《操作系统》

进程与线程

首都师范大学
信息工程学院
霍其润

Processes and Threads

- Processes
- Threads
- Interprocess communication
- Classical IPC problems
- Scheduling

一 进程（Processes）

1、进程的概念引入

What is a process?

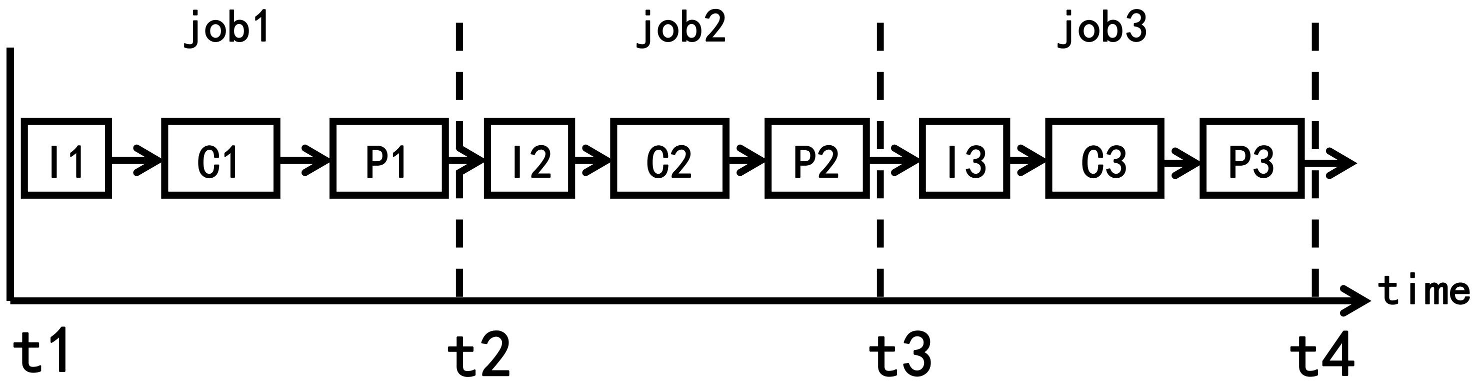
A conceptual model that makes parallelism easier to deal with.

In early, there is one processor, one user, one job.

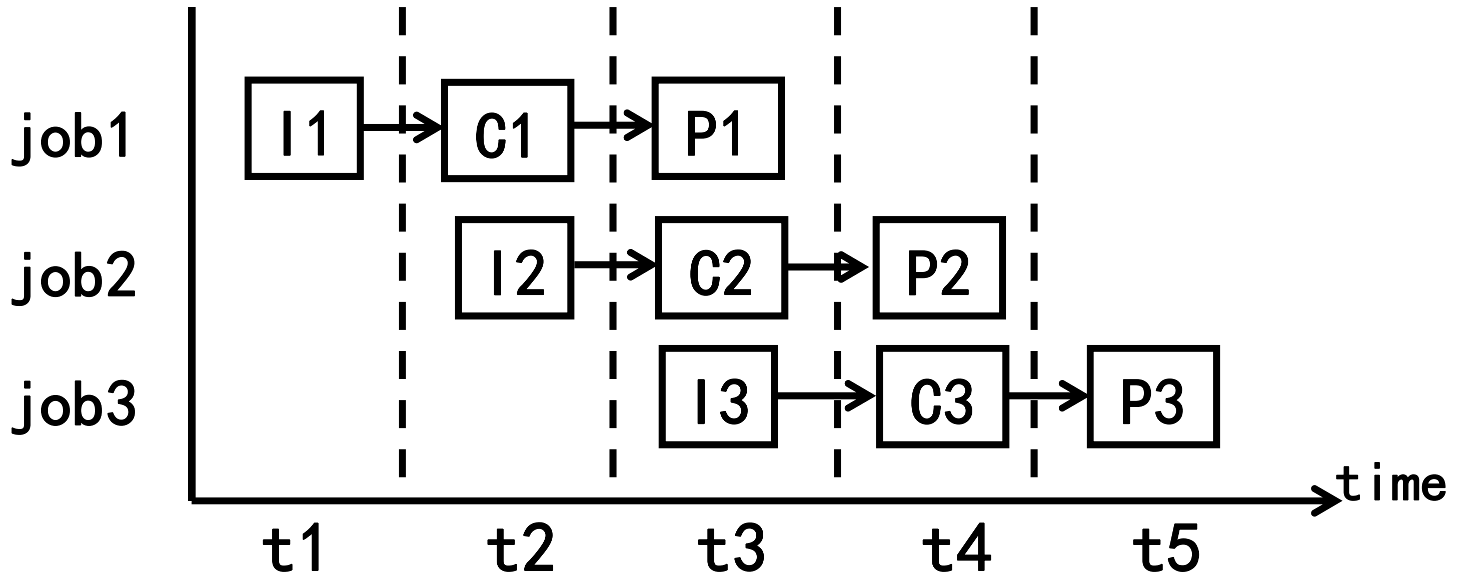
By raising the efficient of processor, we are working at one processor, more user, more jobs.

Parallelism is applied.

Or be called pseudo-parallelism for one processor.

