

# November 19

---

- 5 classes to go!
- No class Wednesday!

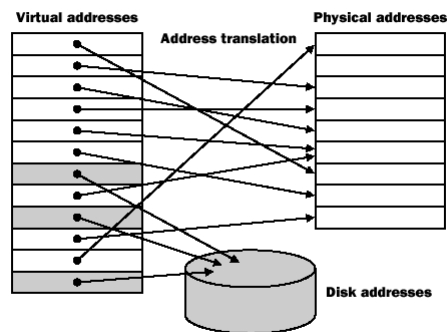
11/19/2001

Comp 120 Fall 2001

1

# Virtual Memory

---



- Main memory is a CACHE for disk
- Advantages:
  - illusion of having more physical memory
  - program relocation
  - protection

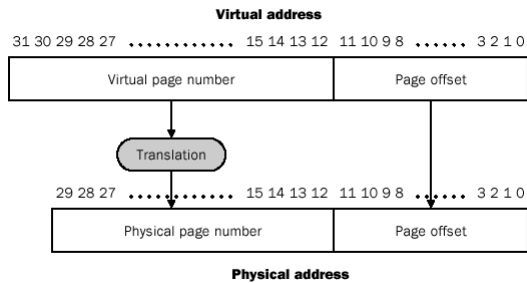
11/19/2001

Comp 120 Fall 2001

2

# Pages: Virtual Memory Blocks

- Page faults: the data is not in memory, retrieve it from disk
  - huge miss penalty, thus pages should be fairly large (e.g., 4KB)
  - reducing page faults is important (LRU is worth the price)
  - can handle the faults in software instead of hardware
  - using write-through is too expensive so we use writeback

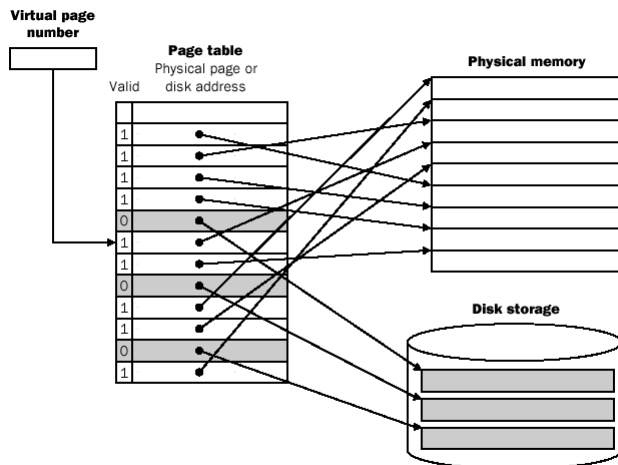


11/19/2001

Comp 120 Fall 2001

3

# Page Tables

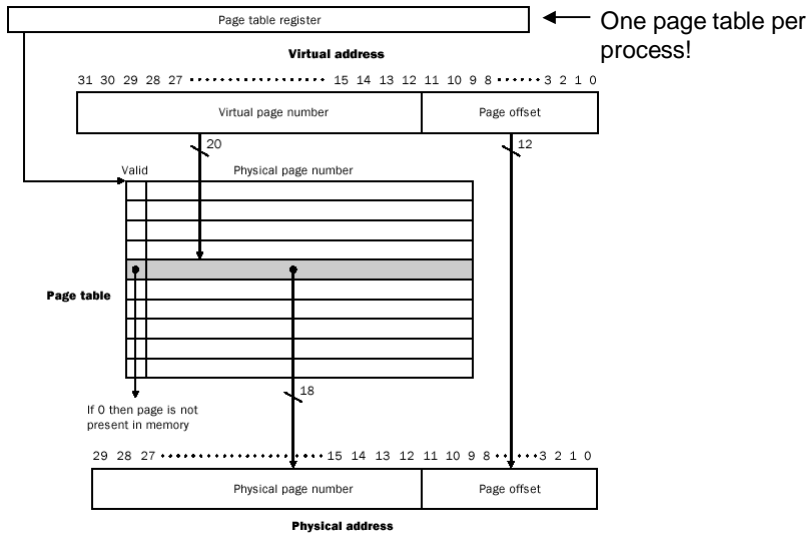


11/19/2001

Comp 120 Fall 2001

4

# Page Tables



11/19/2001

Comp 120 Fall 2001

5

## Where are the page tables?

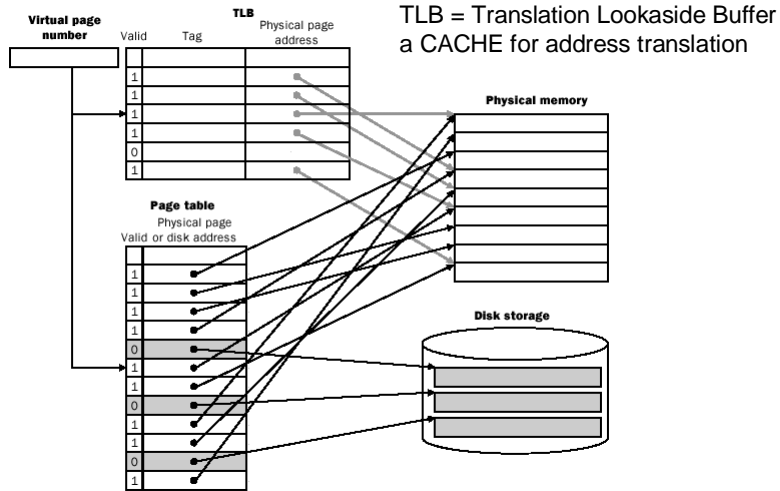
- Page tables are potentially BIG
  - 4kB page, 4MB program, 1k page table entries per program!
    - Powerpoint 14MB
    - Acrobat Distiller 13MB
    - Acrobat 8MB
    - MailCall 8MB
    - HacktiveDesktop 5MB (5 copies!)
    - iCalMinder 4MB
    - iCal 4MB
    - Mulberry 2MB
    - 33 More Processes!
  - May have to page the page tables!
  - Have to look up EVERY address!

