Engineering &
Computer Science
Syracuse University

CIS 657 – Principles of Operating Systems

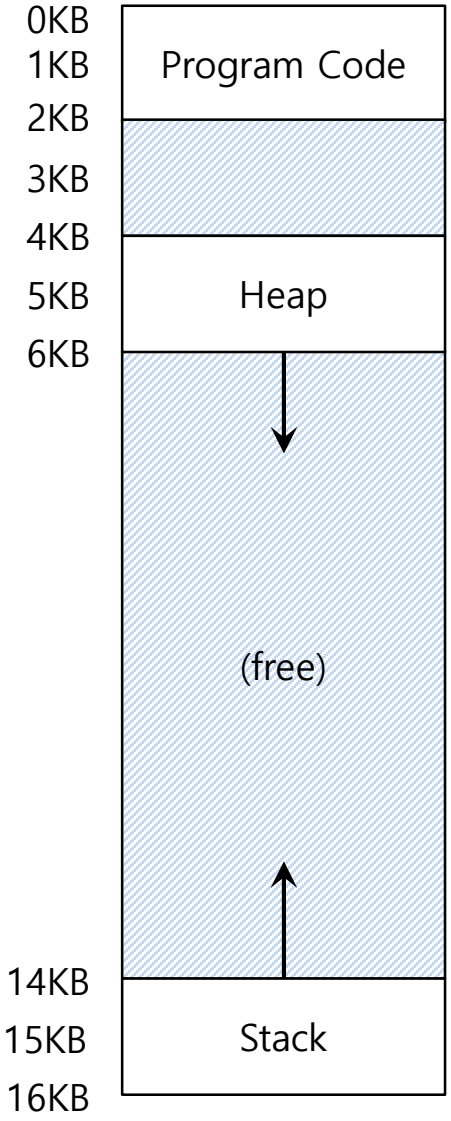
Topic: Memory – Segmentation

Endadul Hoque

Acknowledgement

- Youjip Won (Hanyang University)
- OSTEP book – by Remzi and Andrea Arpaci-Dusseau (University of Wisconsin)

Inefficiency of the Base and Bound Approach



- **Big chunk of "free"** space
- "free" space **takes up** physical memory.
- Hard to run when an address space **does not fit** into physical memory

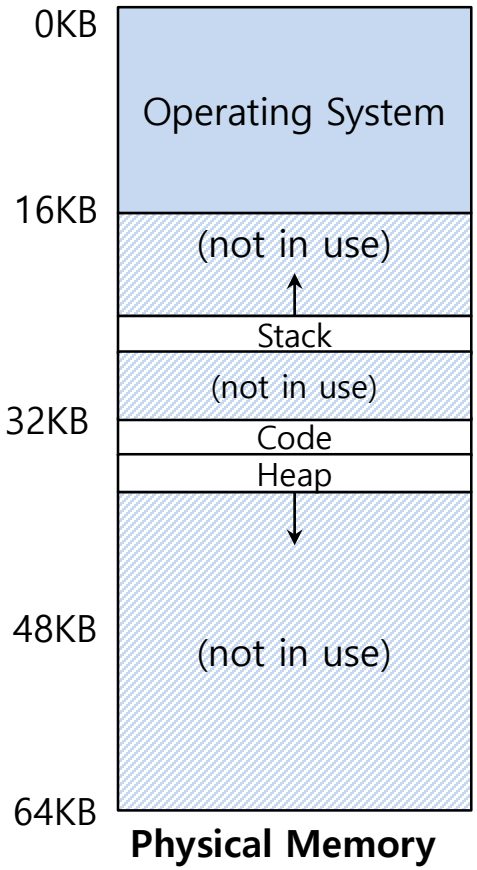
How to support a large address space with (potentially) a lot of free space between stack and heap?

Segmentation

- Segment is just **a contiguous portion** of the address space of a particular length.
  - Logically-different segment: code, stack, heap

- Each segment can be **placed** in **different part of physical memory**.
  - **Base** and **bounds** exist **for each segment**.

