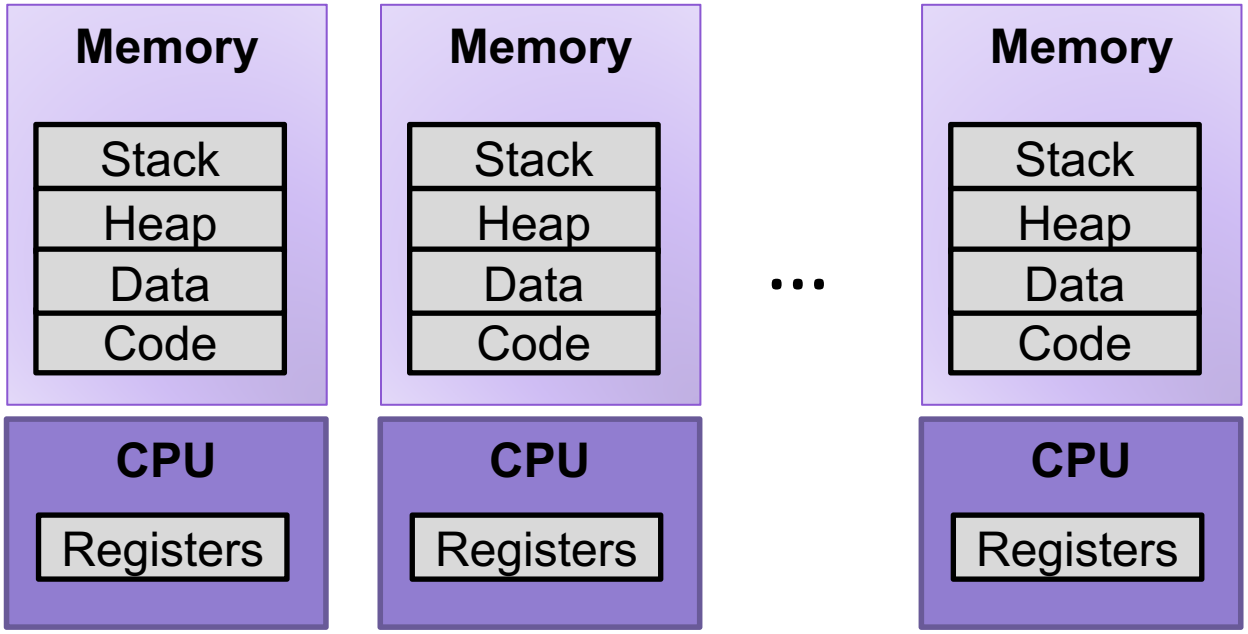
Lecture 17: Virtual Memory

CS 105

Fall 2020

Multiprocessing: The Illusion

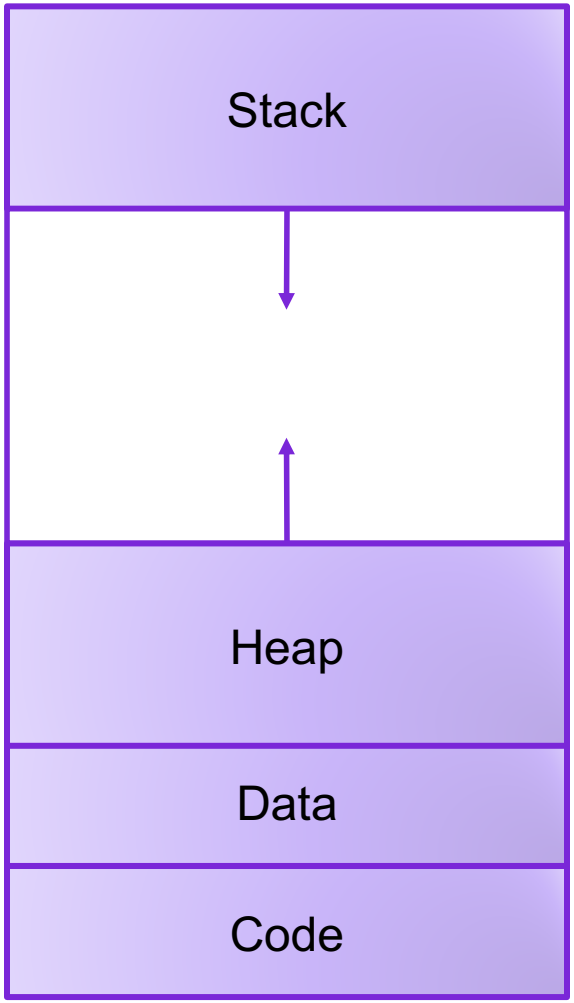


- Process provides each program with two key abstractions:
  - ***Logical control flow***
    - Each program seems to have exclusive use of the CPU
    - Provided by kernel mechanism called ***context switching***
  - ***Private address space***
    - Each program seems to have exclusive use of main memory.
    - Provided by kernel mechanism called ***virtual memory***

