



HCMUS - FIT

Memory

Lecturer: VU THI MY HANG

OPERATING SYSTEM

Plan

- Fundamentals
- Contiguous Memory Allocation
- Swapping
- Issue: Fragmentation

Plan

- **Fundamentals**
- Contiguous Memory Allocation
- Swapping
- Issue: Fragmentation

Fundamentals

Address space

- Virtual (or logical) address: generated by CPU
- Virtual address space: all logical addresses (program address space)
- Physical address: actual address on physical (or main) memory
- Physical address space: all physical addresses (memory address space)

Fundamentals

Address space

0	MOV
4	
8	23
12	
16	ADD

Program address space

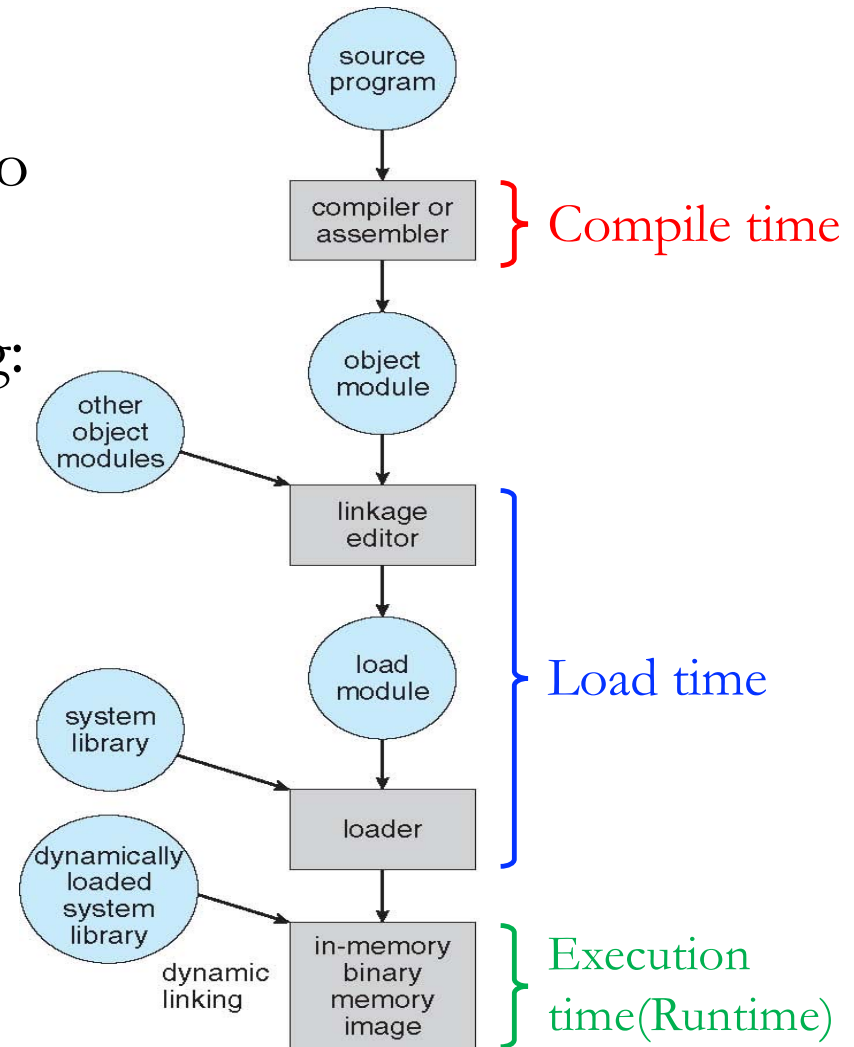
0	
.	
.	
.	
18000	MOV
18004	
18008	23
18012	
18016	ADD
.	
.	
.	

