# Operating System

## Lecture 5: Process Concept & Process scheduling

Manoj Kumar Jain

M.L. Sukhadia University Udaipur

# Outline

- Process Concept
- Process Scheduling

# Process Concept

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms *job* and *process* almost interchangeably.

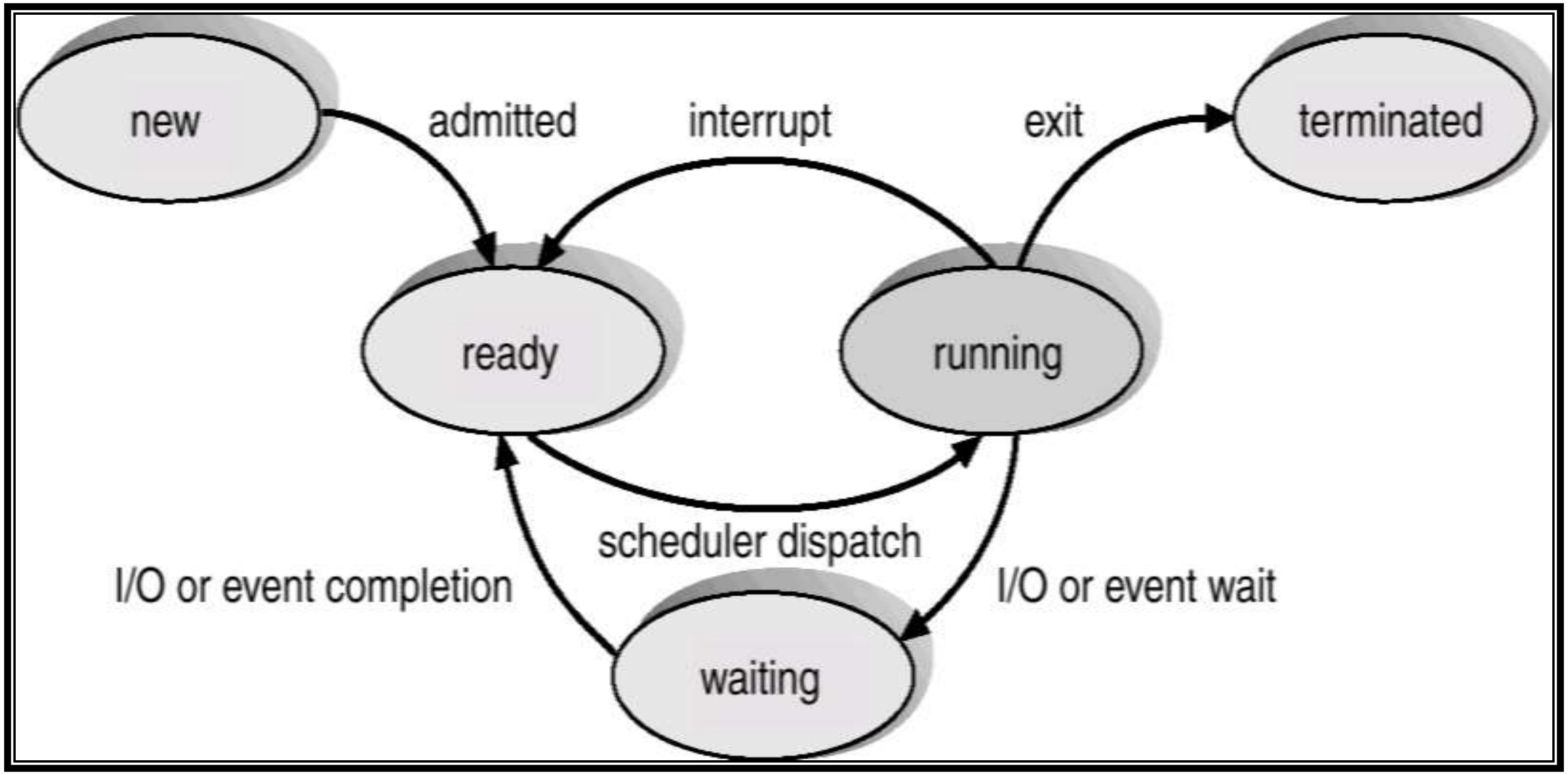
# Process Concept (Cont.)

