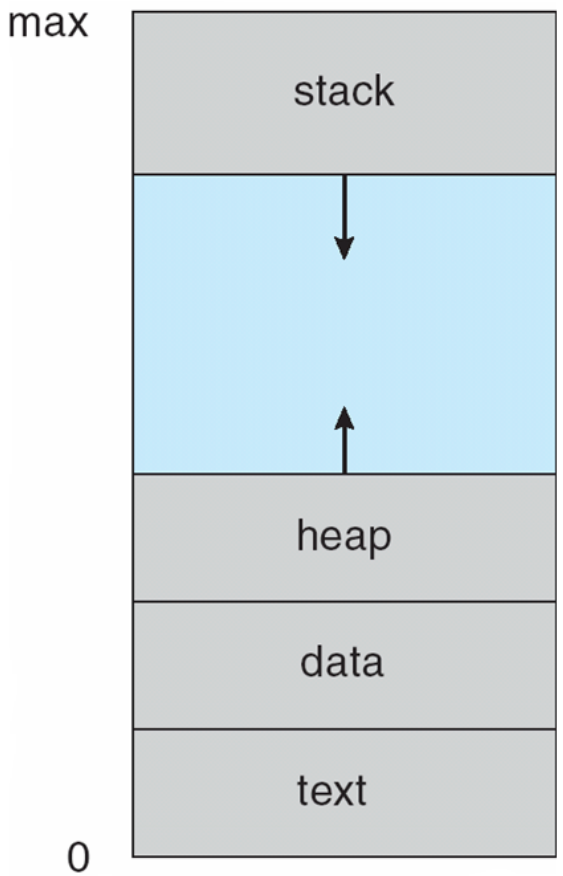
Process Concept

Process Concept

- An operating system executes a variety of programs:
  - Batch system - jobs
  - Time-shared systems - user programs or tasks
- Process - a program in execution; process execution must progress in sequential fashion
- A process includes:
  - program (text) and program counter (PC)
  - stack
  - data section

