

# Processes (2)

---

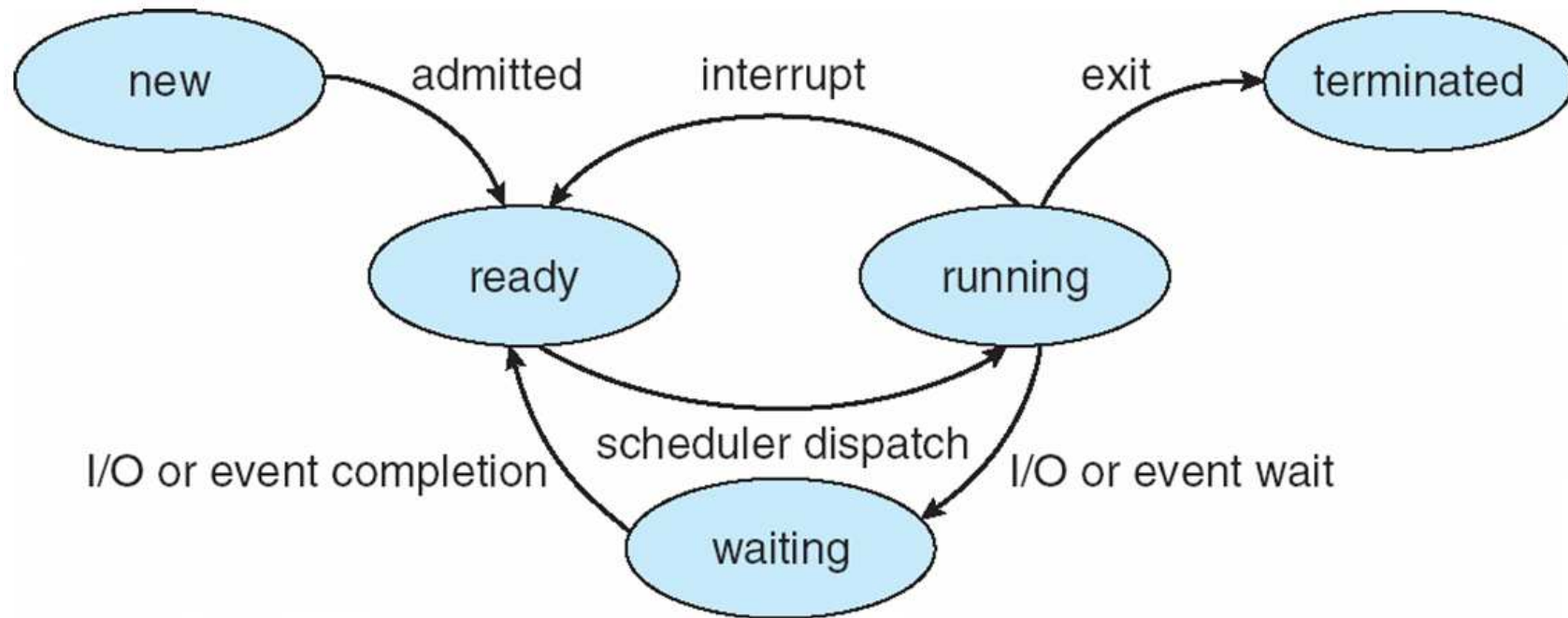
Dr. Jun Zheng  
CSE325 Principles of Operating  
Systems  
9/4/2019



# Process State

- ❑ As a process executes, it changes **state**
  - ❑ **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 Execution State