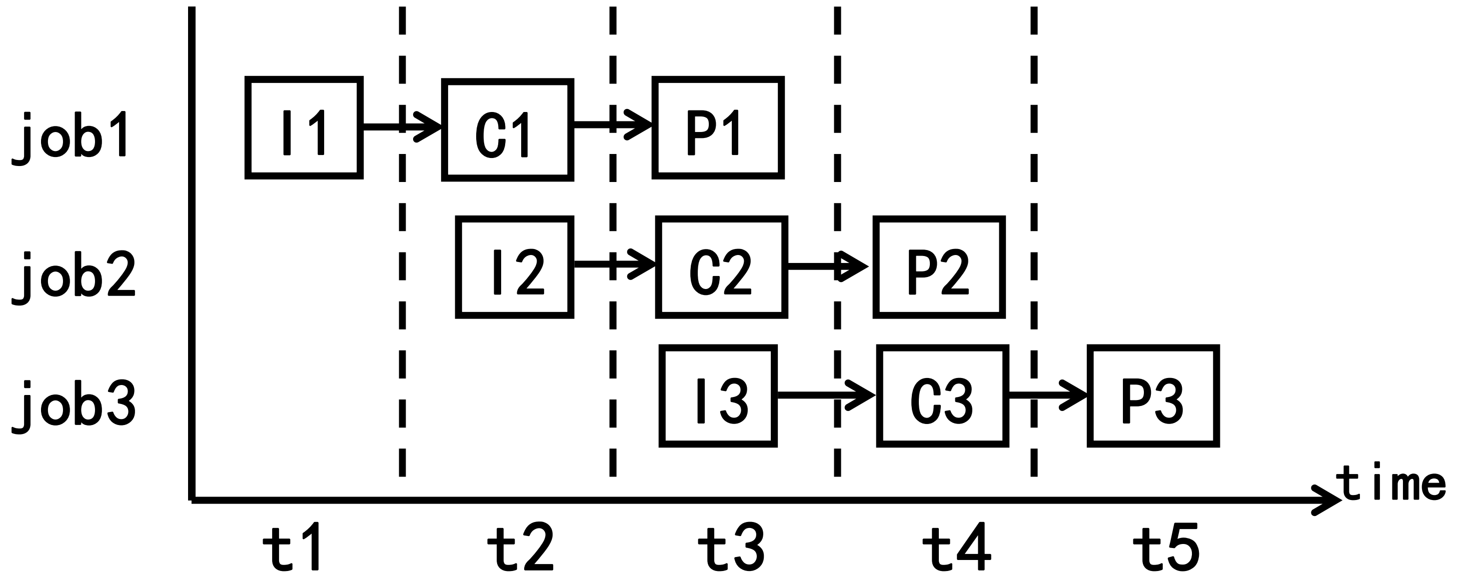


- 早期单道程序技术中，程序为顺序执行方式
  - 顺序性；封闭性；可再现性；与时间无关性
- 可见，一个程序只对应一种执行过程，资源的分配和管理也是面向程序的。



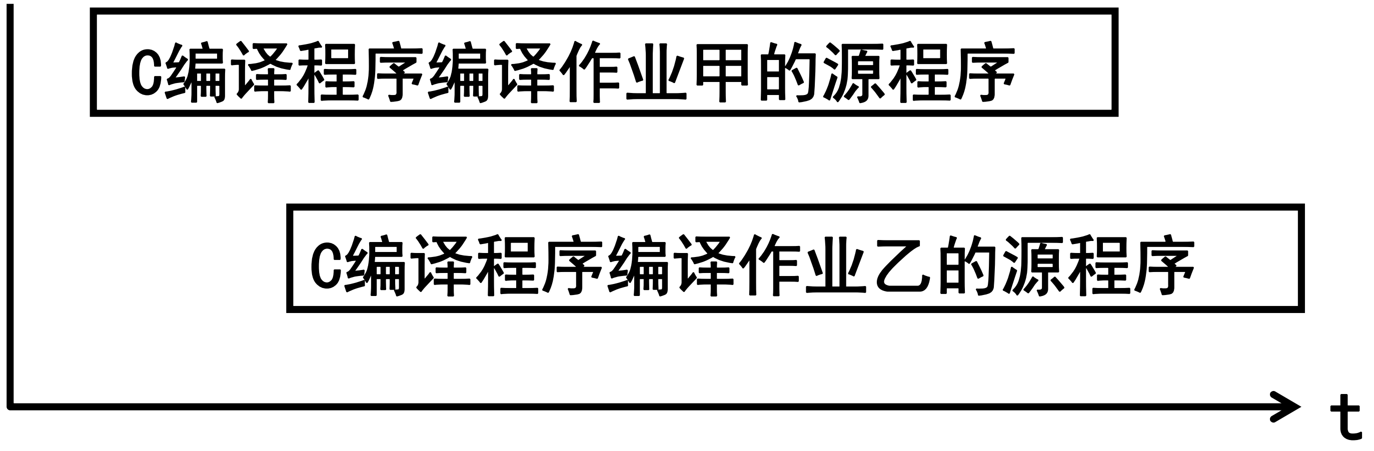
- 多道程序技术中，程序的执行方式改为并发执行
  - 并发性；异步性；开放性；不可再现性

```
begin integer N;
    N:=0;
    cobegin
        program A:
            begin
                L1: ……;
                 N:=N+1;
                 goto L1;
            end;
        program B:
            begin
                L2: ……;
                print(N);
                N:=0;
                goto L2;
            end;
    coend;
end;
```



- 多道程序技术中，程序的执行方式改为并发执行
  - 并发性；异步性；开放性；不可再现性
- 可见，程序与其执行过程不再一一对应，即一个程序可以对应多个执行过程。 资源的分配和管理应面向程序/执行过程呢？

- ✓ 特别是，当并发执行的是同一个程序的情况下，以程序做为资源分配和管理的对象，就很难描述。



显然在多道系统中，资源的分配不能再以程序为对象，而应为程序的执行过程——进程。

Processes

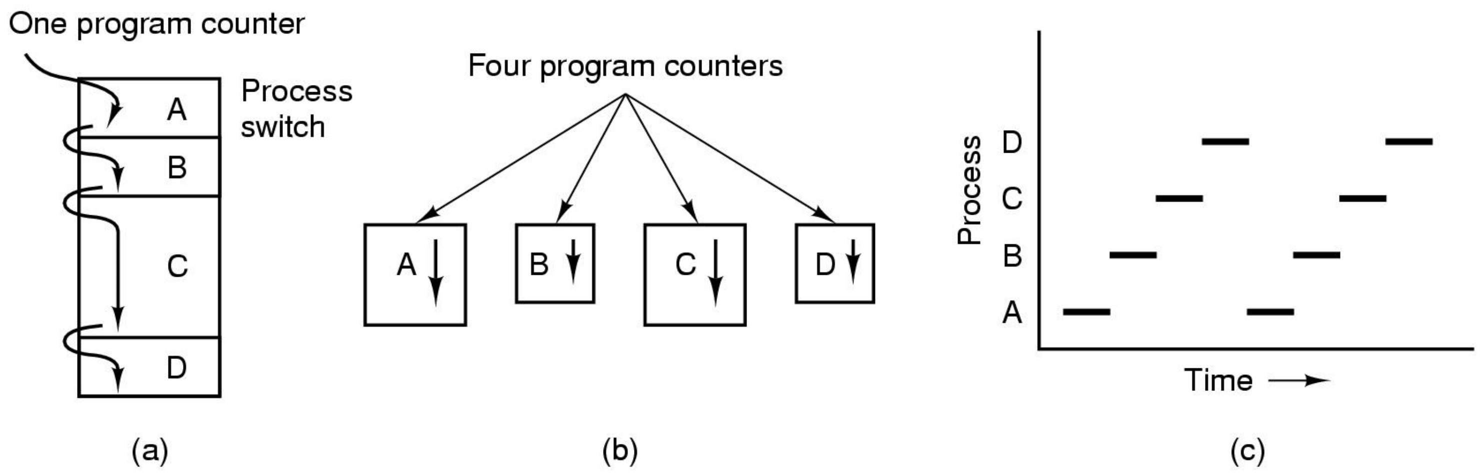
A process is just an instance of an executing program.