- Process – a program in execution; process execution must progress in sequential fashion.
- A process includes:
  - Program code
  - Program counter
  - Processor register contents
  - Process stack (to store temporary data like method parameters and return address and local variables)
  - Data section (to store global variables)

# Process State

- As a process executes, it changes *state. The state of a process is defined in part by the current activity of that process.*
  - **new**: The process is being created.
  - **running**: Instructions are being executed.
  - **waiting**: The process is waiting for some event to occur.
  - **ready**: The process is waiting to be assigned to a processor.
  - **terminated**: The process has finished execution.

# Diagram of Process State



# Process Control Block (PCB)

Information associated with each process.

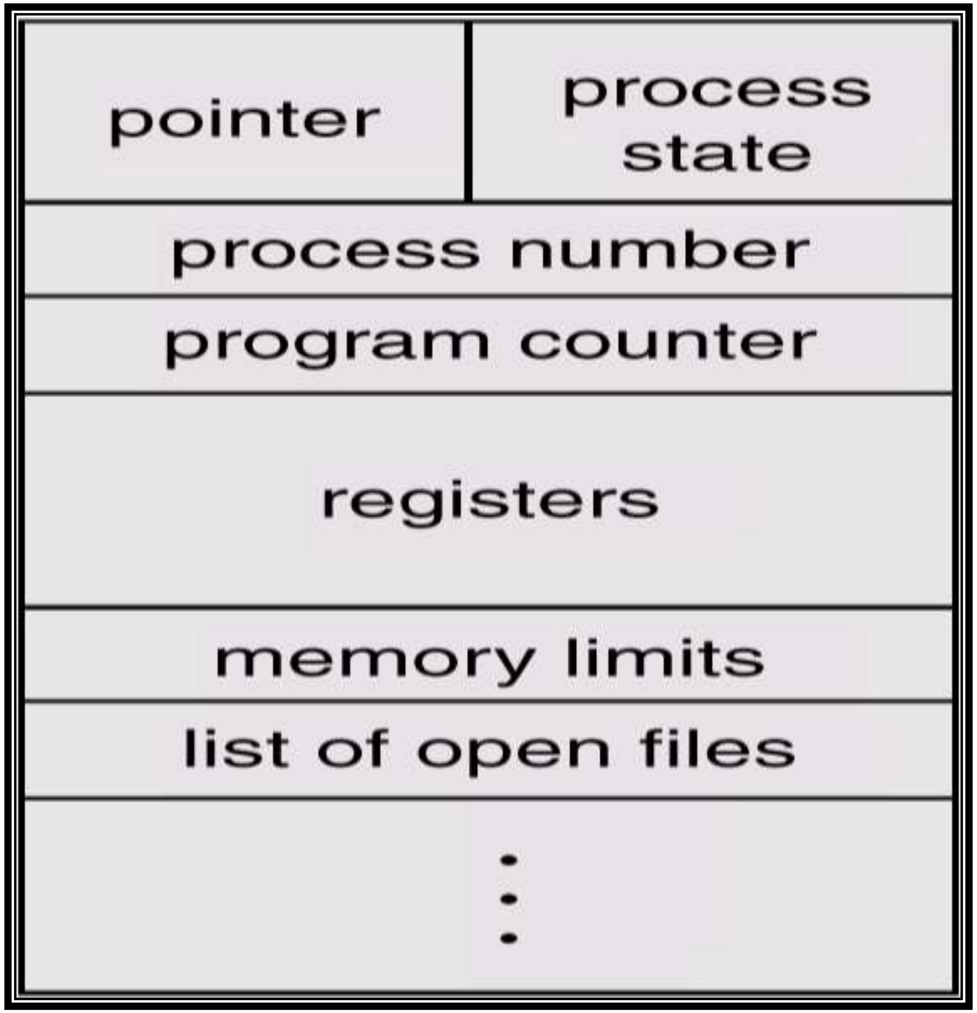
- Process state (new, ready, running, waiting, halted)
- Program counter (next instruction to be executed for this process)
- CPU registers (accumulators, index registers, stack pointers and GPR’s)
- CPU scheduling information (process priority, pointers to scheduling queues, any other scheduling parameters)

# Process Control Block (PCB) (Cont.)

Information associated with each process.

- Memory-management information (value of base and limit registers, page or segment tables)
- Accounting information (amount of CPU and real time used, time limits, account numbers, job or process number ….)
- I/O status information (list of I/O devices allocated to this process, a list of open files ….)

