Paging and Page Replacement

CS 241

Paging

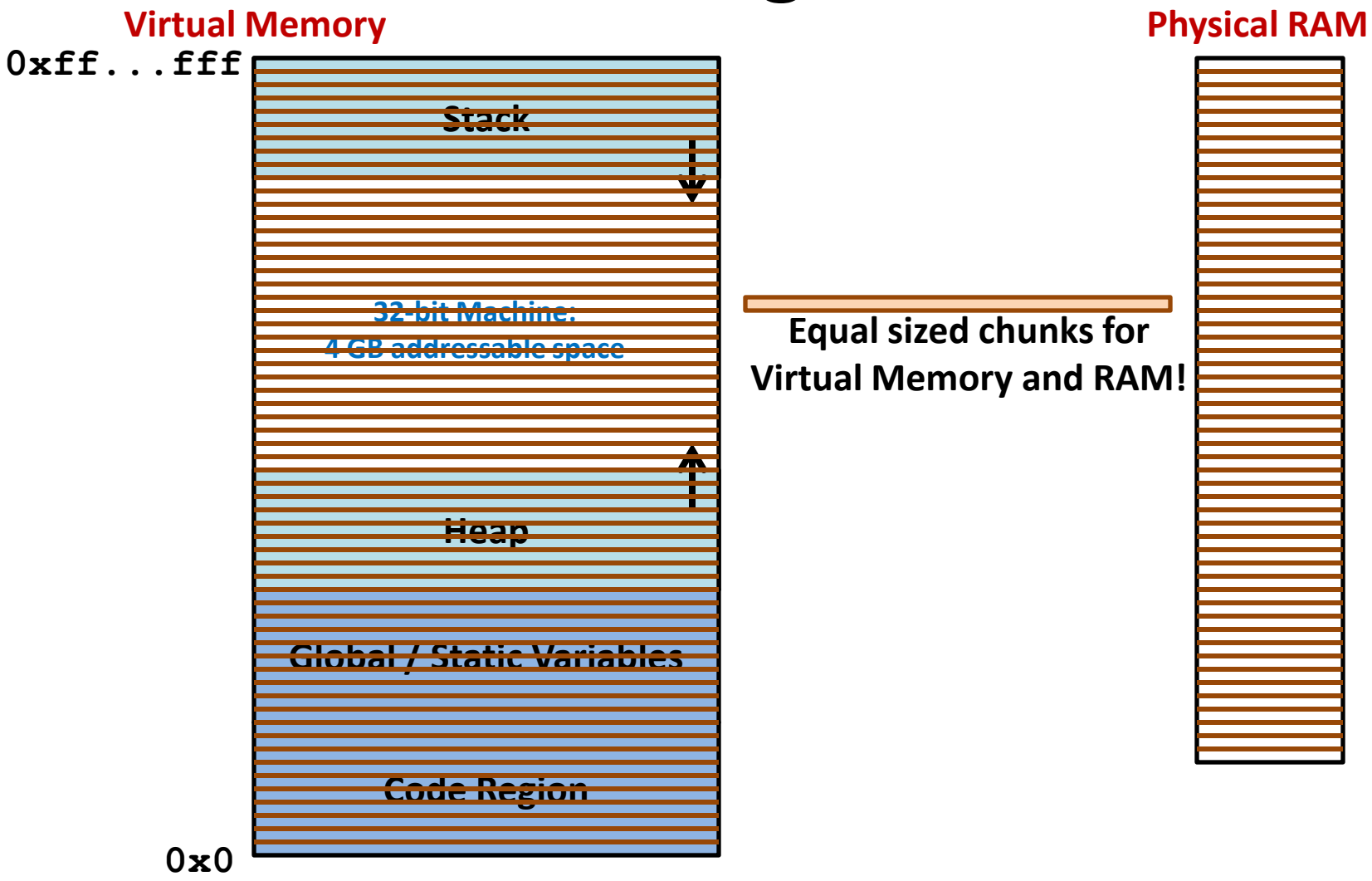
- Motivation:
  - Segmentation maps entire regions of memory to RAM, what happens if we scale this down?
- Answer:
  - Paging

Page

- Divide up **both** the **virtual address space** and the **physical RAM** into equal size chunks.
  - Each of these equal size chunks are called a **page**.