进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动，它是系统进行资源分配和调度的一个独立单位。

一 进程（Processes）

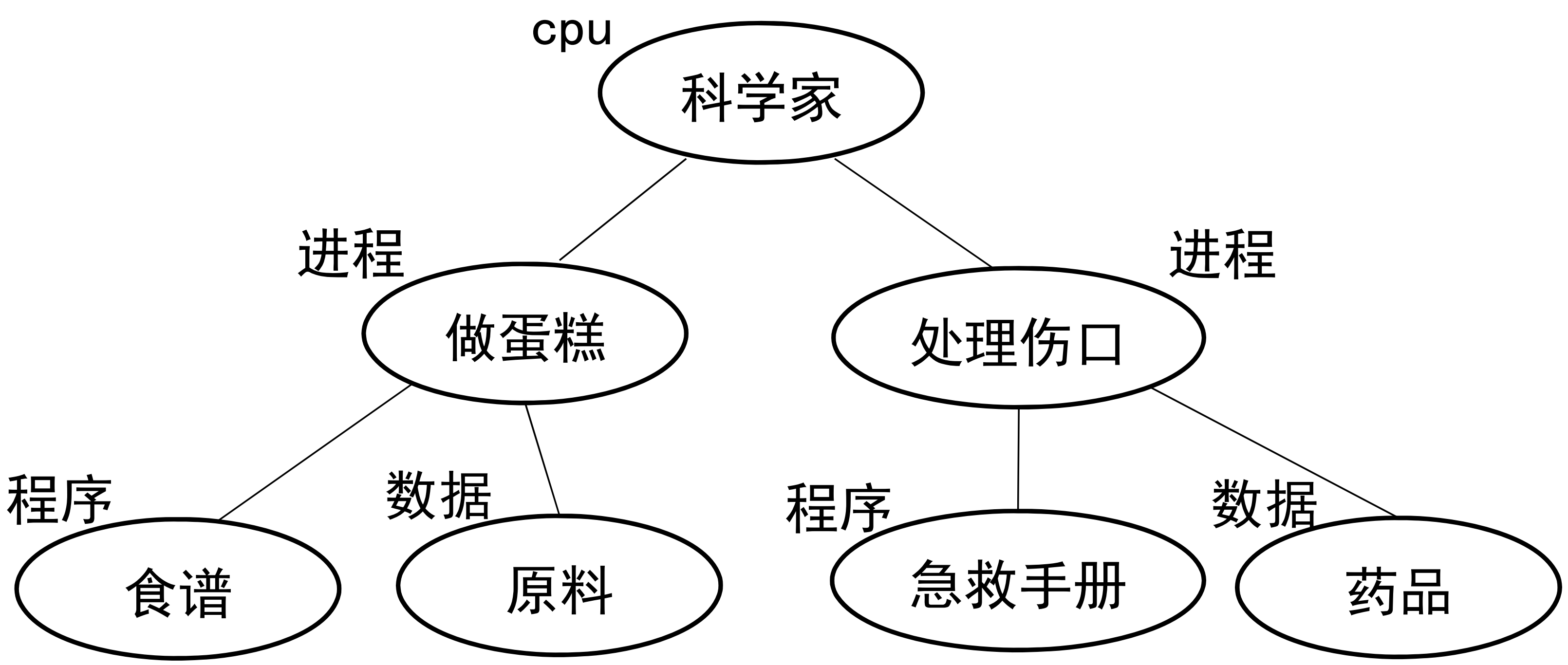
2、进程模型

The Process Model



- Multiprogramming of four programs
- Conceptual model of 4 independent, sequential processes
- Only one program active at any instant

The Difference Between Process and Program



动态性

并发性

The Difference Between Process and Program

- Process is dynamic, and the program is static
- Process is temporary, the program is permanence
- The elements of Process and program is different
- The relationships of Process and program are complex

Process Creation

Principal events that cause process creation

1. System initialization
2. Execution of a process creation system call by a running process
3. User request to create a new process
4. Initiation of a batch job

Process Termination

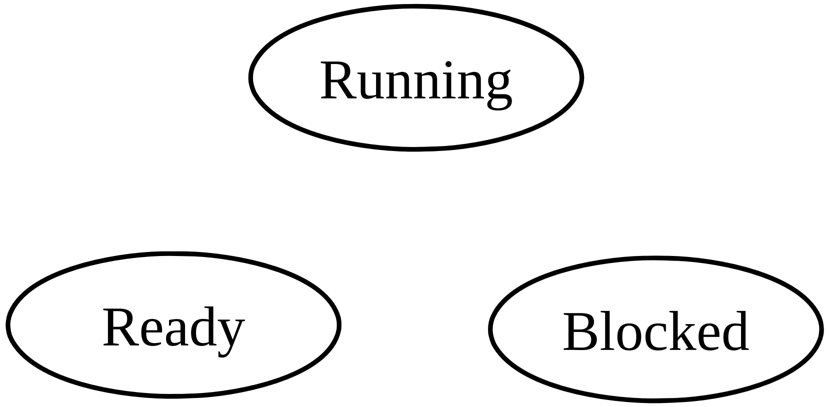
Conditions which terminate processes

1. Normal exit (voluntary)
2. Error exit (voluntary)
3. Fatal error (involuntary)
4. Killed by another process (involuntary)