(Physical or Main) Memory

Fundamentals

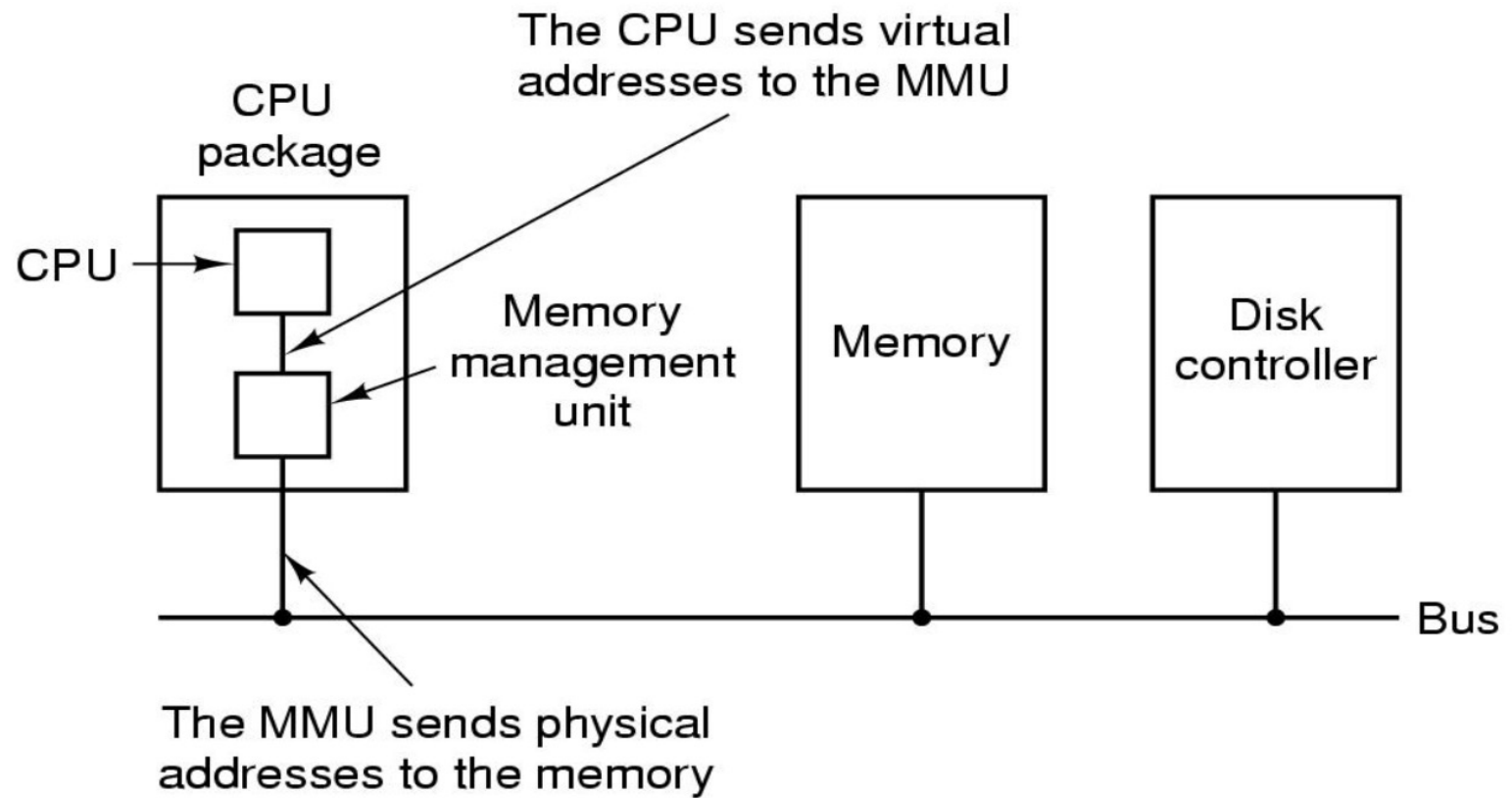
Address binding (Relocation)

- Mapping from one address space to another address space
- Address binding can be done during:
 - ✓ Compile time
 - ✓ Load time
 - ✓ Execution time



Fundamentals

Memory Management Unit (MMU)



Fundamentals

Memory Management

- Allocation/Deallocation
- Address binding (Relocation)
- Protection
- Sharing

Plan

- Fundamentals
- **Contiguous Memory Allocation**
- Swapping
- Issue: Fragmentation

Contiguous Memory Allocation

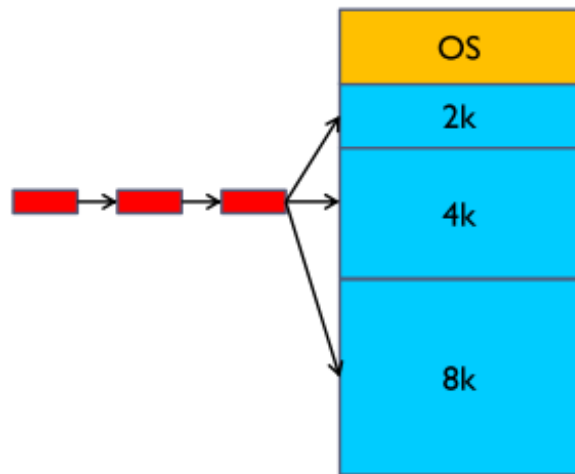
Characteristics

- Process: indivisible and must be loaded into a contiguous partition in the main memory
- (Physical/Main) memory: divided into different partitions before (fixed-partitions) or during (variable partitions) process loading

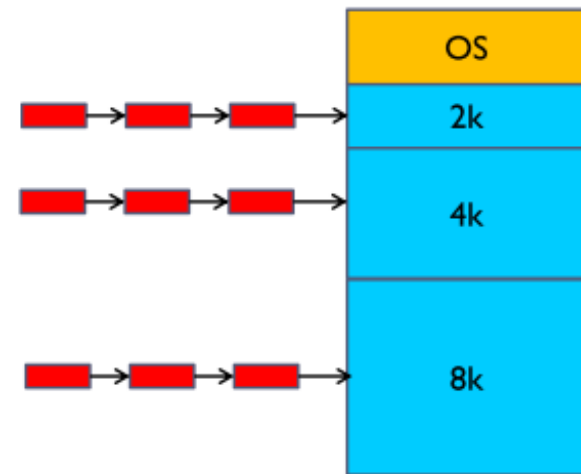
Contiguous Memory Allocation

Fixed-partitions

- Fixed-partitions: created before process loading
- Partition size may not be suitable to the process' needs



