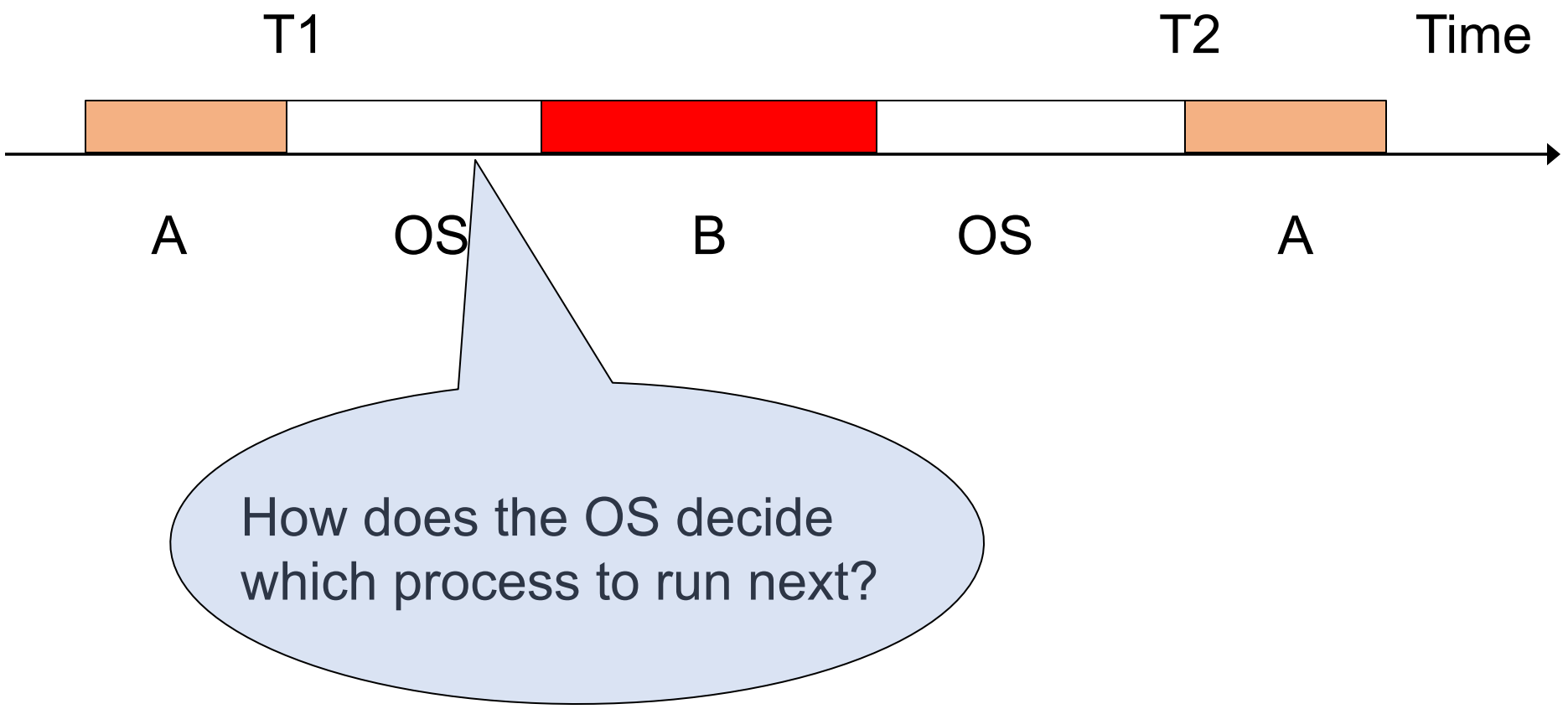
Operating Systems Principles

CPU Management (Processes)

How Does the OS Help Multiple Processes Share the Same CPU?

