Process Concept

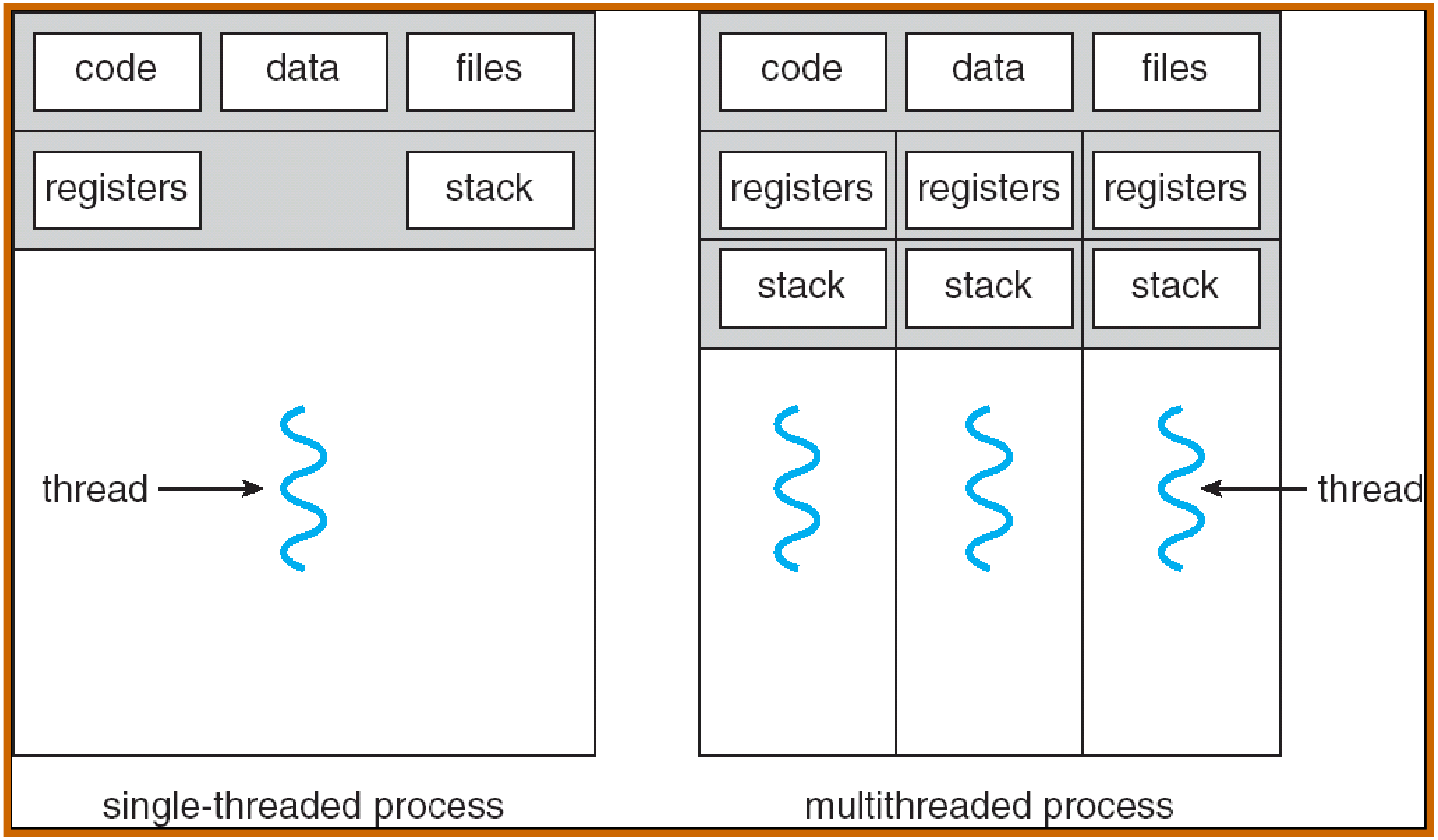
Program, Process, Processor

- A progam is a set of instructions. A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- Processor is the device which executes programs.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources

Threads

- A thread is predefined instance of a process. A thread performs a specific function. A process may consist a set of threads
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Single and Multithreaded Processes