- Pages must always be of size $\mathbf{2^N}$.
  - Eg: 1 KB, 2 KB, 4 KB, 8 KB, etc.

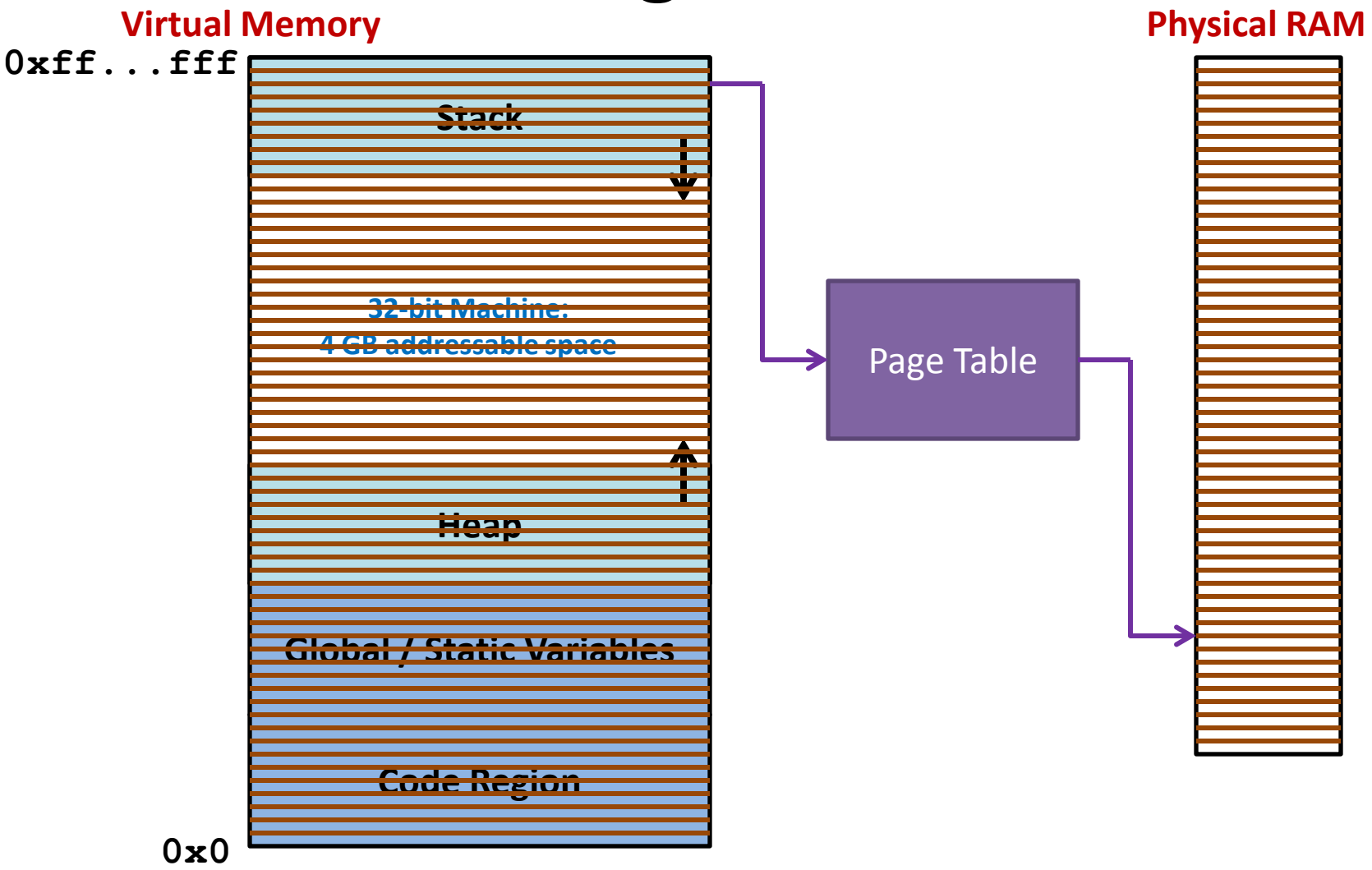
Page


Virtual Memory
Physical RAM
0xff...fff
Stack
32-bit Machine:
4 GB addressable space
Equal sized chunks for
Virtual Memory and RAM!
Heap
Global / Static Variables
Code Region
0x0