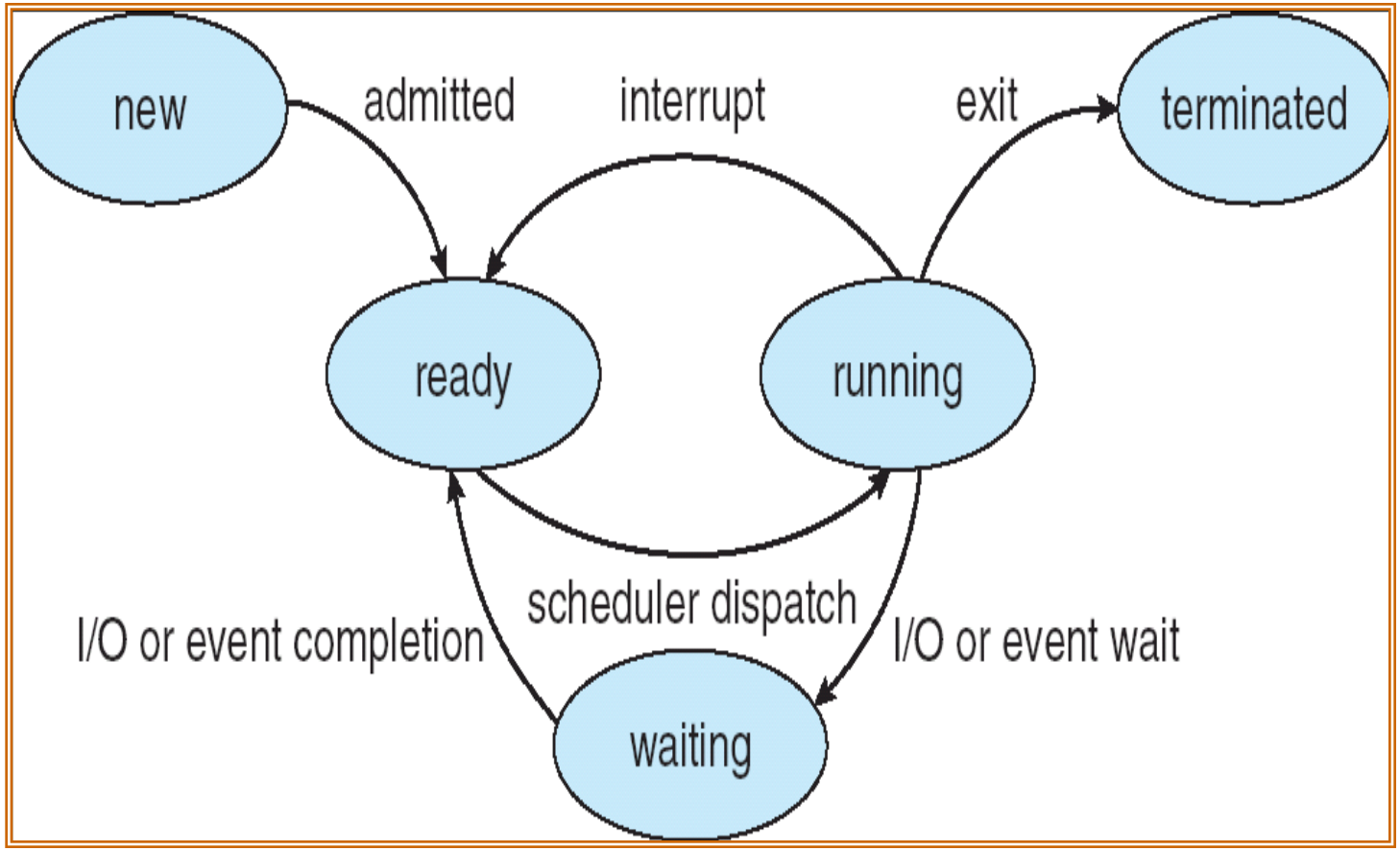
Overheads of Process Switch

- Direct
  - "Wasted" instructions of outgoing process
  - The time spent switching context
- Indirect
  - Cache pollution
  - TLB flush



- Done by the CPU scheduler within the OS
- Today's topic

Not All Processes May be Willing/Able to "Run"



Process Control Block (PCB)

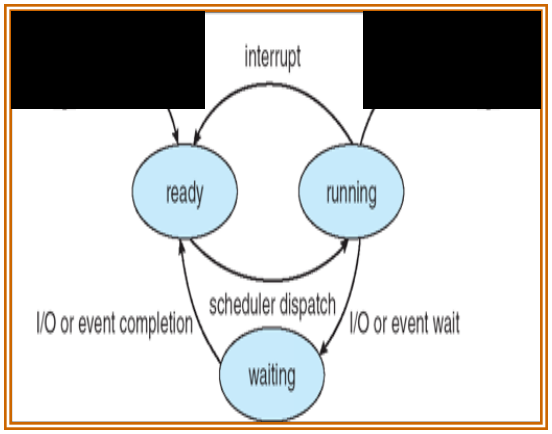
- A data structure in the operating system's address space where it maintains a variety of information about the process
  - Example of information we learnt in last class: Values held in various registers when the process was context switched out
  - Another important piece of information held by the OS in a PCB: what the scheduling-related "state" of a process is (new, ready, running, waiting, terminated)