**Fixed-partition
with unique waiting queue**

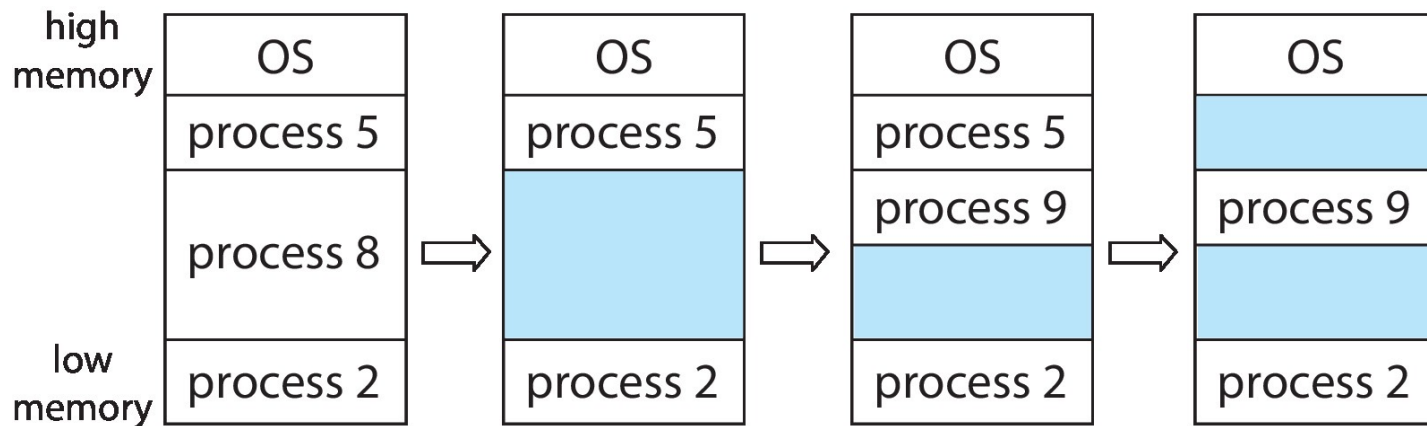


**Fixed-partition
with multiple waiting queues**

Contiguous Memory Allocation

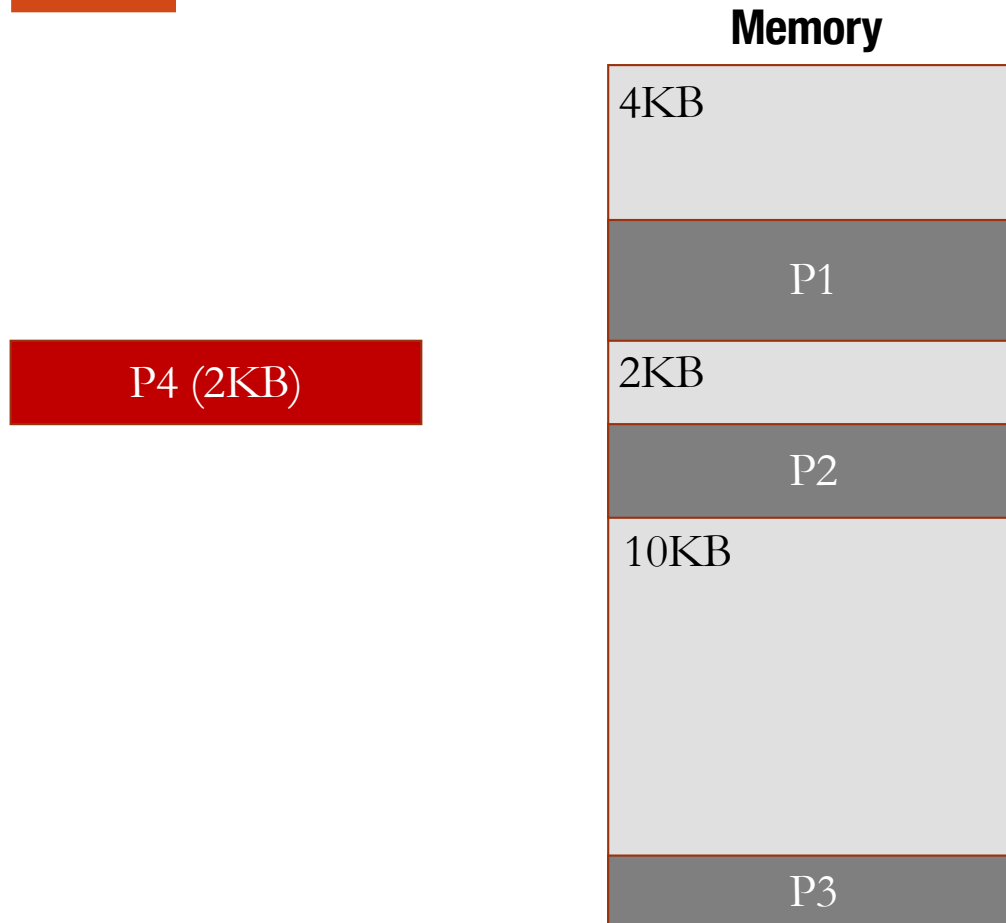
Variable-partitions

- Variable partitions: created during process loading
- Hole: block of available memory
- Partition allocated to a process is fit for the process' needs