Placing Segment In Physical Memory



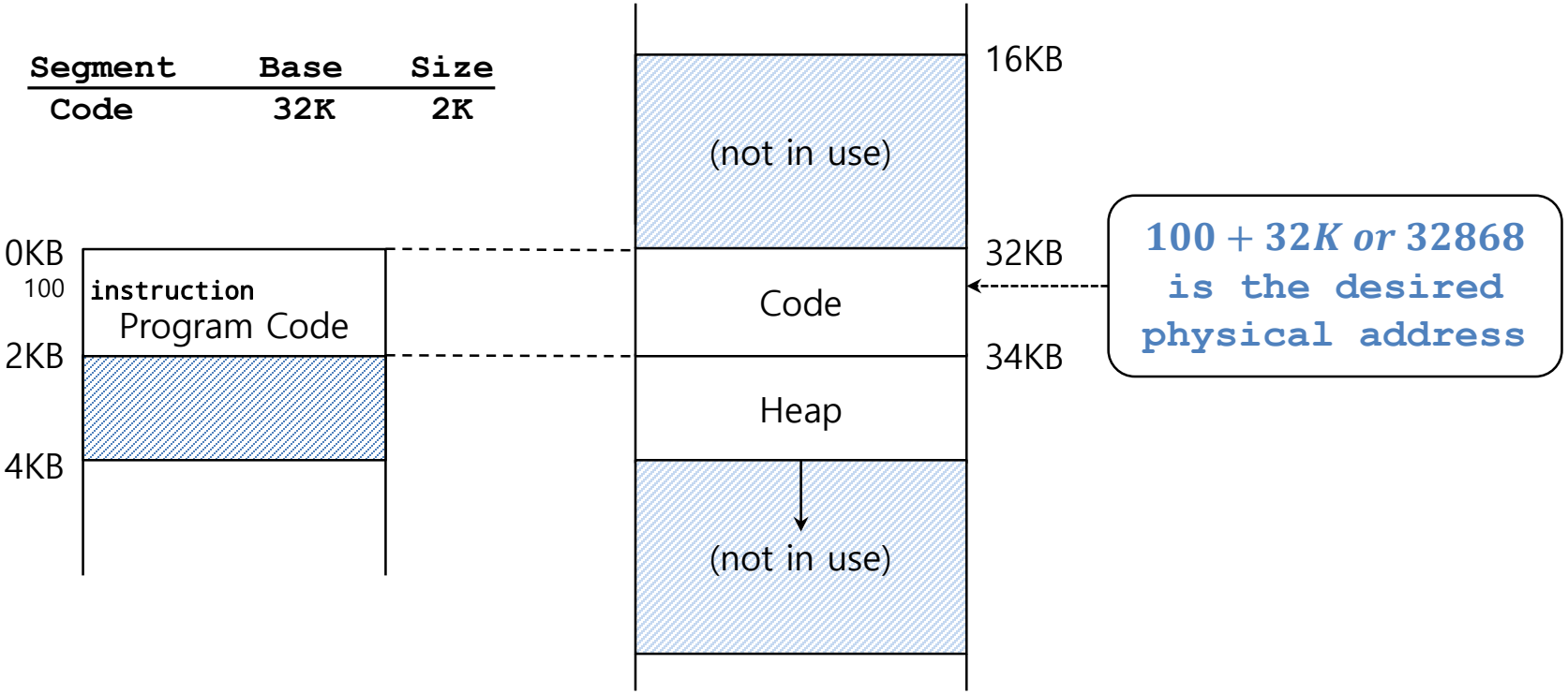
Segment	Base	Size
Code	32K	2K
Heap	34K	2K
Stack	28K	2K

