Operating Systems

Lecture 15

Memory Management

May 27, 2013

http://web.uettaxila.edu.pk/CMS/SP2013/seOSbs/

Agenda for Today

- Overlays
- Swapping
- Contiguous Memory Location
  - Multiprogramming with fixed tasks (MFT)
  - Multiprogramming with variable tasks (MVT)
- Paging
- Addressing and address translation in paging

Overlays

- Allow a process to be larger than the amount of memory allocated to it
- Keep in memory only those instructions and data that are needed at any given time

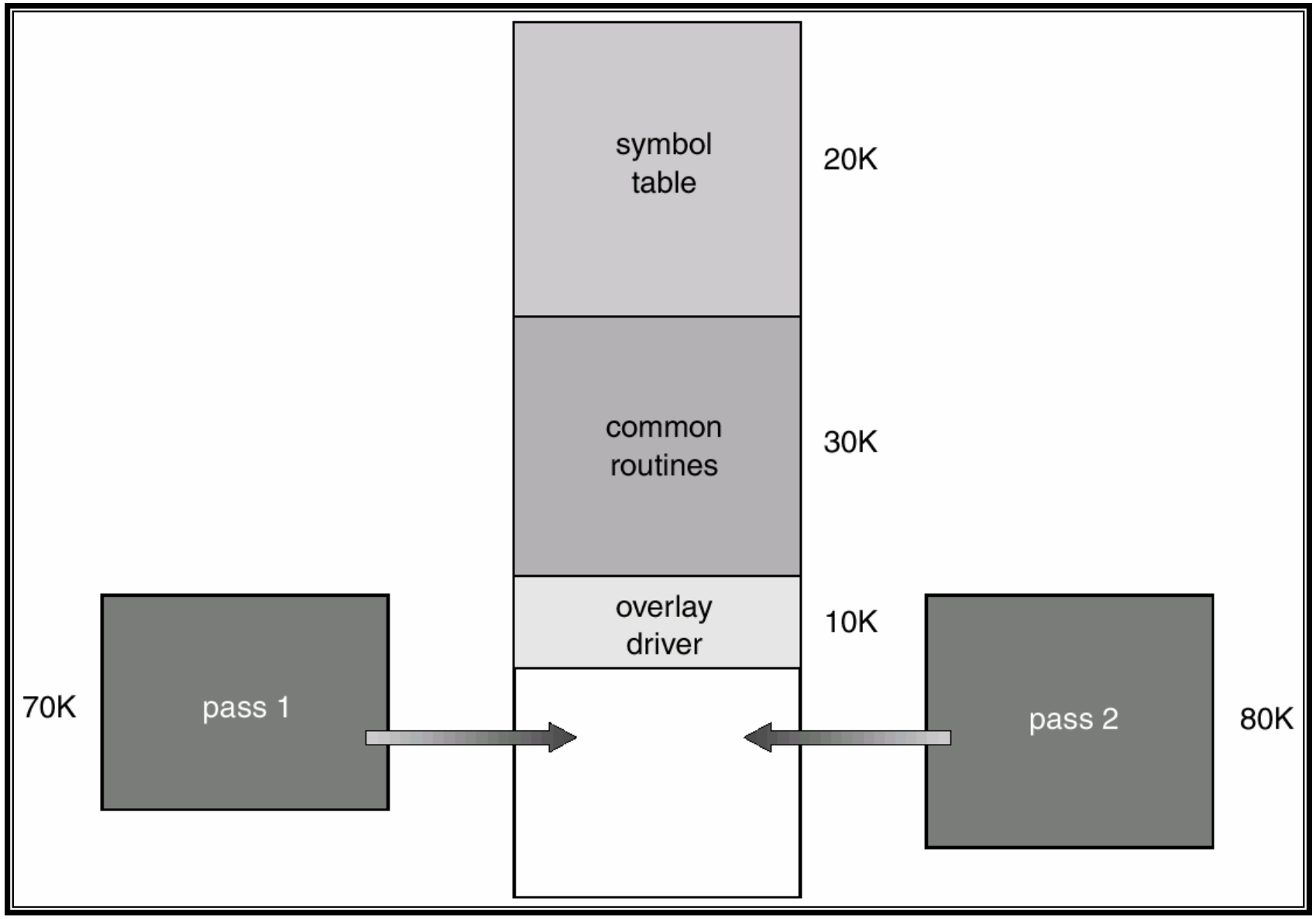
Overlays

- When other instructions are needed, they are loaded into the space occupied previously by instructions that are no longer needed
- Implemented by user
- Programming design of overlay structure is complex and not possible in all cases

Overlays Example

- 2-Pass assembler/compiler
- Available main memory: 150k
- Code size: 200k
  - Pass 1 …….. 70k
  - Pass 2 …….. 80k
  - Common routines …..30k
  - Symbol table …… 20k
  - Overlays Driver …….10K