Contiguous Memory Allocation

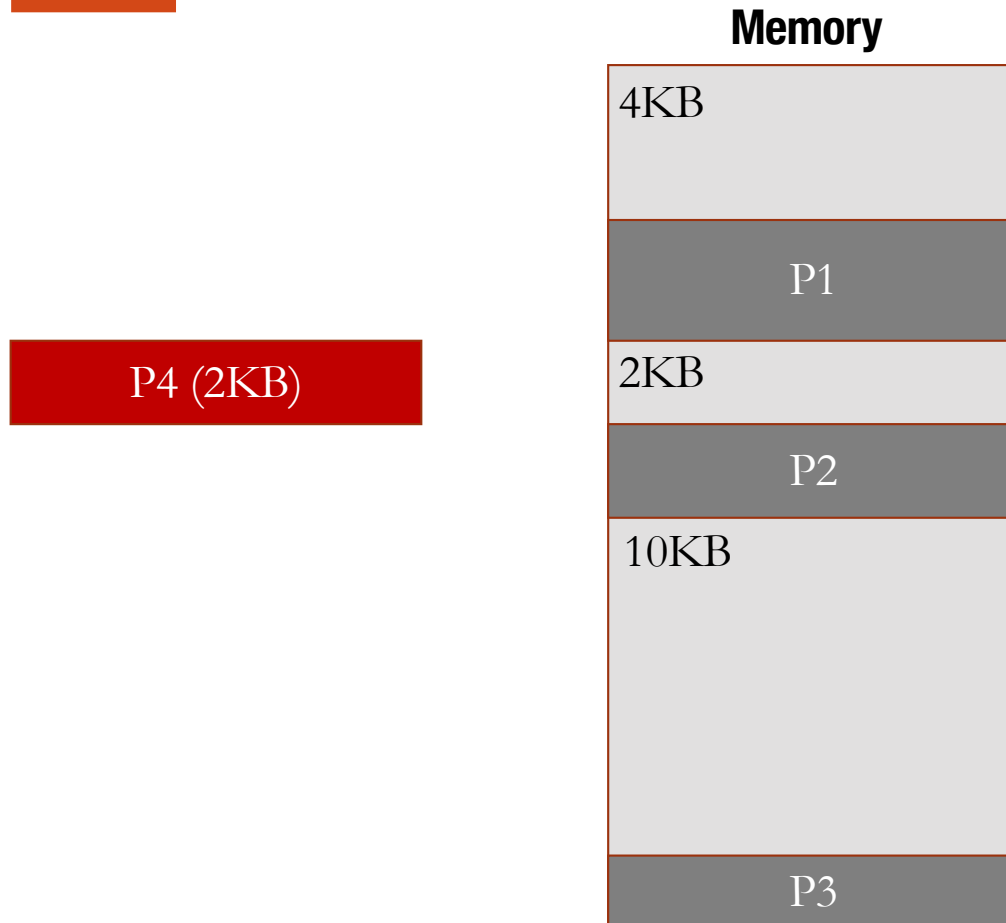
Memory Allocation Strategies



- **First-fit:** the process will be placed in the **first** free space which can contain it