Segment Register Values

Address Translation on Segmentation

$$physical\ address = offset + base$$

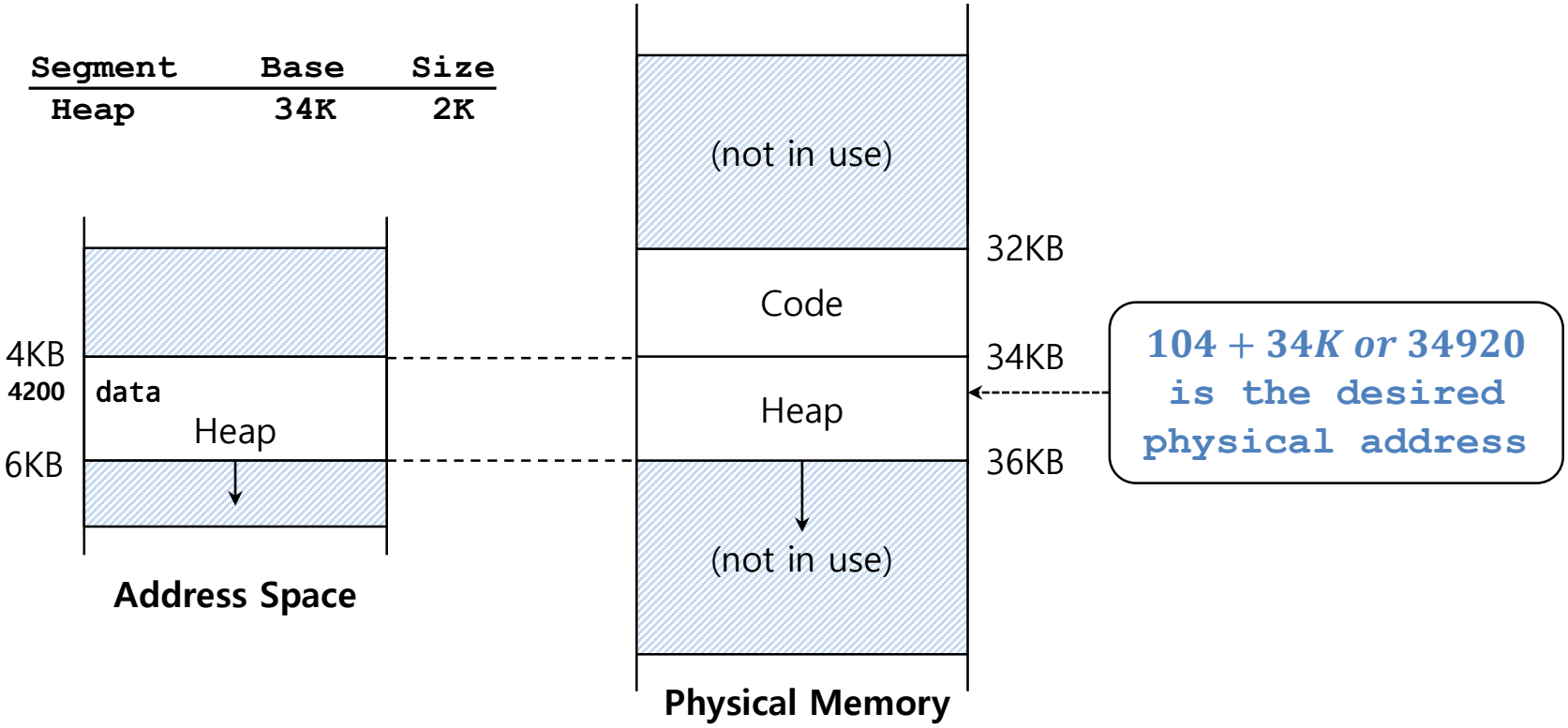
- The `offset` of virtual address 100 is `100`.
  - The code segment **starts at virtual address 0** in address space.



Address Translation on Segmentation

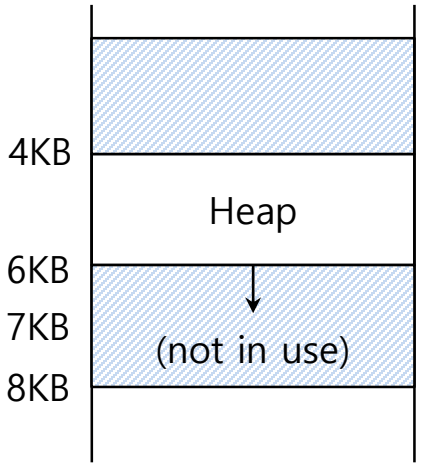
***Virtual address* + *base* is not the correct physical address.**

- The `offset` of virtual address 4200 is `104`.
  - The heap segment **starts at virtual address 4096** in address space.



Segmentation Fault or Violation

- If an **illegal address** such as 7KB which is beyond the end of heap is referenced, the OS raises/causes **segmentation fault**.
  - The hardware detects that address is **out of bounds**.



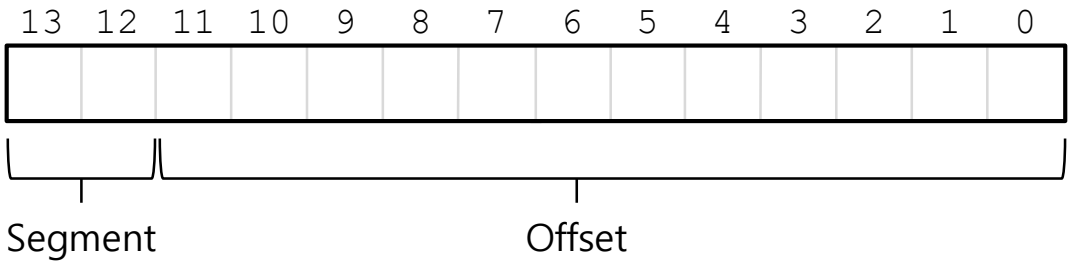
Address Space

Referring to Segment

- The hardware uses segment registers during translation
- How does it know **the offset** into a segment?
- How does it know to **which segment** an address refers?

