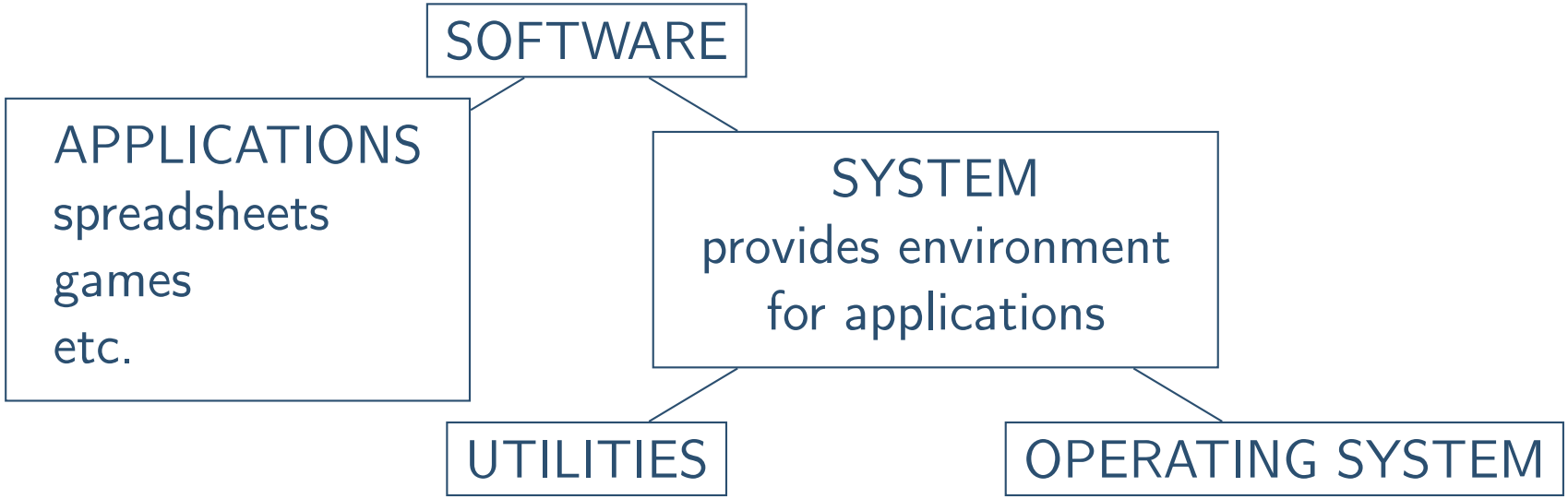
Operating systems

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

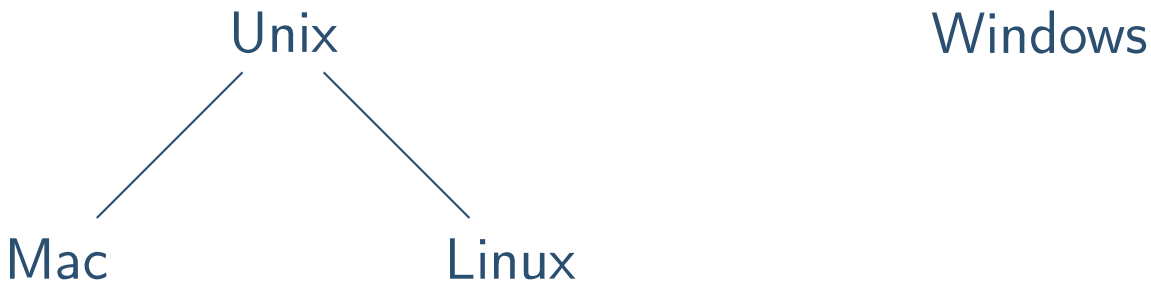


Utilities — unclear boundaries with other things
anti-virus program, formatting a disk, operations with resources, cryptography
browser — no (Internet Explorer?)

Operating systems

User interface = shell

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



Basic functions

Basic functions in kernel

1. File manager
   - directories (folders) — organization
   - path — ∼joan/WWWpublic/intro/14slide5.pdf
   - allows access, checks rights
2. Device drivers
   - printer, screen, mouse, etc.
   - communicate with controllers
   - interrupts or polling (checking status word regularly)

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 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
  - running
  - ready, can continue
  - sleeping, 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 ? 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. OS 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