## Example:

```
void main(){  
    printf("Hello World");  
}
```

## State Sequence:

new

ready

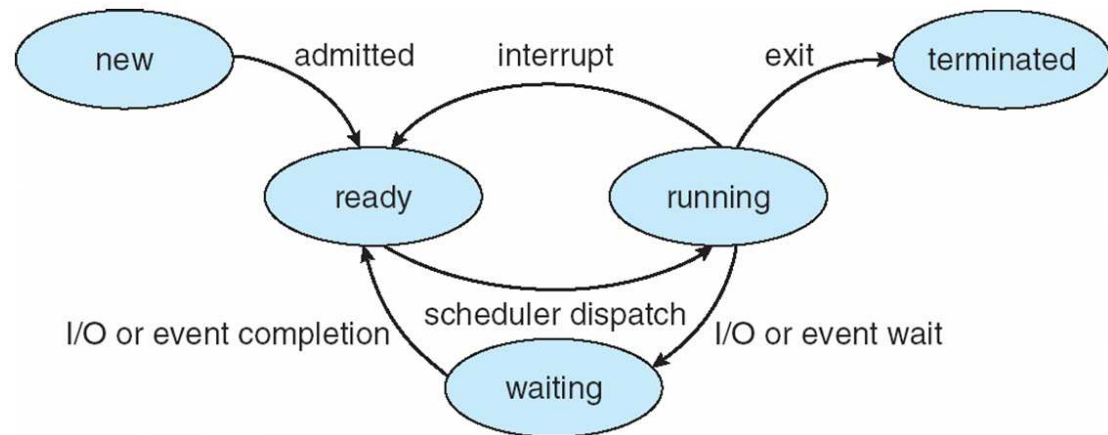
running

waiting (I/O)