一 进程（Processes）

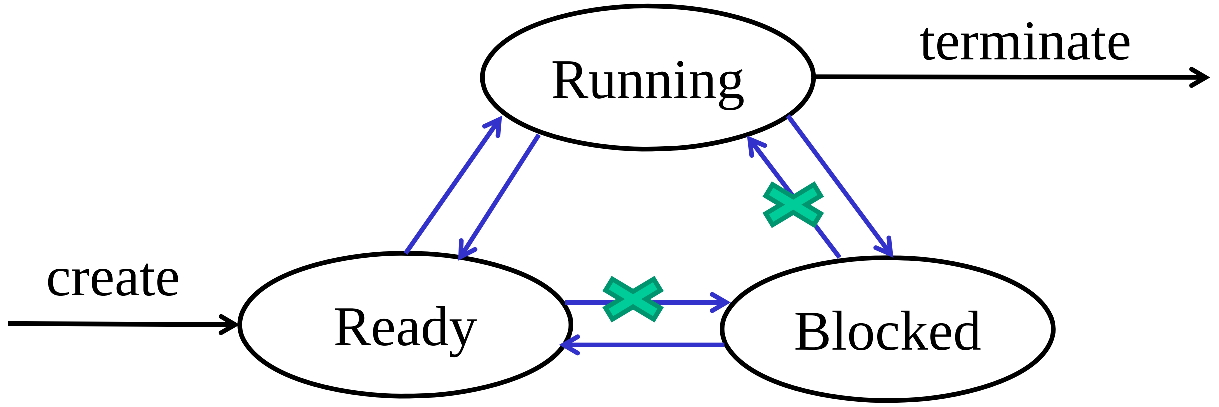
3、进程的状态

Process States



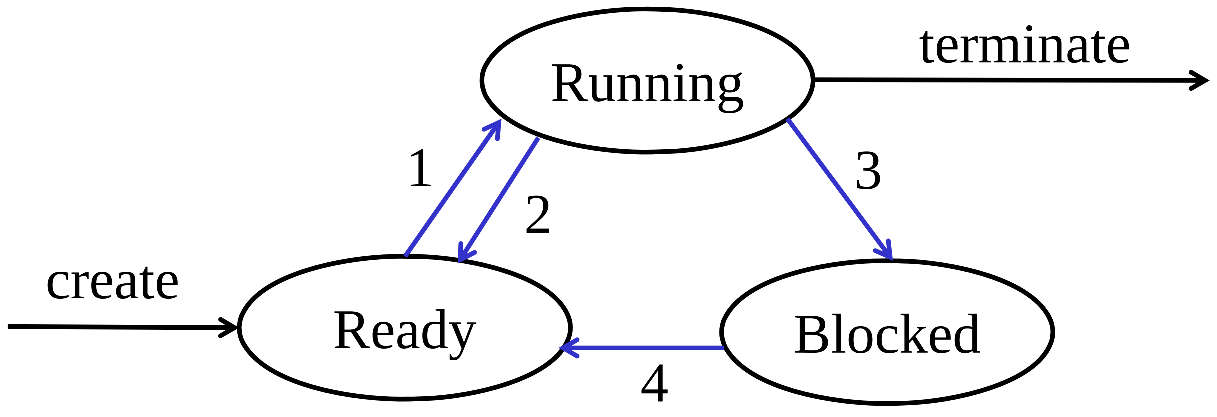
- Possible process states
  - Running
  - Ready
  - Blocked

Process States



- Transitions between states shown

Process States

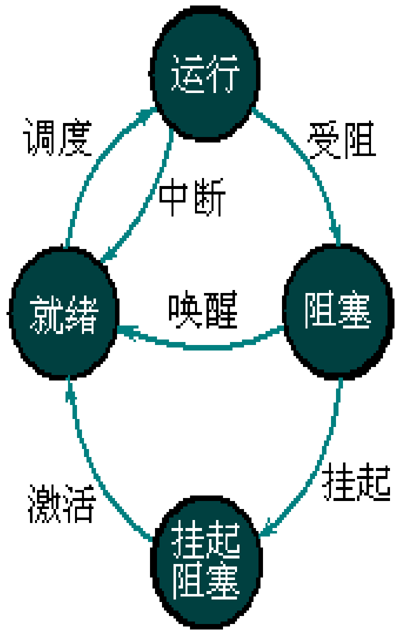


- Transitions between states shown
  1. Scheduler picks this process
  2. Scheduler picks another process
  3. Process blocks for input
  4. Input becomes available

