EE6253 - Operating Systems & Network Programming

O.G.Y.N. Gamlath
Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna

Lecture 07 – Memory Management (Reading Material)

Intended Learning Outcomes

- Introduction to Memory
- Logical and Physical address space
- Static and Dynamic Loading
- Swapping
- Contiguous Memory Allocation
- Memory Allocation
- Fragmentation
- Paging

Introduction to Memory Management

- The task of subdividing the memory among different processes is called Memory Management.

- The main aim of memory management is to achieve efficient utilization of memory.

Logical and Physical Address Space

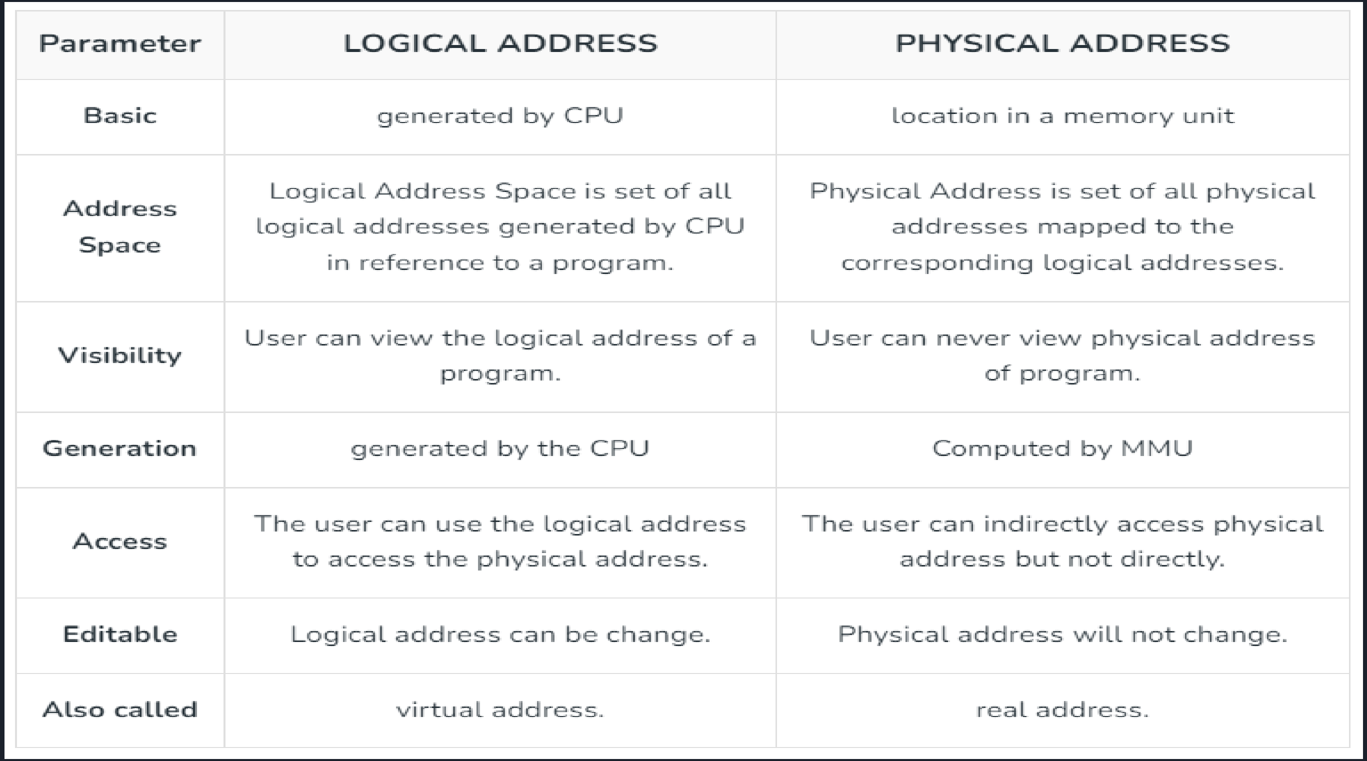
Logical Address Space:

- Address generated by the CPU is known as a Logical Address.
- Logical address space can be defined as the size of the process. A logical address can be changed.