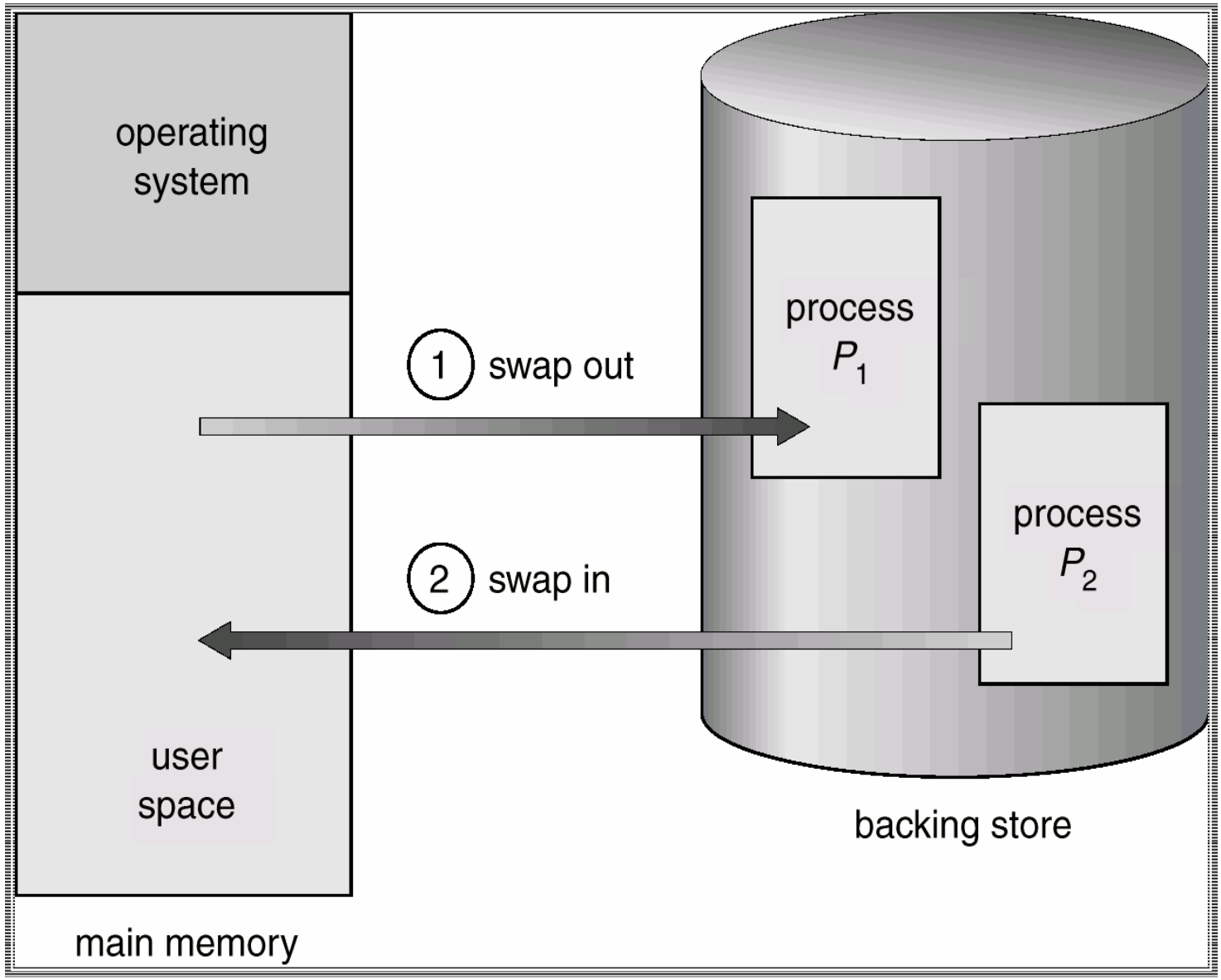
Overlays Example



Swapping

- Swap out and swap in (or roll out and roll in)
- Major part of swap time is transfer time; the total transfer time is directly proportional to the *amount* of memory swapped
- Large context switch time

Swapping


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

Cost of Swapping

- Process size = 1 MB
- Transfer rate = 5 MB/sec
- Swap out time = 1/5 sec
  = 200 ms
- Average latency = 8 ms
- Net swap out time = 208 ms
- Swap out + swap in = 416 ms