# Process Control Block (PCB)



pointer

process state

process number

program counter

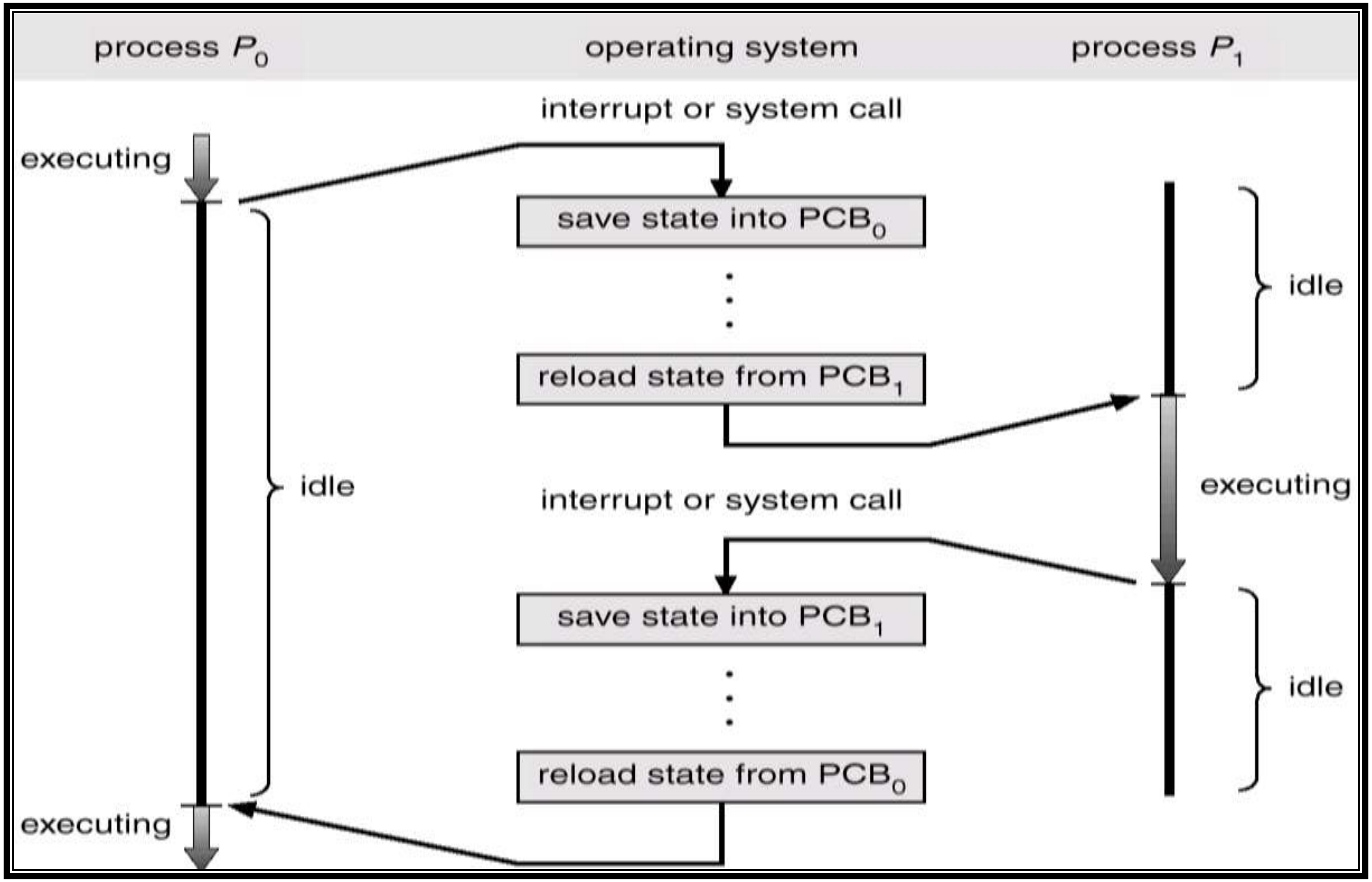
registers

memory limits

list of open files

•
•
•

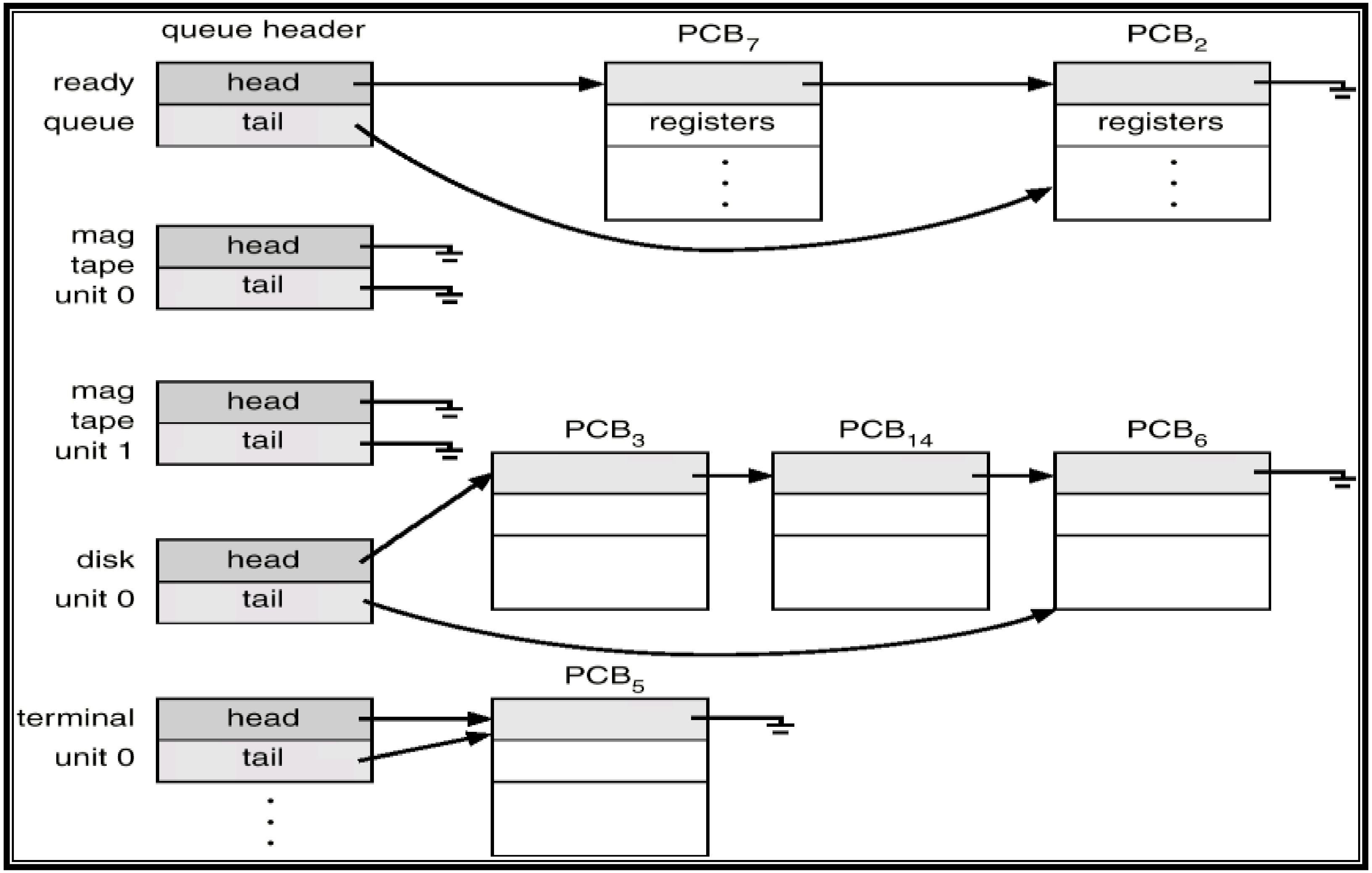
# CPU Switch From Process to Process



# Process Scheduling Queues

- Job queue – set of all processes in the system.
- Ready queue – set of all processes residing in main memory, ready and waiting to execute.
- Device queues – set of processes waiting for an I/O device.
- Process migration between the various queues.