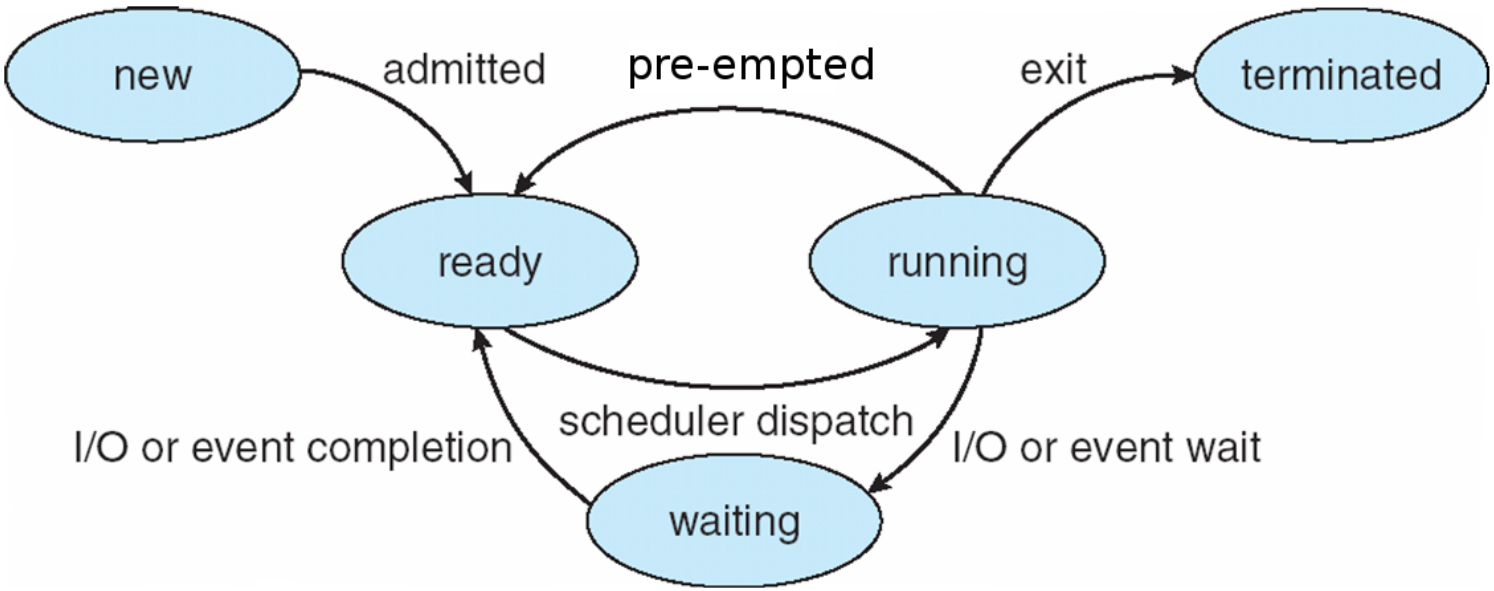
Process in Memory



Process States

- 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

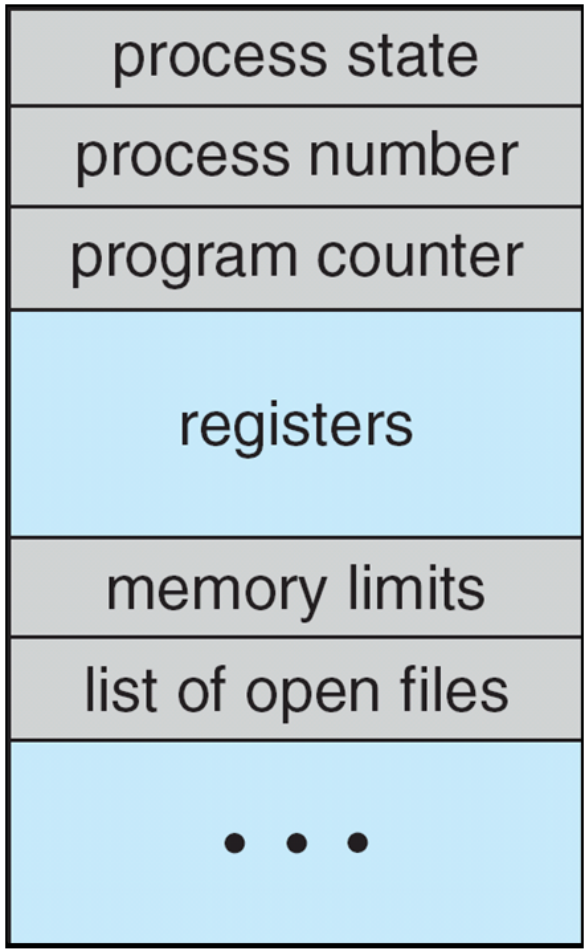
Process States



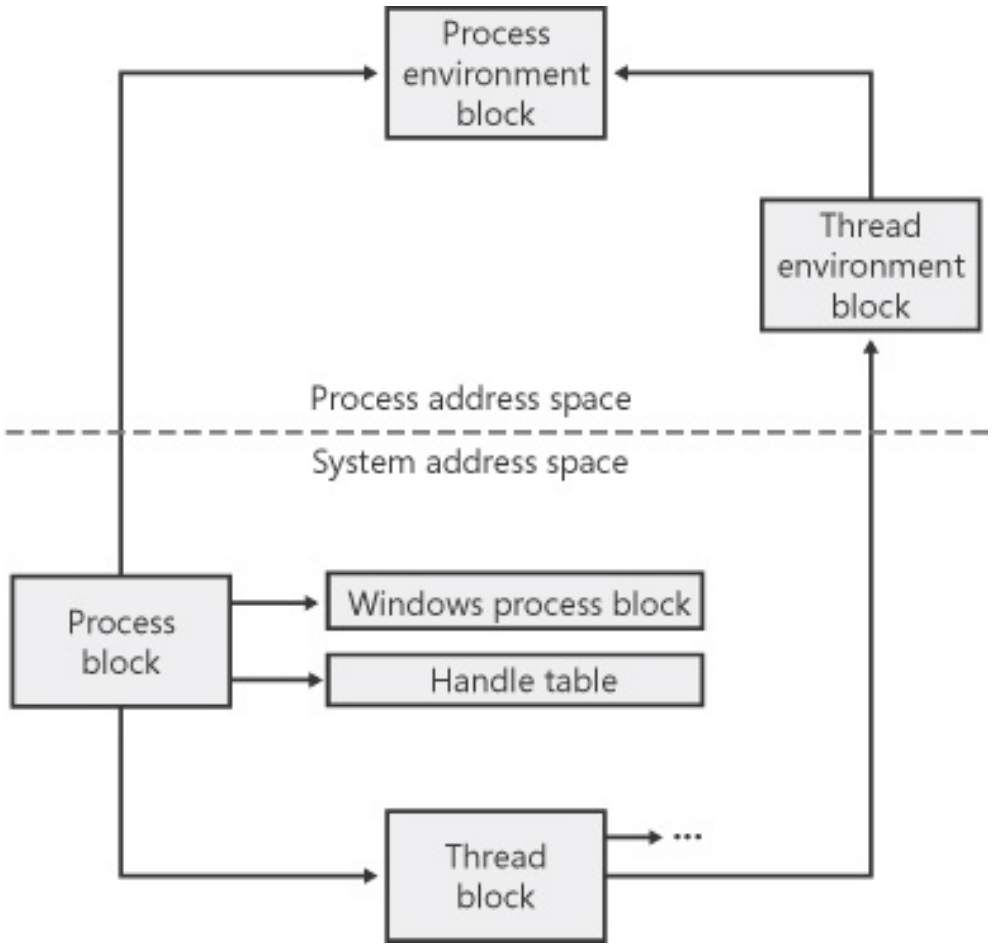
Process Control Block

- Information associated with each process, which is stored as various fields within a kernel data structure:
  - Process state
  - Program counter
  - CPU registers
  - CPU scheduling information
  - Memory-management information
  - Accounting information
  - I/O status information

