Operating Systems!

Memory: Approaches to Memory Management

Prof. Travis Peters

Montana State University

CS 460 - Operating Systems

Fall 2020

https://www.cs.montana.edu/cs460

Some diagrams and notes used in this slide deck have been adapted from Sean Smith's OS courses @ Dartmouth. Thanks, Sean!

Today

- Announcements
  - ~~Yalnix~~ Project Updates
    - You can choose your topic!
    - Review grade breakdown
    - Milestones (proposal, presentation, final submission)
    - Three (3) types of *evaluations!*
  - *Exam 1:* ***Nice work! Working on grading...***

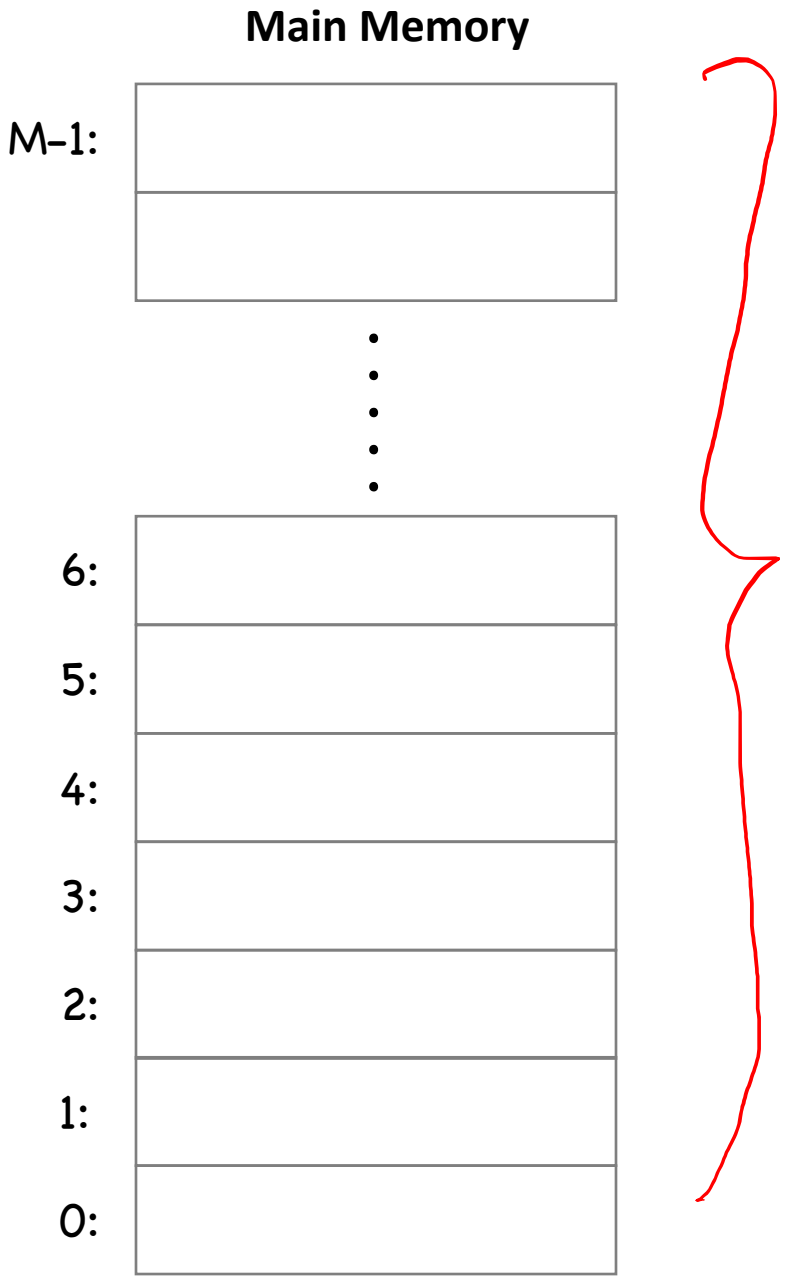
- Upcoming Deadlines
  - PA3 ***(Group Assignment)*** due ***Sunday [10/18/2020] @ 11:59 PM (MST)***
  - Project Proposal due ***Sunday [10/25/2020] @ 11:59 PM (MST)***