Contiguous Memory Allocation

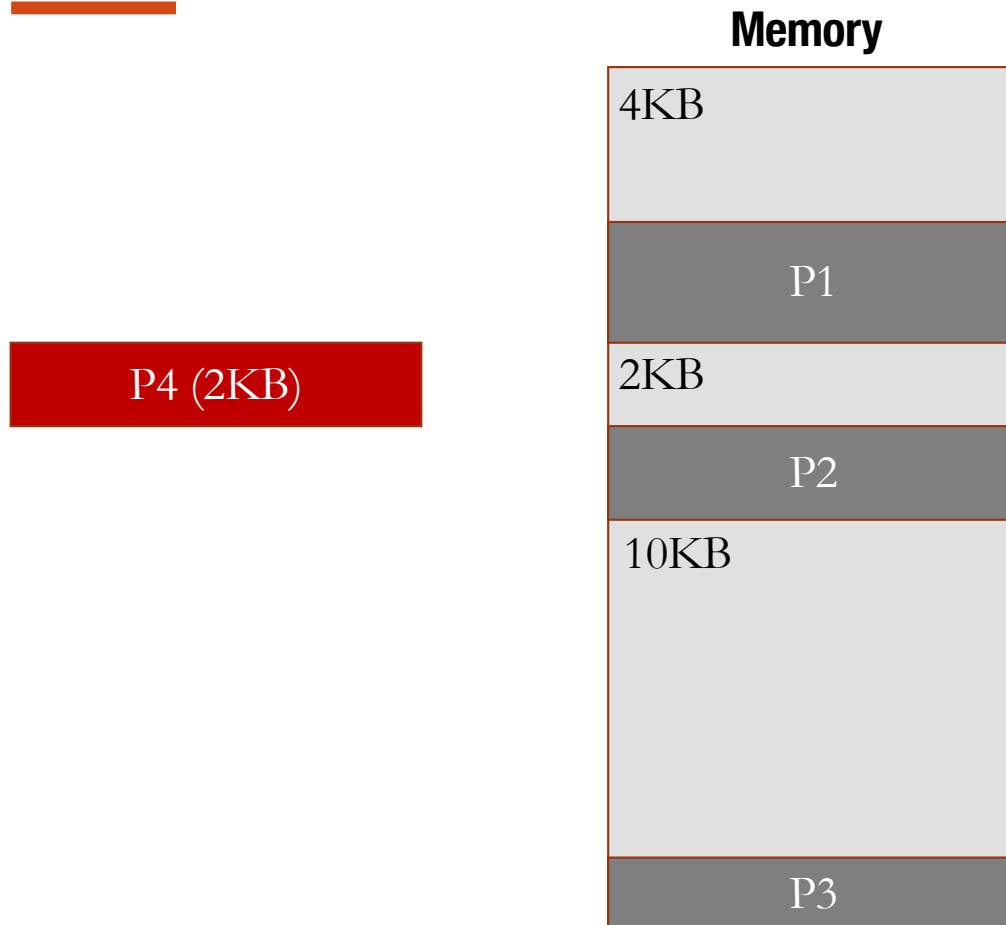
Memory Allocation Strategies



- **Best-fit:** the process will be placed in the **smallest** free space which can contain it

Contiguous Memory Allocation

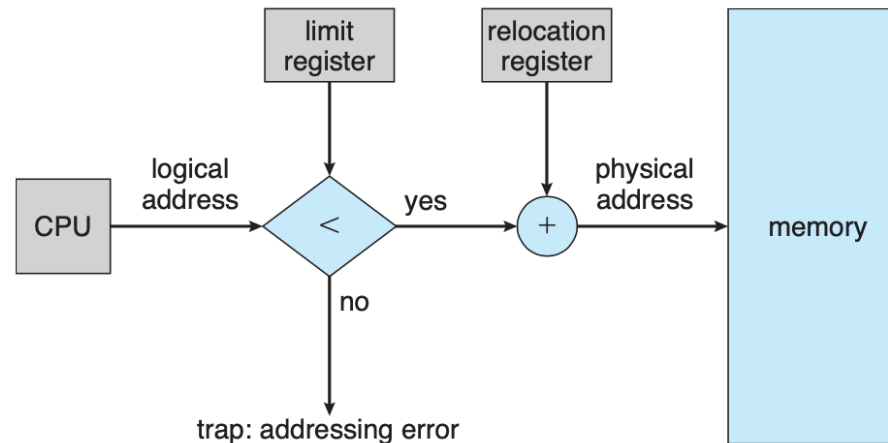
Memory Allocation Strategies



- **Worst-fit:** the process will be placed in the **largest** free space which can contain it

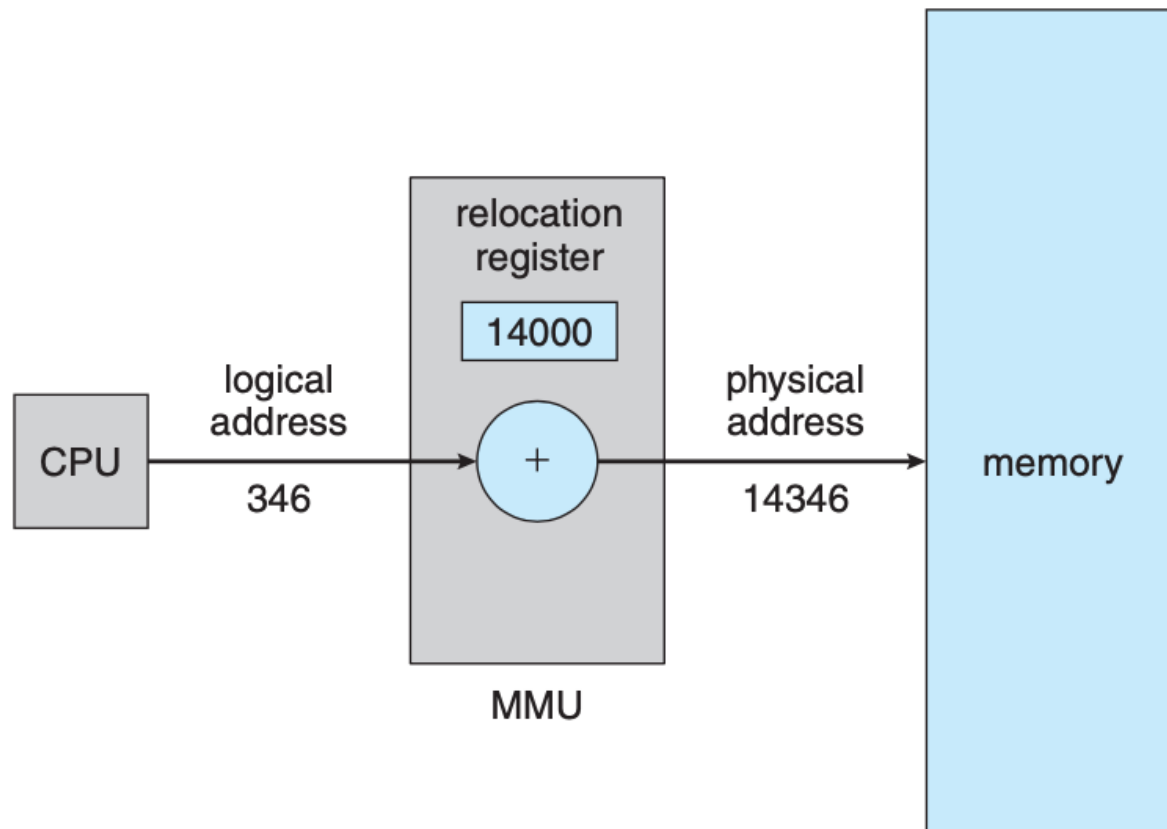
Address Protection

- **Relocation register** (base register): lower physical address limit (address min) of the memory region occupied by the process while it is executing
- **Limit register**: upper physical address limit (address max) of the memory region occupied by the process while it is executing