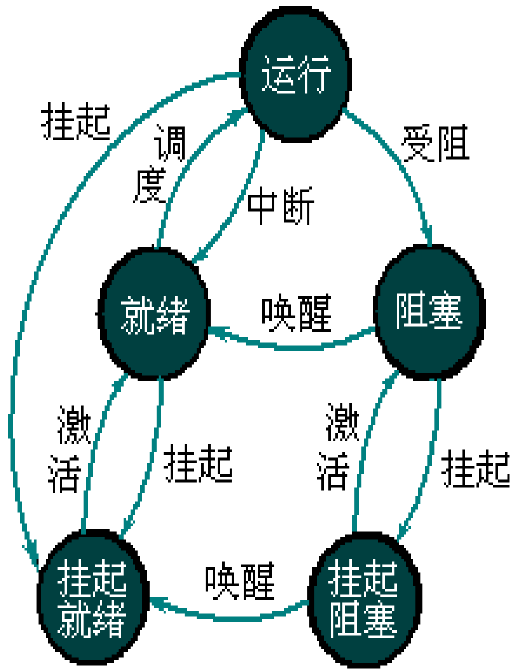
? 当前系统并发10个进程，问处于各种状态的进程分别最多和最少的个数。

– Ready: ~
– Running: ~
– Blocked: ~

含有挂起状态的进程变迁图



（a）一个挂起状态



（b）两个挂起状态

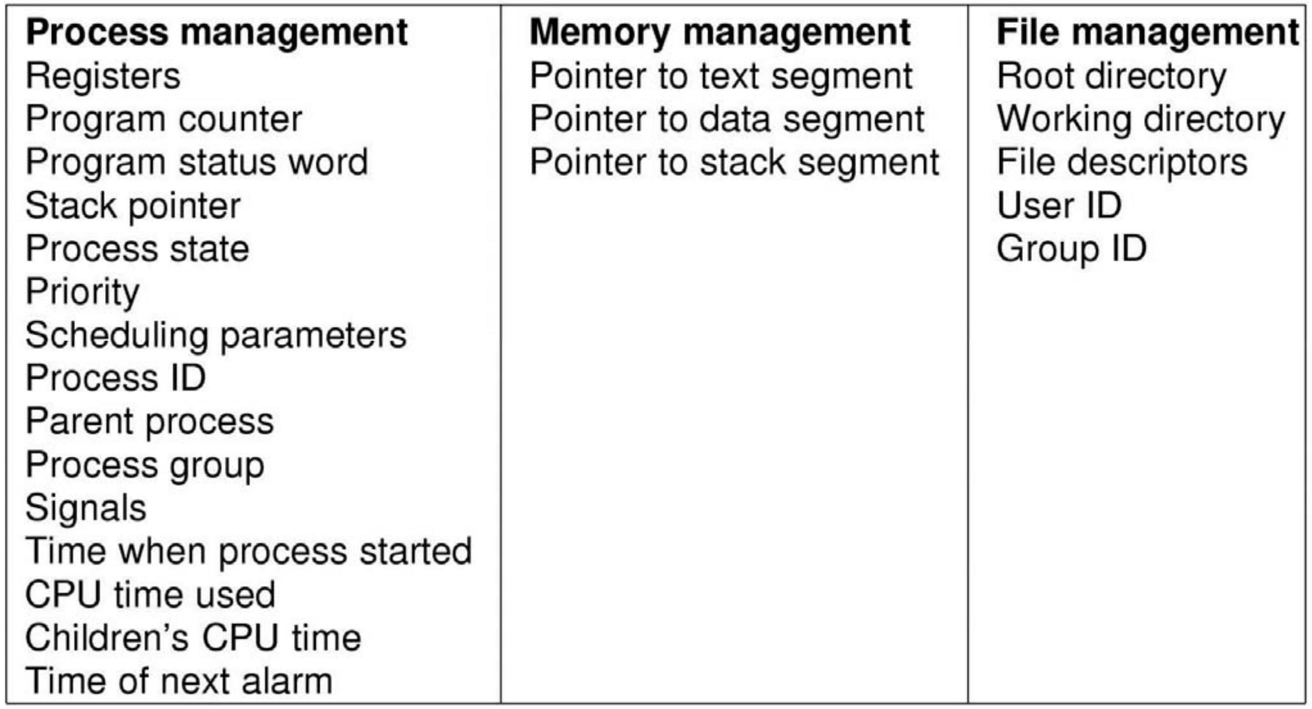
一 进程（Processes）

4、进程的实现

Implementation of Processes (1)

- Elements: code + data + ?
- Process Table (Process Control Block)
  - A memory block which record the information of process, allocated and maintained by OS

Implementation of Processes (2)



Process management	Memory management	File management
Registers	Pointer to text segment	Root directory
Program counter	Pointer to data segment	Working directory
Program status word	Pointer to stack segment	File descriptors
Stack pointer		User ID
Process state		Group ID
Priority		
Scheduling parameters		
Process ID		
Parent process		
Process group		
Signals		
Time when process started		
CPU time used		
Children’s CPU time		
Time of next alarm		

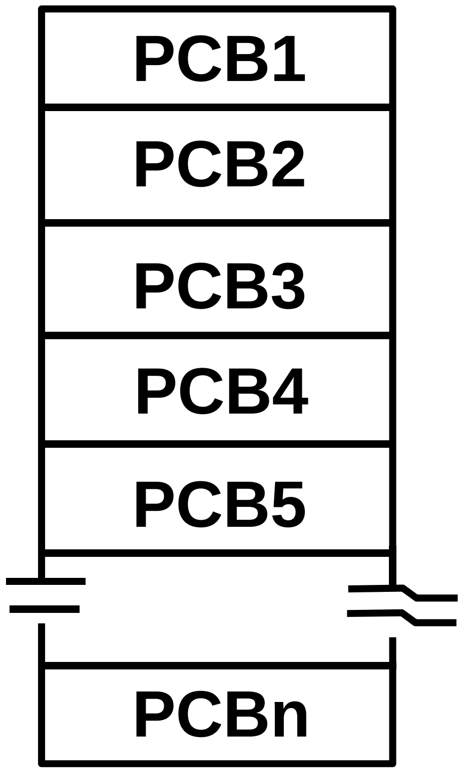
Fields of a process table entry

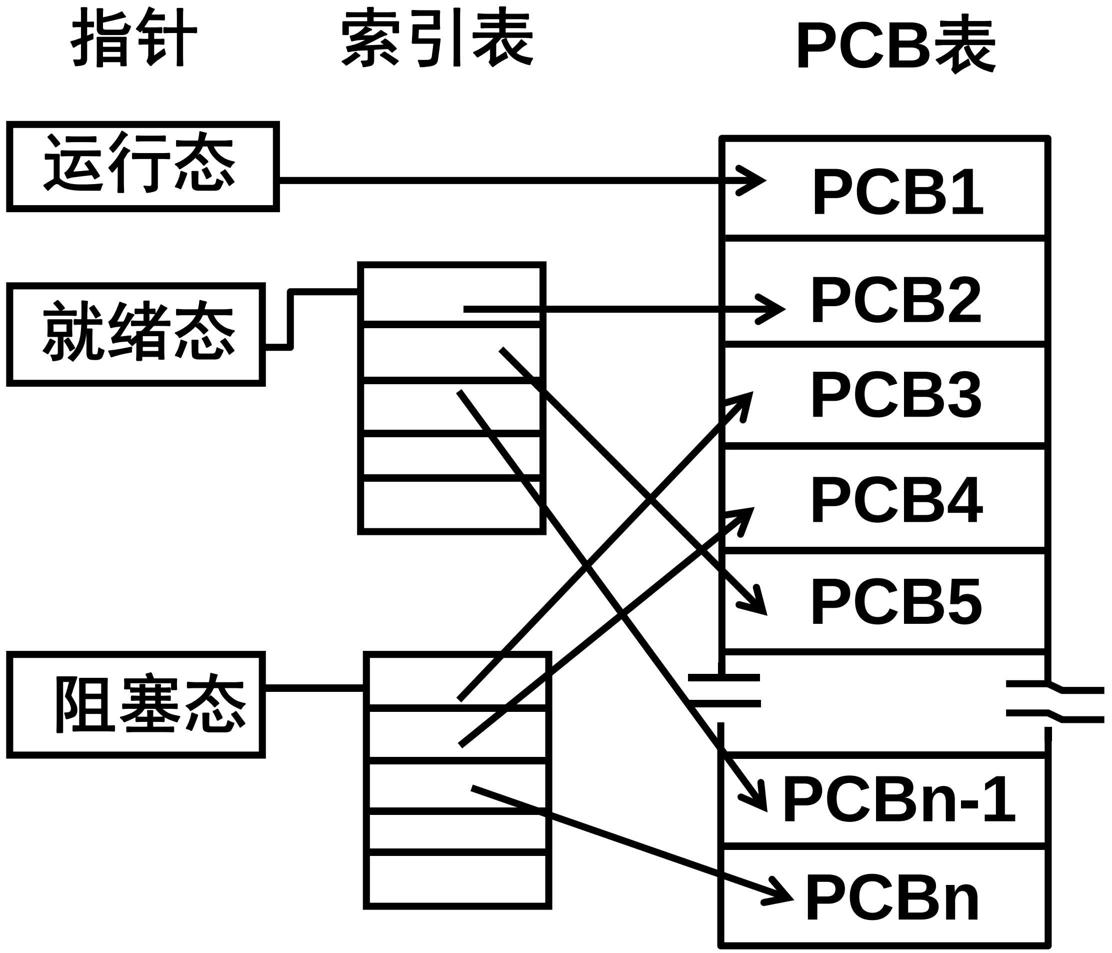
Implementation of Processes (1)