# Ready Queue And Various I/O Device Queues



# Process Scheduling

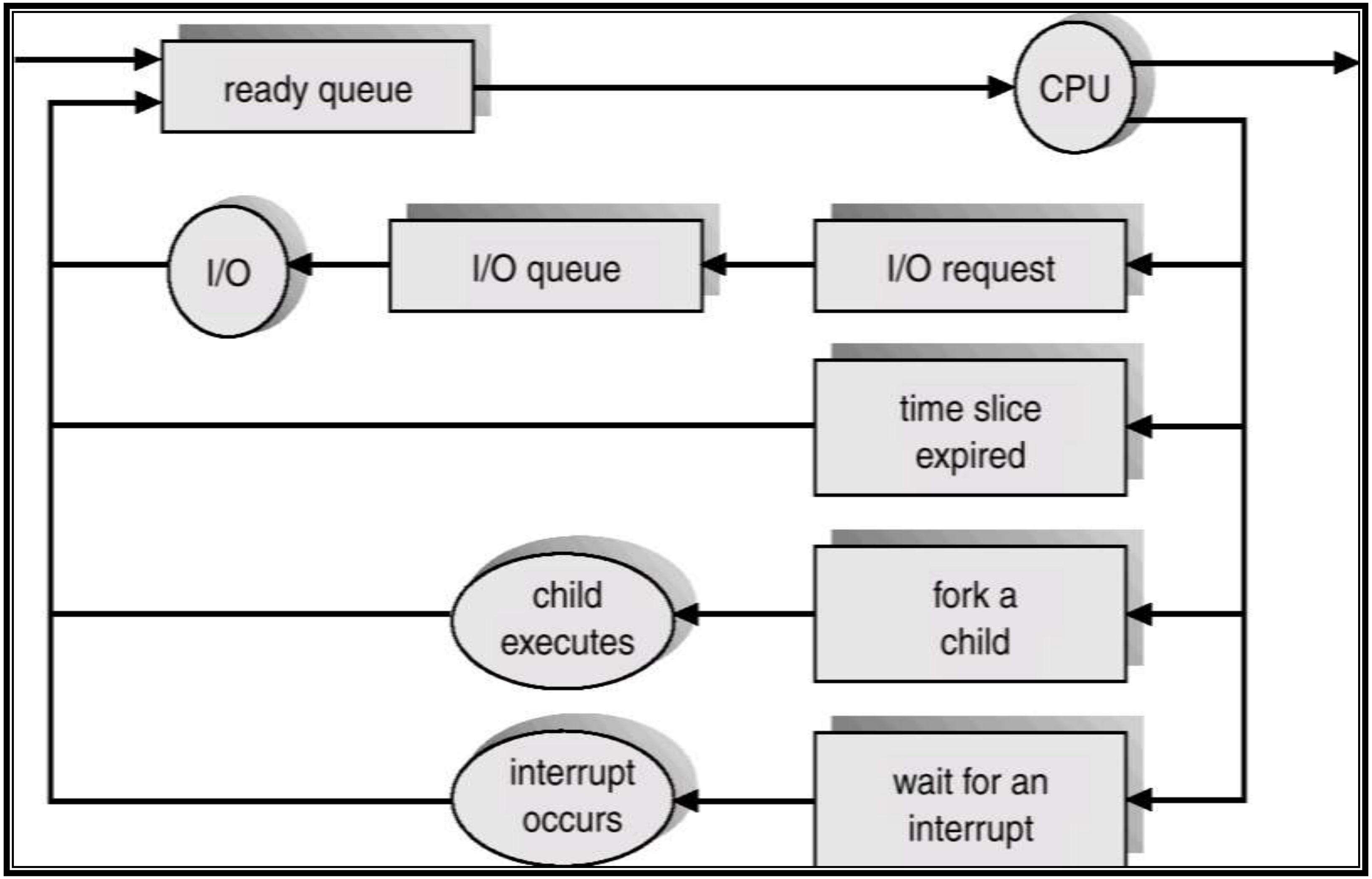
A new process is initially put in the ready queue. It waits in this queue until it is selected for execution (or dispatched). Once the process is assigned to the CPU and is executing, one pf several events could occur.

- The process could issues an I/O request, and then be placed in an I/O queue.
- The process could create a new sub process and wait for its termination.
- The process could be removed forcibly from the CPU, as a result of an interrupt, and be put back in the ready queue.

# Process Scheduling (Cont.)

In first two cases, the process eventually switches from the waiting state to the ready state, and then put back in the ready queue. A process continues this cycle until it terminates, at which time it is removed from all queues and has its PCB and other resources de-allocated.