Physical Address Space:

- Address seen by the memory unit is commonly known as a Physical Address.
- Set of physical addresses corresponding to these logical addresses is known as Physical address space.
- Physical address is computed by MMU. Physical address always remains constant.

Logical and Physical Address Space



Parameter	LOGICAL ADDRESS	PHYSICAL ADDRESS
Basic	generated by CPU	location in a memory unit
Address Space	Logical Address Space is set of all logical addresses generated by CPU in reference to a program.	Physical Address is set of all physical addresses mapped to the corresponding logical addresses.
Visibility	User can view the logical address of a program.	User can never view physical address of program.
Generation	generated by the CPU	Computed by MMU
Access	The user can use the logical address to access the physical address.	The user can indirectly access physical address but not directly.
Editable	Logical address can be change.	Physical address will not change.
Also called	virtual address.	real address.

Static and Dynamic Loading

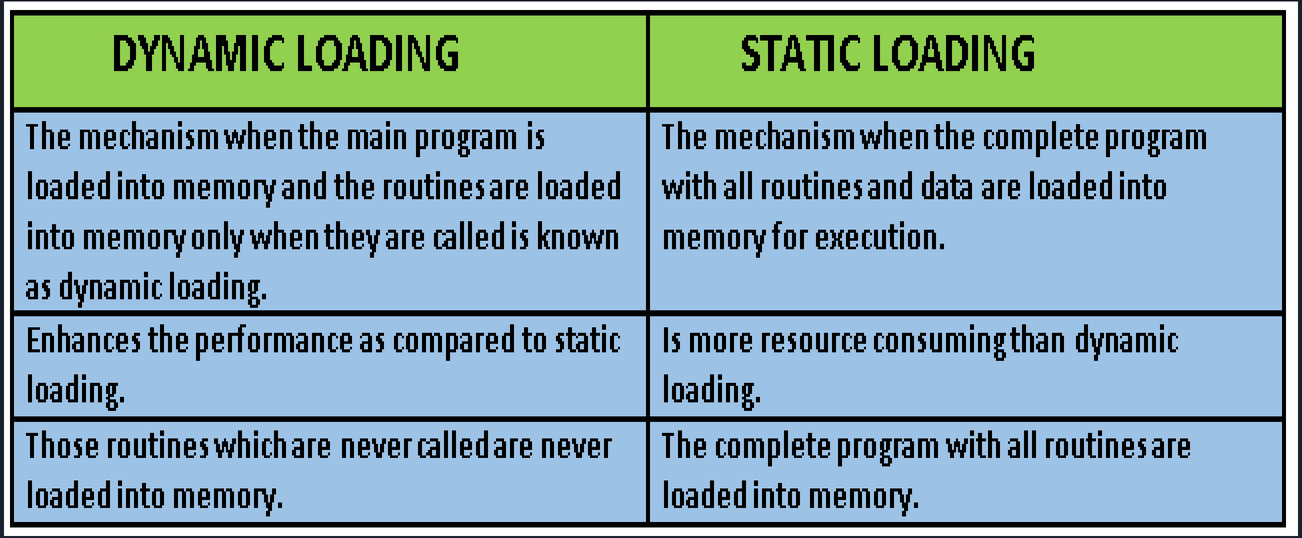
Static Loading:

- Static loading is the process of loading the complete program into the main memory before it is executed.

Dynamic Loading:

- Entire program and all data of a process must be in physical memory to execute the process.
- Process size is restricted by the amount of physical memory available.
- Dynamic loading is utilized to ensure optimal memory consumption.
- In dynamic loading, a routine is not loaded until it is invoked.
- All of the routines are stored on disk in a reloadable load format.