Multiprocessing: The Reality

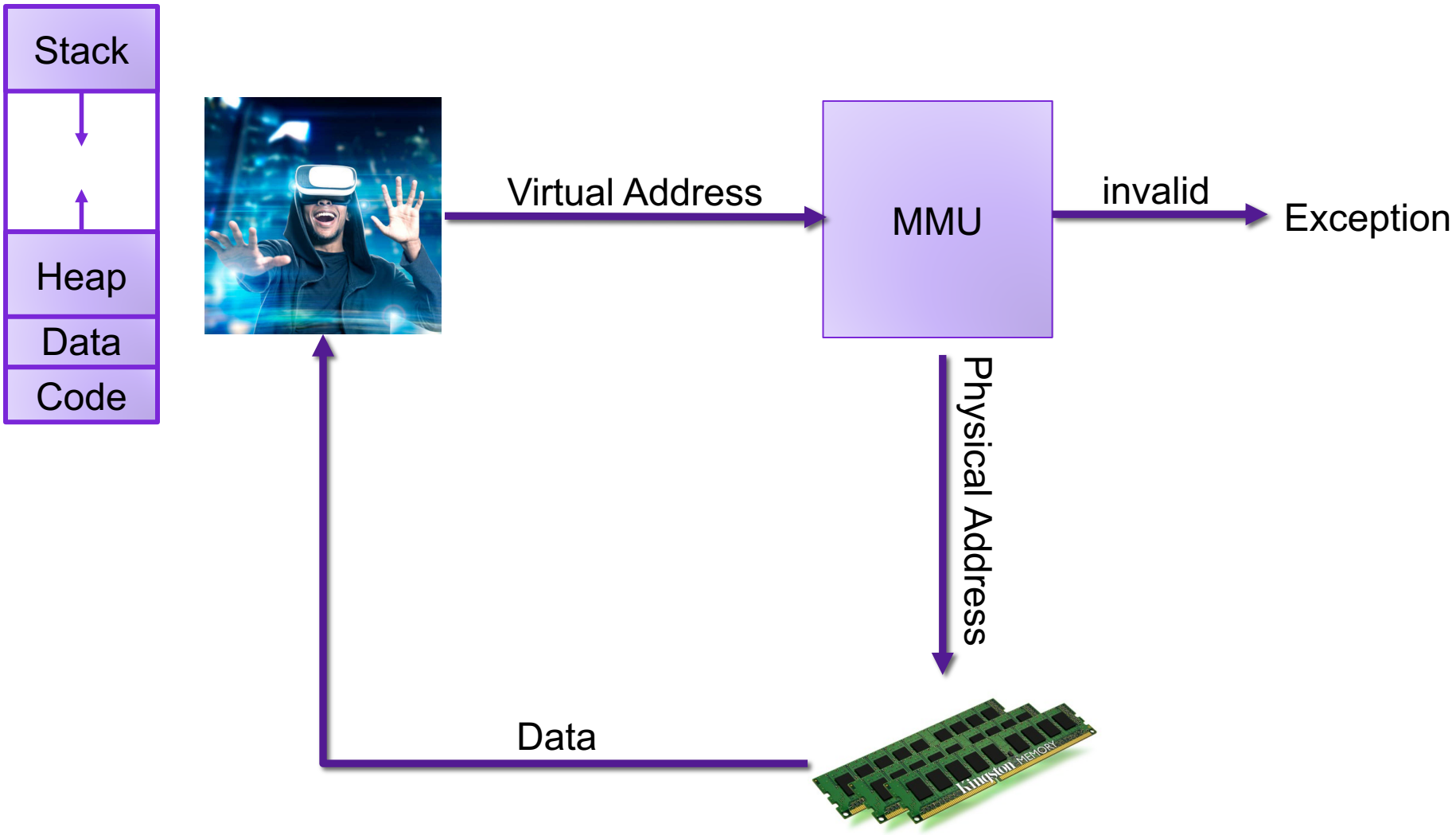
- Computer runs many processes simultaneously
- Running program “top” on Mac
  - System has 123 processes, 5 of which are active
  - Identified by Process ID (PID)

