INTRODUCTION TO COMPUTER SCIENCE

Dr. Yasmine El-Glaly

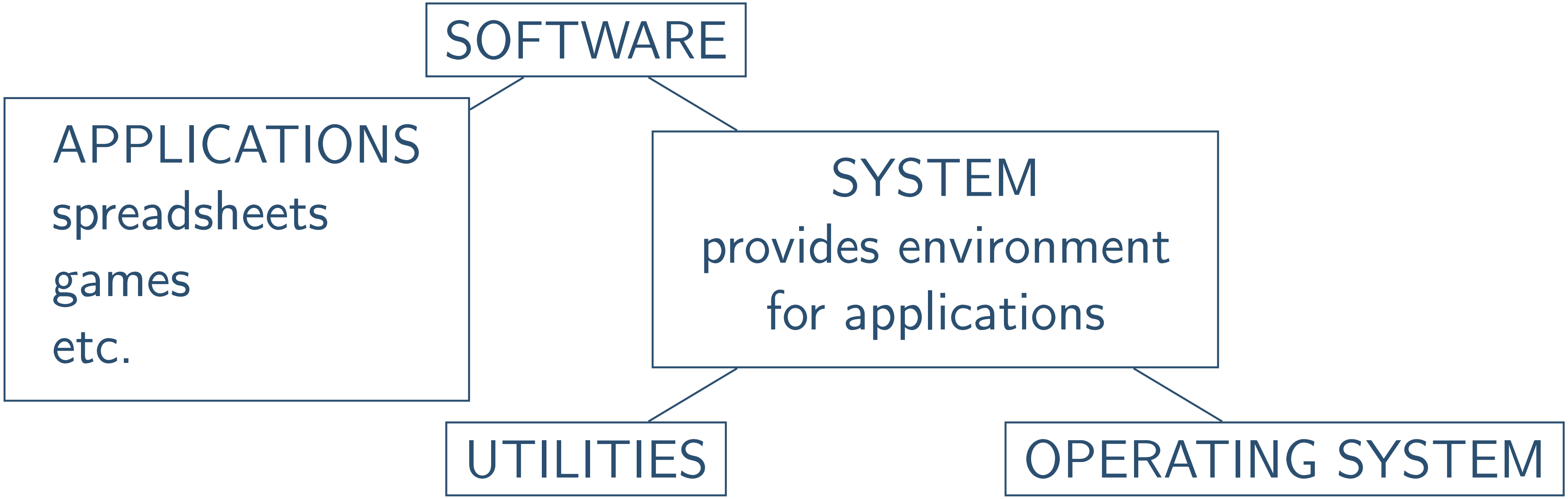
Fall 2013

What is OS?

- An **operating system** is the software that controls the overall operation of a computer.
- It provides the means by which a user can store and retrieve files,
- provides the interface by which a user can request the execution of programs.

- *It is a computer's operating system that transforms the computer hardware into a useful tool.*
- *Examples:*
  - *Windows*
  - *UNIX*
    - *Mac*
    - *Linux (developed by a Finish student in 1991)*

Operating systems



SOFTWARE

APPLICATIONS
spreadsheets
games
etc.