Page Table

- A **page table** provides a mapping between each **virtual page** and the **physical page** in RAM.

Page Table


Virtual Memory
Physical RAM
0xff...fff
Stack
32-bit Machine:
4 GB addressable space
Page Table
Heap
Global / Static Variables
Code Region
0x0

How does the Page Table work?

- Given a size of a page, you can always divide a virtual memory address into two pieces:
  - **Page Number / Page Table Index**: What virtual page are we on?
  - **Page Offset**: How far are we inside the page?

How does the Page Table work?

- Example
  - **4 KB** pages

- Virtual Address:

`0x38940392`

- **Virtual Page Number?**
- **Virtual Page Offset?**

How does the Page Table work?

- Example
  - **256 B** pages

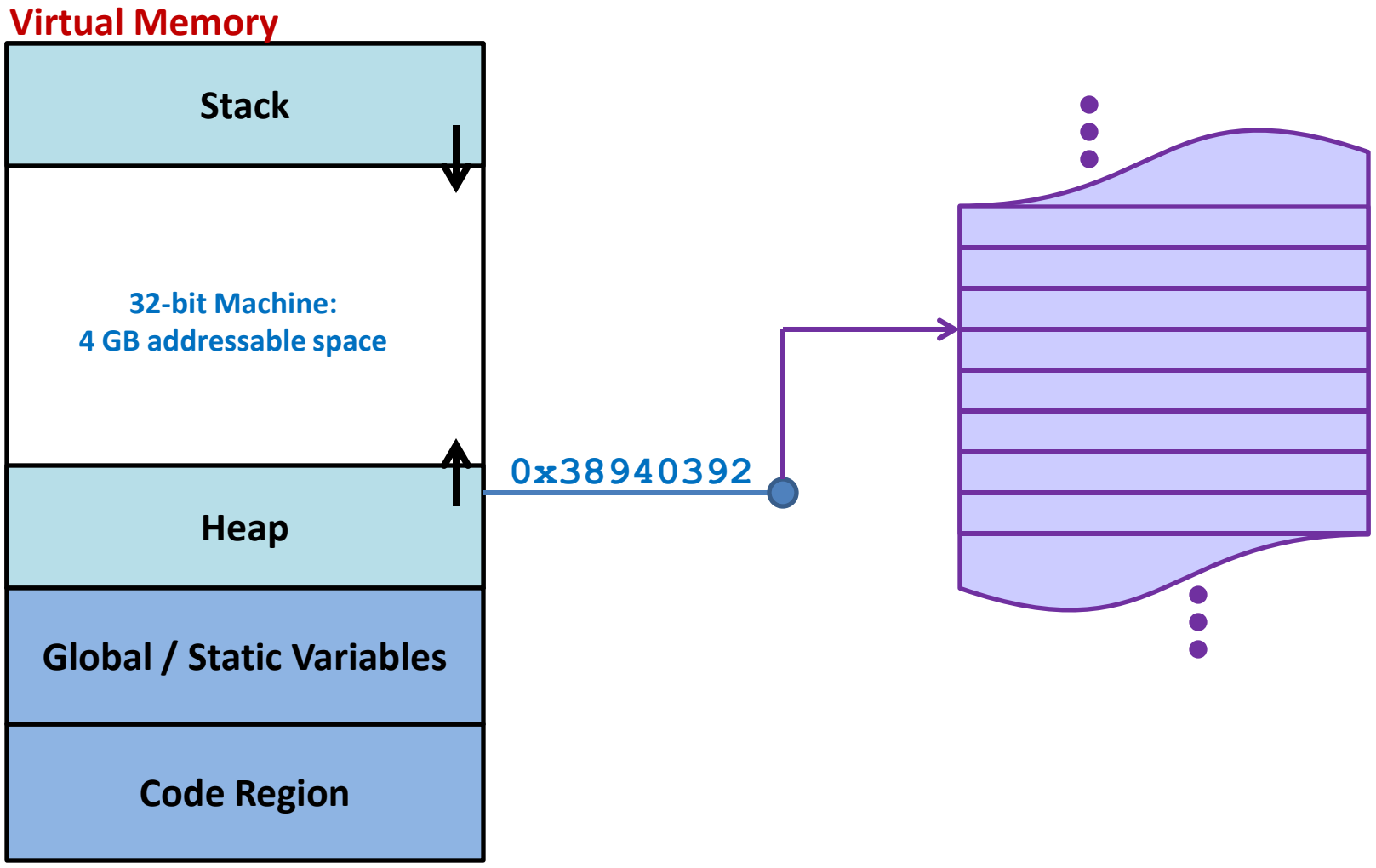
- Virtual Address:

`0x38940392`

- **Virtual Page Number?**
- **Virtual Page Offset?**

Page Table

Assume 4 KB pages.

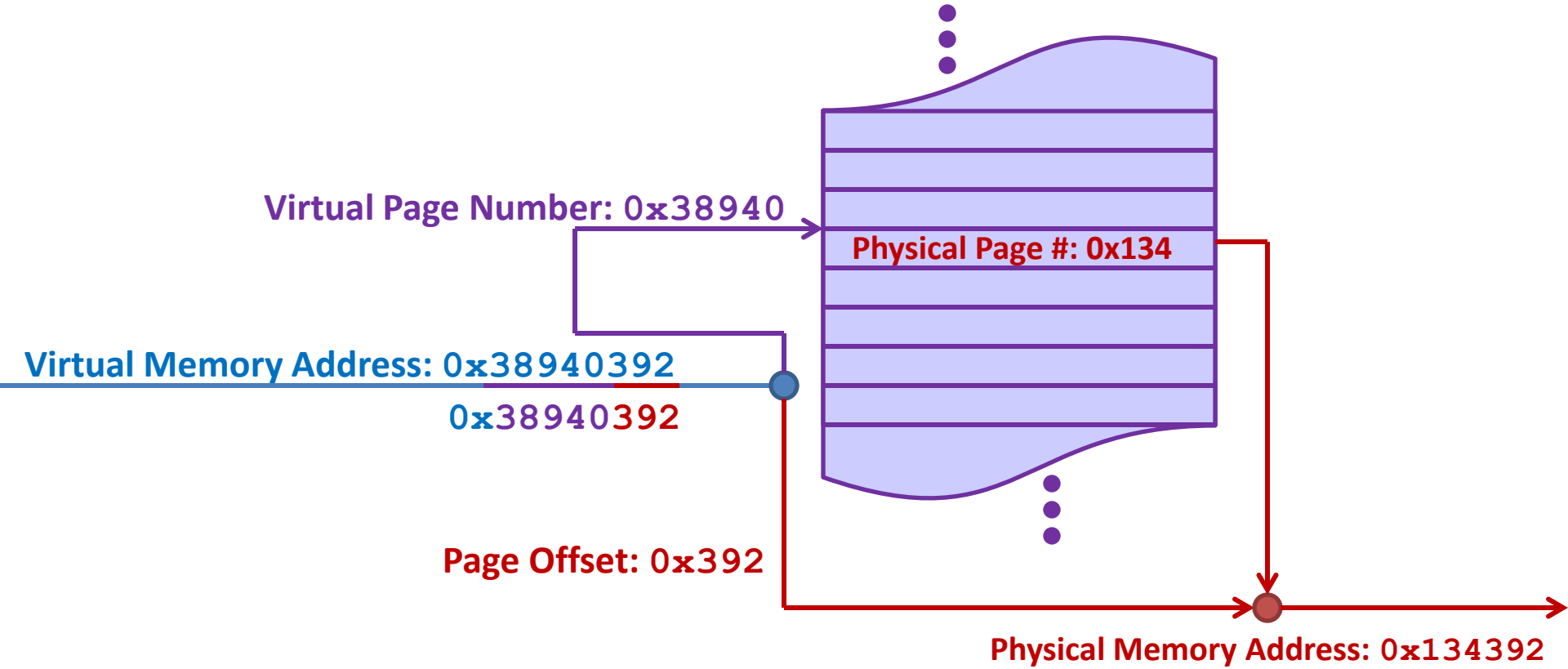


Page Table Entries

- What is the most important thing in each entry in the Page Table, or each **Page Table Entry (PTE)**?

Page Table

Assume 4 KB pages.



Modern Systems…

- A modern economy laptop may have 2 GBs of RAM…
  - …may not be enough RAM for all those programs you run! :(

Resident Bit

- **Solution**: Allow the page table to translate both to RAM and to a hard drive!

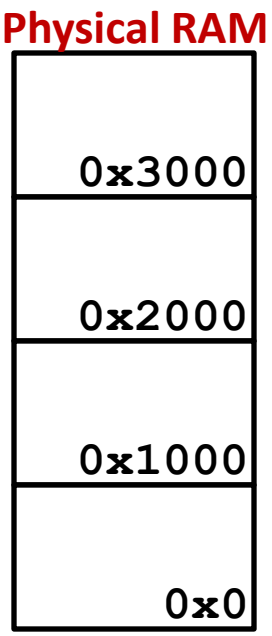
- How?
  - **Resident Bit**: A single bit in every page table entry.
    - **1**: **Resident in RAM**, use the physical page number to translate the virtual address to a physical address.