Today *(cont.)*

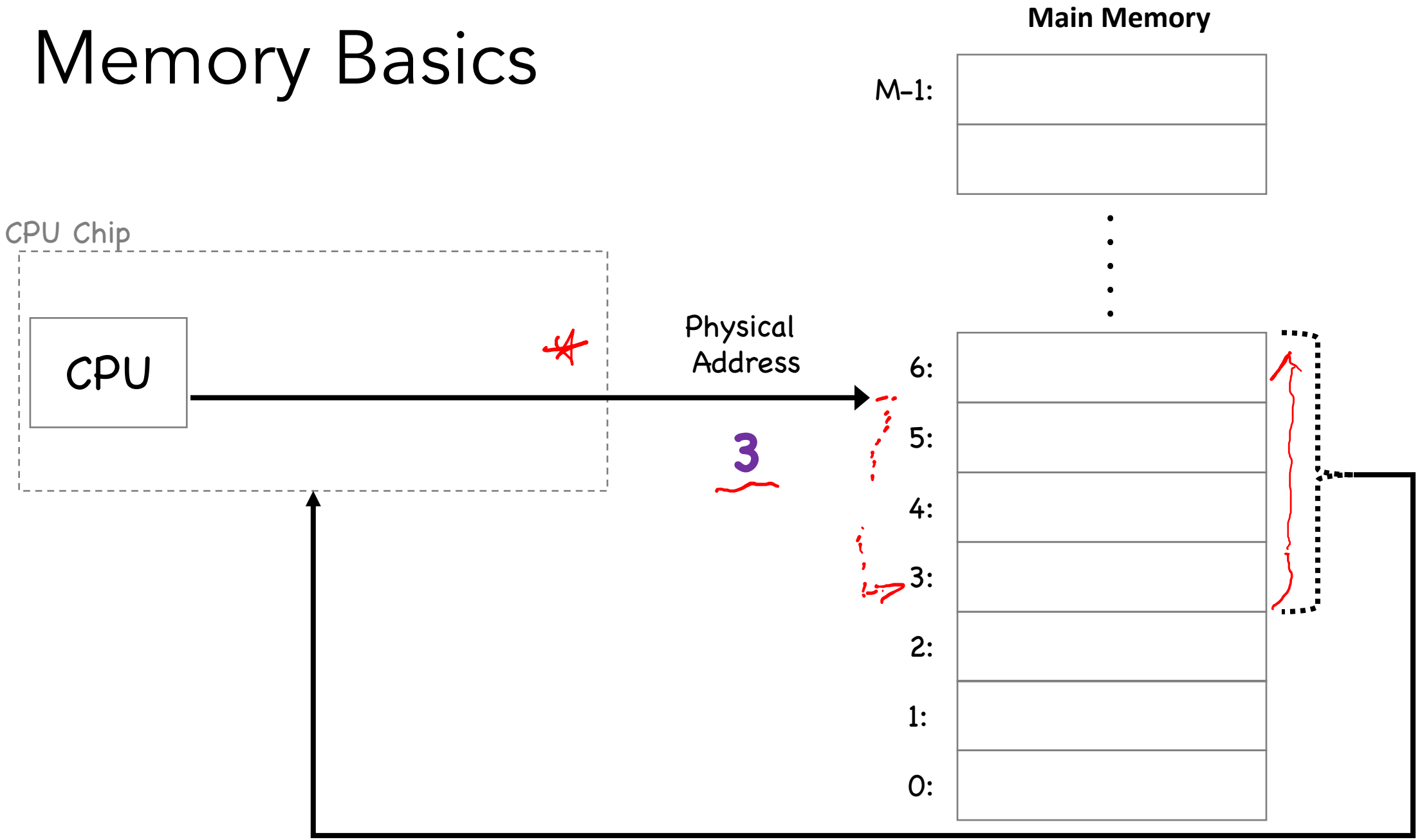
- Agenda
  - Revisiting Memory Basics
  - Logical vs. Physical Addresses
  - Contiguous Memory & "Fit" Strategies
  - Discontiguous Memory

