Example machine language

Instructions:

Op-code	Operands	Meaning
1	RXY	Load reg R from memory cell XY
2	RXY	Load reg R with value XY
3	RXY	Store contents of reg R in cell XY
4	$0RS$	Move contents of reg R to reg S
5	RST	Add two's compl. contents of reg S to reg T; store result in R
6	RST	Foating point add
7	RST	OR
8	RST	AND
9	RST	XOR
A	$R0X$	Rotate reg R X bits to right
B	RXY	Jump to XY if $c(R) = c(0)$
C	000	HALT

Note operands are hexadecimal.

Example machine language

One word (cell) is 1 byte.
One instruction is 16 bits.

Machine cycle:

- fetch — get next instr., increment program counter by 2
- decode
- execute (instr)

Example machine language

Example: check if low-order 4 bits of value in reg 1 = 0

2000	load	load zero into reg 0
220F	load	load string 00001111 into reg 2
8312	AND	c(reg 1) AND c(reg 2) —> reg 3 — masking
B3XY	JMP	jump to address XY if c(reg 3) = c(reg 0)

Example machine language

How can we complement a byte in reg 1?

A. load 11 in register 2; OR 3,1,2;

B. load FF in register 2; OR 3,1,2;

C. load 00 in register 2; XOR 3,1,2;

D. load 11 in register 2; XOR 3,1,2;

E. load FF in register 2; XOR 3,1,2;

Vote at m.socrative.com. Room number 415439.

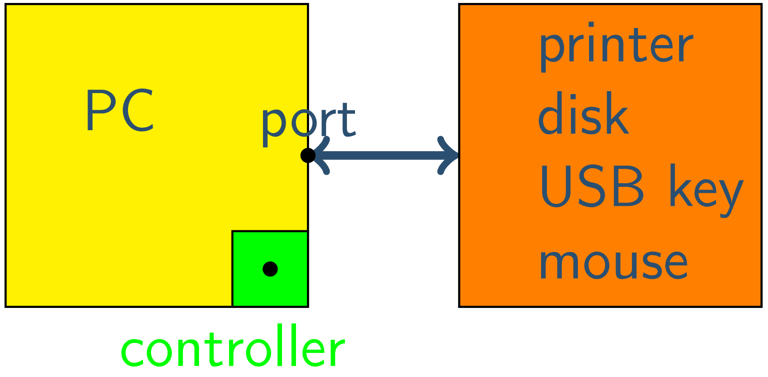
Computer architecture

RISC — reduced instr. set — fast per instr. — cell phones
CISC — complex instruction set — easier to program — PC

Clock

- coordinates activities
- faster clock → faster machine cycle
- Hz — one cycle per second
- MHz — mega Hz (1 million Hz)
- GHz — giga Hz (1000 MHz)
- flop — floating point ops / sec
- benchmark — program to run on different machines for comparison

External devices



motherboard — main circuit board (with CPU, memory)

controller — on motherboard or plugged into motherboard
To reduce number — universal serial bus (USB) or FireWire
Serial — 1 bit at a time (vs. parallel) — fast for short distances

DMA — CPU not involved after starting
(read sector of disk)

If everything uses bus, von Neumann bottleneck.

External devices

Initial connection

- handshaking (also for protocols)
- often status word — is printer OK, paper out, jam,...

Communication rates

- bits per second (bps) / bytes per second (Bps)
- Kbps — standard phone lines
- Mbps — 1,000,000 bps — USB, FireWire 100s of Mbps
- Gbps — 1,000,000,000 bps

External devices

(Time-division) multiplexing

	telephone voice	data from computer	telephone voice	. . .	

data from computer can be modem, xDSL, cable TV

bandwidth – max rate
broadband – high rate

Making computers faster

Example Machine Language
External Devices
Computer Architecture
Operating Systems
Processes