    - **0**: Not resident in RAM, but it is **on disk**. Instead of a physical page number, a disk page number of the data.

A Really Simple System…

- For an example, we have a system with some really simple properties:
  - 4 KB pages
  - Only 16 KB of RAM
    - Eg: Only 4 total pages.

- **Q**: How many bits are in a physical memory address?

A Really Simple System…



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

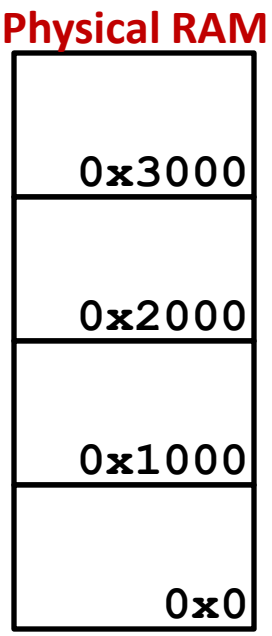
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

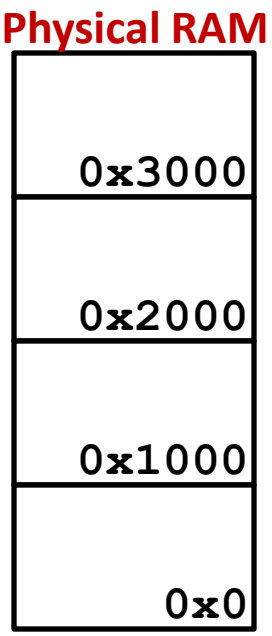
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

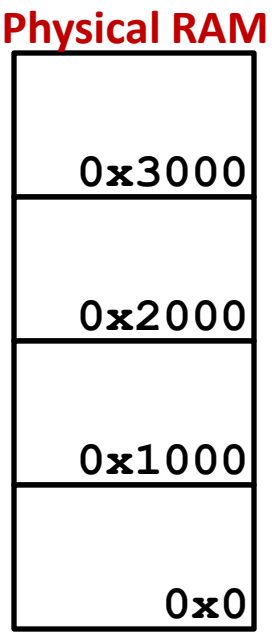
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