  ➔ Virtual Memory

Memory Basics

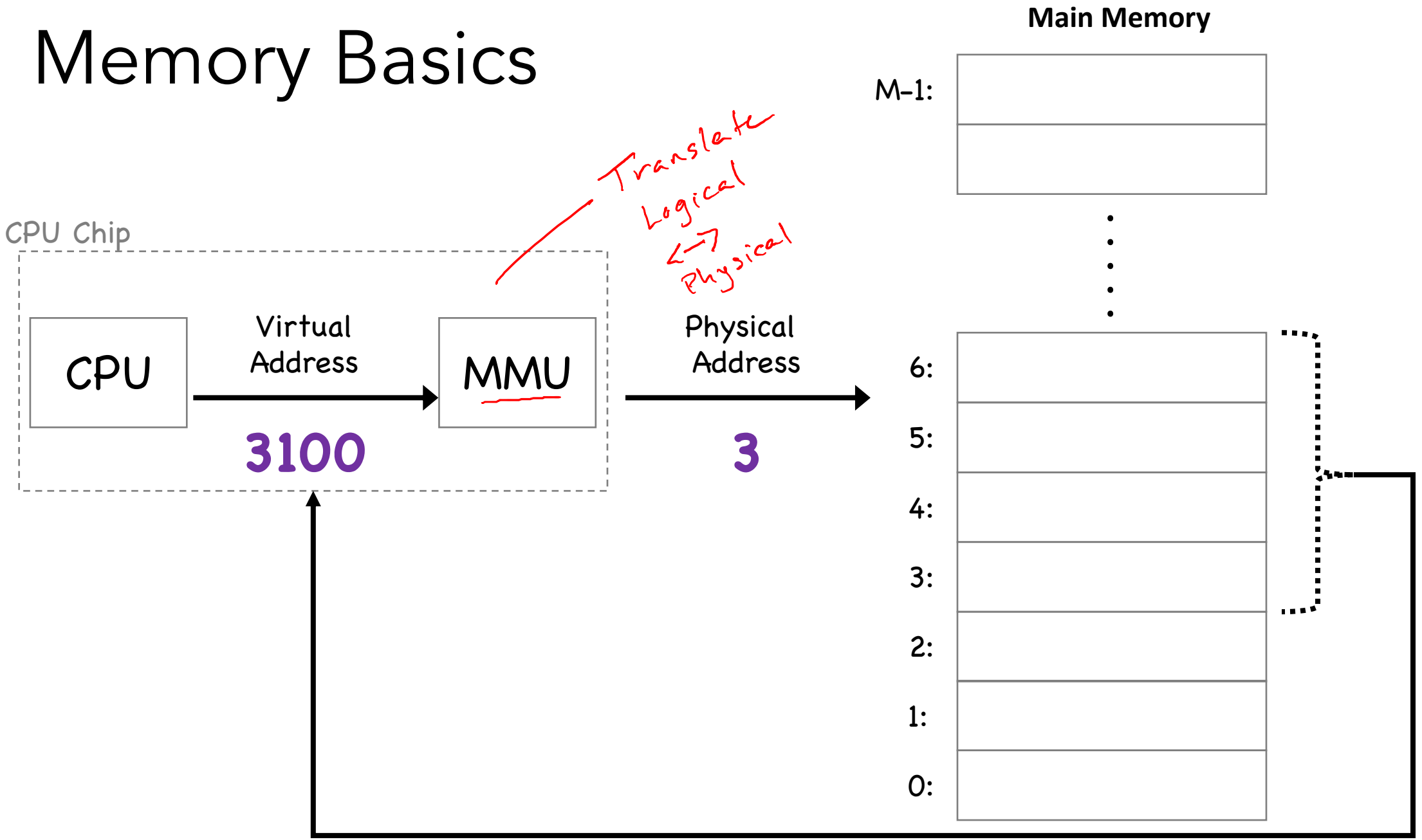


Memory Basics



Logical vs. Physical

Memory Basics



Questions	As a unit	In pieces
Where does it go?		
How does it get there?		

	As a unit	*In pieces*
Where does it go?	Dynamic Allocation	
How does it get there?		

Memory Partitioning: A Progression

The textbook presents a nice (thorough) progression:

- Fixed Partitioning
- Dynamic Partitioning
- Paging
- Segmentation
- Paging and Segmentation

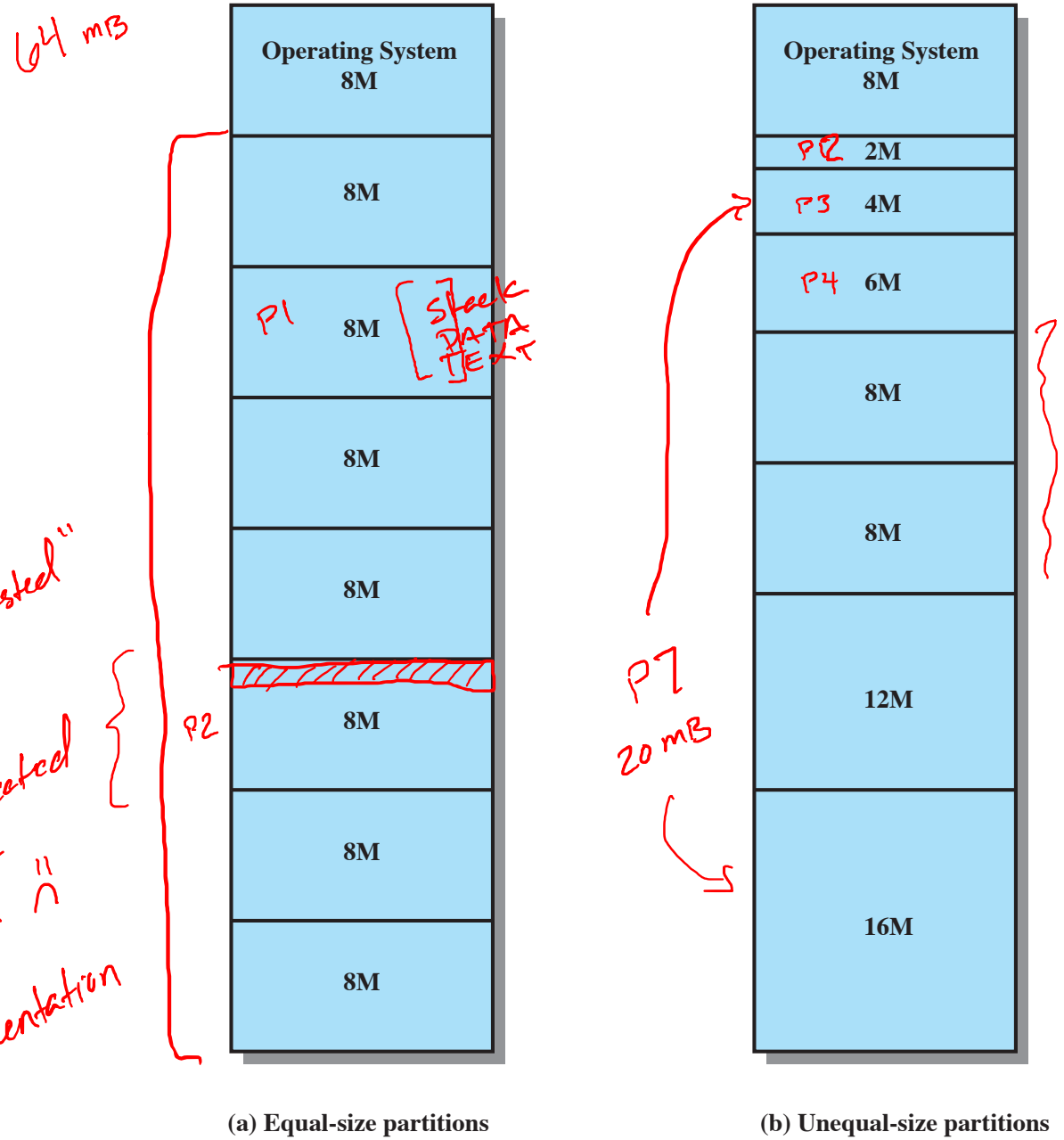
Technique	Description	Strengths	Weaknesses
Fixed Partitioning	Main memory is divided into a number of static partitions at system generation time. A process may be loaded into a partition of equal or greater size.	Simple to implement; little operating system overhead.	Inefficient use of memory due to internal fragmentation; maximum number of active processes is fixed.
Dynamic Partitioning	Partitions are created dynamically, so that each process is loaded into a partition of exactly the same size as that process.	No internal fragmentation; more efficient use of main memory.	Inefficient use of processor due to the need for compaction to counter external fragmentation.
Simple Paging	Main memory is divided into a number of equal-size frames. Each process is divided into a number of equal-size pages of the same length as frames. A process is loaded by loading all of its pages into available, not necessarily contiguous, frames.	No external fragmentation.	A small amount of internal fragmentation.
Simple Segmentation	Each process is divided into a number of segments. A process is loaded by loading all of its segments into dynamic partitions that need not be contiguous.	No internal fragmentation; improved memory utilization and reduced overhead compared to dynamic partitioning.	External fragmentation.
Virtual Memory Paging	As with simple paging, except that it is not necessary to load all of the pages of a process. Nonresident pages that are needed are brought in later automatically.	No external fragmentation; higher degree of multiprogramming; large virtual address space.	Overhead of complex memory management.
Virtual Memory Segmentation	As with simple segmentation, except that it is not necessary to load all of the segments of a process. Nonresident segments that are needed are brought in later automatically.	No internal fragmentation, higher degree of multiprogramming; large virtual address space; protection and sharing support.	Overhead of complex memory management.

