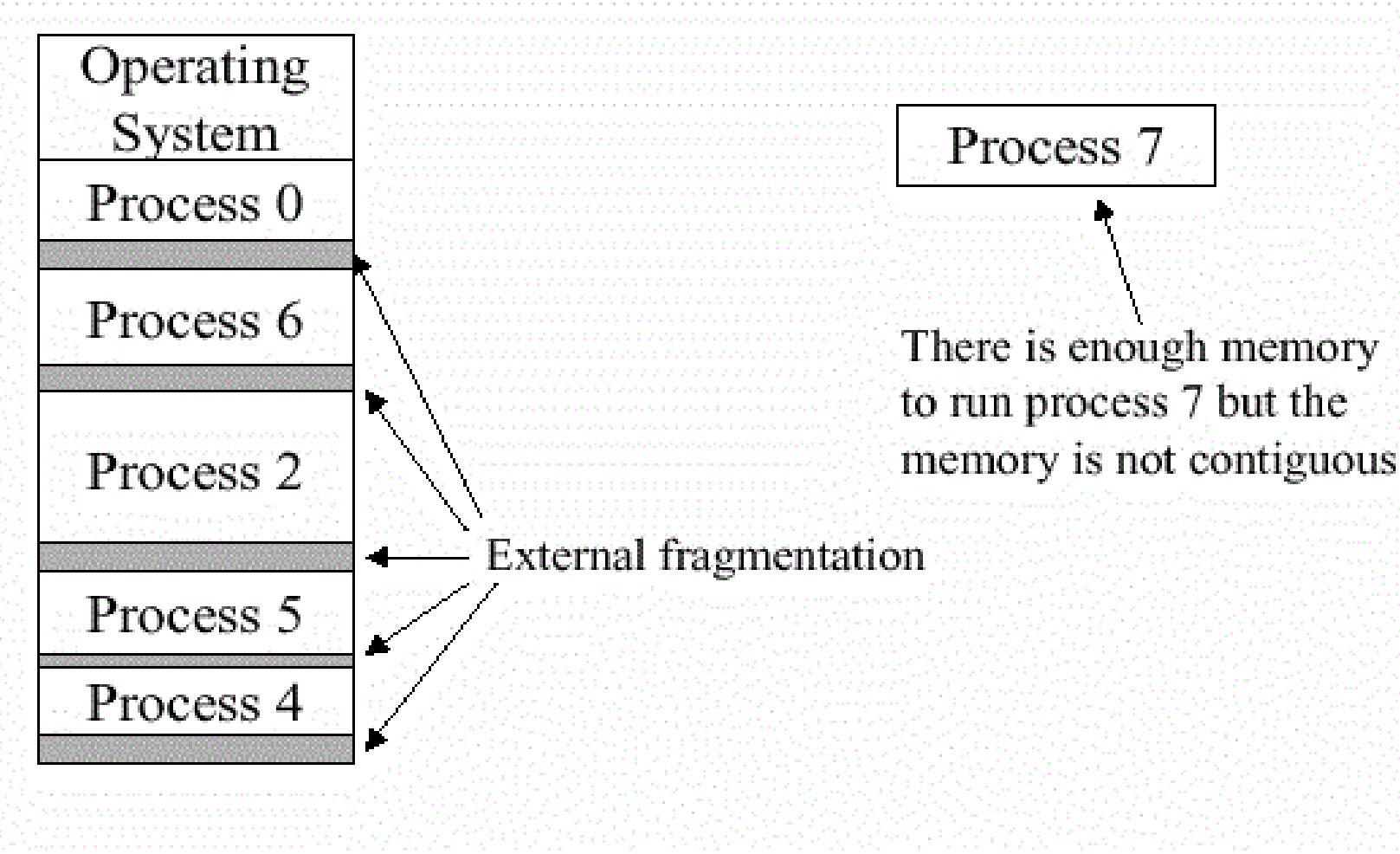
- **External fragmentation**

It exists when enough to the memory space exists to satisfy a request, but it is not contiguous; storage is fragmented into a

Internal fragmentation



External fragmentation



Dynamic Partitioning

- Partitions are created dynamically so that each process is loaded into a partition of exactly the same size as that of process.
- In dynamic partition , there is no internal fragmentation.
- It Support multi-programming.
- External fragmentation is available.