Virtual Memory Goals

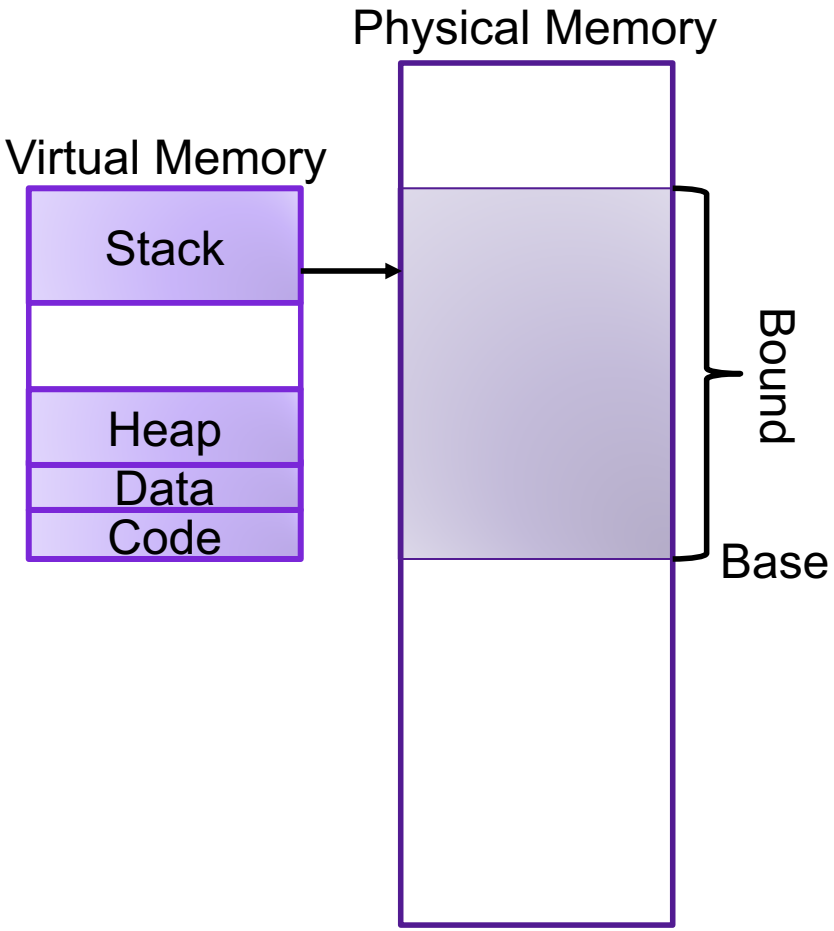


- **Isolation:** don't want different process states collided in physical memory
- **Efficiency:** want fast reads/writes to memory
- **Sharing:** want option to overlap for communication
- **Utilization:** want best use of limited resource
- **Virtualization:** want to create illusion of more resources