- Elements: code + data + ?
- Process Table (Process Control Block)
  - A memory block which record the information of process, allocated and maintained by OS
- OS creates process, allocates memory, I/O devices, files, and so on to user program.
- Create the PCB(the number is limited)
  - Always kept in OS kernel, visited by using system call

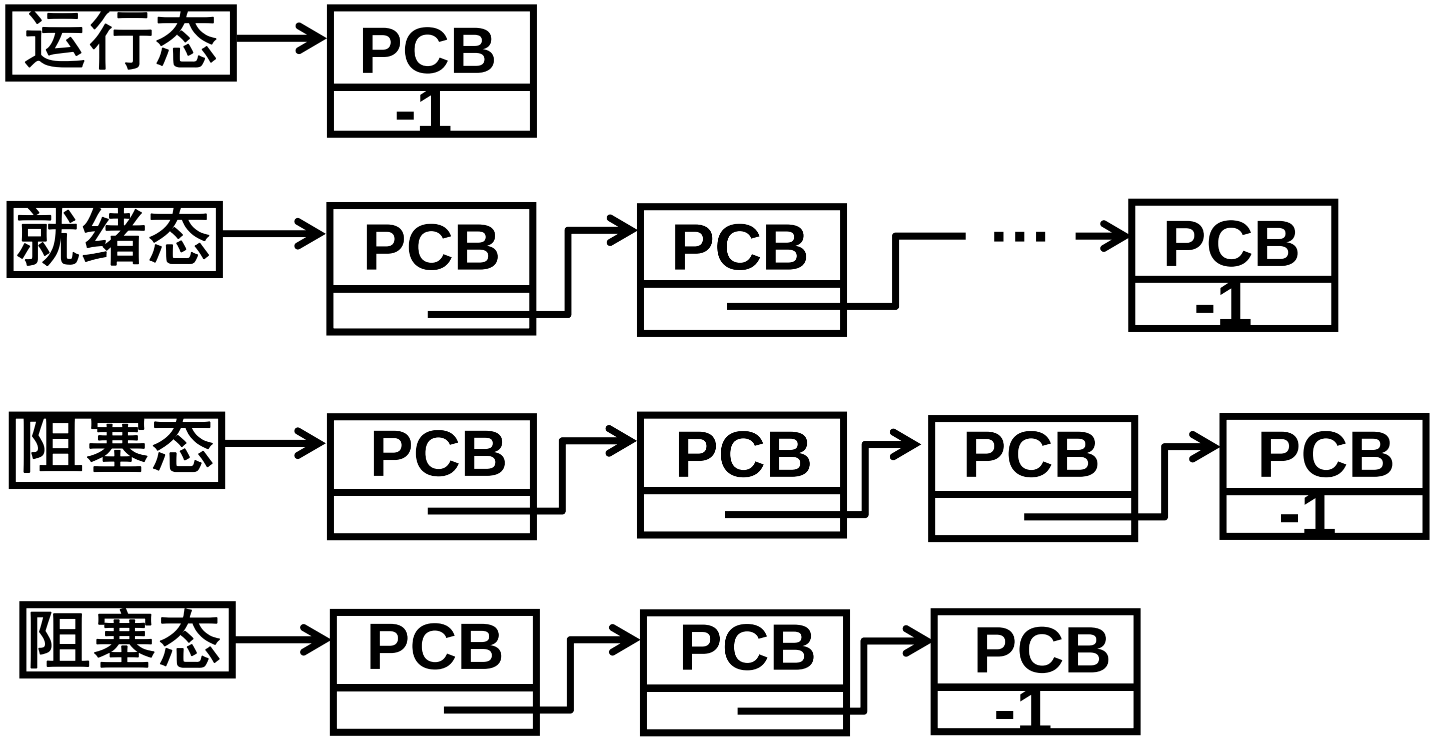
- 为了便于对进程进行管理，系统把各个进程的PCB集中存放在内存的指定区域，并组织在一起形成PCB表。
- 不同的操作系统采用不同的PCB表组织结构，PCB表的物理组织结构直接关系到系统的效率。
- 常用的有三种：线性表结构、索引表结构和链接表结构。