Advantages of Dynamic partitioning

- Supports multi-programming.
- Efficient usage of memory.
- Since partitions change dynamically, there is no internal fragmentation.

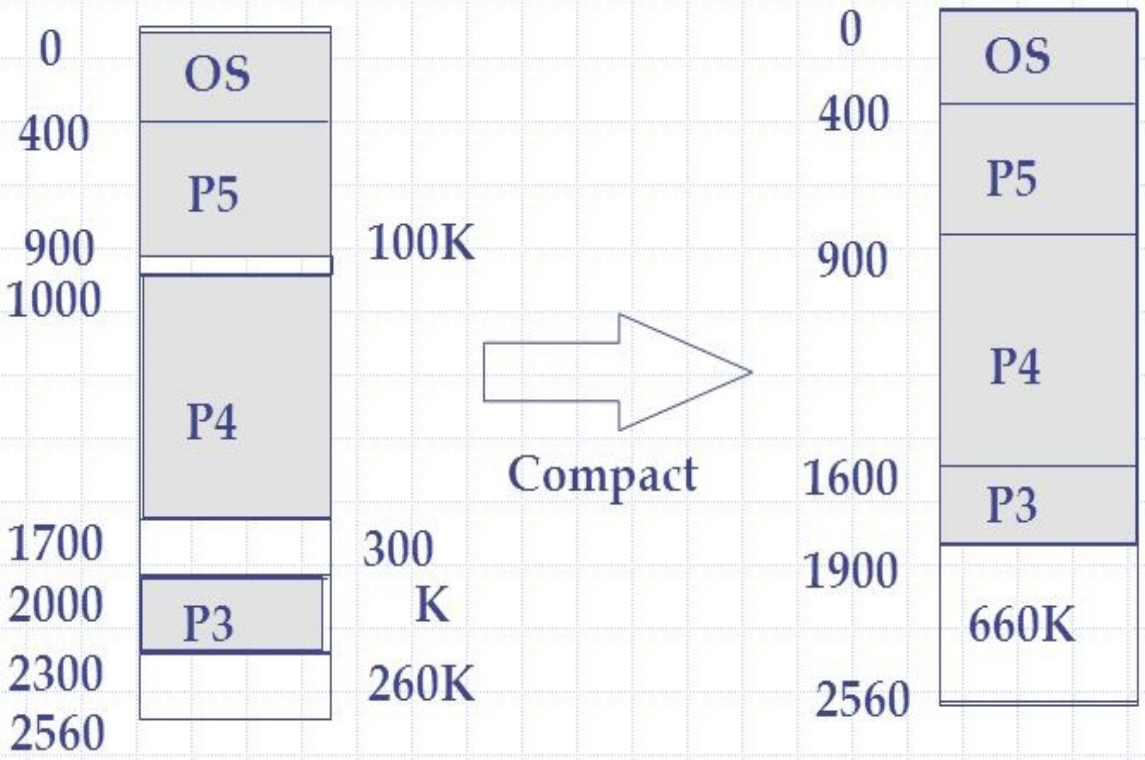
Disadvantages of dynamic allocation

- External fragmentation

Compaction

- Compaction is a technique of combining all the free spaces together into a large block by pushing all the process downwards as far as possible.

Compaction


0
OS
400
P5
900
100K
1000
P4
1700
300
2000
K
P3
2300
260K
2560
Compact
0
OS
400
P5
900
P4
1600
P3
1900
660K
2560

Difference between fixed and dynamic partition

FIXED	DYNAMIC
Main memory is divided into a number of static partitions	Partitions are created dynamically
Internal fragmentation	External fragmentation
Supports multi-programming	Supports multi-programming
Easy to implement	Easy to implement