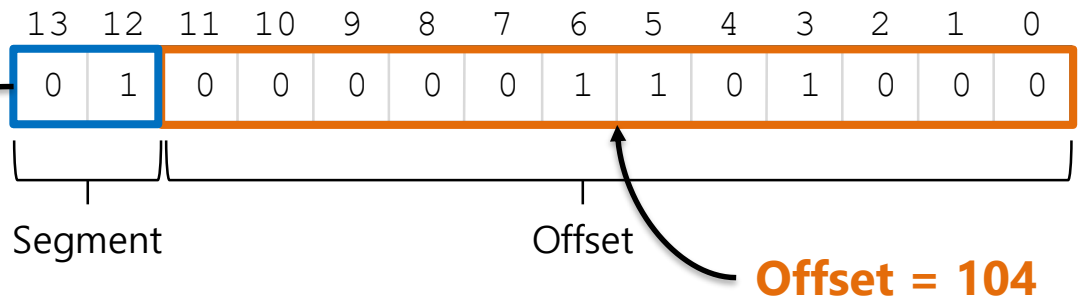
Referring to Segment

- **Explicit approach**
  - Chop up the address space into segments based on the **top few bits** of virtual address.



- Example: virtual address 4200 (01000001101000)

Segment	bits
Code	00
Heap	01
Stack	10
-	11



Referring to Segment

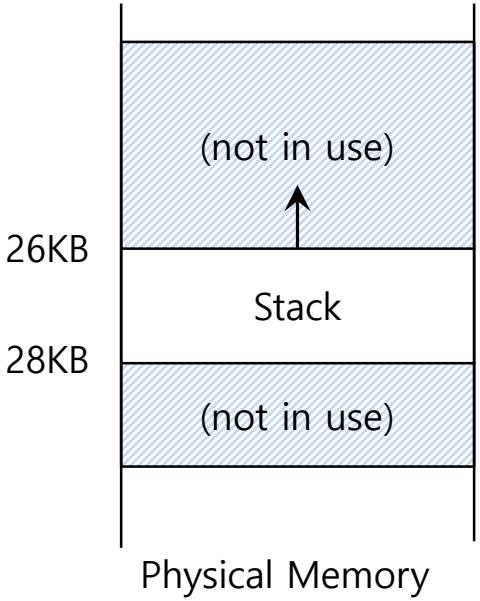
```
1   // get top 2 bits of 14-bit VA
2   Segment = (VirtualAddress & SEG_MASK) >> SEG_SHIFT
3   // now get offset
4   Offset = VirtualAddress & OFFSET_MASK
5   if (Offset >= Bounds[Segment])
6       RaiseException(PROTECTION_FAULT)
7   else
8       PhysAddr = Base[Segment] + Offset
9       Register = AccessMemory(PhysAddr)
```

Pseudocode

- SEG_MASK = 0x3000(110000000000000)
- SEG_SHIFT = 12
- OFFSET_MASK = 0xFFF (001111111111111)

Referring to Stack Segment

- Stack grows **backward**.
- **Extra hardware support** is needed.
  - The hardware checks which way the segment grows.
  - 1: positive direction, 0: negative direction



Segment Register(with Negative-Growth Support)

Segment	Base	Size	Grows Positive?
Code	32K	2K	1
Heap	34K	2K	1
Stack	28K	2K	0

Support for Sharing

- Segment can be **shared between address** spaces
  - Between multiple processes running the same program
  - **Code sharing** is still in-use in systems today
  - OS "secretly" shares (read-only) memory
- Extra hardware support is needed for **Protection bits.**
  - **A few more bits** per segment to indicate **permissions** of **read, write** and **execute**.