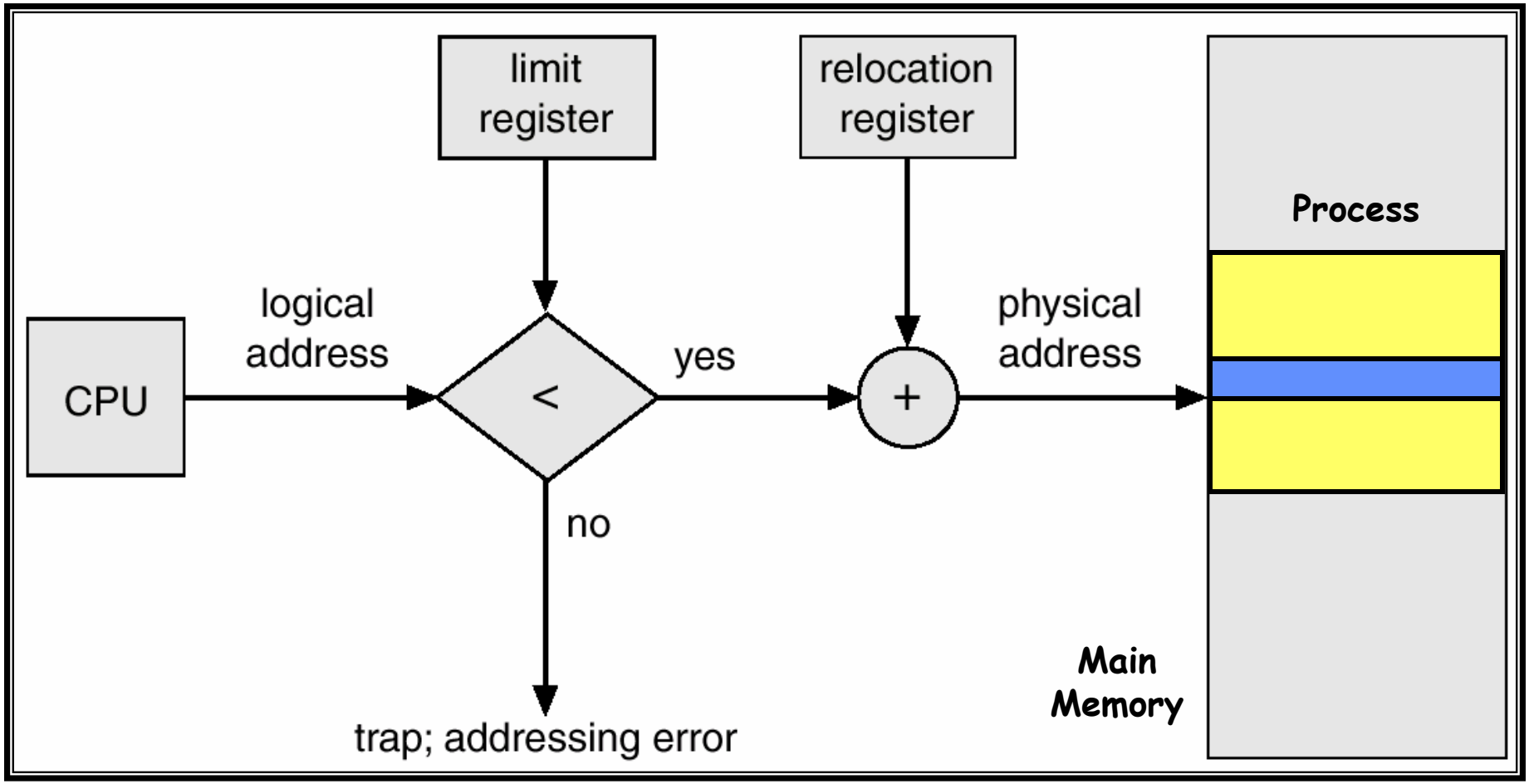
Issues with Swapping

- Quantum for RR scheduler
- Pending I/O for swapped out process
- User space used for I/O
- Solutions
  - Don't swap out processes with pending I/O
  - Do I/O using kernel space

Contiguous Allocation

- Kernel space, user space
- A process is placed in a single contiguous area in memory
- Base (re-location) and limit registers are used to point to the smallest memory address of a process and its size, respectively.

Contiguous Allocation



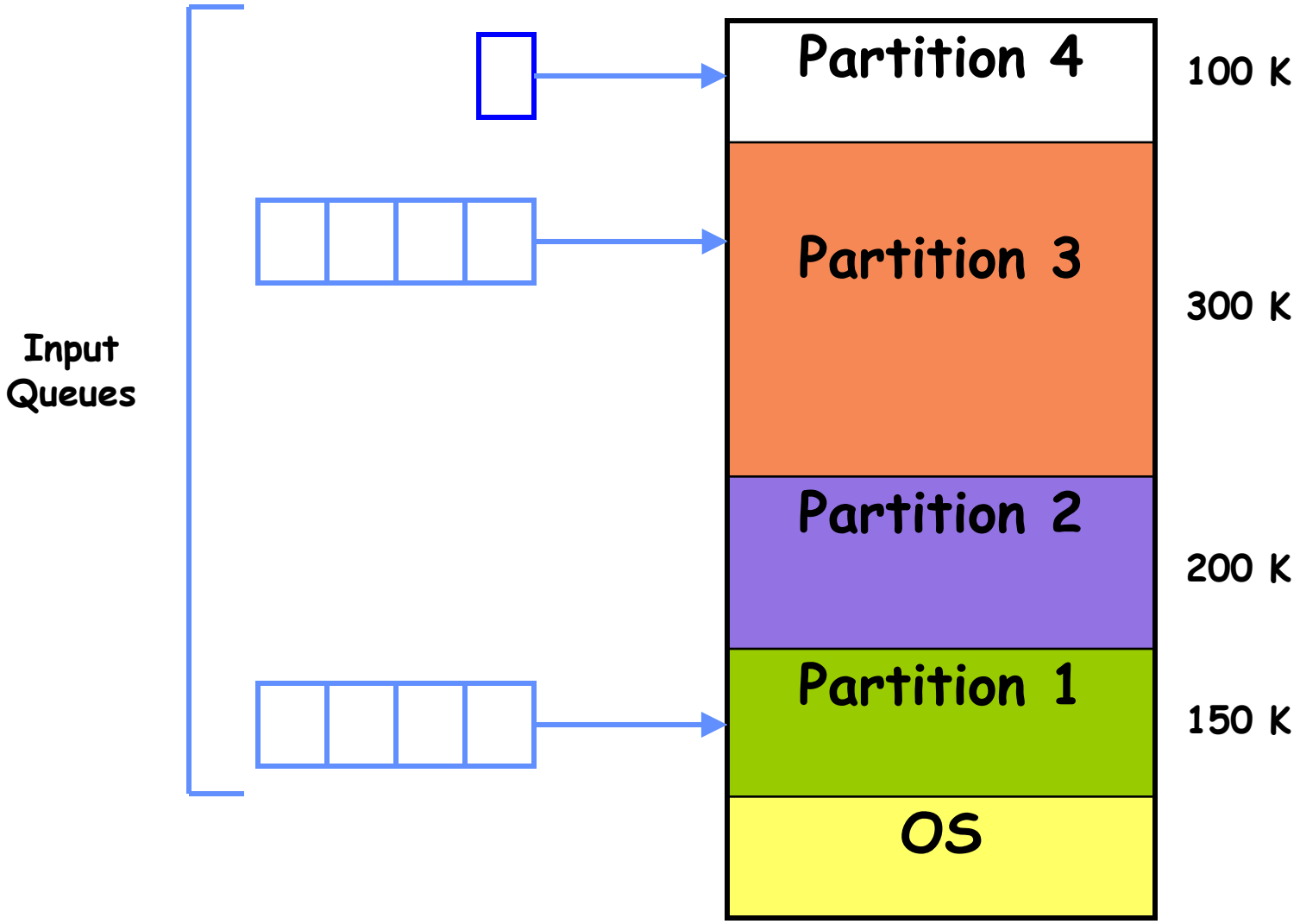
MFT

- Multiprogramming with fixed tasks (MFT)
- Memory is divided into several fixed-size partitions.
- Each partition may contain exactly one process/task.