11/19/2001

Comp 120 Fall 2001

6

## Making Address Translation Fast



11/19/2001

Comp 120 Fall 2001

7

## What is in the page table?

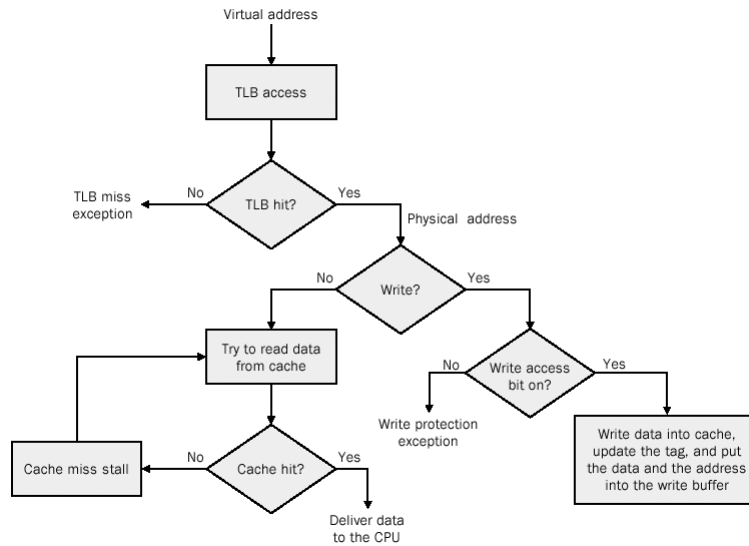
- Address = upper bits of physical memory address OR disk address of page if not in memory
- Valid = bit, set if page is in memory
- Use = bit, set when page is accessed
- Protection = bit (or bits) to specify access permissions
- Dirty = bit, set if page has been written

11/19/2001

Comp 120 Fall 2001

8

## Integrating TLB and Cache



11/19/2001

Comp 120 Fall 2001

9

## Program Relocation?

We want to run multiple programs on our computer “simultaneously”

To start a new program

Without Virtual Memory:

We have to modify all the address references to correspond to the range chosen. This is “relocation”.

With Virtual Memory:

EVERY program can pretend that it has ALL of memory. TEXT segment always starts at 0, STACK always resides a some huge high address (0xffffffff)

11/19/2001

Comp 120 Fall 2001

10

## Protection?

---

We'd like to protect one program from the errors of another

Without Virtual Memory (old Macs, win3-)

One program goes bad (or the programmer makes a mistake) and kills another program or the whole system!

With Virtual Memory (new Macs, win95+)

Every program is isolated from every other. You can't even NAME the addresses in another program.

Each page can have read, write, and execute permissions

11/19/2001

Comp 120 Fall 2001

11

## Some Issues

---

- Processor speeds continue to increase very fast
  - much faster than either DRAM or disk access times
- Design challenge: dealing with this growing disparity
- Trends:
  - synchronous SRAMs (provide a burst of data)
  - redesign DRAM chips to provide higher bandwidth or processing
  - restructure code to increase locality
  - use prefetching (make cache visible to ISA)

11/19/2001

Comp 120 Fall 2001

12

## What these things have in common

Question 1: Where can a block be placed?

Question 2: How is a block found?

Question 3: Which block should be replaced on a cache miss?

Question 4: What happens on a write?

11/19/2001

Comp 120 Fall 2001

13

## Where can a block be placed?

- Direct Mapped Cache: only 1 place for any block (many blocks map to the same place)
- 2-Way Set Associative: 2 places for any block
- 4-Way Set Associative: 4 places for any block
- Fully Associative: anywhere
- Virtual Memory: anywhere

11/19/2001

Comp 120 Fall 2001

14

## How is a block found?

---

- Direct mapped cache: compute the index
- Set associative cache: compute the index, then search
- Fully associative cache: search all cache entries
- Virtual memory: separate lookup table

11/19/2001

Comp 120 Fall 2001

15

## Which block is replaced on miss?

---

- Direct mapped cache have no choice
- Others can use:
  - Random replacement
  - Least Recently Used (LRU)

11/19/2001

Comp 120 Fall 2001

16

## What happens on write?

---

- **Write-through:** write to both the cache and the next lower level
- **Write-back:** write only to the cache, remember that we have to write to the lower level on replacement

11/19/2001

Comp 120 Fall 2001

17

## Misses

---

- Compulsory misses
- Capacity misses
- Conflict misses

11/19/2001

Comp 120 Fall 2001

18

## Classes to go

---

4

11/19/2001

Comp 120 Fall 2001

19