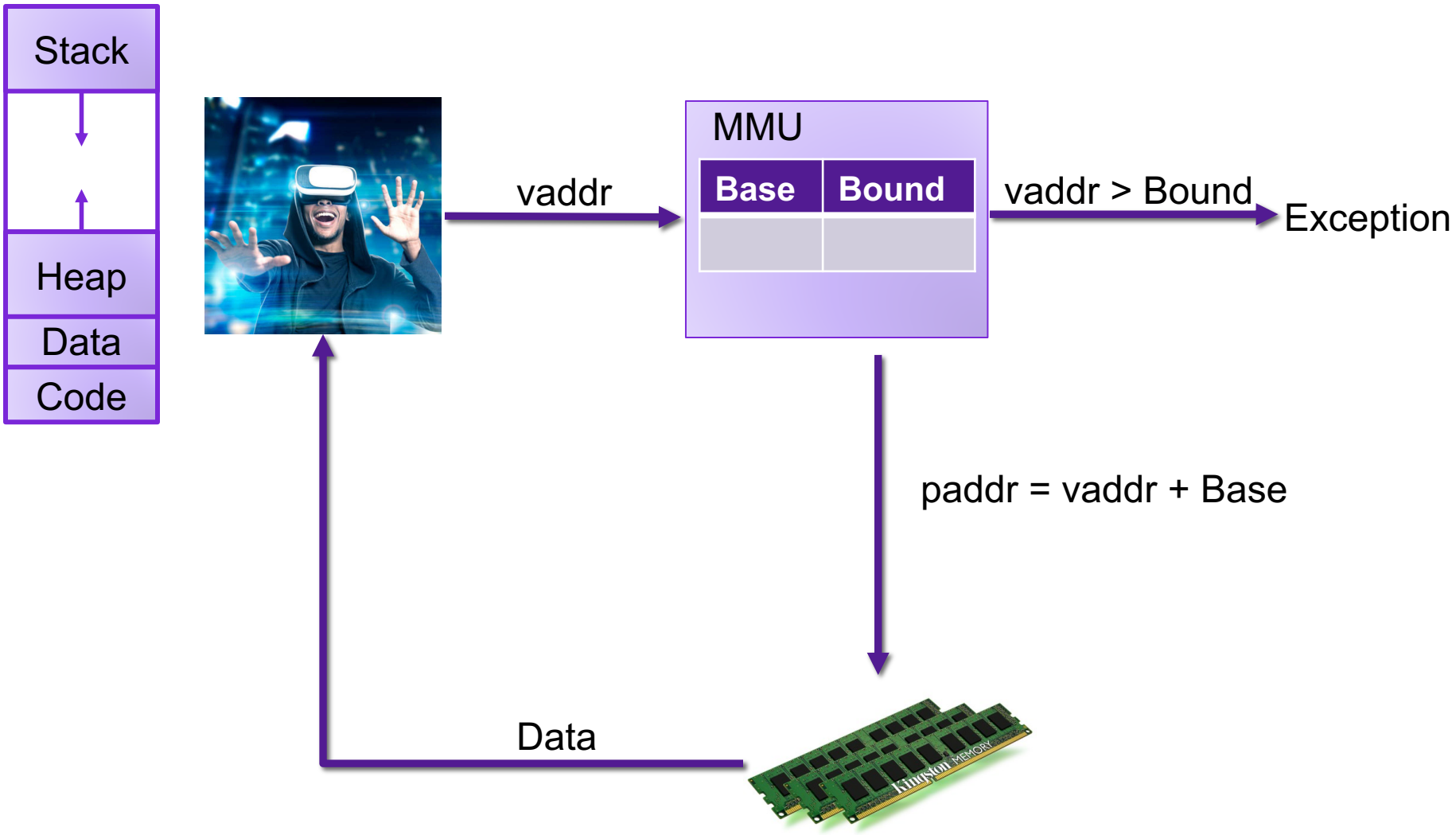
Address Translation



Base-and-Bound



Base-and-Bound



Exercise 1: Base-and-Bound

Assume that you are currently executing a process P with Base 0x1234 and Bound 0x100.

- What is the physical address that corresponds to the virtual address 0x47?
- What is the physical address that corresponds to the virtual address 0x123?

Exercise 1: Base-and-Bound

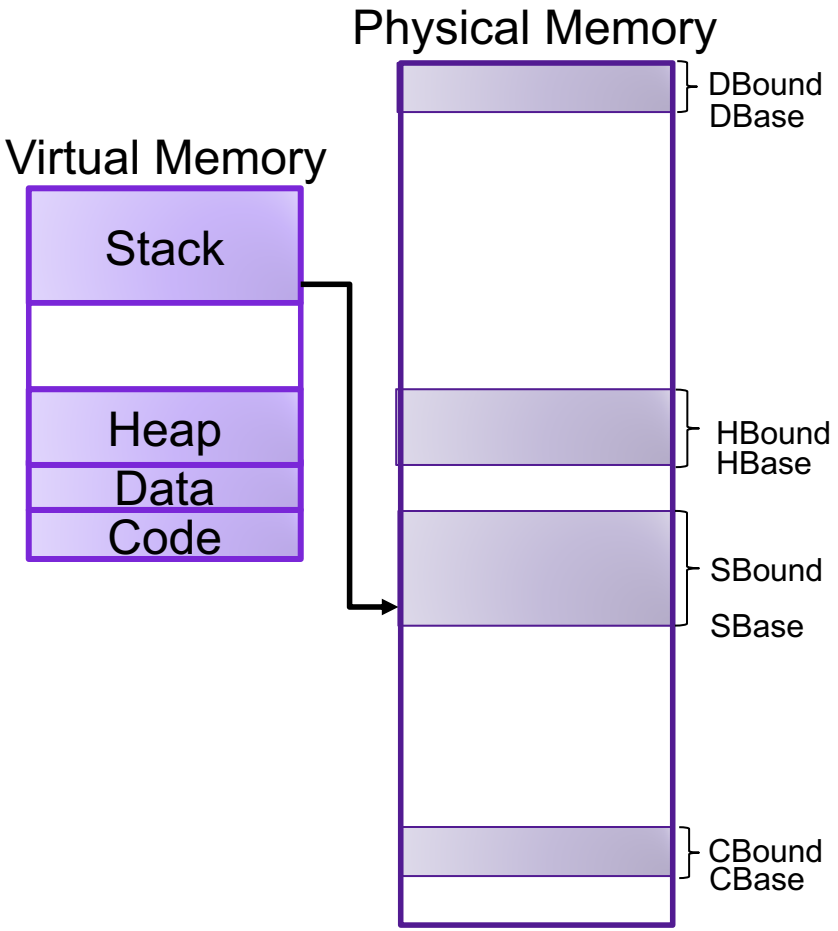
Assume that you are currently executing a process P with Base 0x1234 and Bound 0x100.