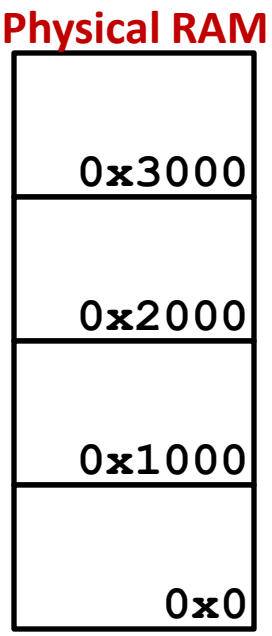
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

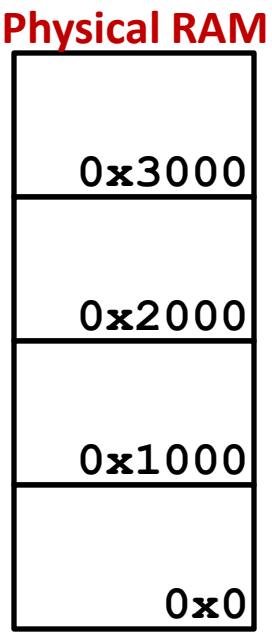
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

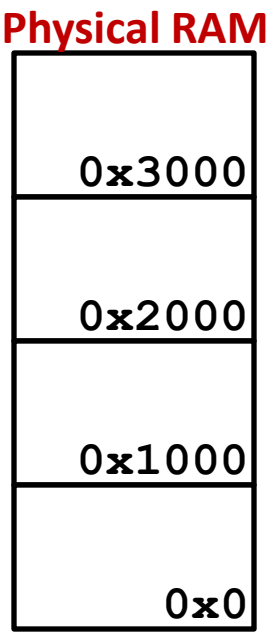
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

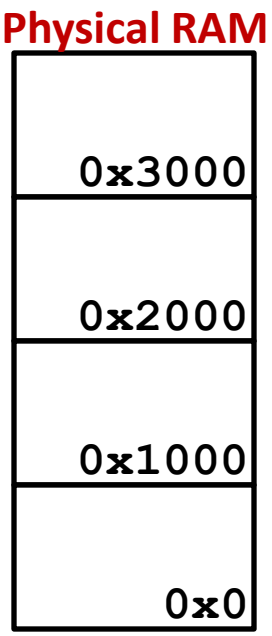
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

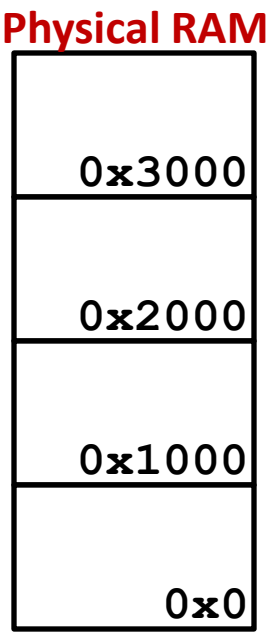
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



A Really Simple System…

Virtual Page #: 17 →

Virtual Page #: 33 →

Virtual Page #: 40 →

Virtual Page #: 17 →

Virtual Page #: 43 →

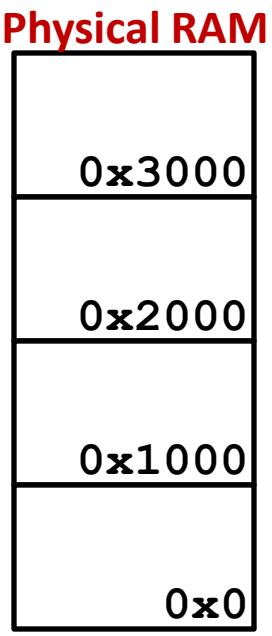
Virtual Page #: 8 →

Virtual Page #: 99 →

Virtual Page #: 33 →

Virtual Page #: 99 →

Virtual Page #: 17 →



Five Page Replacement Algorithms

- **Optimal**
- **FIFO**
- **LRU**
- **LFU**
- **MRU**

Five Page Replacement Algorithms

- **Optimal**
- **FIFO**
- **LRU**
- **LFU**
- **MRU**