**第二章．进程与线程**

1. multiprogramming: A mode of operation that provides for the interleaved execution of two or more programs by a single processor.

2. concurrency: Allows multiple applications to run at the same time

3.process model

4. process:

1) process creation

* System Initialization
* Execution of a process creation system call (fork())
* User request to create a new process
* Initiation of a batch job

2)process termination

* Normal exit (voluntary)
* Error exit (voluntary)
* Fatal error (involuntary)
* Killed by another process (involuntary)

3)process status:running,ready,blocked

4)process implementation

OS maintains a PCB for each process ,PCB contains process iditification,process state information ,process control information

5.thtead

1)user-level threads

advantages: ULTs can run on any OS;Thread context switch is much faster than process context switch; Thread context switch is much faster than process context switch; Better scalability

issues: Most system calls are blocking and the kernel blocks processes. So all threads within the process will be blocked.

6.thread vs process

* Process
  + **Unit of resource ownership** with respect to the execution of a single program
  + Can encompass **more than one thread of execution**
  + Processes are largely independent
* Thread
  + **Unit of execution**
  + Belongs to a process
  + Threads are part of the same “job” and are actively and closely cooperating

**对顺序进程的理解：**

**不管是否具有并行性，一个进程看起来是不被中断的执行完，当发生上下文切换时，如果必须显示地保存或重新装入相应的计算状态，则会破坏顺序进程模型。**

**阻塞系统调用与非阻塞系统调用：**

**系统调用分为两种，一种是阻塞的，一种是非阻塞的。**

**阻塞系统调用与read(),fread(),getc(),write()这些I/O操作相关，它们在调用完成时才会返回，进程（或线程）在阻塞系统调用时会被转到阻塞状态。**

**非阻塞系统调用（for I/O）是异步的I/O，比较复杂，一旦I/O操作开始，调用就会返回，并且调用程序继续进行，一旦I/O操作完成，一个中断就会被发给该调用程序。**

**中断：指当出现需要时，CPU暂时停止当前程序的执行转而执行处理新情况的程序和执行过程。**