MFT

- Boundaries for partitions are set at boot time and are *not* movable.
- An input queue per partition
- The degree of multiprogramming is bound by the number of partitions.

MFT


Input
Queues
Partition 4
Partition 3
Partition 2
Partition 1
OS
100 K
300 K
200 K
150 K

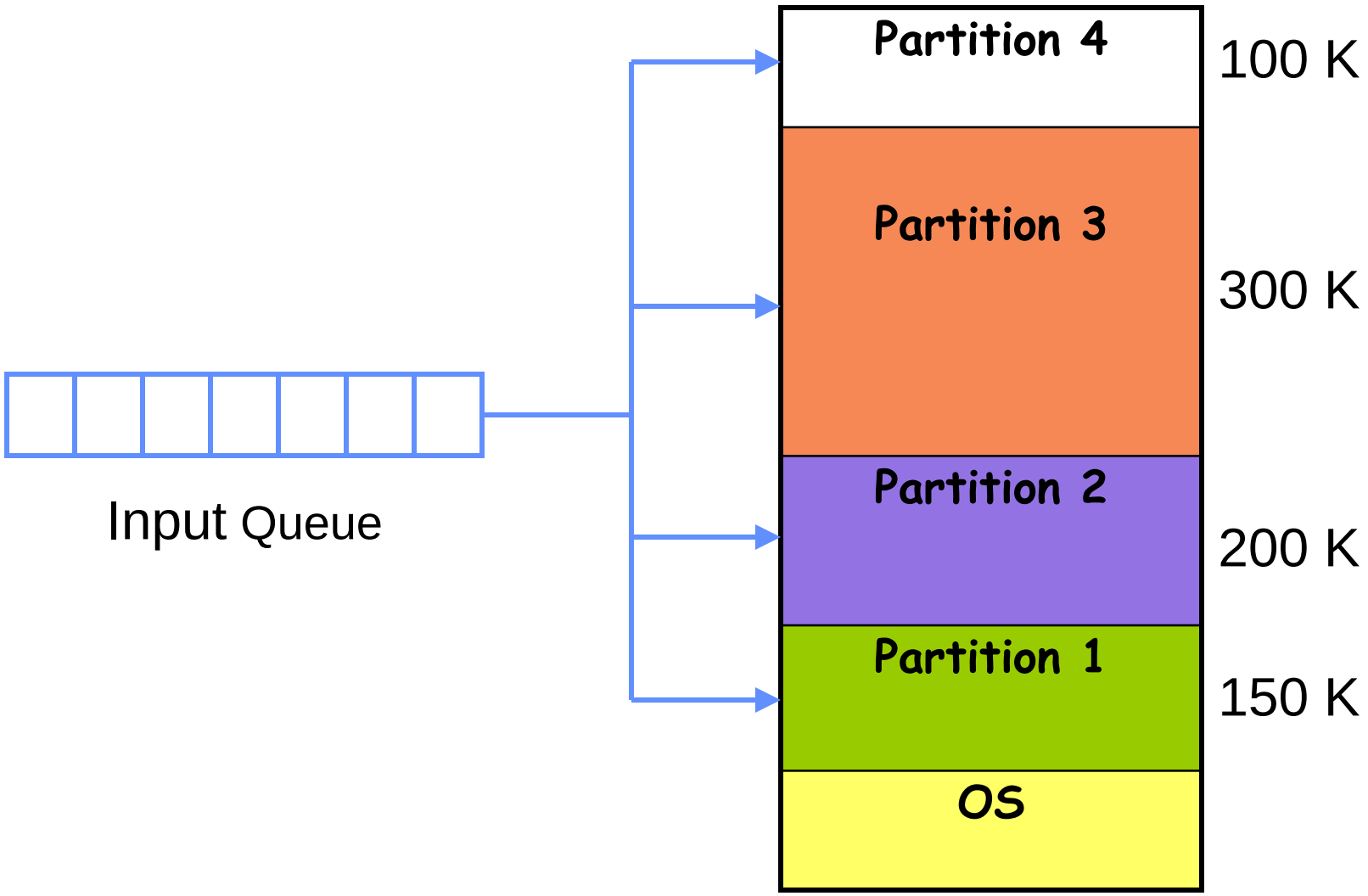
MFT With Multiple Input Queues

- Potential for wasted memory space—an empty partition but no process in the associated queue
- Load-time address binding

MFT With Single Input Queue

- Single queue for all partitions
  - Search the queue for a process when a partition becomes empty
  - *First-fit*, *best-fit*, *worst-fit* space allocation algorithms