Process Control Block

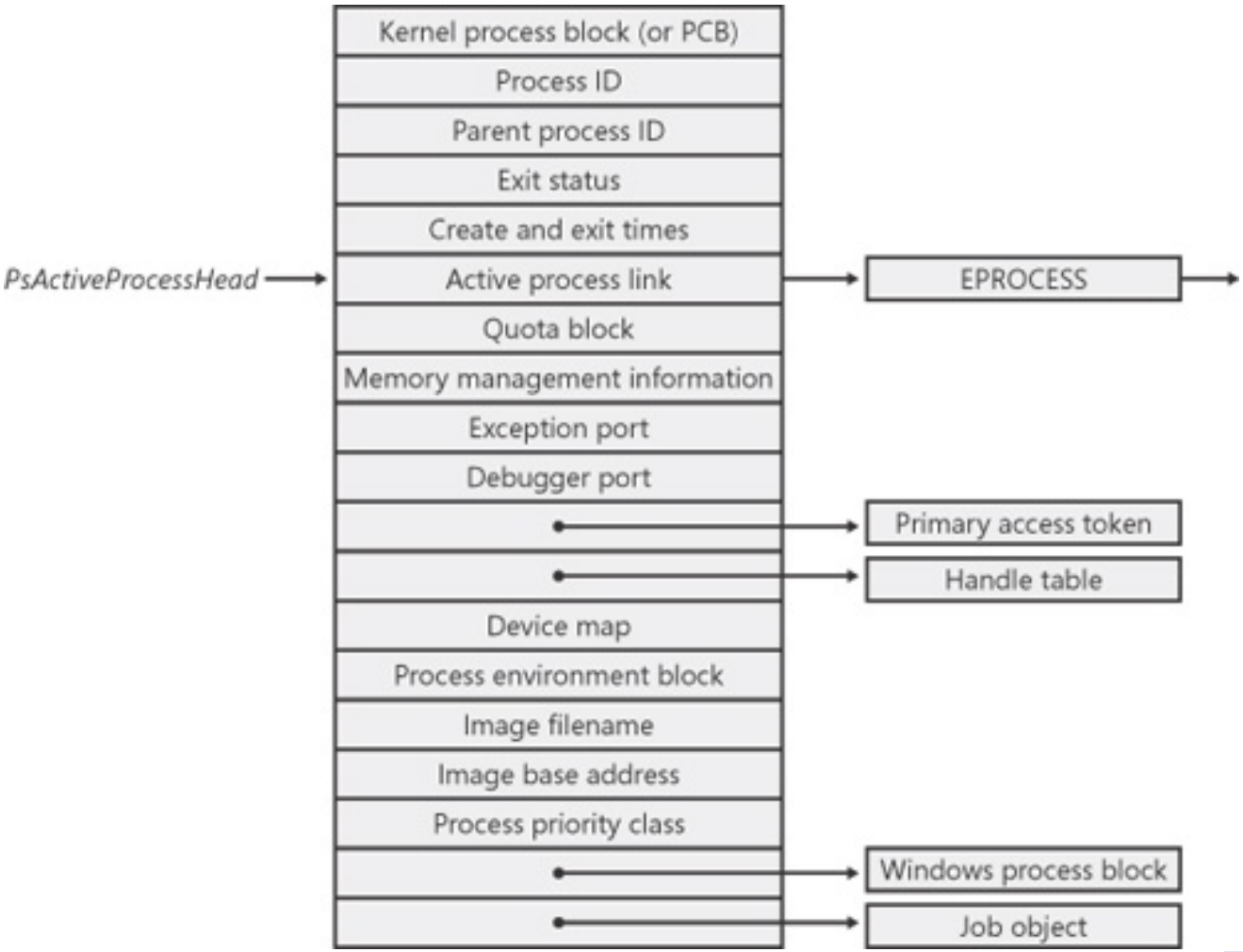


Process Control Blocks (Windows)