Static and Dynamic Loading

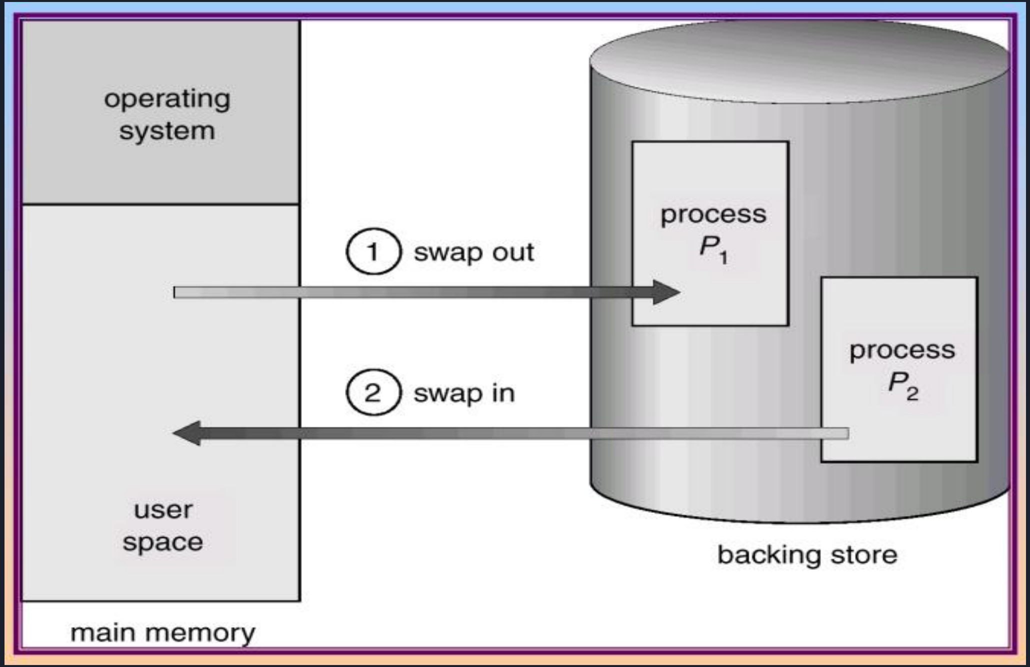


DYNAMIC LOADING	STATIC LOADING
The mechanism when the main program is loaded into memory and the routines are loaded into memory only when they are called is known as dynamic loading.	The mechanism when the complete program with all routines and data are loaded into memory for execution.
Enhances the performance as compared to static loading.	Is more resource consuming than dynamic loading.
Those routines which are never called are never loaded into memory.	The complete program with all routines are loaded into memory.

Swapping

- It is a process of swapping a process temporarily into a secondary memory from the main memory, which is fast compared to secondary memory.
- A swapping allows more processes to be run and can be fit into memory at one time.
- Swapping is also known as roll-out, because if a higher priority process arrives and wants service, the memory manager can swap out the lower priority process and then load and execute the higher priority process.
- After finishing higher priority work, the lower priority process swapped back in memory and continued to the execution process.

Swapping



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

Contiguous Memory Allocation

- Allocation of memory becomes an important task in OS.
- The memory is usually divided into two partitions: one for the OS and one for the user processes.
- We normally need several user processes to reside in memory simultaneously.
- Therefore, we need to consider how to allocate available memory to the processes that are in the input queue waiting to be brought into memory.
- In adjacent memory allotment, each process is contained in a single contiguous segment of memory.

Memory Allocation

- To gain proper memory utilization, memory allocation must be allocated efficient manner.

- One of the simplest methods for allocating memory is to divide memory into several fixed-sized partitions and each partition contains exactly one process.

Multiple partition allocation:

- In this method, a process is selected from the input queue and loaded into the free partition.