- What is the physical address that corresponds to the virtual address 0x47? **0x127b**
- What is the physical address that corresponds to the virtual address 0x123? **invalid**

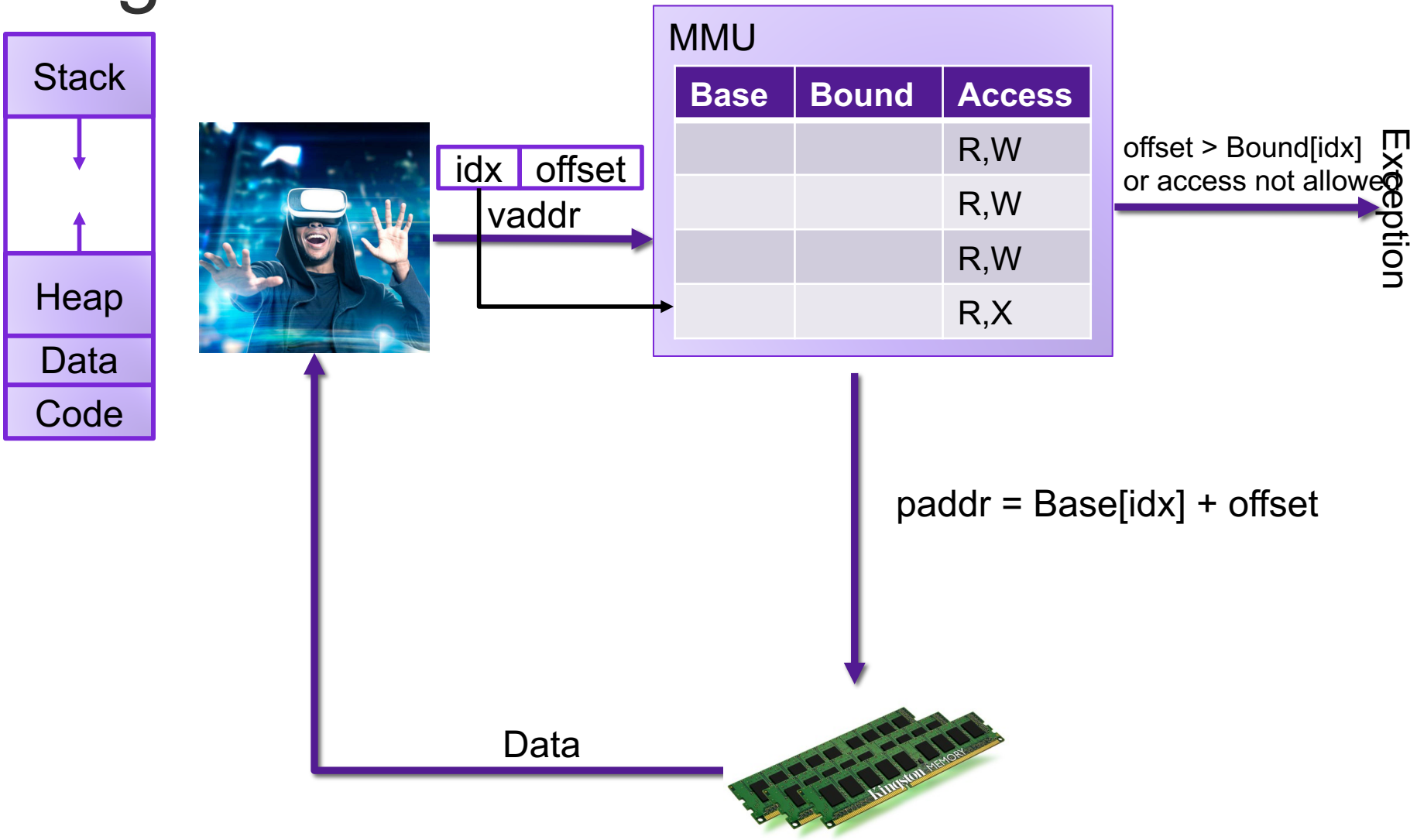
Evaluating Base-and-Bound

- **Isolation:** don't want different process states collided in physical memory
- **Efficiency:** want fast reads/writes to memory
- **Sharing:** want option to overlap for communication
- **Utilization:** want best use of limited resource
- **Virtualization:** want to create illusion of more resources

Segmentation



Segmentation



Exercise 2: Segmentation

Assume that you are currently executing a process P with the following segment table:

Base	Bound	Access
0x4747	0x80	R,W
0x2424	0x40	R,W
0x0023	0x80	R,W
0x1000	0x200	R,X

- What is the physical address that corresponds to the virtual address 0x001?
- What is the physical address that corresponds to the virtual address 0xD47?