SYSTEM
provides environment
for applications

UTILITIES

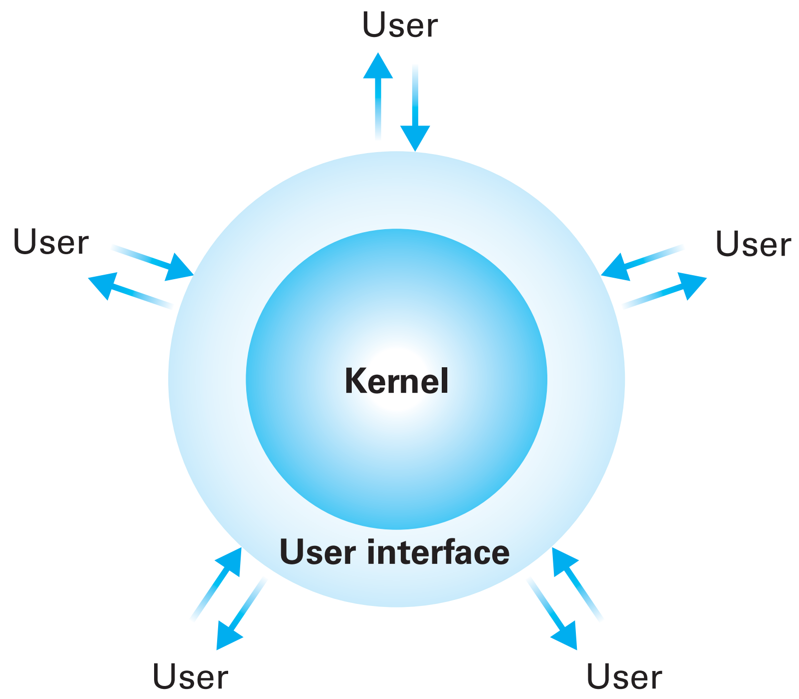
OPERATING SYSTEM

- Utilities — unclear boundaries with other things

anti-virus program, formatting a disk, cryptography

Operating Systems

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



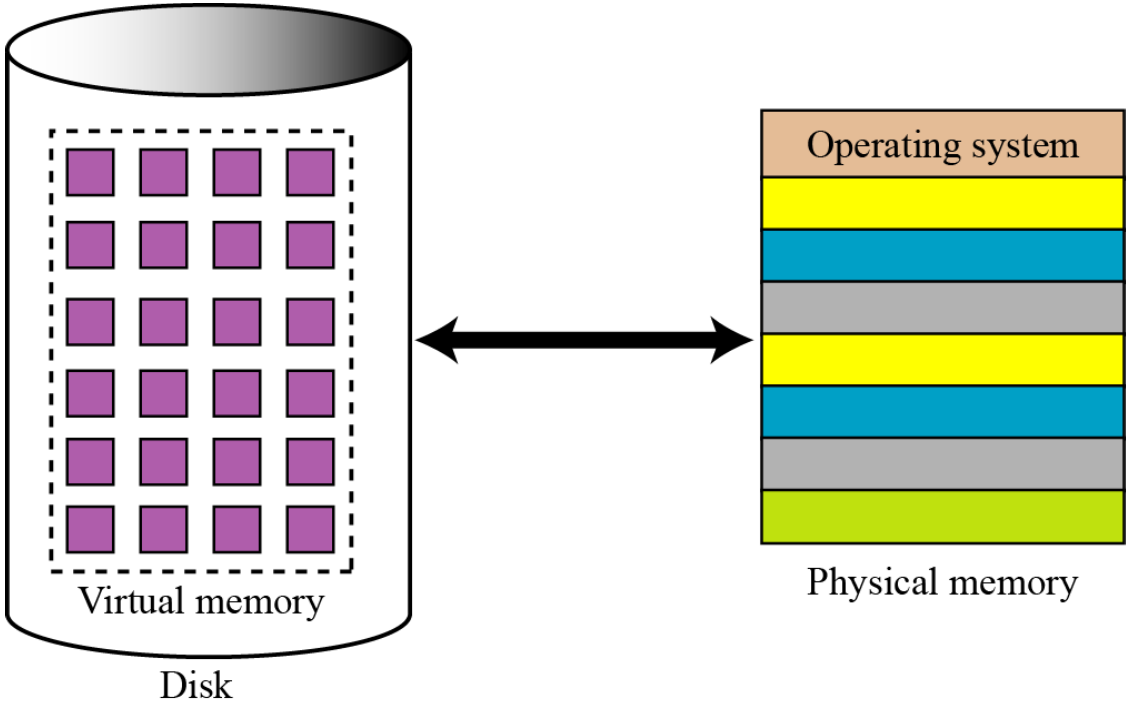
Basic Functions

- Basic functions in kernel
- 1. File manager
  - directories (folders) — organization
  - path —animals\prehistoric\dinos\intro.pdf
  - allows access, checks rights
- 2. Device drivers
  - printer, screen, mouse, etc.
  - communicate with controllers

Basic Functions

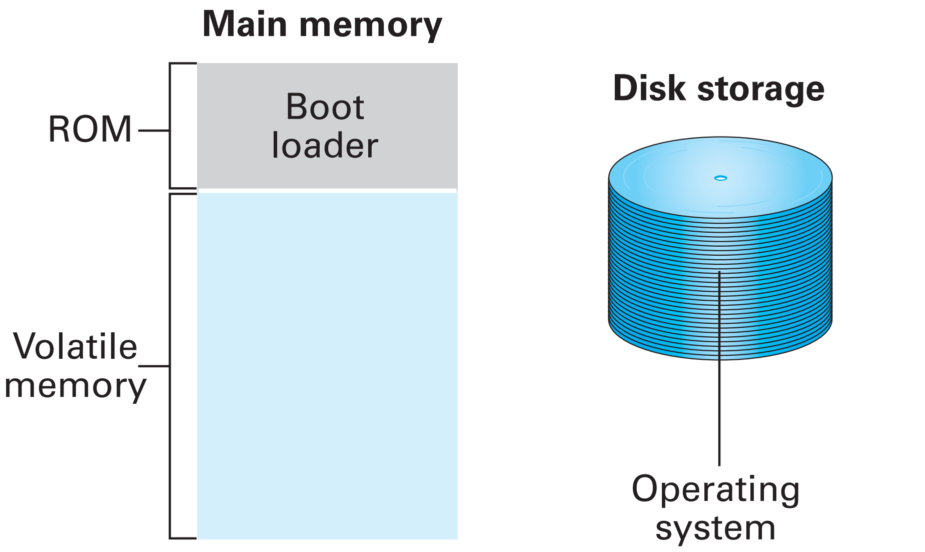
- 3. Memory manager
  - in multiuser or multitask system, much to do
  - virtual memory — if more data than for physical memory
  - store some pages in secondary storage