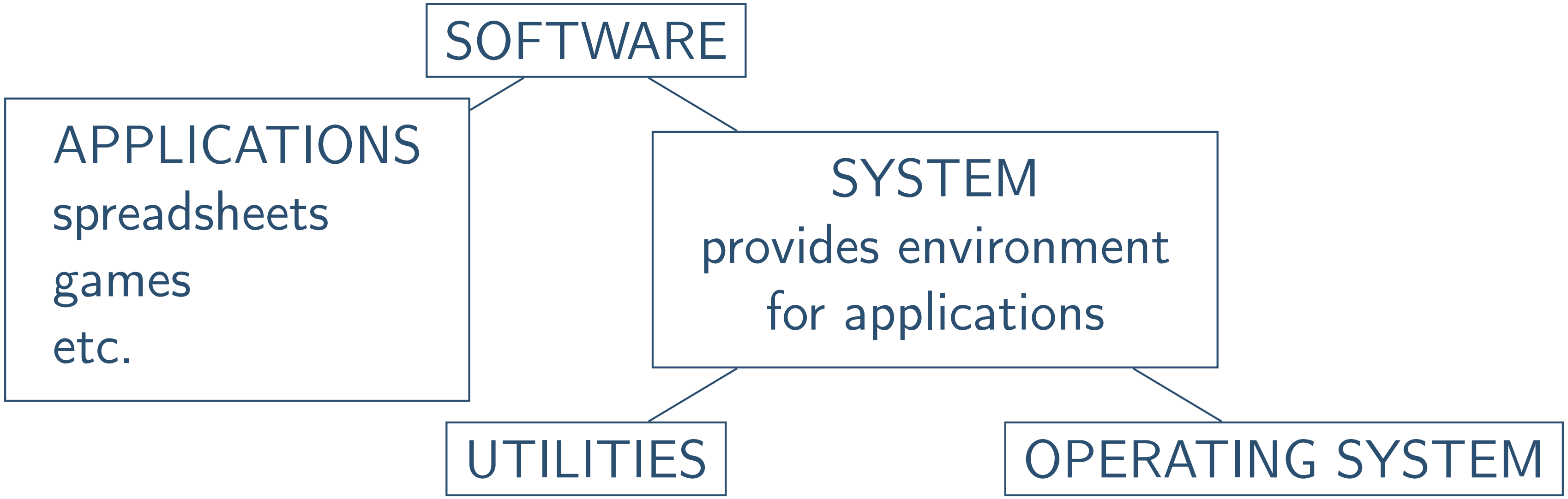
- Pipelining —

ADD	RXY	fetch instruction
ADD	R'X'Y'	decode
ADD	R''X''Y''	perform add
		possibly further divided

- Supercomputers
  — multiprocessor machines now (up to 60,000)
  — SIMD, MIMD
- Multi-core — in single integrated circuit, package
  - dual-core — 2 processors
  - quad-core — 4 processors
  - ...
  - 2 at 2 GHz not as good as 1 at 4 GHz

Operating systems

Operating system — controls operation of computer
controls access to computer's resources

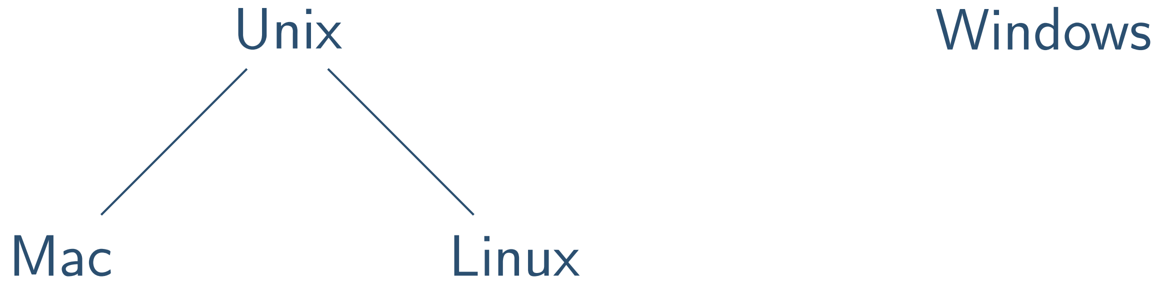


Utilities — unclear boundaries with other things
anti-virus program, formatting a disk, operations with resources, cryptography
browser — no (Internet Explorer?)

Operating systems

User interface = shell

- Command window
- GUI — graphical user interface
  icons, clicking, windows manager



Unix

Mac

Linux

Windows

Basic functions

Basic functions in kernel

1. File manager
   - directories (folders) — organization
   - path — ∼joan/WWWpublic/intro/13slide4.pdf
   - allows access, checks rights
2. Device drivers
   - printer, screen, mouse, etc.
   - communicate with controllers

Basic functions

3. Memory manager
   - in multiuser or multitask system, much to do
   - virtual memory — if more data than for physical memory
   - store some pages in physical memory
     — if used often, leave there — paging is slow
4. Scheduler and dispatcher
   — giving time slices to different tasks or users
5. Bootstrap
   - bootstrap program (boot loader) in ROM (non-volatile)
   - loads rest of OS from disk into main memory (volatile)

Processes

program — instructions
process — execution of program
— 2 users use use same program = 2 processes

process state

- value of program counter
- values in other registers
- values in memory
- used to restart a process

Scheduler

OS must

- give needed resources to processes
  — space in memory, files, devices, etc.
- make sure processes don't interfere with each other
- let processes exchange info if needed

Scheduler

The scheduler maintains a process table, with info for each process:

- memory locations assigned
- priority of process
- status of process
  - ready
  - can continue
  - waiting — for external event
    — completion of read from disk, etc.

Dispatcher

- gets scheduled processes executed by time sharing
- chooses highest priority (given by scheduler)
- gives each process its time slice
- changing processes — process switch/ context switch
  - caused by interrupt
  - dispatcher sets timer to cause interrupt
  - interrupt handler
    - transfers control from process to dispatcher
    - saves and restores process state
    - machine language designed for it

Competition among processors

Allocating access to resources

- sections of code — device driver for printer
- memory addresses

1 process at a time

Competition among processors

flag $\boxed{?}$

0 –	clear	OK
1 –	set	in use

Problem:

Process 1	Is flag clear?
	Yes
interrupt	
Process 2	Is flag clear?
	Yes
	set flag
	use printer
interrupt	
Process 1	set flag
	use printer

Competition among processors

Possible solutions:

1. OK disables interrupts when checking flag
   — re-enables after done with set

2. test-and-set instruction
   — no interrupts in middle of single instruction

The flag is a semaphore (railway signals).
Used to protect critical regions (of code) which require mutual exclusion.

Competition among processors

Another problem:

- Process 1 and Process 2 each need same 2 resources (printer and disk).
- Process 1 gets 1 resource.
- Process 2 gets the other.
- Neither process can continue. — Deadlock

Competition among processors

Deadlock can occur if:

1. There is competition for non-shareable resources
2. Resources requested on partial basis
   — after getting some, may request more
3. Can't take resources back

Possible solutions:

- Deadlock detection and correction — remove condition 3
- Spooling
  - device driver saves data (for printer)
  - sends data later
    — process continues as if printing completed