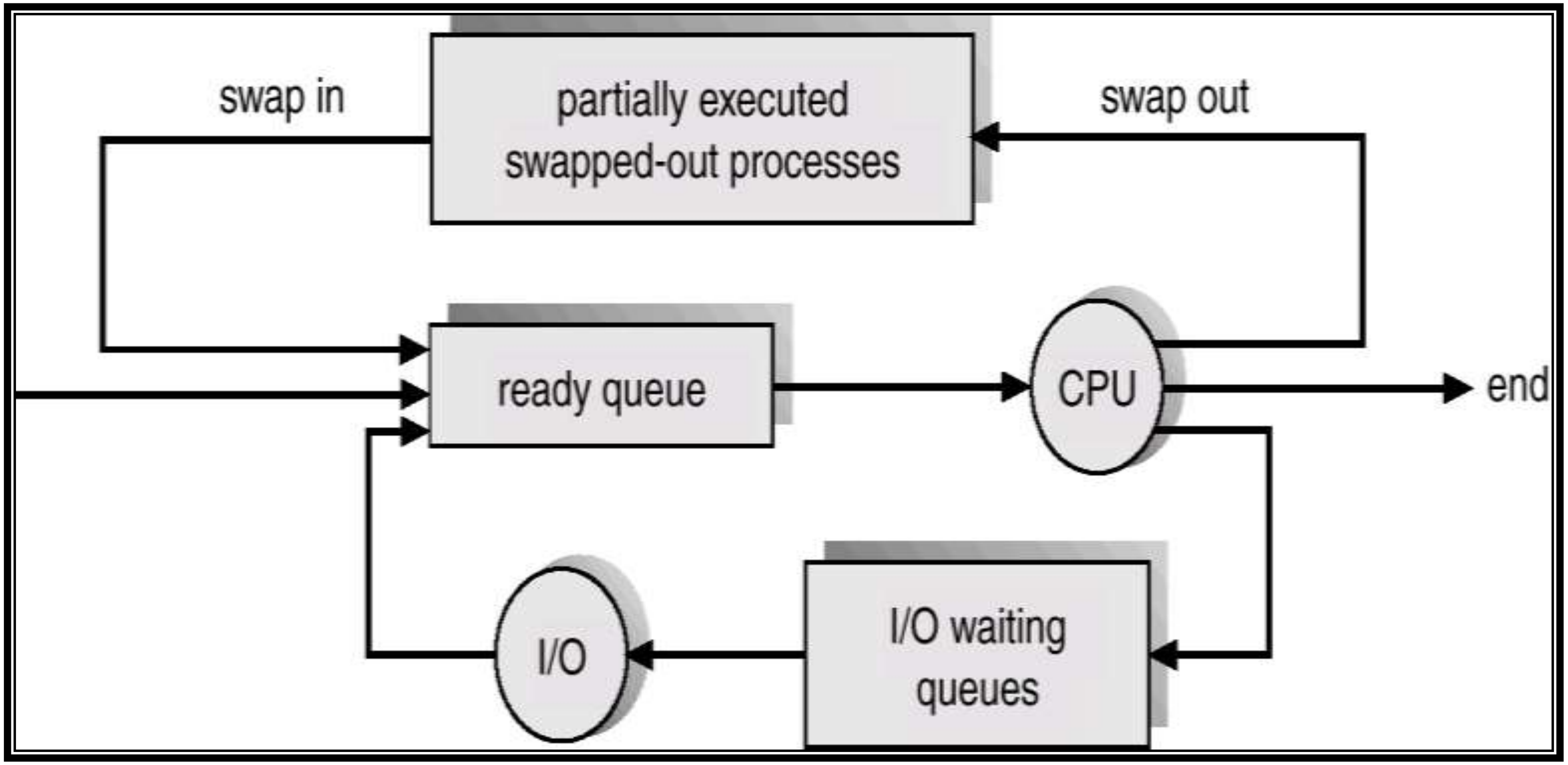
# Representation of Process Scheduling



# Schedulers

- Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue.
- Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU.

# Addition of Medium Term Scheduling



# Schedulers (Cont.)

- Short-term scheduler is invoked very frequently (milliseconds) ⇒ (must be fast).
- Long-term scheduler is invoked very infrequently (seconds, minutes) ⇒ (may be slow).
- The long-term scheduler controls the *degree of multiprogramming.*

# Schedulers (Cont.)

- Processes can be described as either:
  - I/O-*bound process* – spends more time doing I/O than computations, many short CPU bursts.
  - *CPU-bound process* – spends more time doing computations; few very long CPU bursts.

# Context Switch

- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process.
- Context-switch time is overhead; the system does no useful work while switching.
- Time dependent on hardware support.

# Thanks