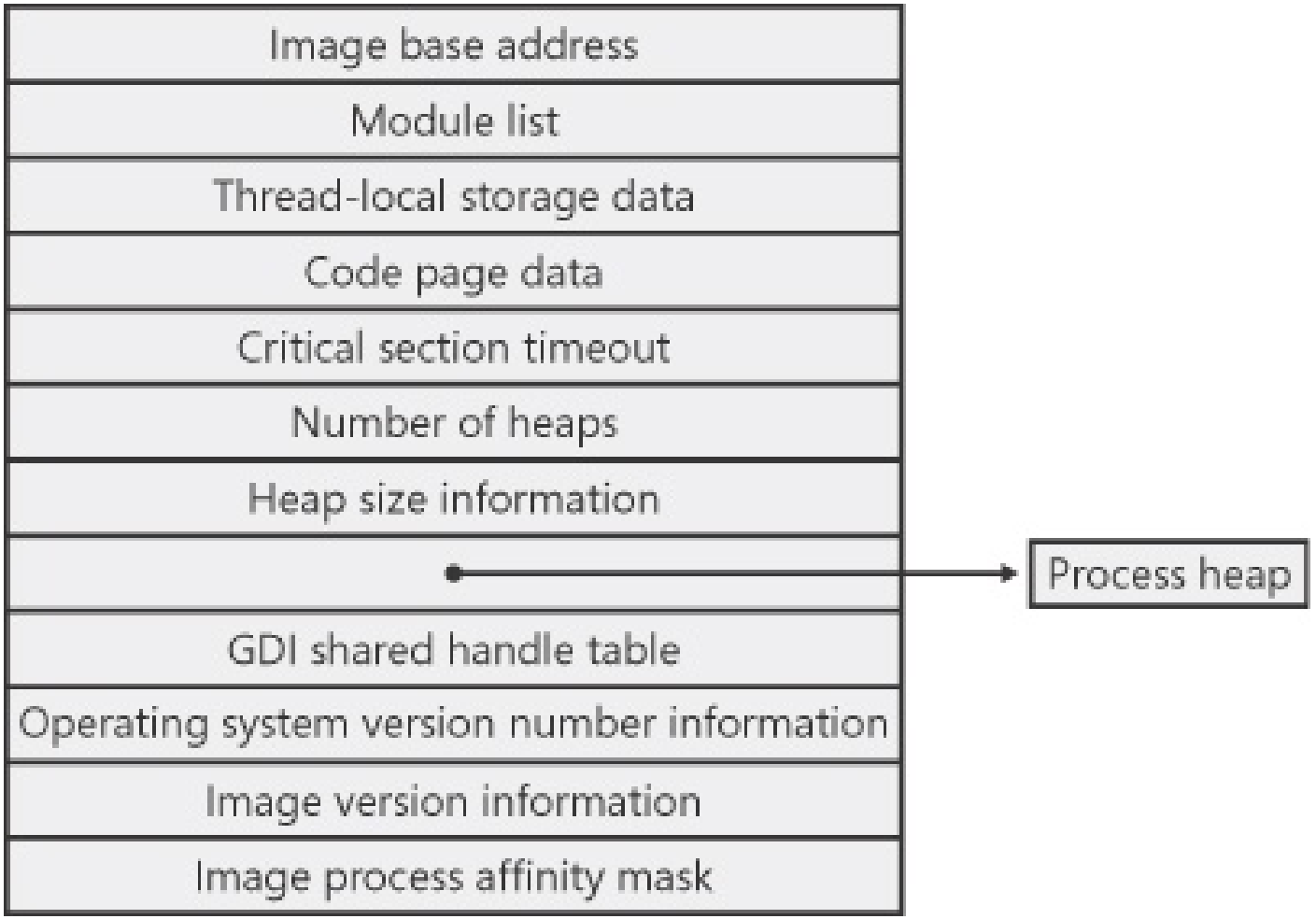


https://www.microsoftpressstore.com/articles/article.aspx?p=2233328

Process Block (EPROCESS)



Process Environment Block (PEB)



taskstruct (Linux)

```
struct task_struct {
        volatile long state;
        void *stack;
       pid_t pid;
       struct list_head tasks;
       ...
};
```

Can inspect data in /proc/pid

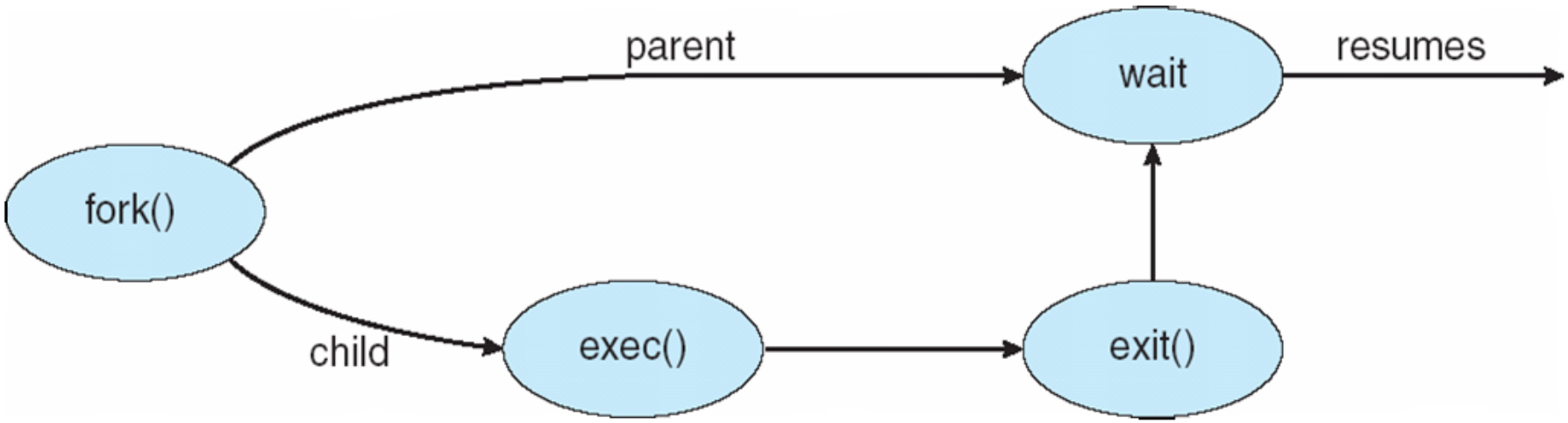
Process Creation

- 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
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- Execution
  - Parent and children execute concurrently
  - Parent waits until children terminate