Contiguous Memory Allocation

Address Binding



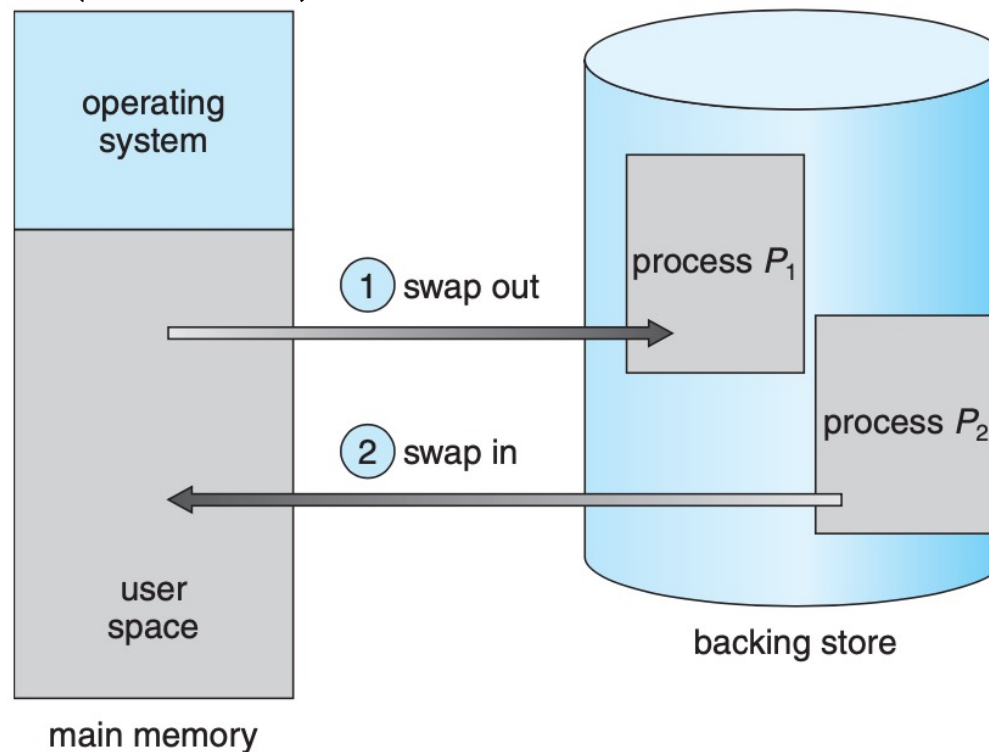
Plan

- Fundamentals
- Contiguous Memory Allocation
- **Swapping**
- Issue: Fragmentation

Swapping

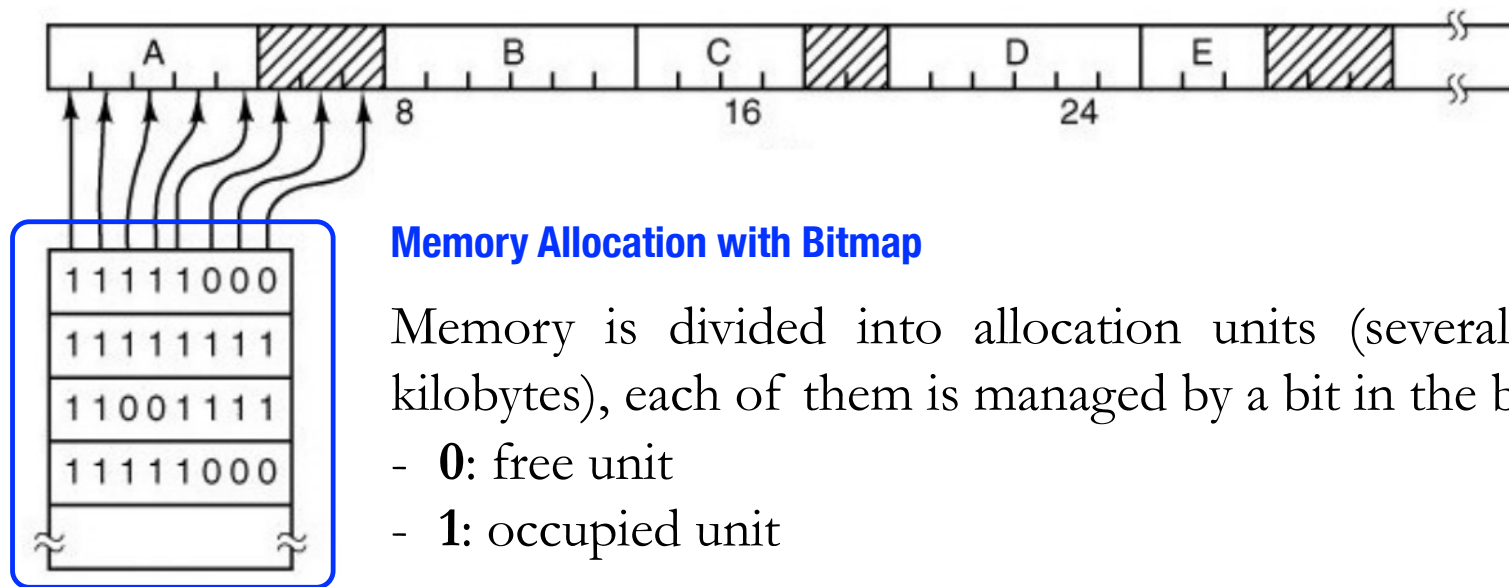
Principles

- Processes can be swapped in and out between memory and backing store (fast disk)