Storage Allocation

“Fixed Partitioning” vs. “Dynamic Allocation”

Fixed Partitioning

- OS occupies a fixed portion of MM
- All other memory available to processes
- **Idea:** divide memory into fixed-sized partitions.
- **Idea:** divide memory into unequal-sized partitions



(a) Equal-size partitions

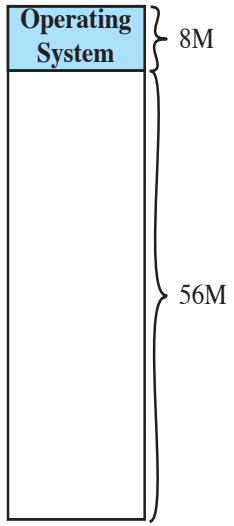
(b) Unequal-size partitions

Storage Allocation

"Fixed Partitioning" vs. **"Dynamic Allocation"**

Dynamic Storage Allocation

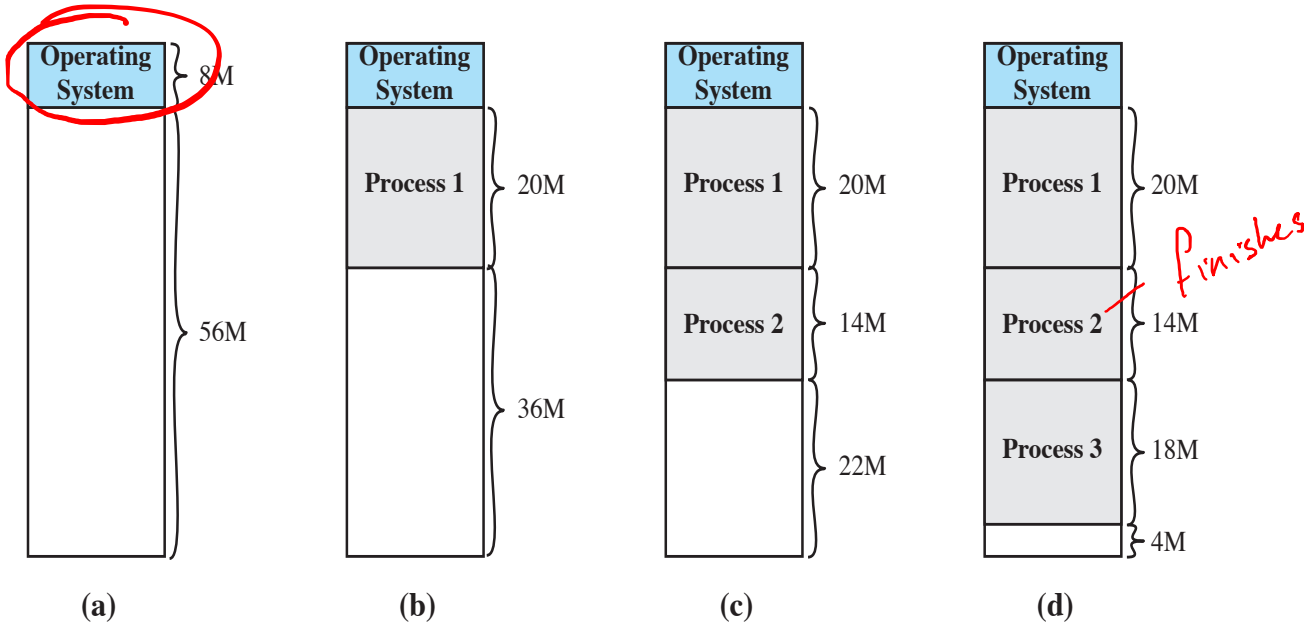
- Partitions are of **variable length** and **number**
- Each process is allocated *exactly* as much memory as it needs