ready

running

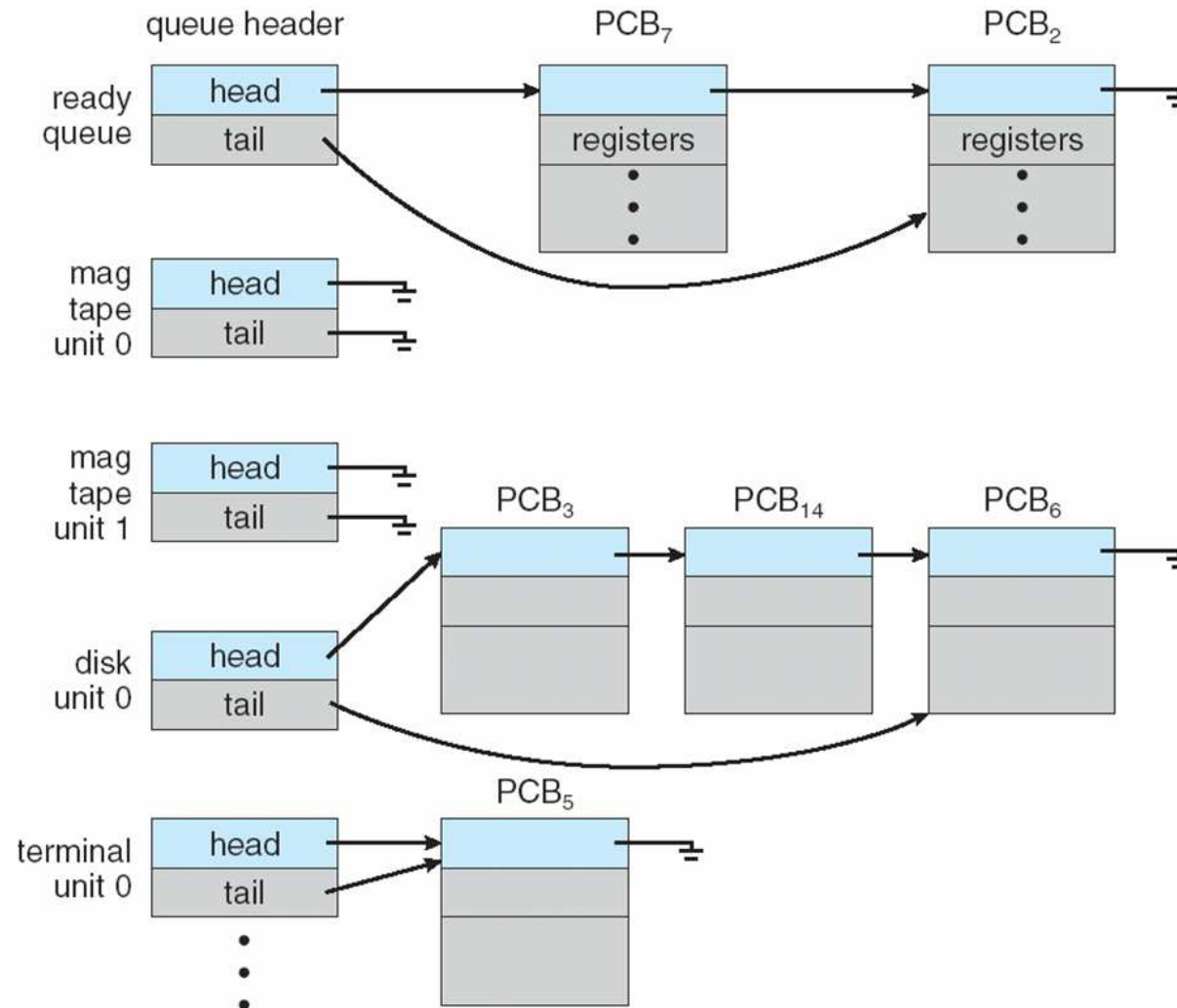
terminated



# Process Scheduling

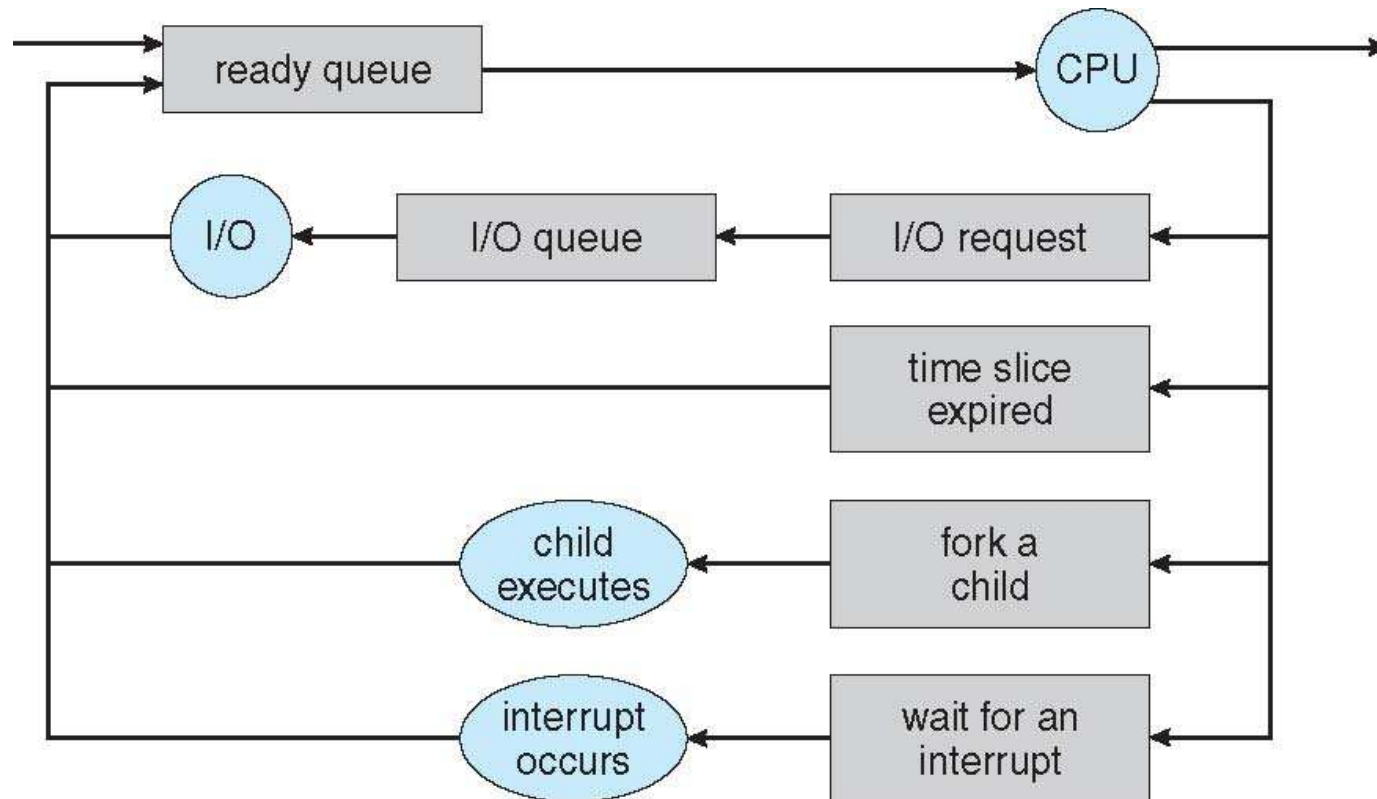
- ❑ Maximize CPU use, quickly switch processes onto CPU for time sharing
- ❑ **Process scheduler** selects among available processes for next execution on CPU
- ❑ Maintains **scheduling queues** of processes
  - ❑ **Job queue** – set of all processes in the system
  - ❑ **Ready queue** – set of all processes residing in main memory, ready and waiting to execute
  - ❑ **Device queues** – set of processes waiting for an I/O device
- ❑ Processes migrate among the various queues