Exercise 2: Segmentation

Assume that you are currently executing a process P with the following segment table:

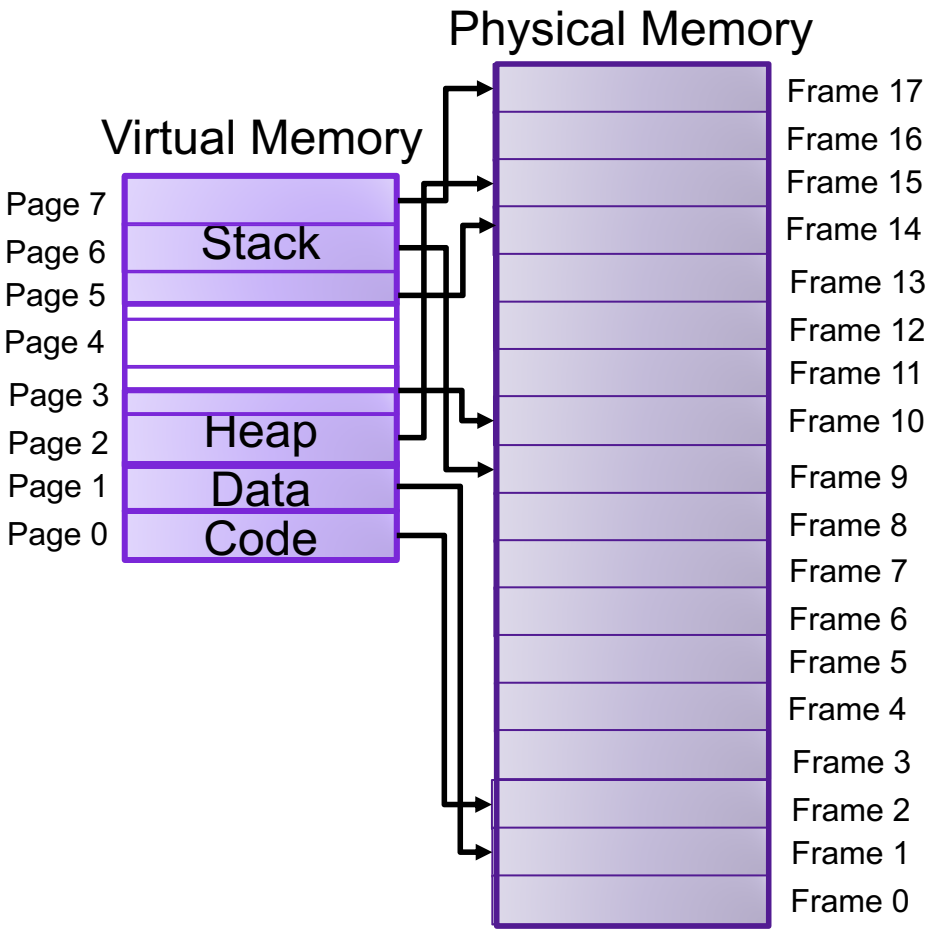
Base	Bound	Access
0x4747	0x80	R,W
0x2424	0x40	R,W
0x0023	0x80	R,W
0x1000	0x200	R,X

- What is the physical address that corresponds to the virtual address 0x001? | 00 | 0000000001 | **0x4748**
- What is the physical address that corresponds to the virtual address 0xD47? | 11 | 0101000111 | **0x1147**