Segment Register Values(with Protection)

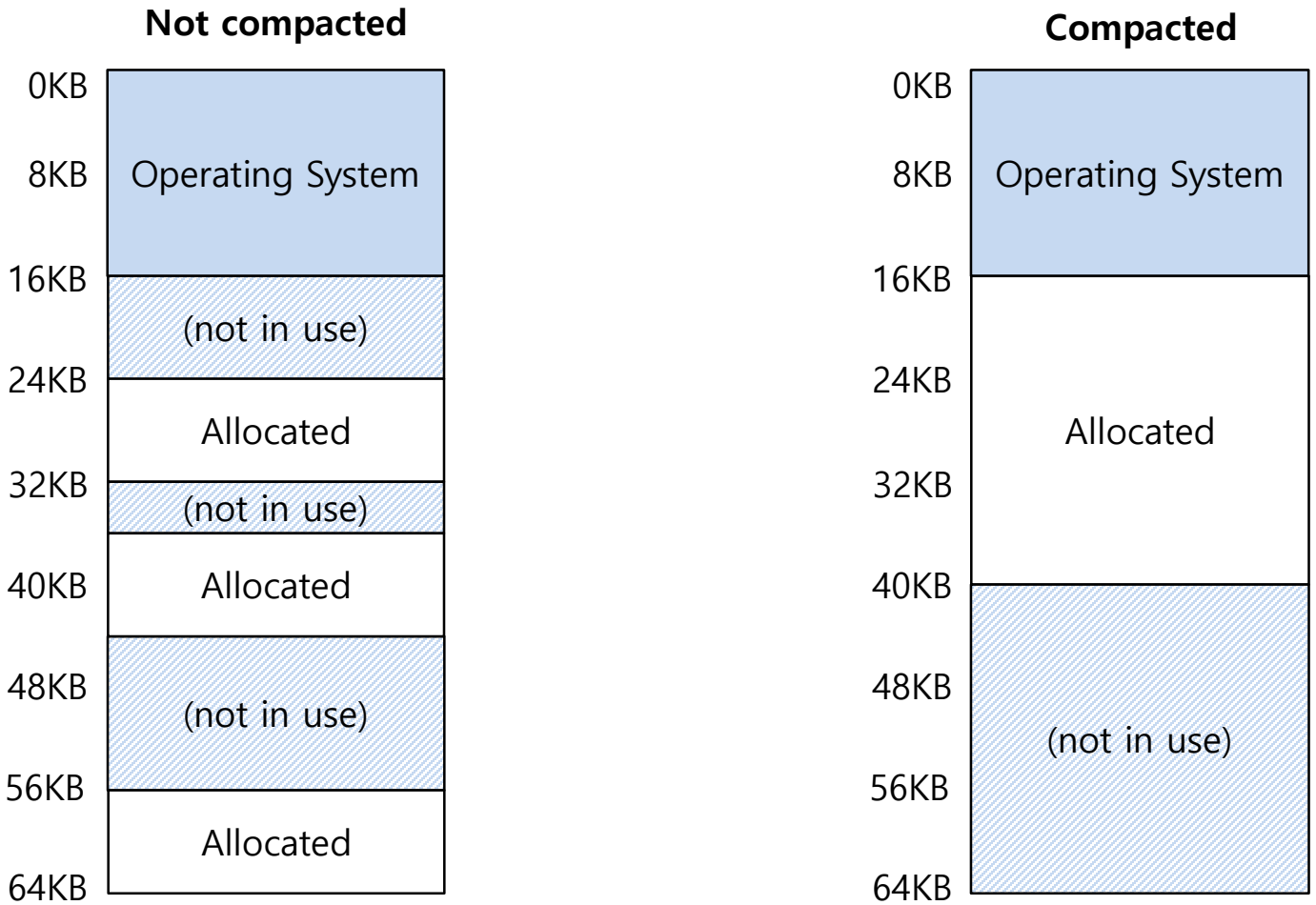
Segment	Base	Size	Grows Positive?	Protection
Code	32K	2K	1	Read-Execute
Heap	34K	2K	1	Read-Write
Stack	28K	2K	0	Read-Write

OS support: Fragmentation

- **External Fragmentation**: little holes of **free spaces** in physical memory that make it difficult to allocate new segments.
  - There is **24KB free**, but **not in one contiguous** segment.
  - The OS **cannot** satisfy the **20KB request**.

- **Compaction**: **rearranging** the exiting segments in physical memory.
  - Compaction is **costly**.
    - **Stop** running process.
    - **Copy** data to somewhere.
    - **Change** segment register value.

Memory Compaction



Reading Material

- **Chapter 16** of OSTEP book – by Remzi and Andrea Arpaci-Dusseau (University of Wisconsin) http://pages.cs.wisc.edu/~remzi/OSTEP/vm-segmentation.pdf

Questions?