— if used often, leave there — paging is slow

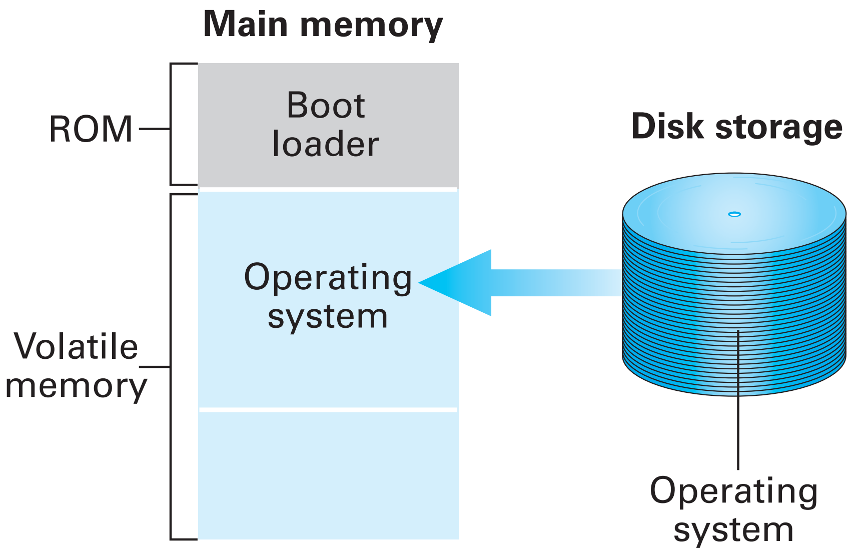


Basic Functions

- 4. Scheduler and dispatcher
  — giving time slices to different tasks or users
- 5. Bootstrap (booting)
  - bootstrap program (boot loader) in ROM (non-volatile)
  - loads rest of OS from disk into main memory (volatile)



Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.



Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

Ch.3: OS

- Coordinating the Machine’s Activities
- Handling Competition Among Processes
- Security

The Concept of a Process

- program — instructions

process — execution of program

— 2 users use same program = 2 processes

- The current status of the activity is called the **process state.** This state includes:
    - value of program counter
    - values in other registers
    - values in memory

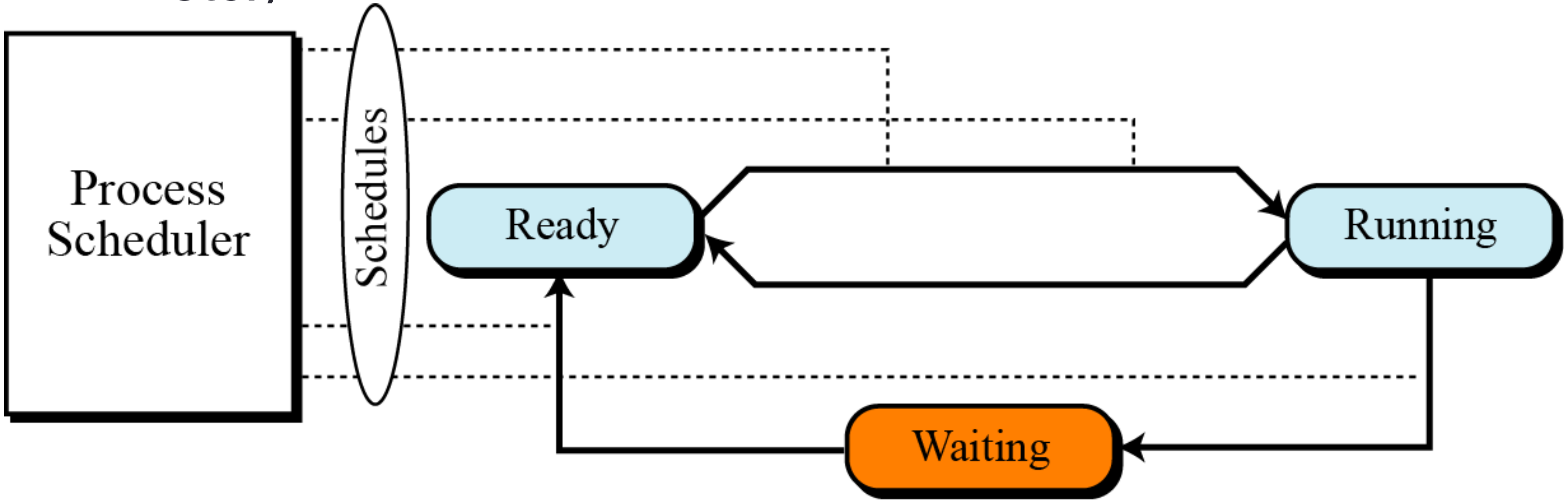
Scheduler

- Multitasking computers are running many processes
- 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
    - Running
    - waiting — for external event (completion of read from disk, etc.)



Dispatcher

- gets scheduled processes executed by multitasking
- 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 Processes

- Allocating access to resources
  - sections of code — device driver for printer
  - memory addresses

1 process at a time

Competition among Processes

flag $\boxed{?}$

0 –	clear	OK
1 –	set	in use

Competition among Processes

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 Processes

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 Processes

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 Processes

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

Security

- Self reading

Assignment

- Questions:
- 2, 4, 7, 8, 13, 19, 39, 49