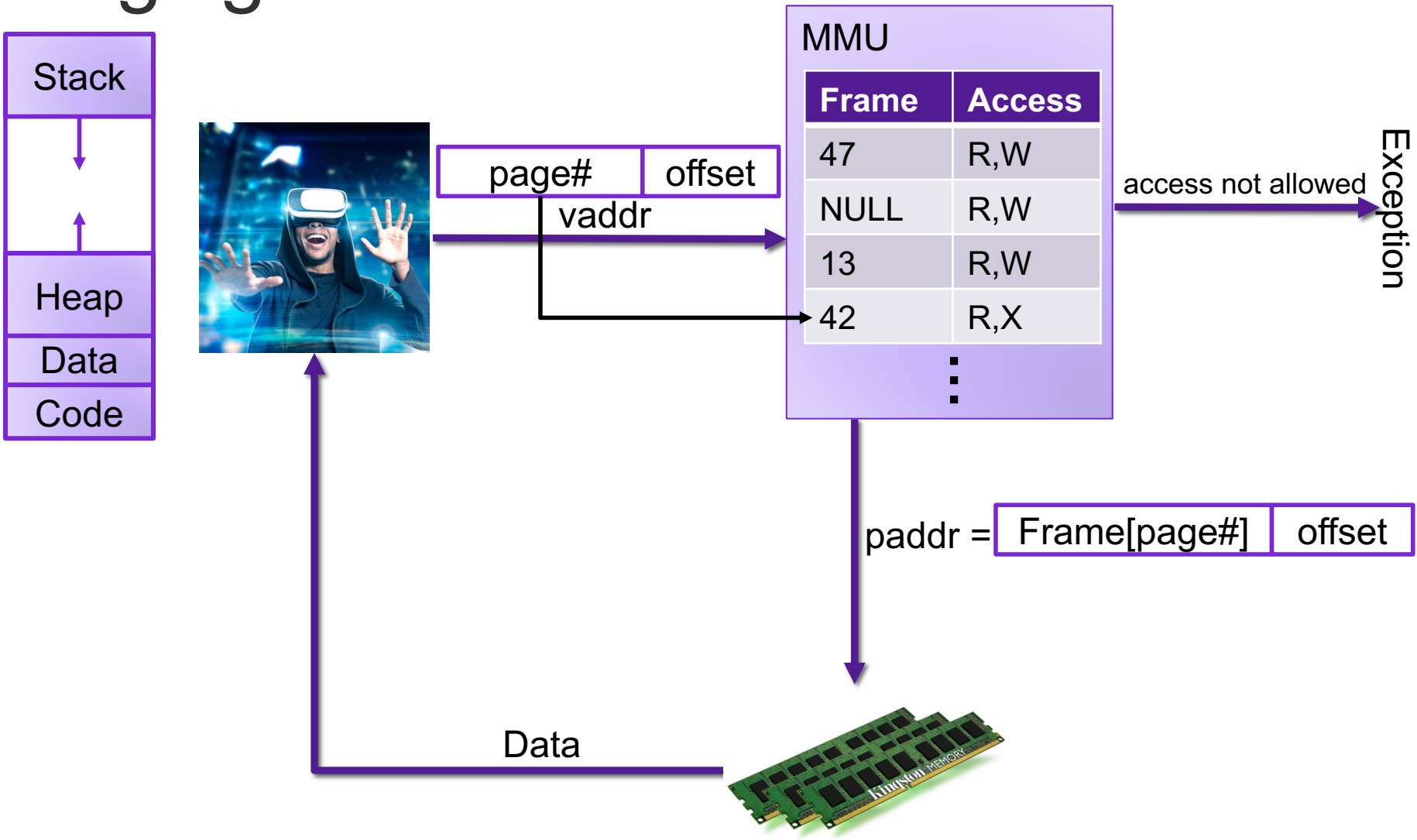
Evaluating Segmentation

- **Isolation:** don't want different process states collided in physical memory
- **Efficiency:** want fast reads/writes to memory
- **Sharing:** want option to overlap for communication
- **Utilization:** want best use of limited resource
- **Virtualization:** want to create illusion of more resources

Paging


Physical Memory
Virtual Memory
Page 7
Page 6
Page 5
Page 4
Page 3
Page 2
Page 1
Page 0
Stack
Heap
Data
Code
Frame 17
Frame 16
Frame 15
Frame 14
Frame 13
Frame 12
Frame 11
Frame 10
Frame 9
Frame 8
Frame 7
Frame 6
Frame 5
Frame 4
Frame 3
Frame 2
Frame 1
Frame 0

Paging


Stack
Heap
Data
Code
page#
offset
vaddr
MMU
Frame
Access
47
R,W
NULL
R,W
13
R,W
42
R,X
access not allowed
Exception
paddr =
Frame[page#]
offset
Data

Exercise 3: Paging

Assume that you are currently executing a process P with the following page table on a system with 16 byte pages:

	Frame	Access
⋮		
0x17	0x47	R,W
0x16	0xF4	R,W
0x15	NULL	R,W
0x14	0x23	R,X
⋮		

- What is the physical address that corresponds to the virtual address 0x147?
- What is the physical address that corresponds to the virtual address 0x16E?

Exercise 3: Paging

Assume that you are currently executing a process P with the following page table on a system with 16 byte pages:

	Frame	Access
0x17	0x47	R,W
0x16	0xF4	R,W
0x15	NULL	R,W
0x14	0x23	R,X

- What is the physical address that corresponds to the virtual address 0x147?

0x237

- What is the physical address that corresponds to the virtual address 0x16E?