# Ready Queue And Various I/O Device Queues



# Representation of Process Scheduling

- **Queueing diagram** represents queues, resources, flows



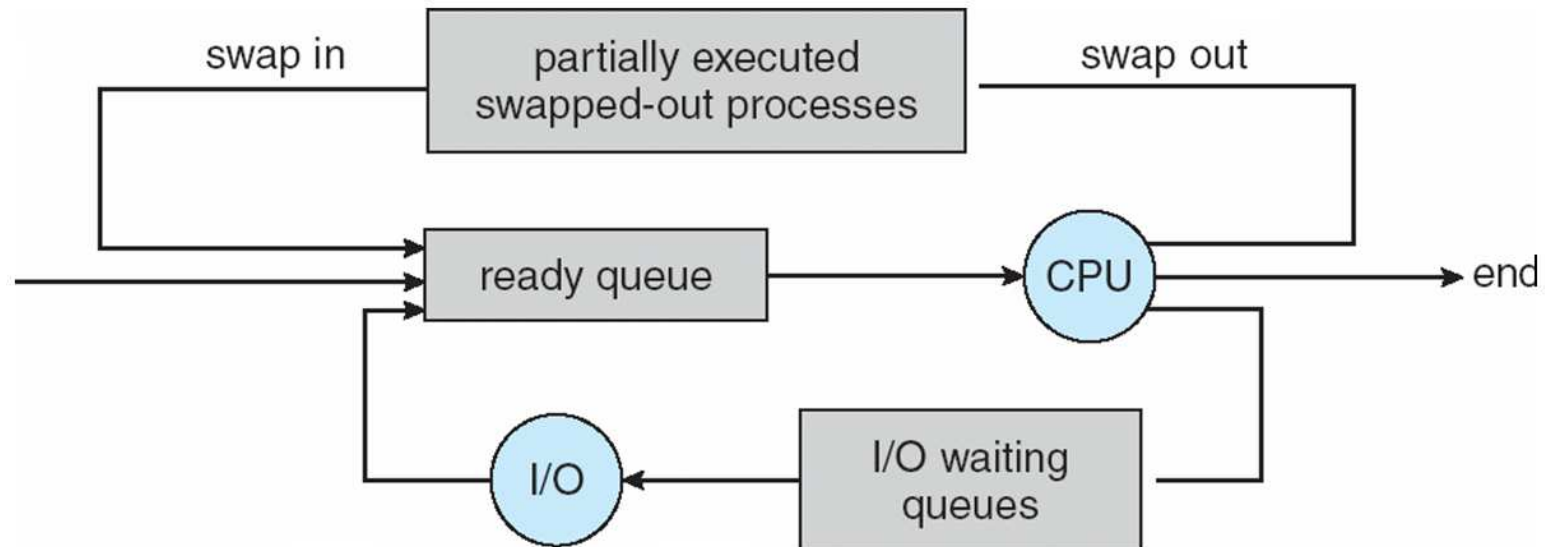
# Schedulers

- ❑ **Short-term scheduler** (or **CPU scheduler**) – selects which process should be executed next and allocates CPU
  - ❑ Sometimes the only scheduler in a system
  - ❑ Short-term scheduler is invoked frequently (milliseconds)  $\Rightarrow$  (must be fast)
- ❑ **Long-term scheduler** (or **job scheduler**) – selects which processes should be brought into the ready queue
  - ❑ Long-term scheduler is invoked infrequently (seconds, minutes)  $\Rightarrow$  (may be slow)
  - ❑ The long-term scheduler controls the **degree of multiprogramming**
- ❑ 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
- ❑ Long-term scheduler strives for good ***process mix***



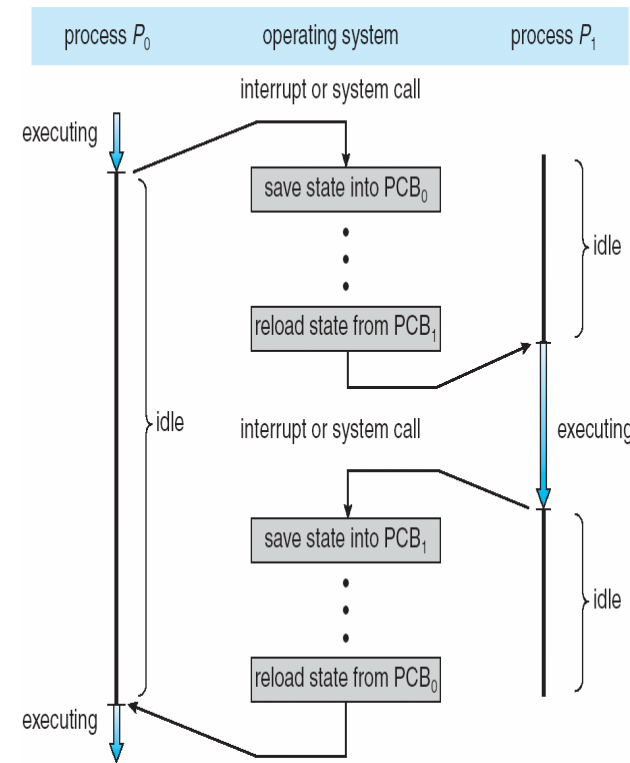
# Addition of Medium Term Scheduling

- **Medium-term scheduler** can be added if degree of multiple programming needs to decrease
  - Remove process from memory, store on disk, bring back in from disk to continue execution: **swapping**



# 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 via a **context switch**
- ❑ **Context** of a process represented in the PCB
- ❑ Context-switch time is overhead; the system does no useful work while switching
  - ❑ The more complex the OS and the PCB → the longer the context switch
- ❑ Time dependent on hardware support
  - ❑ Some hardware provides multiple sets of registers per CPU → multiple contexts loaded at once



# Operations on Processes

- ❑ System must provide mechanisms for:
  - ❑ process creation
  - ❑ process termination

# Process Creation

- ❑ **Parent** process create **children** processes, which, in turn create other processes, forming a **tree** of processes
- ❑ Generally, process identified and managed via a **process identifier (pid)**
- ❑ Resource sharing options
  - ❑ Parent and children share all resources
  - ❑ Children share subset of parent's resources
  - ❑ Parent and child share no resources
- ❑ Execution options
  - ❑ Parent and children execute concurrently
  - ❑ Parent waits until children terminate