MFT With Single Input Queue



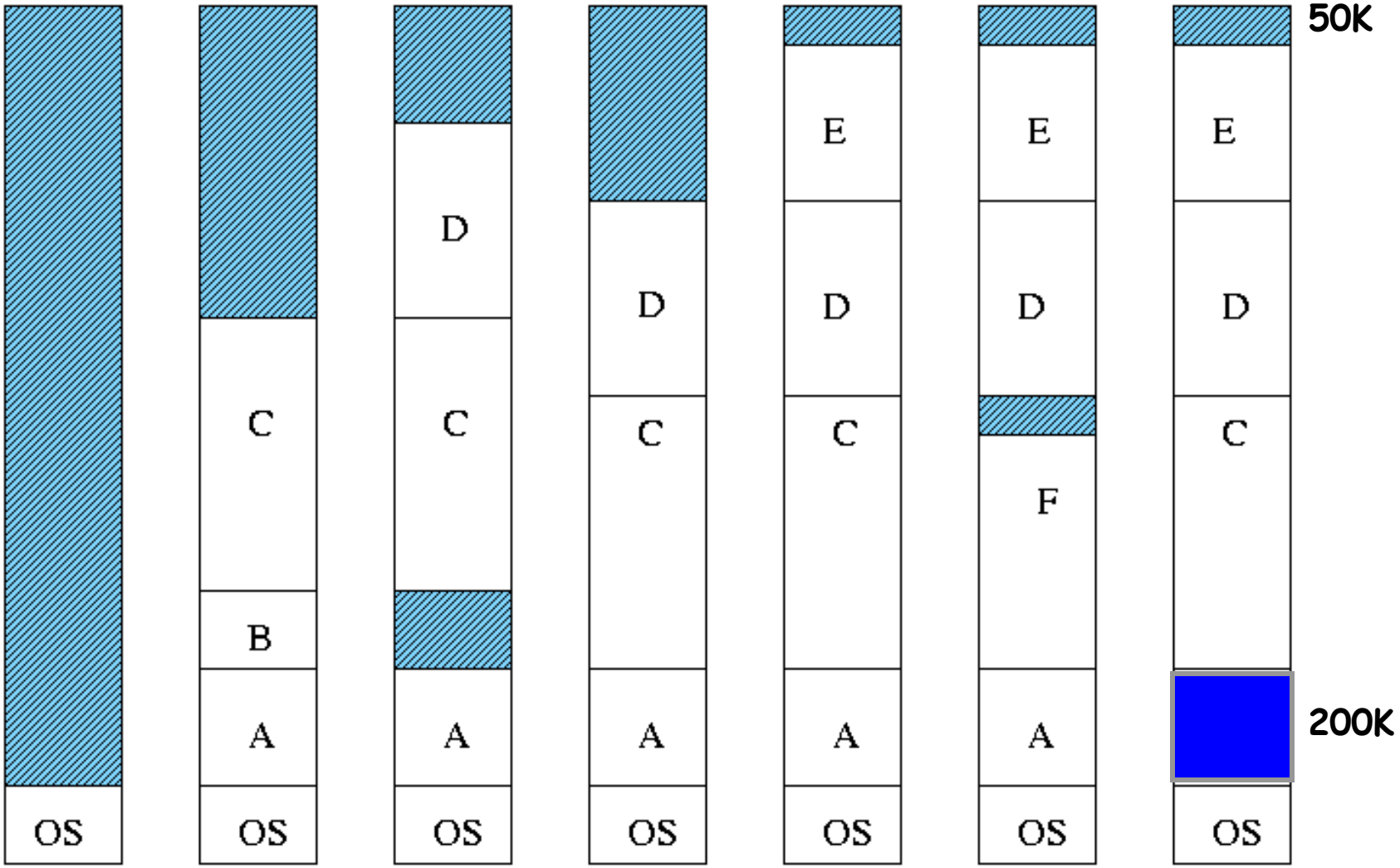
MFT Issues

- Internal fragmentation—wasted space *inside* a fixed-size memory region
- No sharing between processes.
- Load-time address binding with multiple input queues

MVT

- Jobs can move from one partition to another
- Dynamic address binding
- No internal fragmentation
- Introduces external fragmentation

MVT



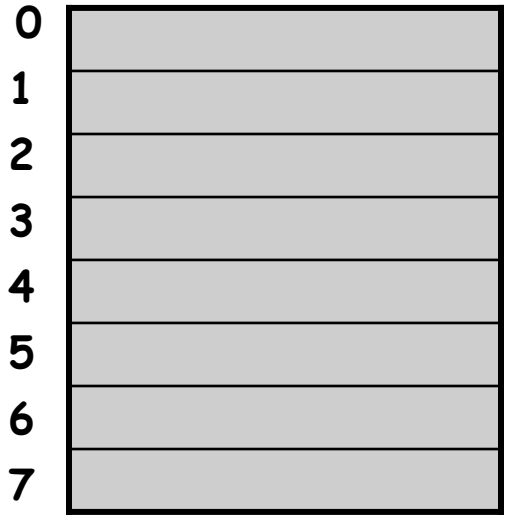
External Fragmentation

- External Fragmentation refers to the state of memory space when total amount of unused memory space exists to satisfy a request but this memory space is not contiguous.

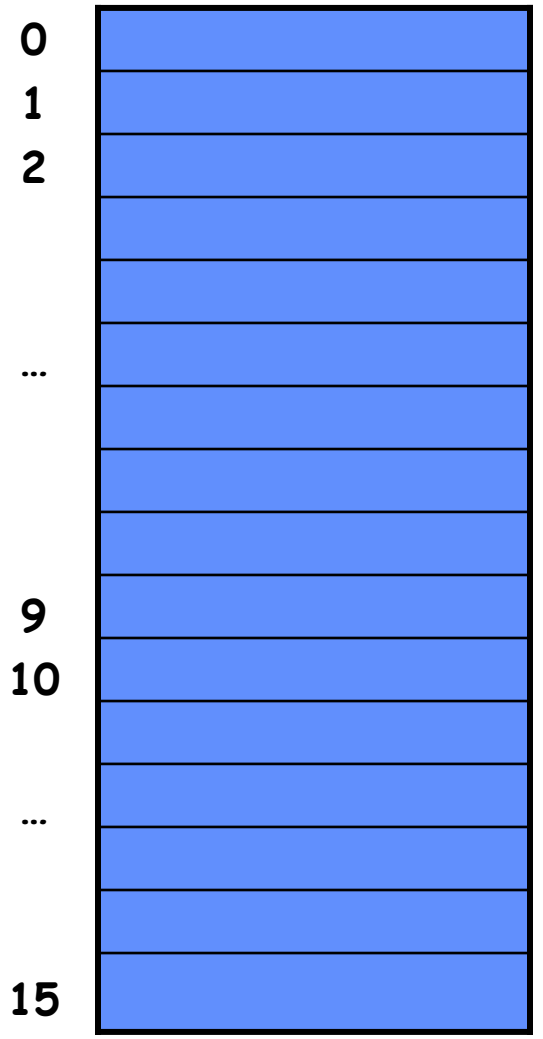
External Fragmentation

- Compact main memory (shuffle processes around to put free space in one contiguous area)
- Slows down execution of currently running processes