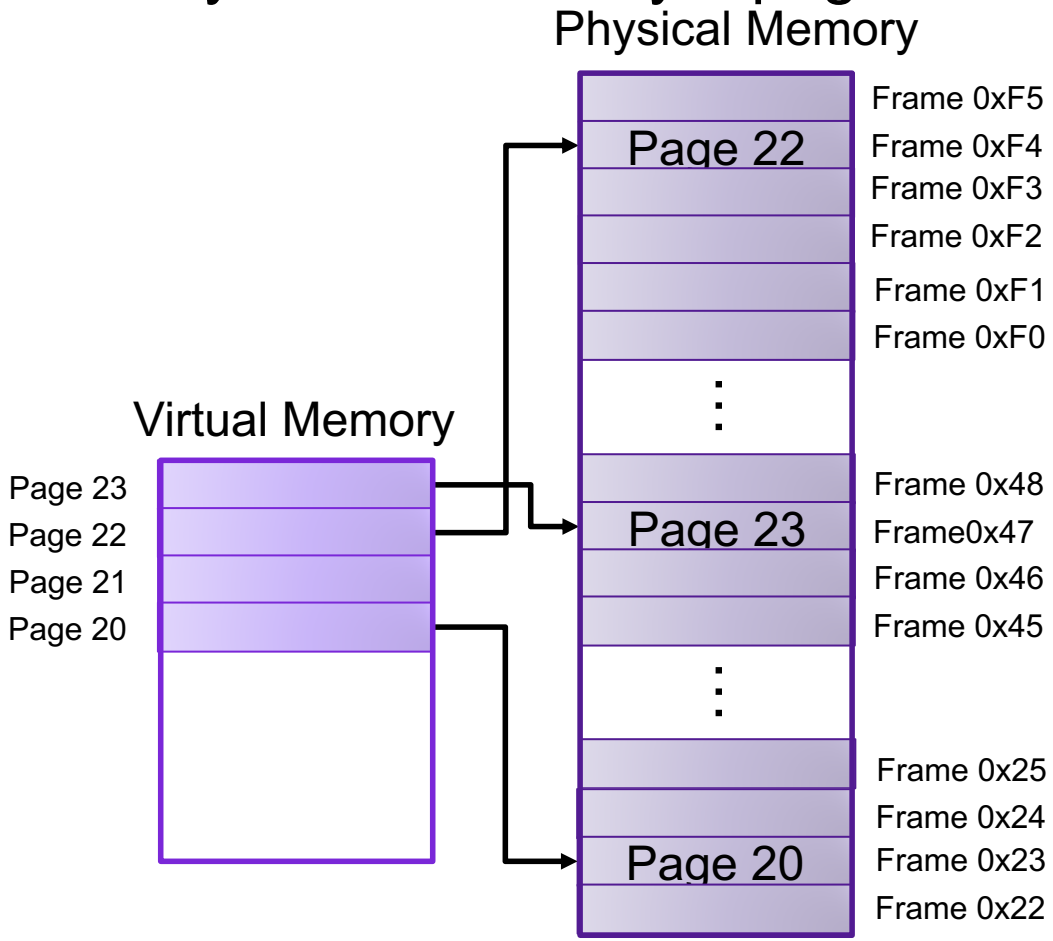
0xF4E

Exercise 3: Paging

Assume that you are currently executing a process P with the following page table on a system with 16 byte pages:

	Frame	Access
⋮		
0x17	0x47	R,W
0x16	0xF4	R,W
0x15	NULL	R,W
0x14	0x23	R,X
⋮		

0x147 → **0x237**



Memory as a Cache

- each page table entry has a valid bit
- for valid entries, frame indicates physical address of page in memory
- a **page fault** occurs when a program requests a page that is not currently in memory
  - takes time to handle, so context switch
  - evict another page in memory to make space (which one?)

MMU

v	Frame	Access
1	47	R,W
0	NULL	R,W
0	13	R,W
1	42	R,X
⋮		

Thrashing

- working set is the collection of a pages a process requires in a given time interval
- if it doesn't fit in memory, program will thrash

Exercise 4: Paging

Assume that you are currently executing a process P with the following page table on a system with 256 byte pages:

	v	Frame	Access
250	1	0x47	R,W
249	1	0x24	R,W
248	0	NULL	R,W
247	0	0x23	R,X

- What is the physical address that corresponds to the virtual address 0xF947?
- What is the physical address that corresponds to the virtual address 0xF700?

Exercise 4: Paging

Assume that you are currently executing a process P with the following page table on a system with 256 byte pages:

	v	Frame	Access
250	1	0x47	R,W
249	1	0x24	R,W
248	0	NULL	R,W
247	0	0x23	R,X

- What is the physical address that corresponds to the virtual address 0xF947? 0xF9 | 0x47 **0x237**
- What is the physical address that corresponds to the virtual address 0xF700? 0xF7 | 0x00 **0xF4E**

Evaluating Paging

- **Isolation:** don't want different process states collided in physical memory
- **Efficiency:** want fast reads/writes to memory
- **Sharing:** want option to overlap for communication
- **Utilization:** want best use of limited resource
- **Virtualization:** want to create illusion of more resources

Exercise 5: Feedback

1. Rate how well you think this recorded lecture worked
    1. Better than an in-person class
    2. About as well as an in-person class
    3. Less well than an in-person class, but you still learned something
    4. Total waste of time, you didn't learn anything
2. How much time did you spend on this video (including exercises)?
3. Do you have any particular questions you'd like me to address in this week's problem session?
4. Do you have any other comments or feedback?