(a)

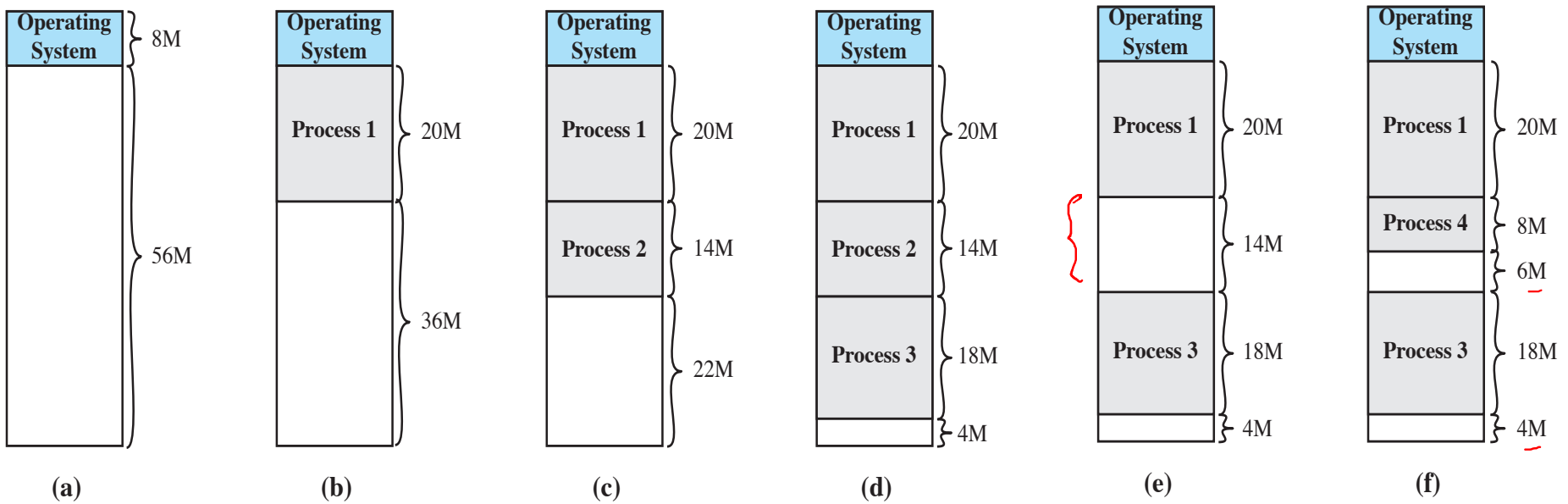
Dynamic Storage Allocation

- Partitions are of **variable length** and **number**
- Each process is allocated *exactly* as much memory as it needs



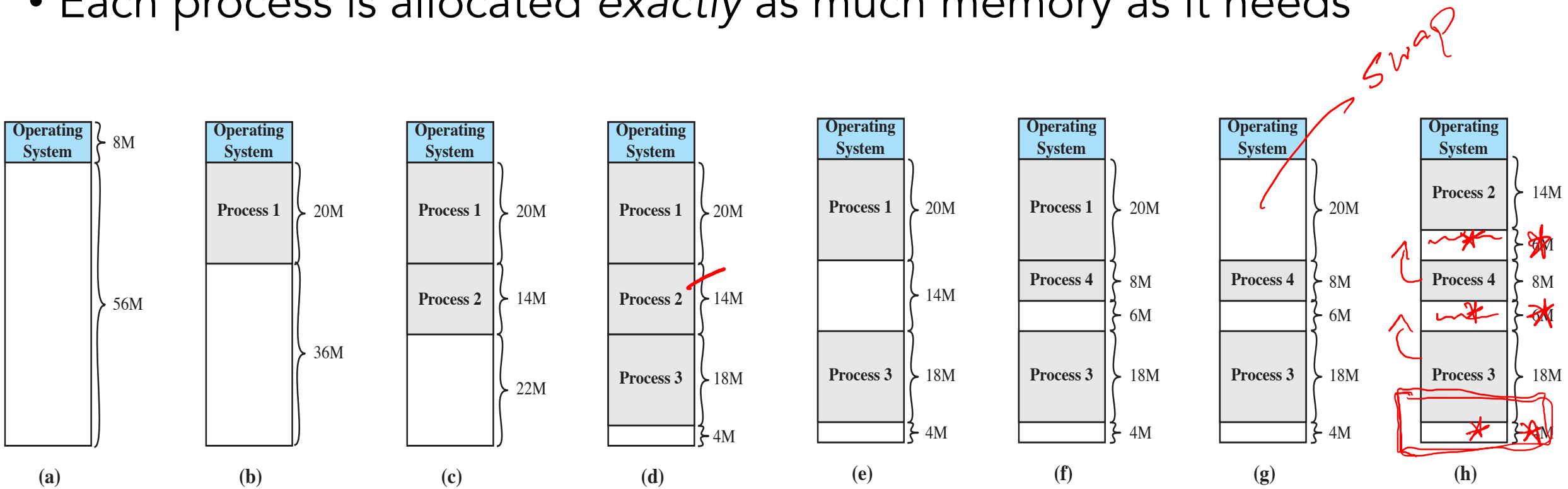
Dynamic Storage Allocation

- Partitions are of **variable length** and **number**
- Each process is allocated *exactly* as much memory as it needs