Swapping

Memory Management with Bitmap



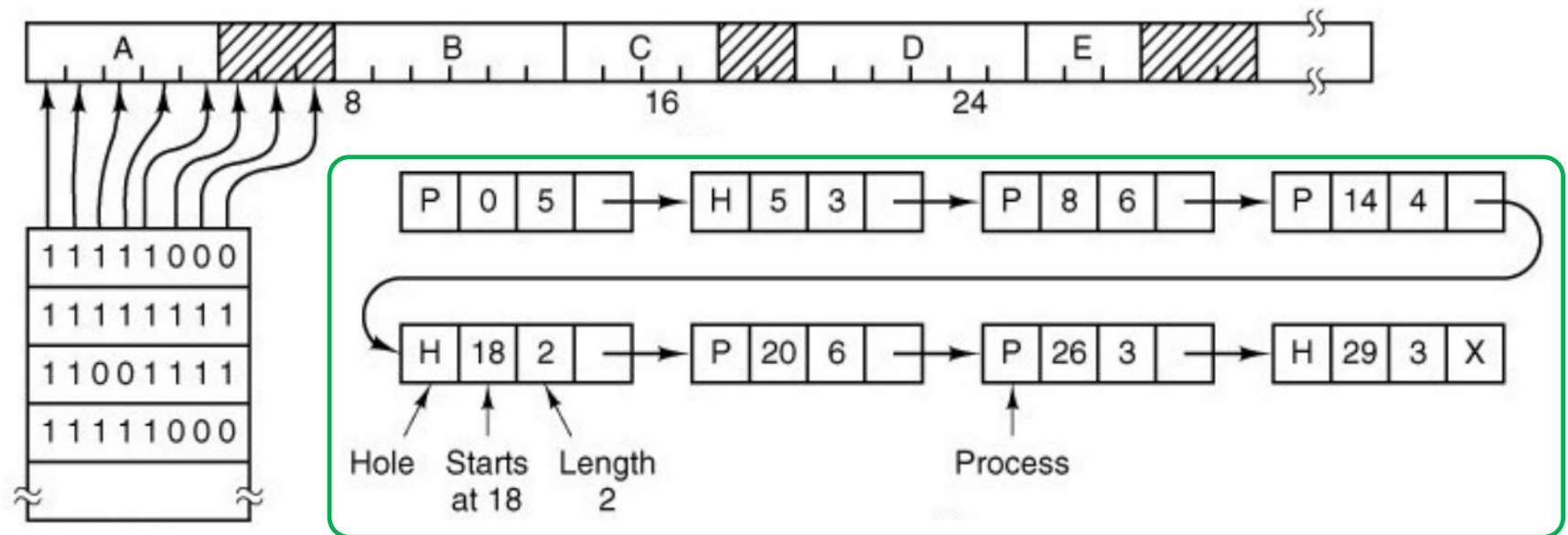
Memory Allocation with Bitmap

Memory is divided into allocation units (several bytes or kilobytes), each of them is managed by a bit in the bitmap:

- **0**: free unit
- **1**: occupied unit

Swapping

Memory Management with Linked List



Memory Allocation with Linked List

Memory allocation is managed by a linked list of segments:

- **H**: free segment (hole between two processes)
- **P**: occupied segment by a process

Swapping

Swap Time

- Assume that:
 - ✓ Process P1: 1024MBs
 - ✓ Transfer rate between memory and backing store: 50MBs/s
- Total swap time = $2 \times (1024 / 50) = 40.96\text{s}$