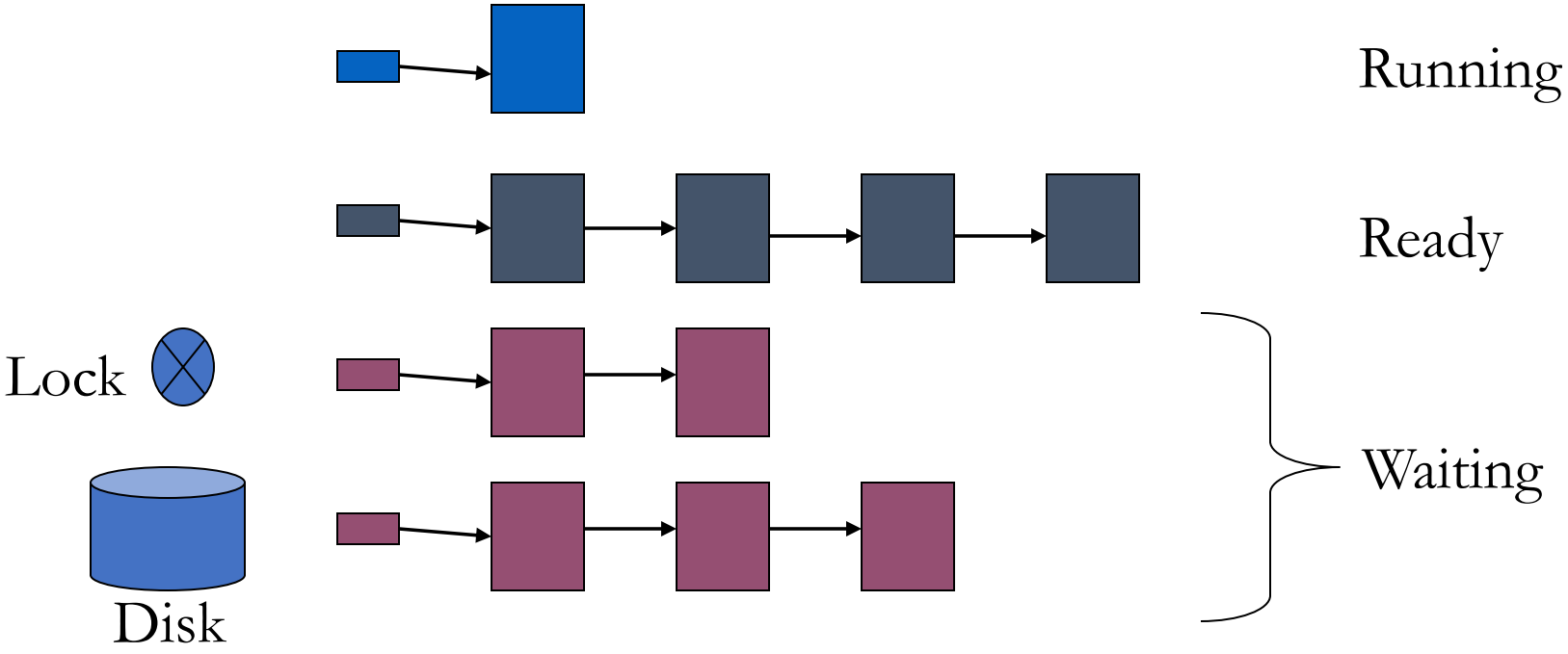
Some data structures that group subsets of PCBs together

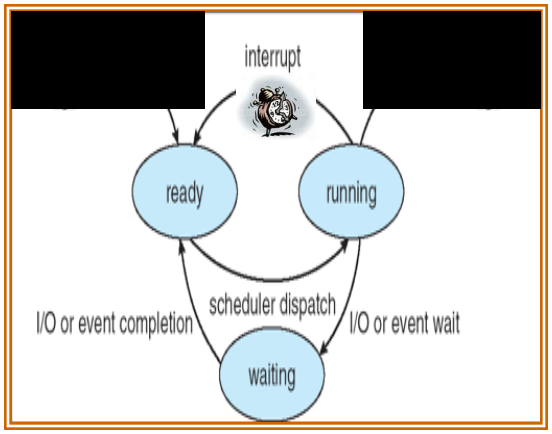
- Queue of ready (runnable) processes
  - Scheduler picks from these
- Queues of waiting (blocked) processes
  - Separate queues for difference devices or software resources (e.g., locks)