Dynamic Storage Allocation

- Partitions are of **variable length** and **number**
- Each process is allocated *exactly* as much memory as it needs



Fragmentation

EXTERNAL

FRAGMENTATION:

UNUSED SPACE THAT CANNOT BE **ALLOCATED**

INTERNAL

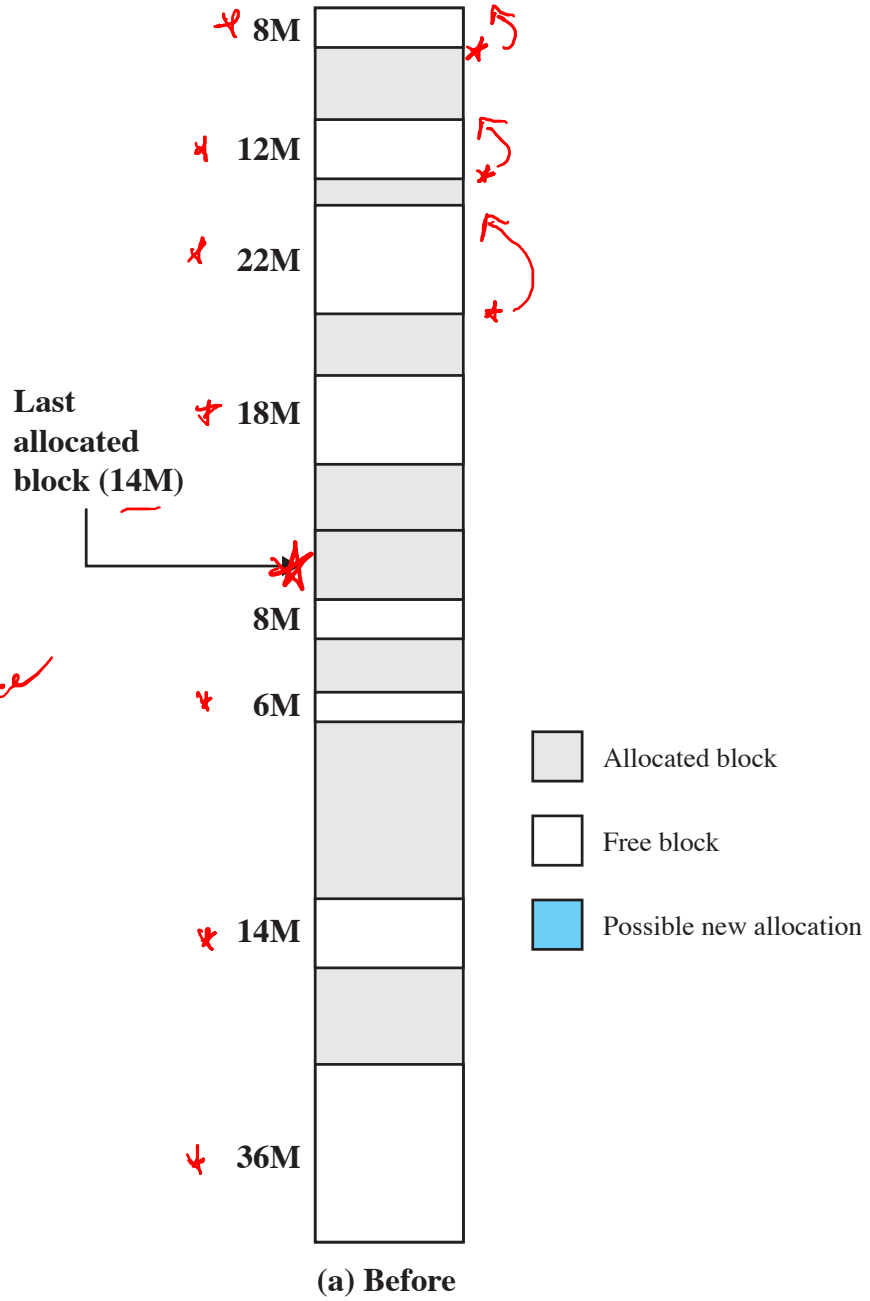
FRAGMENTATION:

ALLOCATED SPACE THAT CANNOT BE **USED**


HAVE YOU TRIED
TURNING IT OFF AND ON AGAIN?
quickmeme.com

Memory Placement

- How to best place ("fit") processes and their memory requirements into available memory?



You Try!