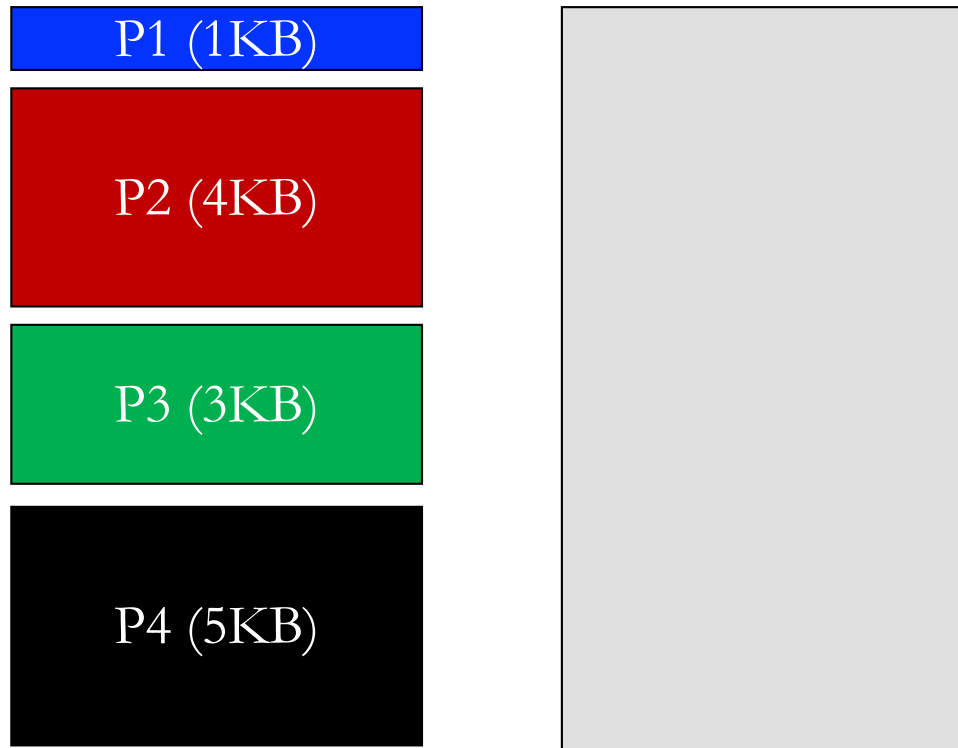
Plan

- Fundamentals
- Contiguous Memory Allocation (without Swapping)
- Swapping
- **Issue: Fragmentation**

Issue: Fragmentation

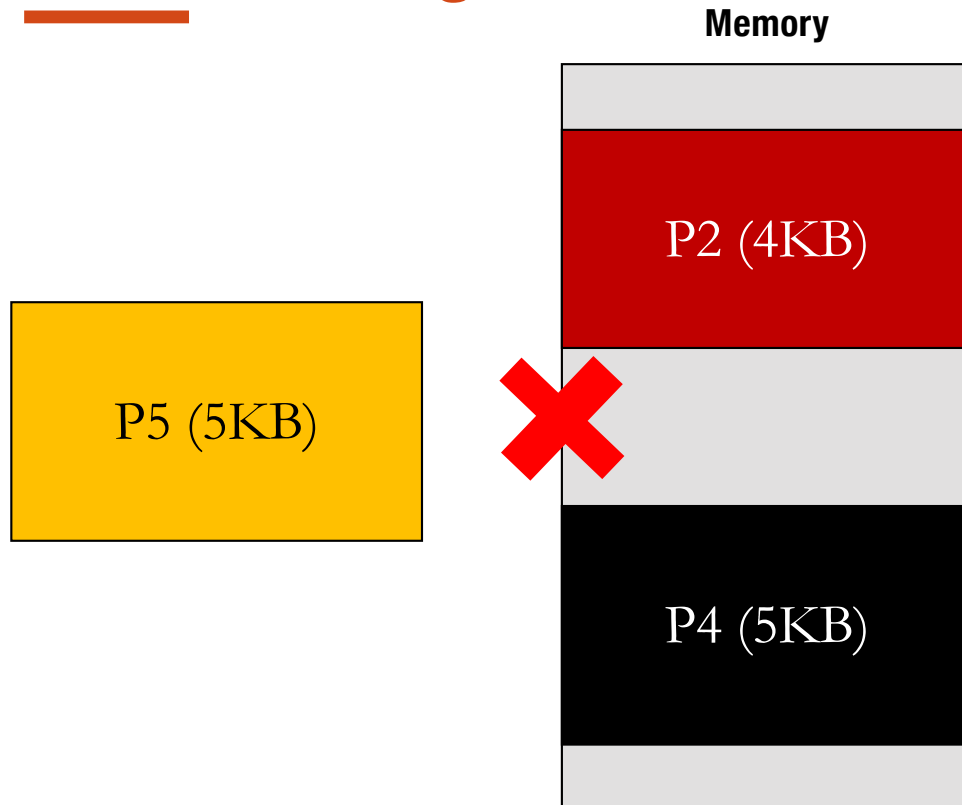
External Fragmentation

Memory



Issue: Fragmentation

External Fragmentation



- **External fragmentation:** free spaces are not contiguous to contain another process
- Solution: **compaction**
 - Technique used to combine free memory spaces into a larger free space

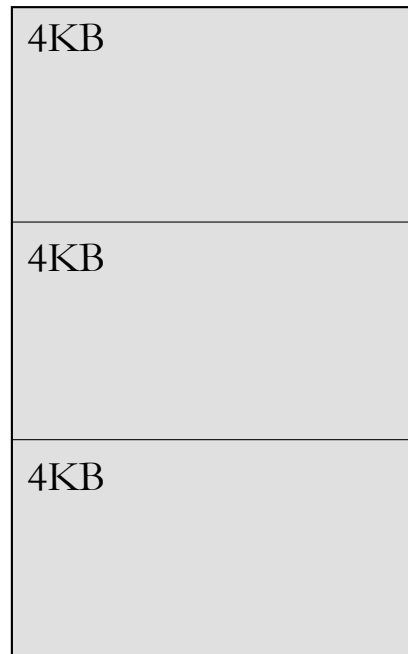
Issue: Fragmentation

Internal Fragmentation

P1 (1KB)

P2 (4KB)

Memory



- **Internal fragmentation:** internal partition allocated to a process can not be used for another process
- Solution: **best-fit allocation**