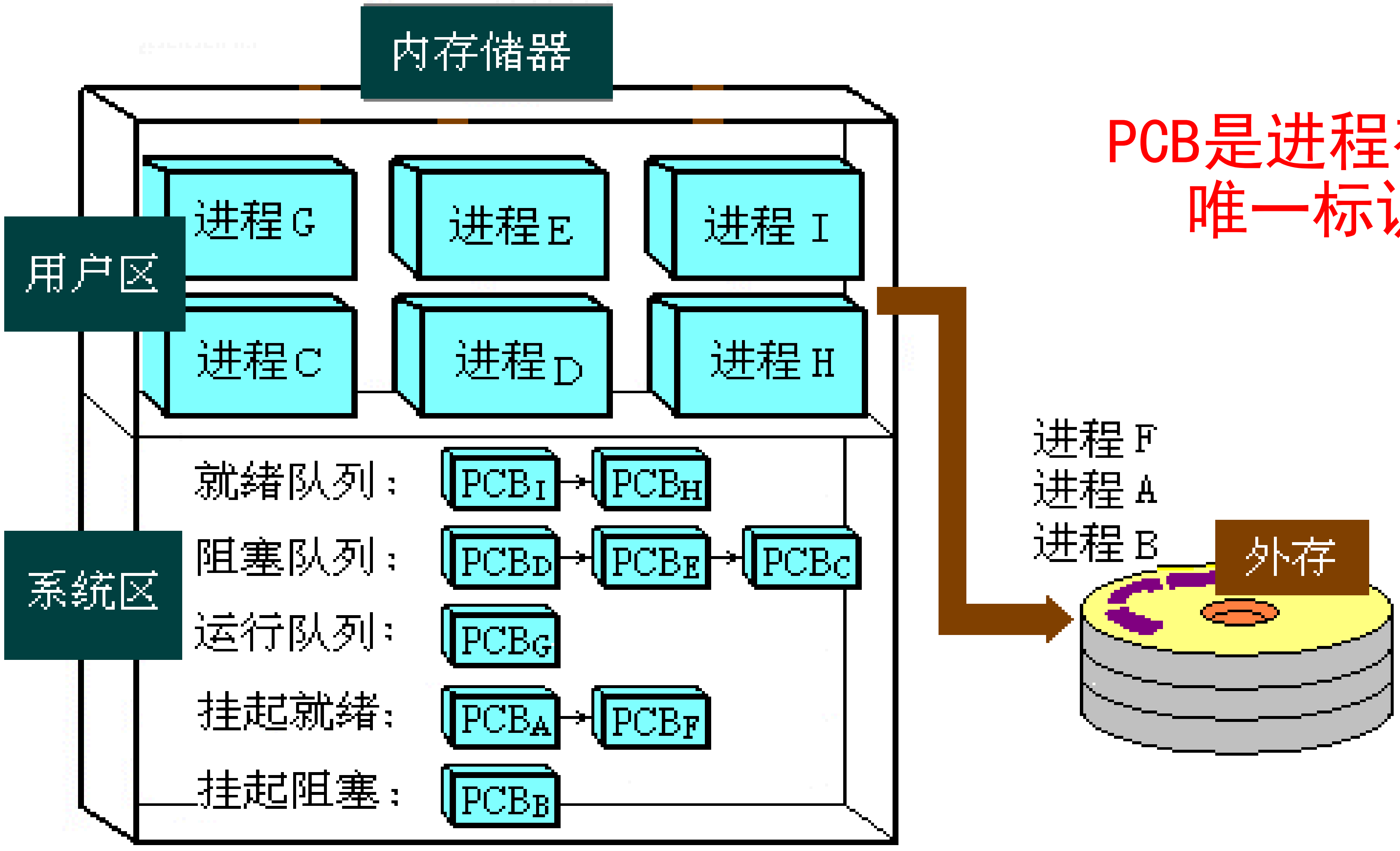
PCB表




指针
索引表
PCB表
运行态
就绪态
阻塞态
PCB1
PCB2
PCB3
PCB4
PCB5
PCBn-1
PCBn

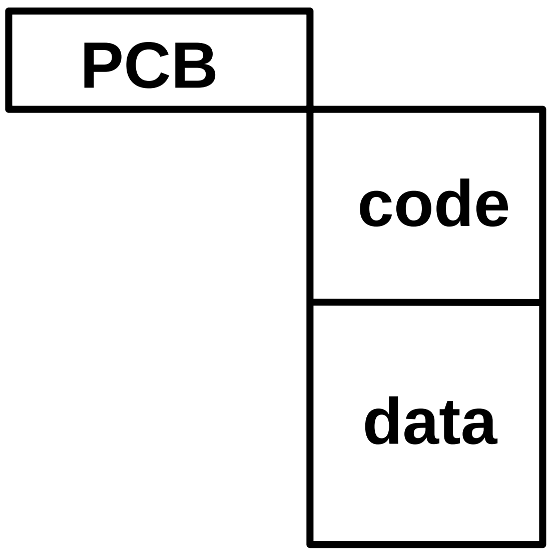
指针



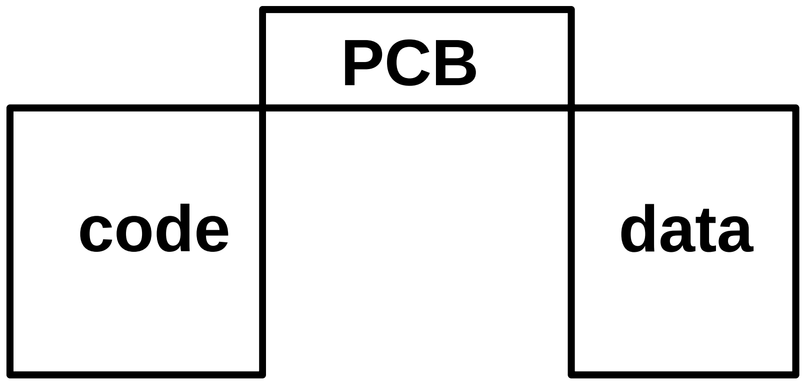


PCB是进程存在的唯一标识！

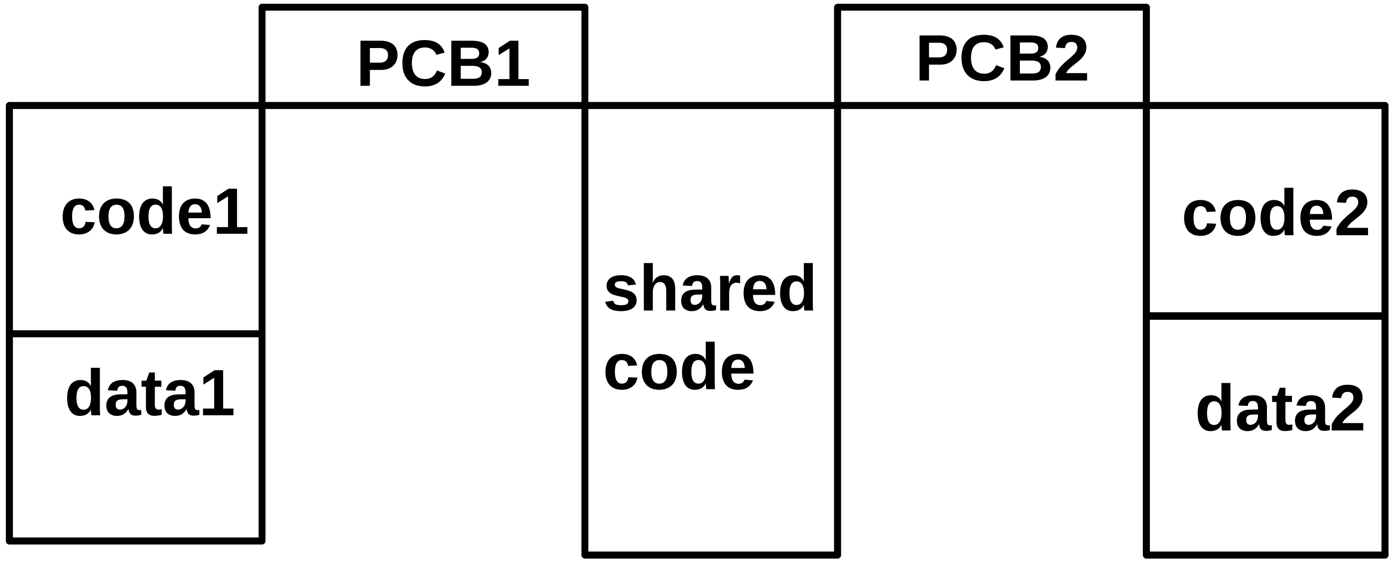
Space of Processes



(A)



(B)