Process Creation

- UNIX examples
  - `fork` system call creates new process
    - will look at fork soon, in one of our practical lectures.
  - exec system call used after a fork to replace the process' memory space with a new program

Process Creation



Process Termination

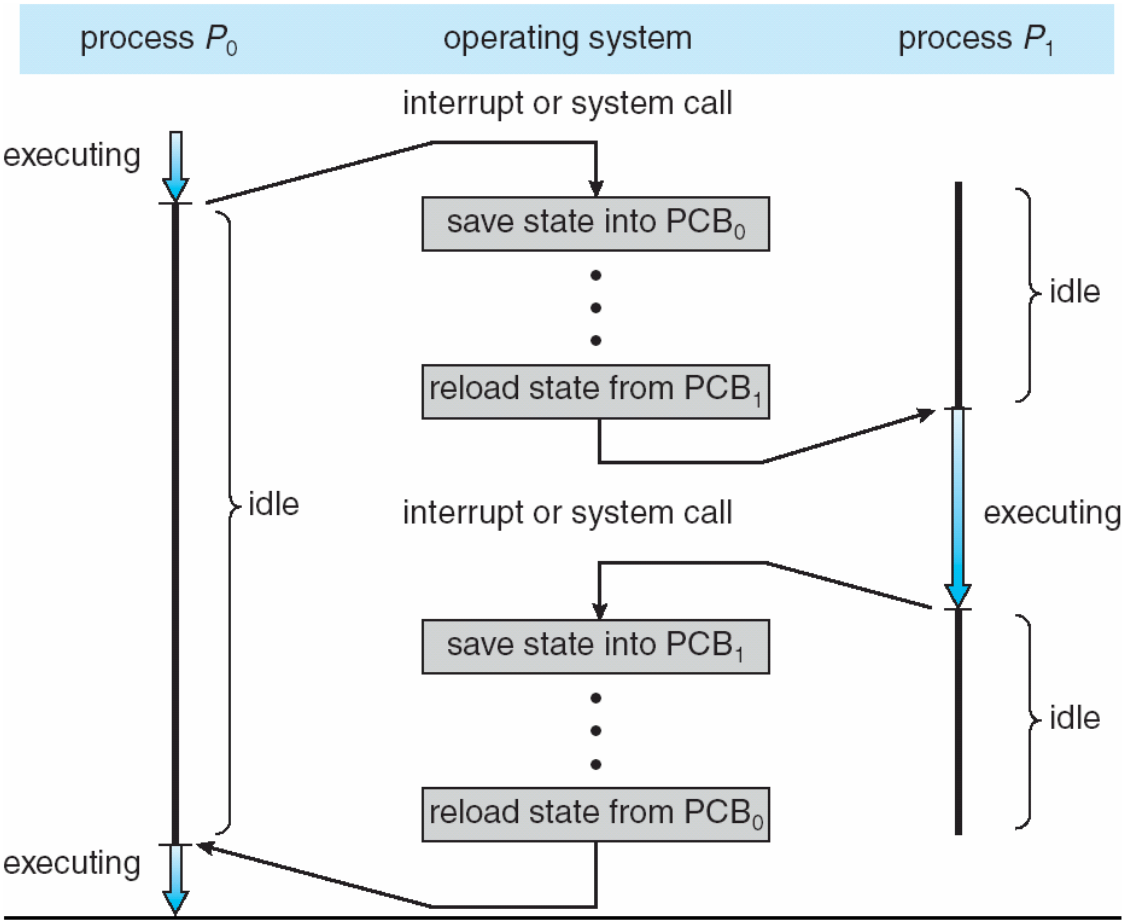
- Process executes last statement and asks the operating system to delete it (exit)
  - Output data from child to parent (via wait)
  - Process' resources are deallocated by operating system
- Parent may terminate execution of children processes (abort)
  - Child has exceeded allocated resources
  -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

Concurrency Through Context Switching

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
- Time dependent on hardware support

Context Switch



Summary

- A process is a program in execution
- Processes are managed by the Operating Systems and can be in various stages (executing, waiting etc.)
- Have system calls for creating processes, as child of existing process
- CPU can be switched from one process to another by OS (context switch)
- Context switches are costly