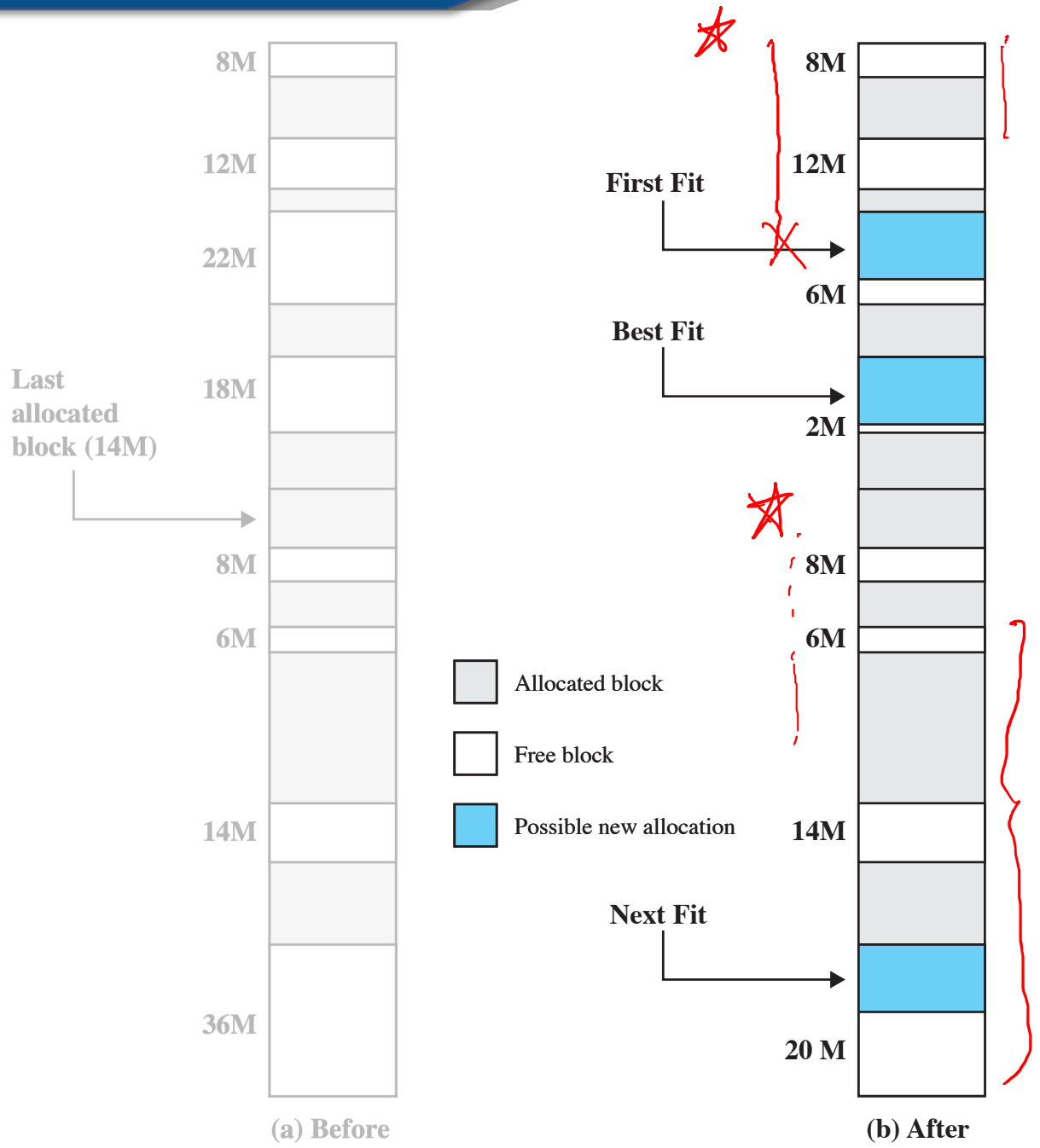
- How to place a new request for a 16MB block?

Memory Placement

- How to best place (“fit”) processes and their memory requirements into available memory?

You Try!