Paging

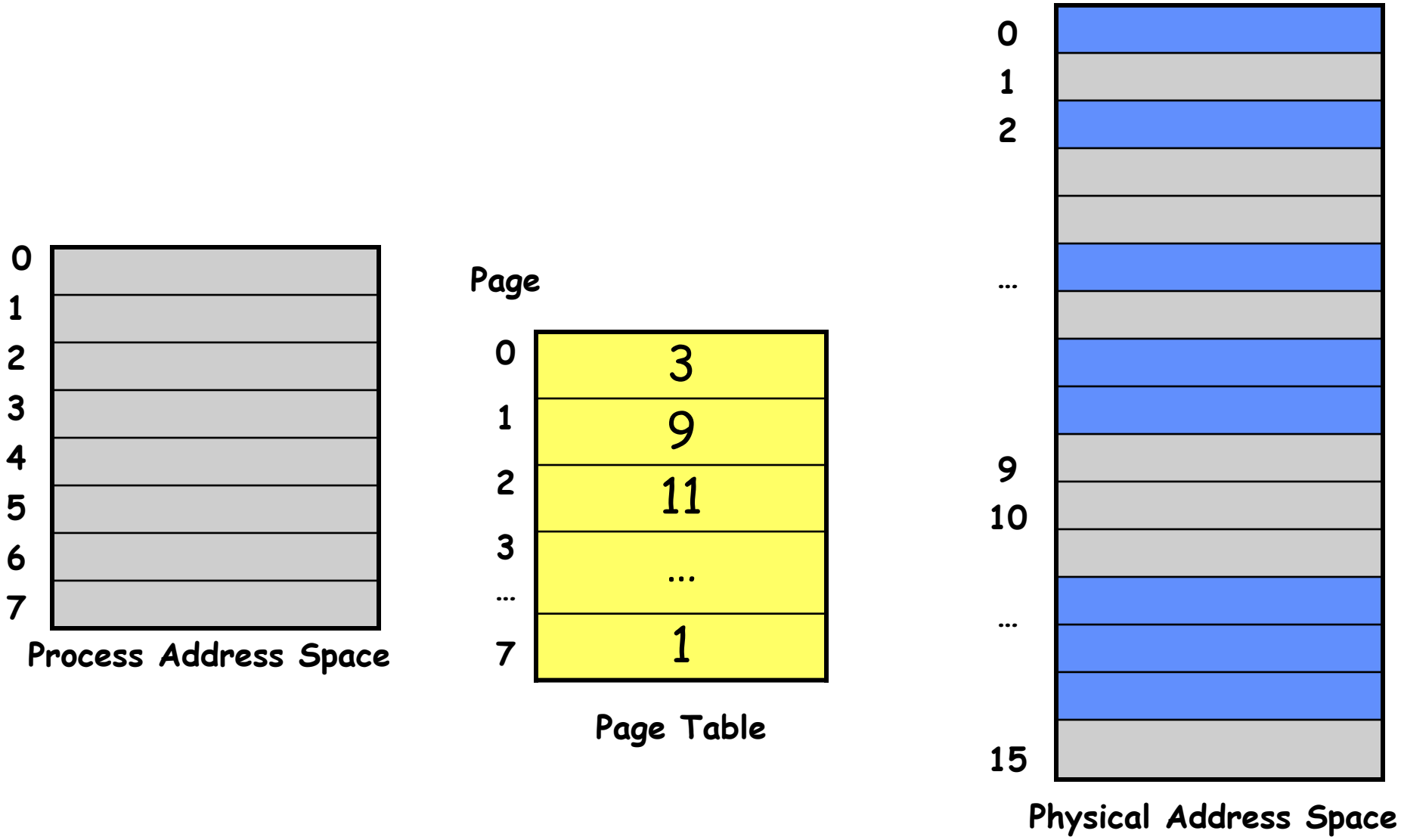


Process Address Space

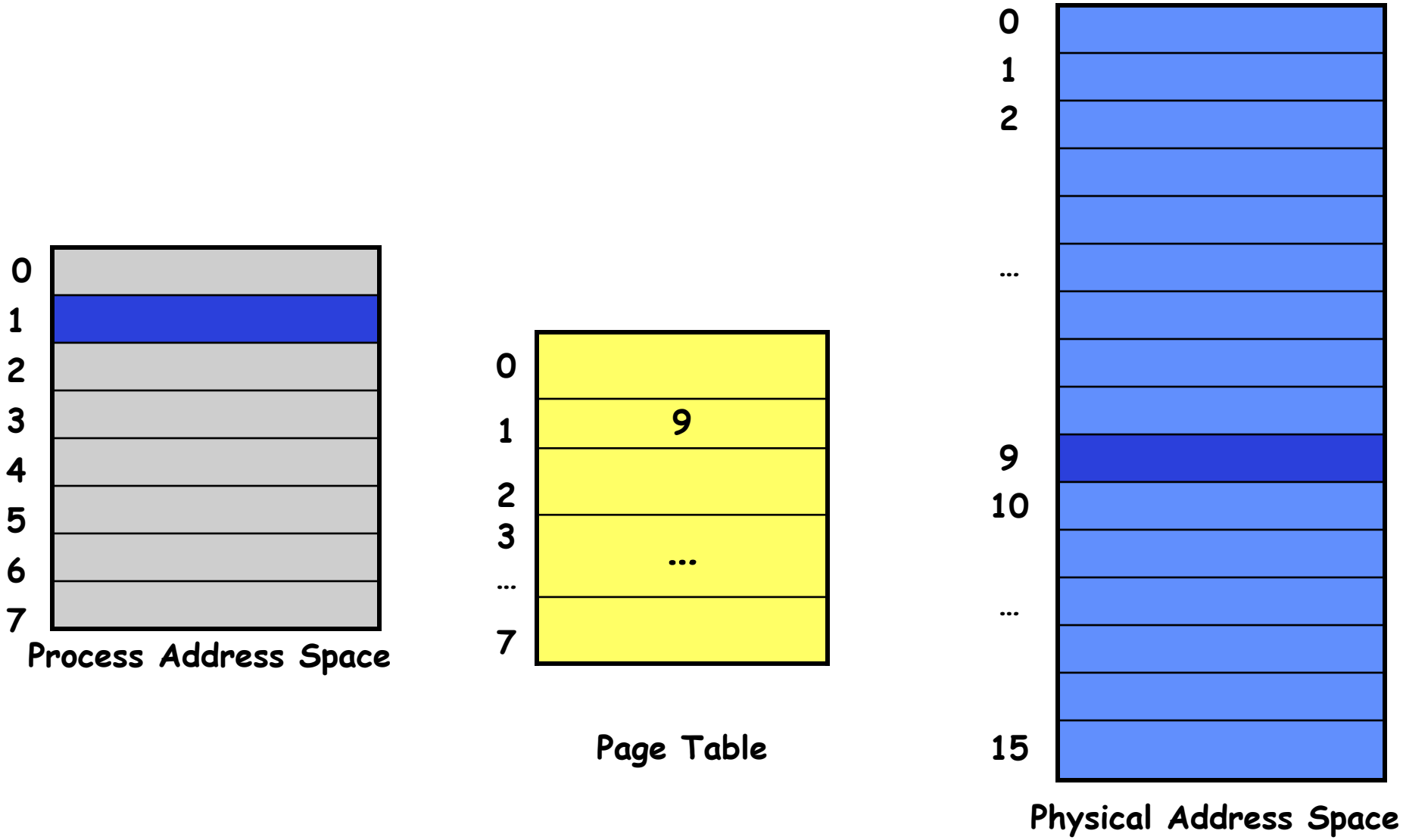


Physical Address Space

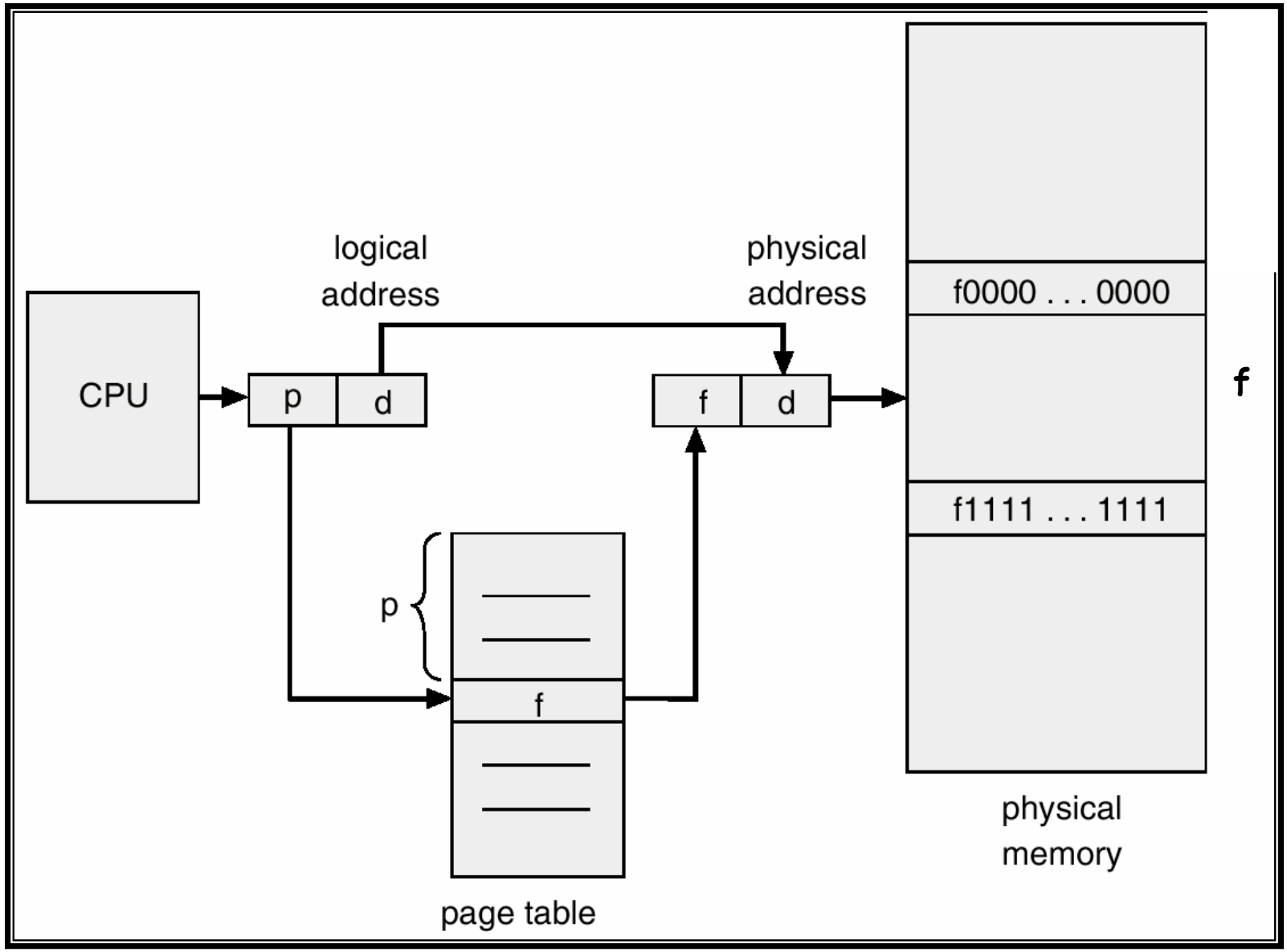
Paging


0
1
2
3
4
5
6
7
Process Address Space
Page
0
1
2
3
...
7
3
9
11
...
1
Page Table
0
1
2
...
9
10
...
15
Physical Address Space

Paging


0
1
2
3
4
5
6
7
Process Address Space
0
1
2
3
...
7
9
...
Page Table
0
1
2
...
9
10
...
15
Physical Address Space

Address Translation