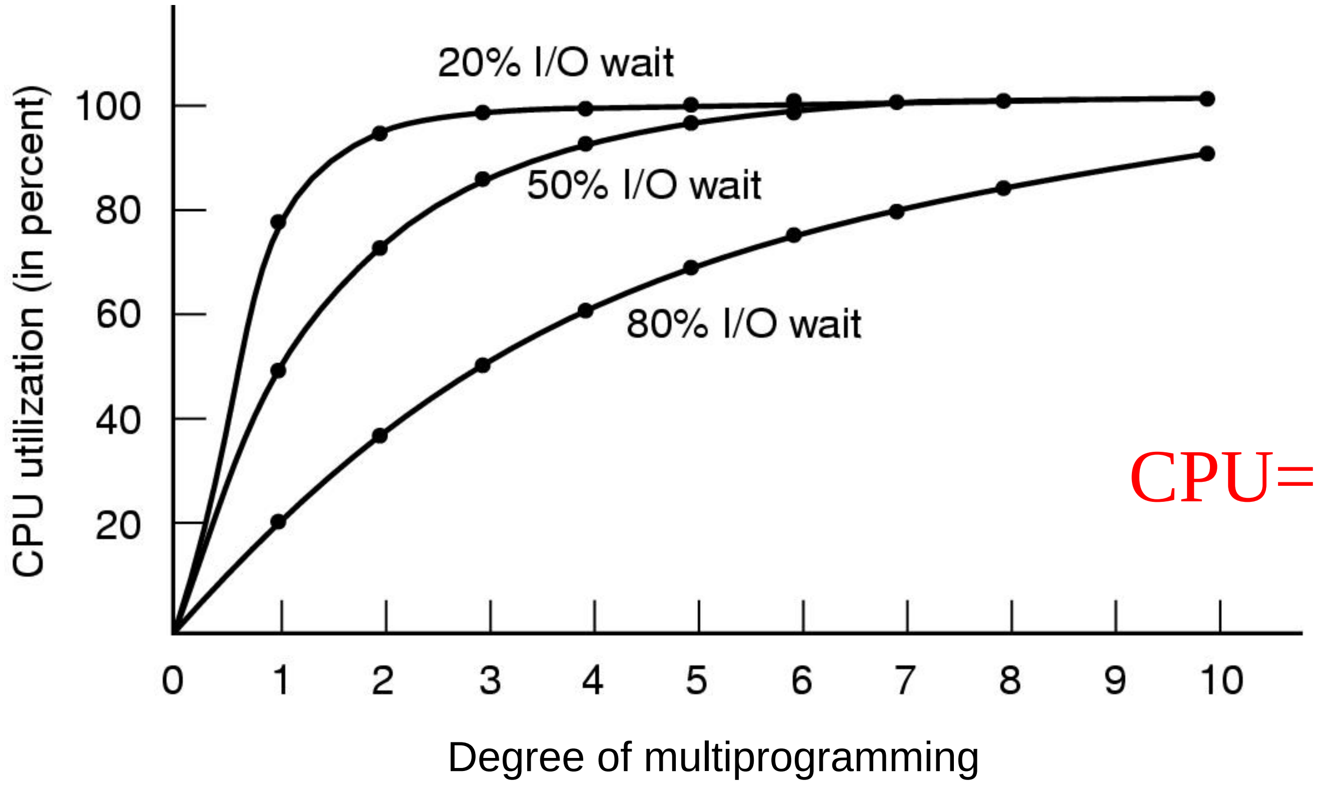


(C)

一 进程（Processes）

5、多道程序设计模型

Modeling Multiprogramming



$CPU=1-p^n$

CPU utilization as a function of number of processes in memory

Example:
A computer has 512MB of memory, with OS taking up 128MB and each user program also taking up 128MB. Three user program to be in memory at once. With 80% average I/O wait, we have a CPU utilization of $1 – 0.8^3$ = 49%.

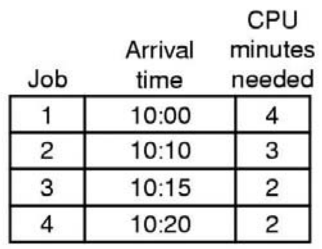
- Adding another 512MB of memory allows the system to go from 3 to 7, thus raising the CPU utilization to $1 – 0.8^7$ = 79%.

  In another word, the additional 512MB will raise the throughput by 30%.

- Adding yet another 512MB would increase CPU utilization only from 79% to $1 – 0.8^{11}$ = 91%.

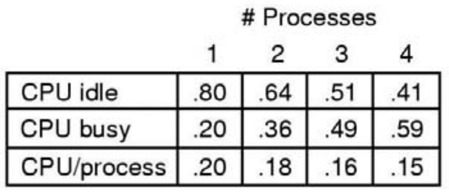
  Thus raising the throughput by only another 12%.

Analysis of Multiprogramming System Performance



Job	Arrival time	CPU minutes needed
1	10:00	4
2	10:10	3
3	10:15	2
4	10:20	2

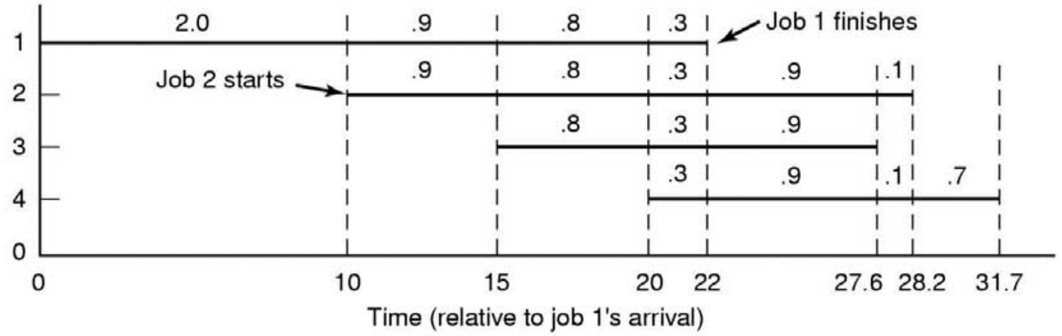
(a)



	# Processes 1	2	3	4
CPU idle	.80	.64	.51	.41
CPU busy	.20	.36	.49	.59
CPU/process	.20	.18	.16	.15

(b)

总CPU利用率提高明显，单个进程降低幅度不大。



(c)

- Arrival and work requirements of 4 jobs
- CPU utilization for 1 – 4 jobs with 80% I/O wait
- Sequence of events as jobs arrive and finish
  - note numbers show amount of CPU time jobs get in each interval

小 结

- 进程的概念特性、三要素；
- 进程和程序的区别；
- 进程的三种基本状态及四种转换；
- PCB—进程表；
- 多道程序并发性能分析。