virtual vs. physical memory
- types of physical memory
- paging

Wed. March 9, 2016

MIPS Memory

next three lectures

virtual memory (program addresses)

physical memory (program addresses)

줌

process” (running program)

How do multiple programs share the same (finite) address space?

How to reconcile different sizes of program vs. physical memory?

virtual address
physical address

514 - 398 - 3740

cell tower
cell phone

Sizes of Memory

2^10 ≈ 1 KB (kilobyte)
2^20 ≈ 1 MB (megabyte)
2^30 ≈ 1 GB (gigabyte)
2^40 ≈ 1 TB (terabyte)
2^50 ≈ 1 PB (petabyte)
2^60 ≈ 1 EB (exabyte)

Floppy disk (1.4 MB)

(obsolete)

CD ~ 1 GB
DVD ~ 4.9 GB

Hard Disk Drive (HDD)

~ 1 TB

magnetic
optical (laser)
We would like to access Memory in one clock cycle. However, there is a tradeoff between the speed and size of physical memory (can't be large and fast).

**Memory Hierarchy**

- **fast and expensive**
  - access in one clock cycle (10⁻⁶ sec)
- **slow and cheap**
  - access in ~10 clock cycles

**Virtual vs. Physical Memory**

- **Typical physical memory types**
  - RAM (SRAM, DRAM)
  - Flash (SSD)
  - Disk (HDD, CD, DVD)

**Paging**

- how to translate (map) virtual to physical?
- page tables
- page fault and page swap

**MIPS Memory**

There is nothing significant about the square tile geometry in this sketch.
Q: How to translate a virtual address to a physical address?

Example: suppose 1 page = $2^{12}$ bytes

Virtual Memory
(4 GB = $2^{32}$ bytes)

$=> 2^{32} / 2^{12} = 2^{20}$

HDD
(e.g. 1 TB = $2^{40}$ bytes)

$=> 2^{40} / 2^{12} = 2^{28}$ pages

How many pages?

Physical Memory

RAM
(e.g. 1 GB = $2^{30}$ bytes)

$=> 2^{30} / 2^{12} = 2^{18}$ pages

A virtual address is 32 bits. These are the program addresses we have been talking about for the last few weeks.

Note that both the user part of Memory and (part of the) kernel part of Memory is paged.

How to translate (map) a virtual address to a physical address?

Again, take example that a page is $2^{12}$ bytes

virtual address

31        ...                             12   11   ...           0

virtual page number

page offset

table lookup

copy

physical address (RAM)  e.g. 1 GB = $2^{30}$ bytes

Page Fault and Page Swap

- When a MIPS program tries to access an address whose physical page is on disk (HDD), we say that a "page fault" occurs. The page first must be brought into main memory (RAM) before the program can access that address.

- If there is no page available in main memory, then some page first must be moved out of main memory, and then the desired page can be moved in main memory. This is called a page swap.

- The page table must be updated (regardless of whether a page is swapped out).

Page swaps are done by a kernel program (OS) called the page fault handler (return to this in lecture 21 -- interrupts).

valid bit

physical page number

physical page number

page offset

"Valid bit" says whether page is in RAM (1) or on HDD (0).