Some food-for-thought related to PCBs

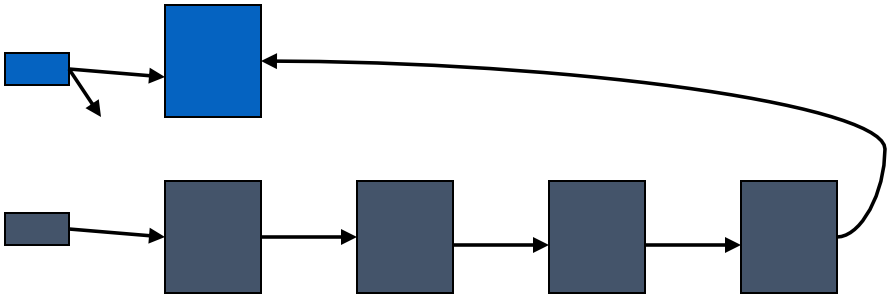
- Sometimes PCB of exited process kept in memory
  - Pop quiz: Why?
- All PCBs in a part of RAM reserved by the kernel for itself and inaccessible to processes
  - Why?
  - This part of RAM initialized during boot-up


interrupt
ready
running
scheduler dispatch
I/O or event completion
I/O or event wait
waiting


Running
Ready
Lock
Waiting
Disk



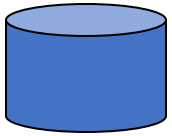
Timer interrupt



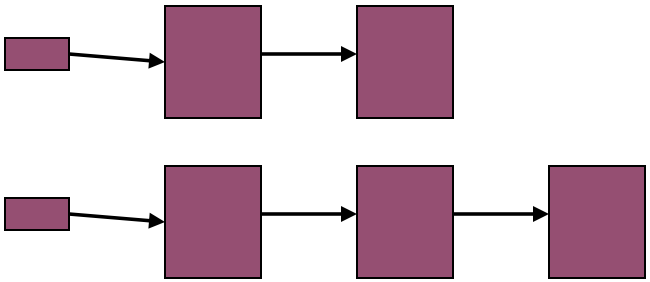