- When the process terminates, the partition becomes available for other processes.

Memory Allocation

Fixed partition allocation:

- In here, OS maintains a table that indicates which parts of memory are available and which are occupied by processes.
- Initially, all memory is available for user processes and is considered one large block of available memory.
- This available memory is known as a “Hole”. When the process arrives and needs memory, we search for a hole that is large enough to store this process.
- If the requirement is fulfilled then we allocate memory to process, otherwise keeping the rest available to satisfy future requests.

Memory Allocation

- While allocating a memory sometimes dynamic storage allocation problems occur, which concerns how to satisfy a request of size n from a list of free holes.

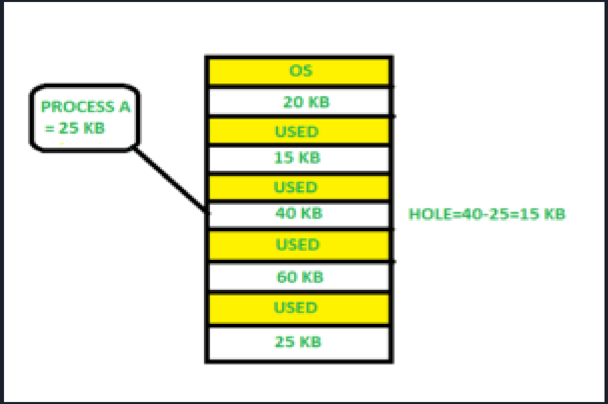
- There are some solutions to this problem:

    - First Fit
    - Best Fit
    - Worst Fit

Memory Allocation

First Fit

- In here, the first available free hole fulfil the requirement of the process allocated.
- In this diagram, a 40 KB memory block is the first available free hole that can store process A (size of 25 KB), because the first 2 blocks did not have sufficient memory space.

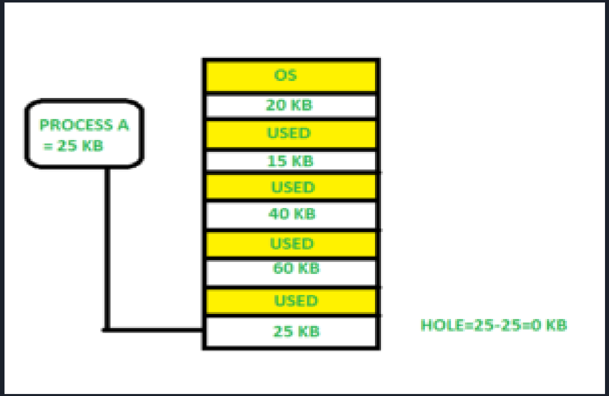


Memory Allocation

Best Fit

- In here, allocate the smallest hole that is big enough to process requirements. For this, we search the entire list, unless the list is ordered by size.

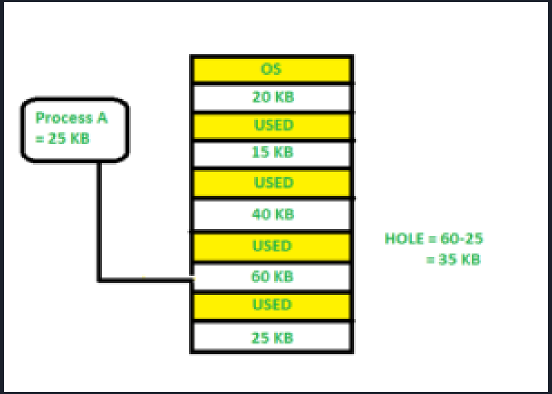
- In here, first, we traverse the complete list and find the last hole 25KB is the best suitable hole for Process A (size 25KB). In this method, memory utilization is maximum as compared to other memory allocation techniques.