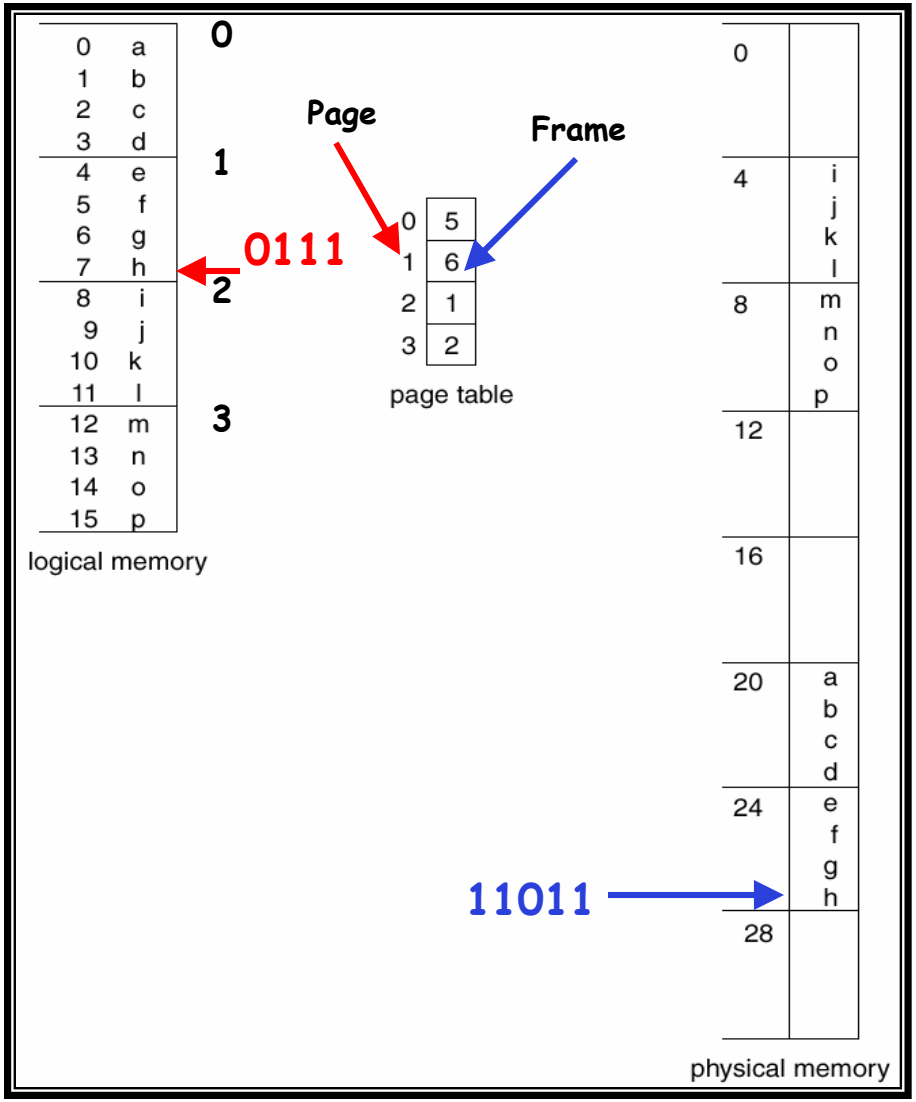


Paging Example

- Page size = 4 bytes
- Process address space
  = 4 pages
- Physical address space
  = 8 frames
- Logical address: (1,3)

  = 0111
- Physical address: (6,3)

  = 11011



Addressing in Paging

- **Logical address space** of 16 pages of 1024 words each, mapped into a physical memory of 32 frames.
- **Logical address** size?
- **Physical address** size?
- Number of bits for p, f, and d?

Addressing in Paging

- No. of bits for **p** = 4 bits
- No. of bits for **f** = 5 bits
- No. of bits for **d** = 11 bits

Addressing in Paging

Logical address size = |p| + |d|
= 4+11
= 15 bits

Physical address size = |f| + |d|
= 5+11
= 16 bits

Page Table Size

Page table size = NP * PTES

where NP is the number of pages in the process address space and PTES is the page table entry size (equal to |f| based on our discussion so far).

Page table size = 16 * 5 bits

= 16 bytes