Process Management Activities

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling

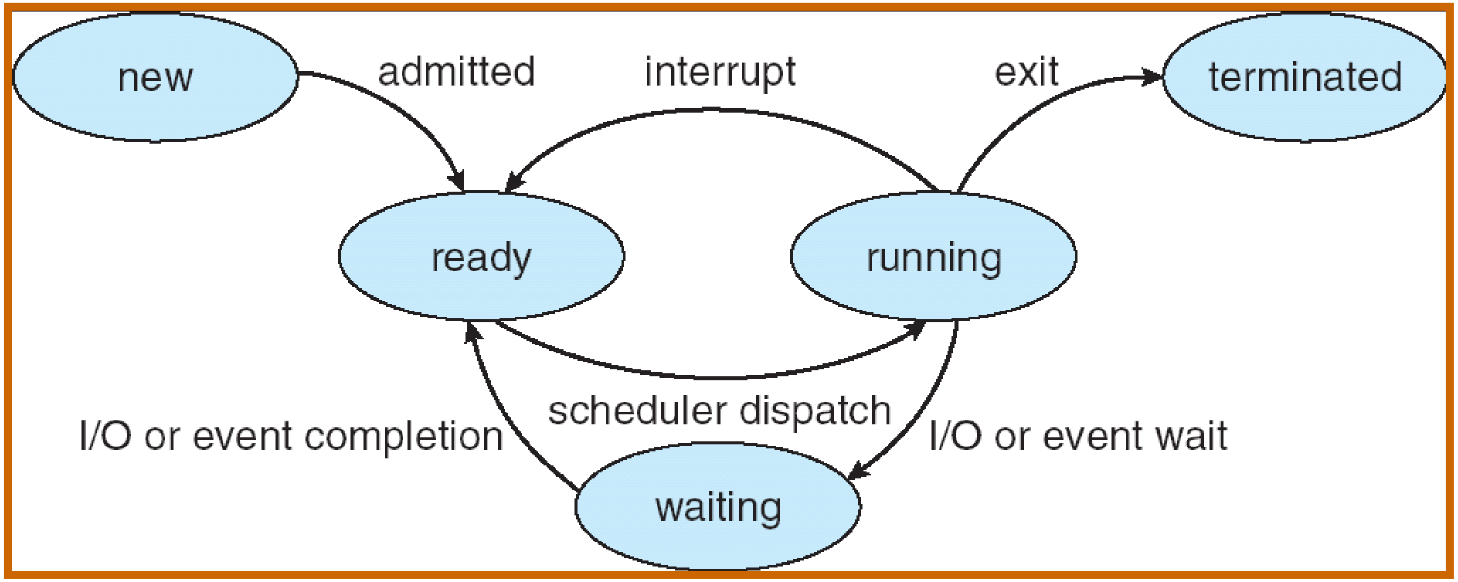
Process Creation

- System Initialization – Boots up
- User related events – click, double click, drag
- Execution of a function call- process creates a child process

Process Termination

- System shut down
- Process termination – completes execution
- Error condition(Voluntary)
- Fatal error(Involuntary
- Killed by another process

Process States



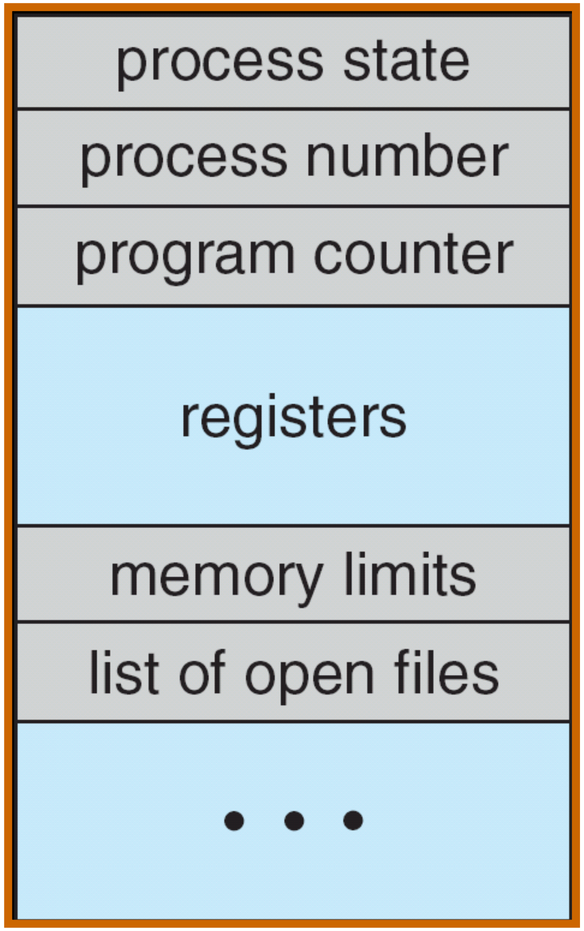
- New – just created, arrived in memory
- Ready – prepared for execution once the CPU becomes available
- Running – Being executed.
- Blocked – has suspended running, requires an I/O device.
- Terminated – Has finished executed, prepared to exit memory.