Running

Ready

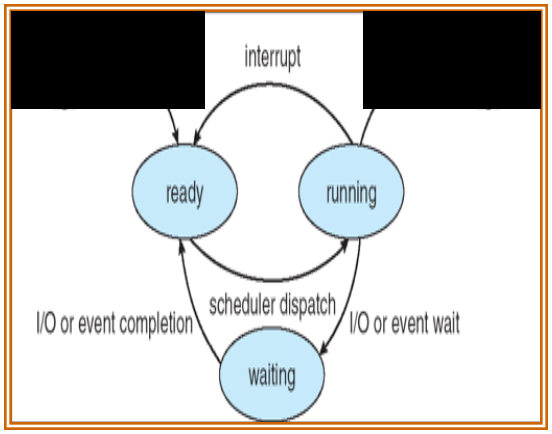
Lock

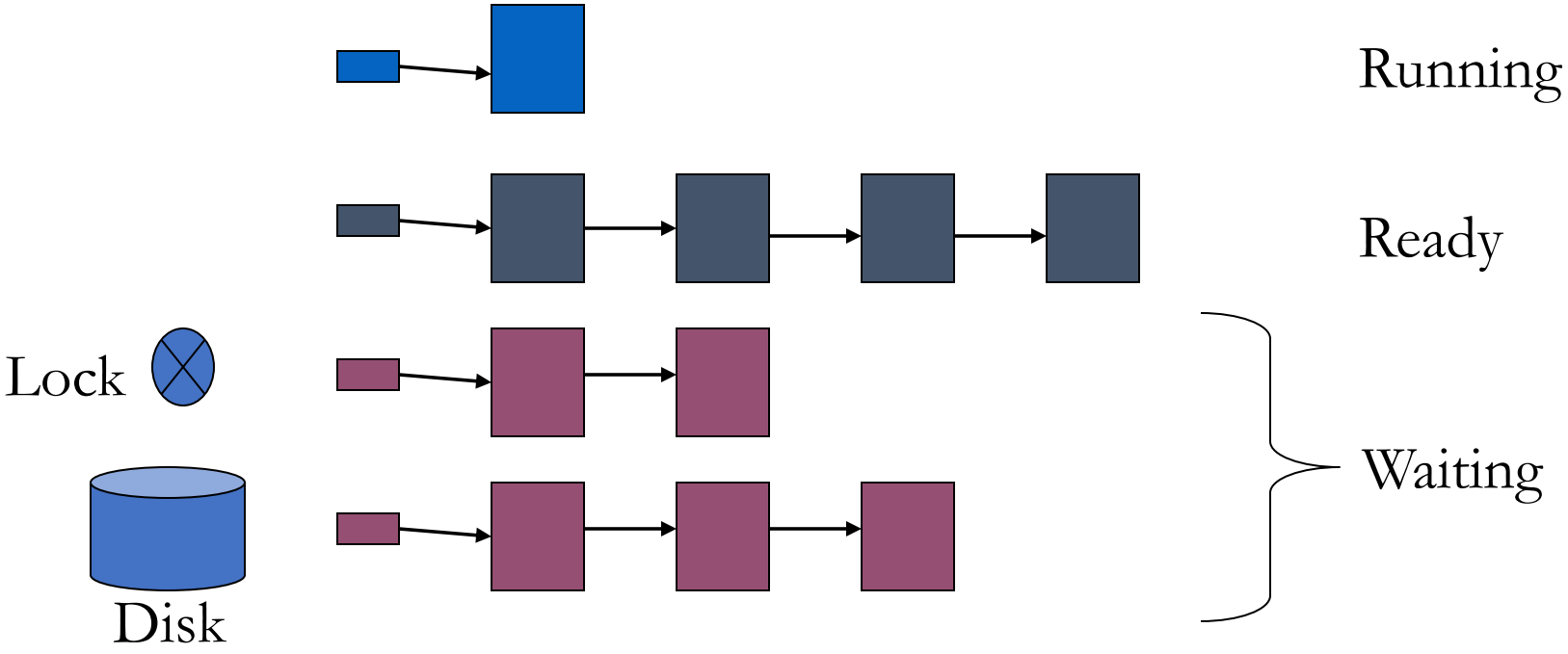


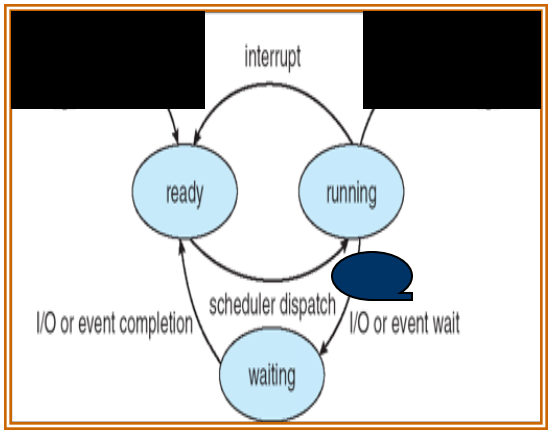
Disk



Waiting


interrupt
ready
running
scheduler dispatch
I/O or event completion
I/O or event wait
waiting


Running
Ready
Lock
Waiting
Disk



I/O system call

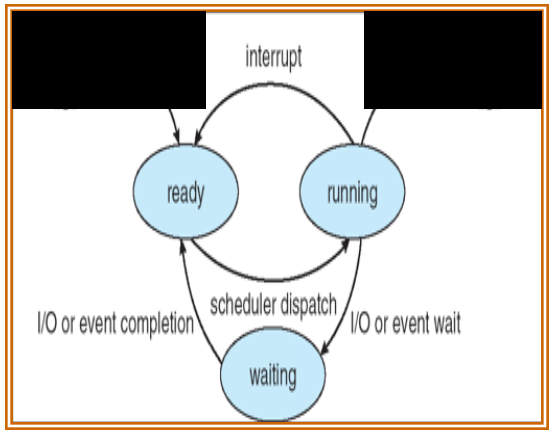
Running

Ready

Lock

Waiting

Disk



Lets pick the second process in the ready queue

OS (scheduler)

Lock

Disk

Running

Ready

Waiting