Memory Allocation

Worst Fit

- In here, allocate the largest available hole to process. This method produces the largest leftover hole.
- Here, Process A (Size 25 KB) is allocated to the largest available memory block which is 60KB. Inefficient memory utilization is a major issue in the worst fit.



fragmentation

- Fragmentation is defined as when the process is loaded and removed after execution from memory, it creates a small free hole.

- These holes can not be assigned to new processes because holes are not combined or do not fulfill the memory requirement of the process.

- Then we must reduce the waste of memory or fragmentation problems.

- In OS two types of fragmentation:

    - ❖ Internal Fragmentation
    - ❖ External Fragmentation

Paging

- Paging is a memory management scheme that eliminates the need for a contiguous allocation of physical memory.
- This scheme permits the physical address space of a process to be non-contiguous.
- A page is a logical memory unit in a program.
- Logical memory is organized into equal-sized pages or blocks.
- A frame is a type of physical memory unit.
- In the concept of paging, physical memory is organized into frames, which are equally sized memory blocks.

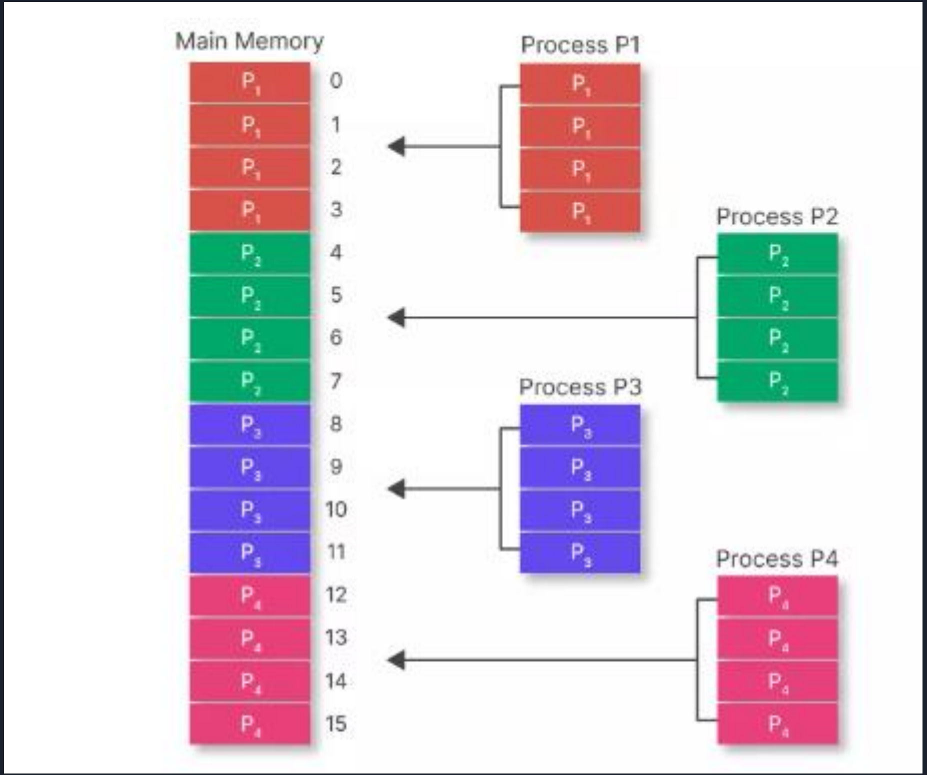
Paging

- The memory size of a new process is determined when it arrives.
- If a process has n pages in local memory, there must be n frames available in the system.
- To get maximum utilization of the main memory, the size of a frame should be the same as the size of a page.
- Paging is mostly used to store non-contiguous portions of a single process.

Paging - Example

- Let’s say the main memory size is 64B and the frame size is 4B then, the number of frames would be 64/4 = 16.
- There are 4 processes.
- The size of each process is 16B and the page size is also 4B then, the number of pages in each process = 16/4 = 4.
- These pages may be stored in the main memory frames in a non-contiguous form, depending on their availability.

Paging - Example



Thank You…