Process Control Block (PCB)

Stores Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

Process Control Block (PCB)



process state

process number

program counter

registers

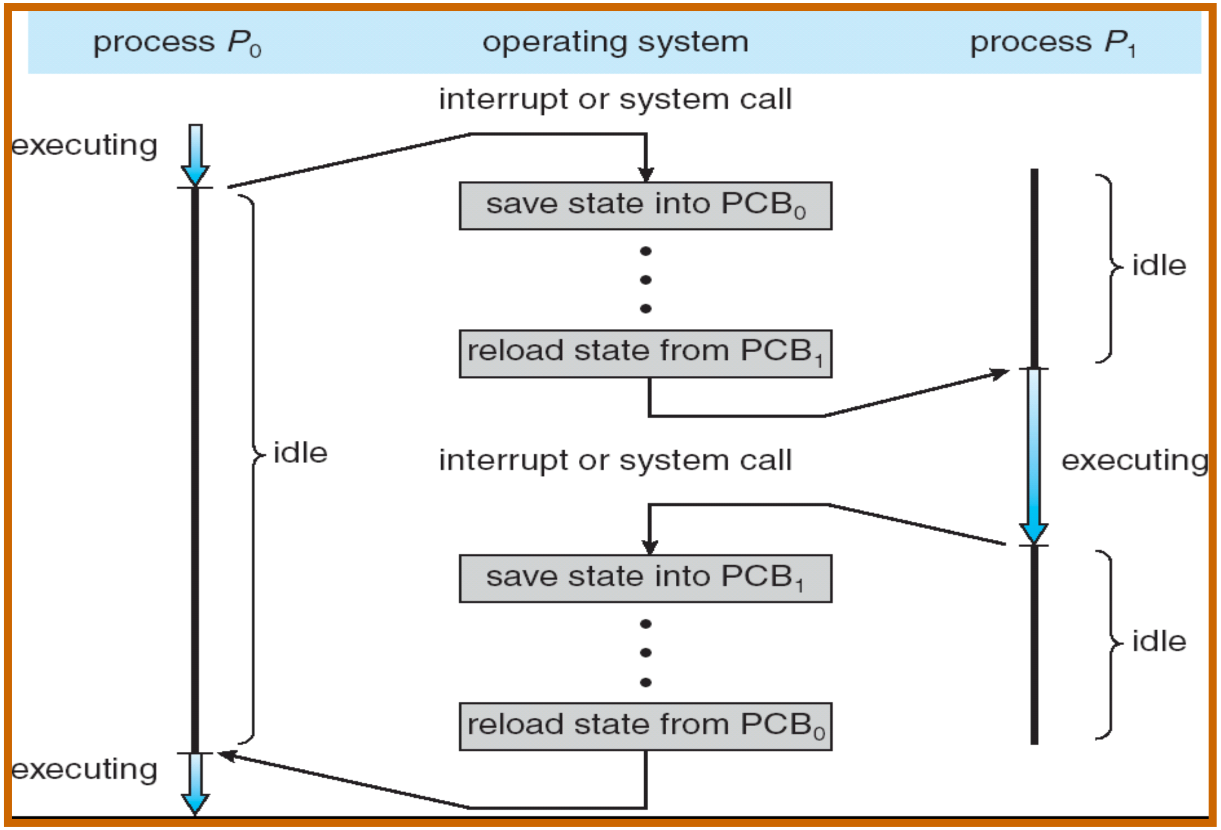
memory limits

list of open files

• • •

Process Interruption

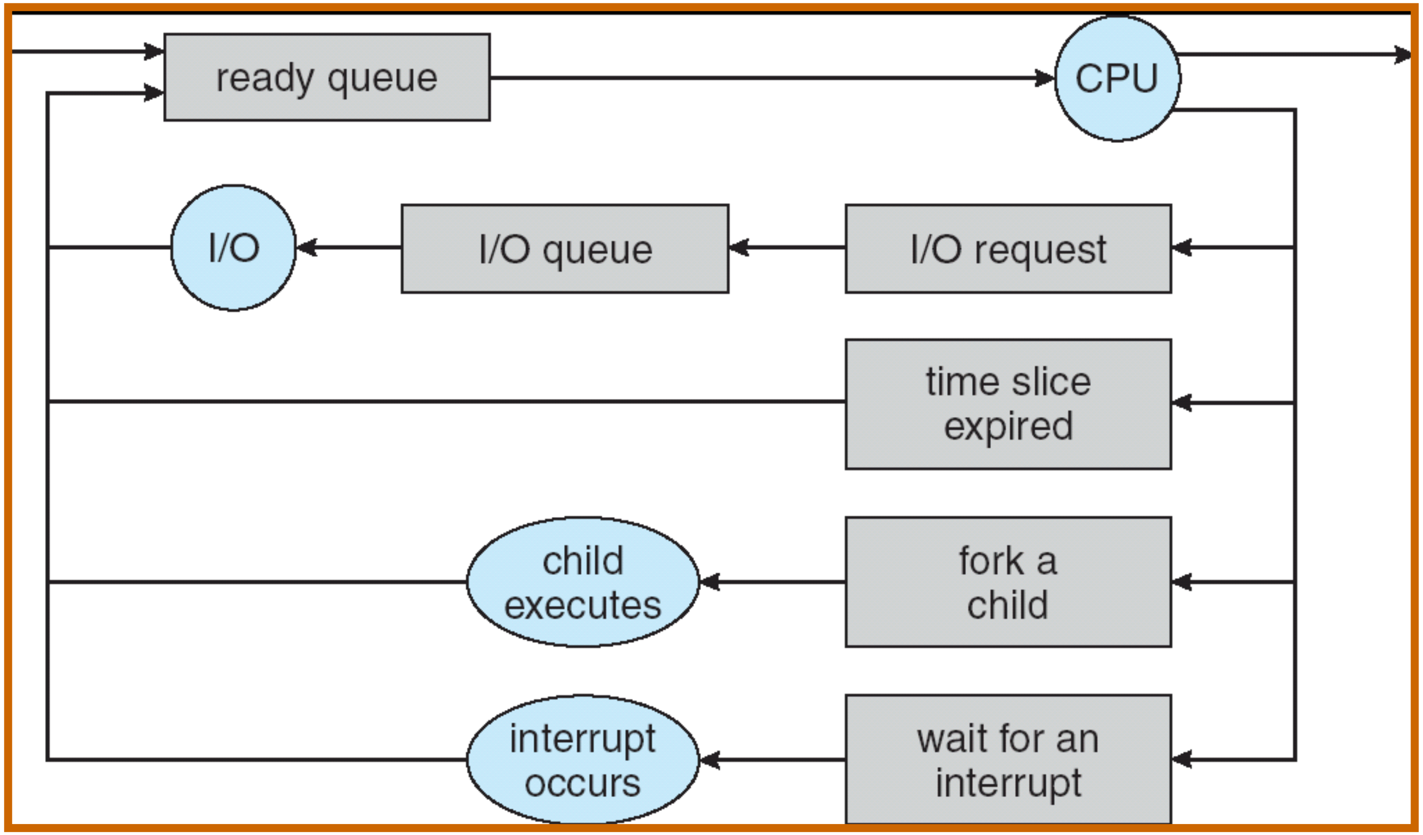
- CPU Switch From Process to Process



Process Scheduling Queues

- **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

Representation of Process Scheduling



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

Process Creation

- Parent process create children processes, which, in turn create other processes, forming a tree of processes
- Resource sharing may be
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- Execution mode may be
  - Parent and children execute concurrently
  - Parent waits until children terminate

Process Termination

- Process executes last statement and asks the operating system to delete it (**exit**)
  - Output data from child to parent (via **wait**)
  - Process’ resources are deallocated by operating system
- Parent may terminate execution of children processes (**abort**) if
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting
    - Some operating system do not allow child to continue if its parent terminates
      - All children terminated - *cascading termination*