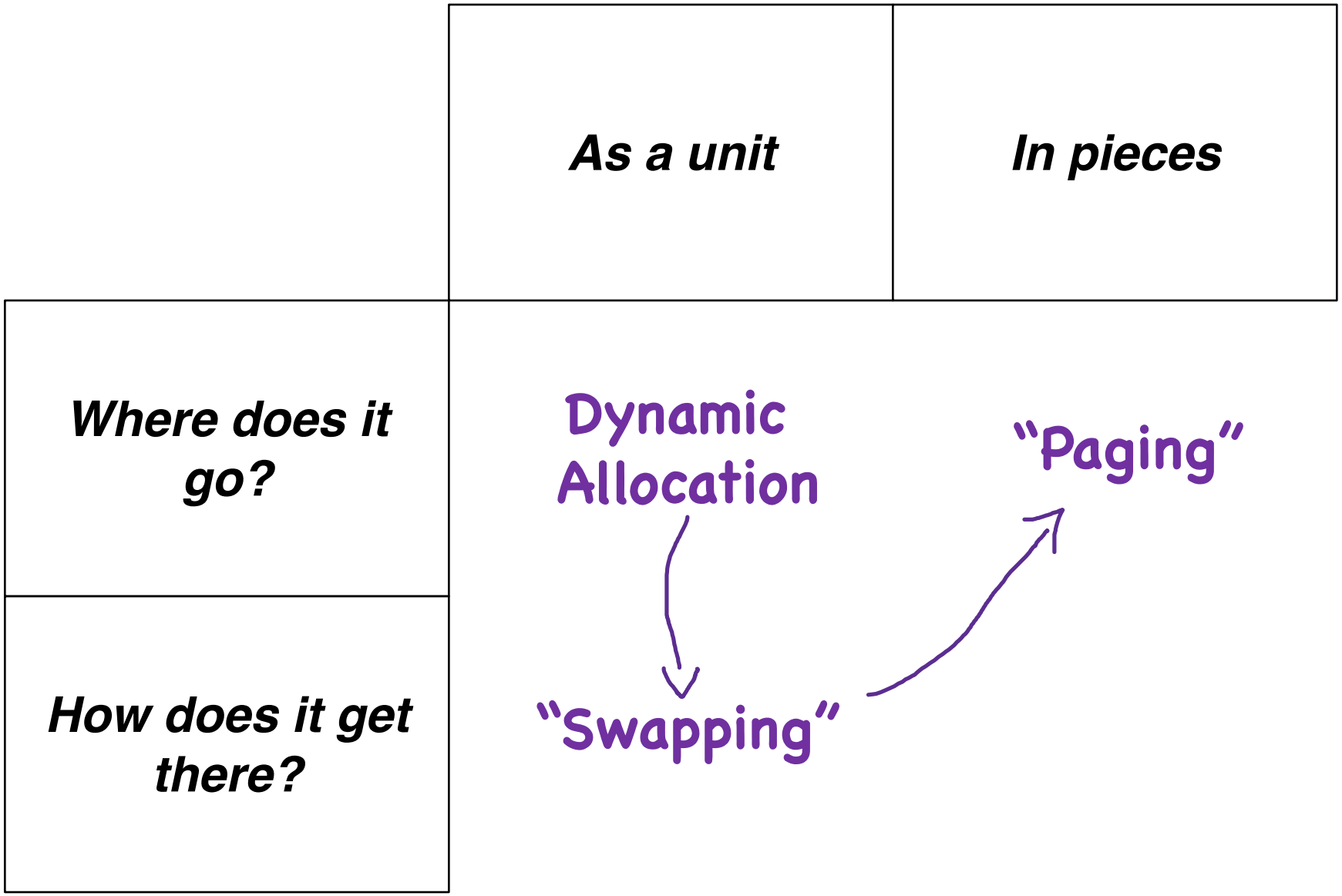
- How to place a new request for a 16MB block?



What if it doesn’t fit?!

What if the memory required for all our processes don't fit in what we have?

	As a unit	*In pieces*
Where does it go?	Dynamic Allocation	
How does it get there?	"Swapping"	

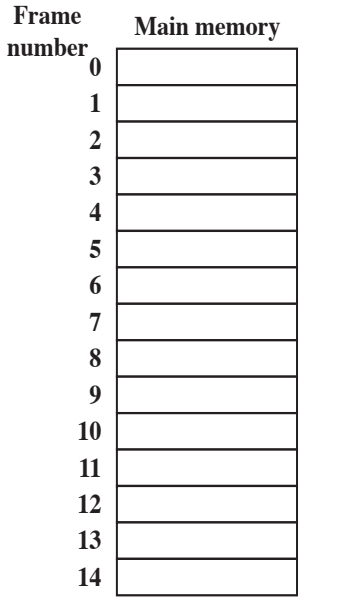

As a unit
In pieces
Where does it go?
How does it get there?
Dynamic Allocation
"Paging"
"Swapping"

Paging

We don’t have to keep everything in a contiguous region!

Paging

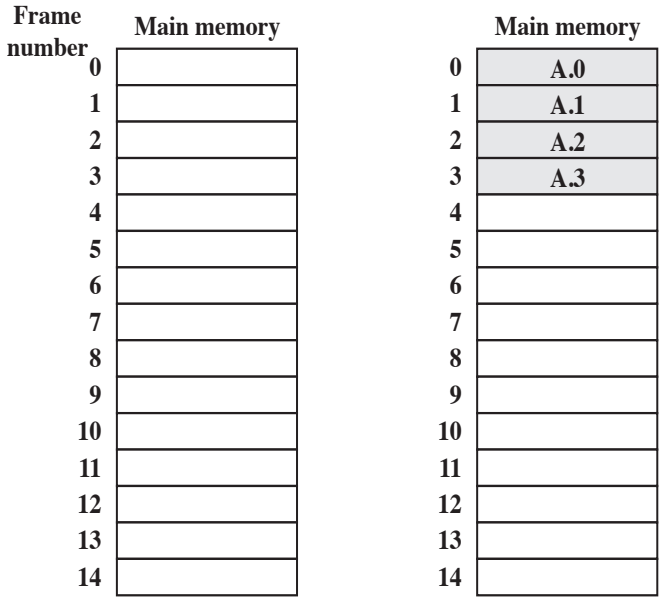
- Partition MM into **small, fixed-size chunks** of the same size
- Assign chunks of processes ("**pages**") into available chunks of MM ("**frames**")
- No more external fragmentation!
- Only minimal internal fragmentation!



(a) Fifteen Available Frames

Paging

- Partition MM into **small, fixed-size chunks** of the same size
- Assign chunks of processes ("**pages**") into available chunks of MM ("**frames**")
- No more external fragmentation!
- Only minimal internal fragmentation!



(a) Fifteen Available Frames (b) Load Process A