# 第三章 进程管理

## 3.1 进程概念

**1、作业和作业步的概念**

①作业：

②作业步：

**2、程序的顺序执行的特点：**

①

②

③

**3、程序并发执行的特点：**

①

②

③

**4、程序并发执行的优缺点：**

优点：

①

②

缺点：

①

②

**5、进程的定义：**

并发执行的程序在执行过程中分配和管理资源的基本单位

**6、进程的特征以及各自的含义：（简答题必考）**

①

②

③

④

⑤

**7、程序和进程的不同：**

①

②

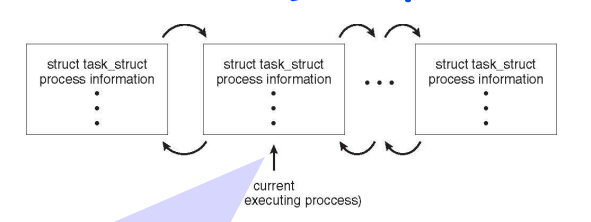
③

**8、进程的状态以及转换图**

**9、进程控制块PCB**

①作用：

10、Linux进程控制块的组织：



Current:指向当前正在执行进程的（）

内核如何修改当前进程的状态：（）

## 3.2 进程调度

**1、上下文切换**

概念：

**2、进程上下文**

**概念：**

保存于：

3、上下文切换的时间

①存粹的开销：

②影响因素：（）、（）

## 3.3 进程控制

### 3.3.1进程创建

**1、过程描述：（进程图画一遍）**

2、bash：（）；P：（）

**3、init进程：**

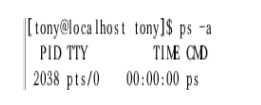
① pid==（）

② 启动（）

**4、ps命令**

①ps -a :

作用：



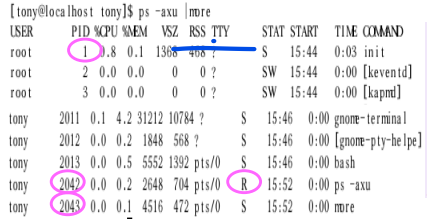
PID:

TTY:

TIME:

CMD:

②ps -axu|more



USER:

%CPU:

%MEM:

VSZ:

RSS:

STAT:进程的状态； R、S(D)、T、Z

**5、Linux下的进程创建fork**

①fork（）执行2次返回

建立一个子进程与自己独立的并发的运行

②fork（）的返回值

-1：进程创建失败

>0：返回值为子进程的PID，在父进程的上下文中

0:在子进程的上下文中

③fork之后父子进程哪一个先执行不确定，取决于内核调度；如果想要让父子进程按照一定的顺序执行，则需要使用**进程同步的系统调用**

**6、fork中父子进程的关系**

①子进程继承父进程的相关属性（**说明不是完全相同**）

子进程的地址空间复制父进程的地址空间

②父子进程之间的区别：（）、（）

③数据段的内容

子进程刚创建时，复制父进程的（）

父子进程可以修改各自的（）而互不影响

7、通常在系统调用fork()之后，有个进程使用系统调用exec()，以用新程序来取代进程的内存空间。系统调用exec()加载二进制文件到内存中（破坏了包含系统调用exec()的原来程序的内存内容），并开始执行。采用这种方式，这2个进程之间能相互通信，并能够按各自方法运行。**父进程能创建更多子进程，或者在子进程运行时没有什么可做时，那么它采用系统调用wait()把自己移出就绪队列，直到子进程终止**。因为调用exec（）用新程序覆盖了进程的地址空间，所以调用exec()除非出现错误，不会返回控制。

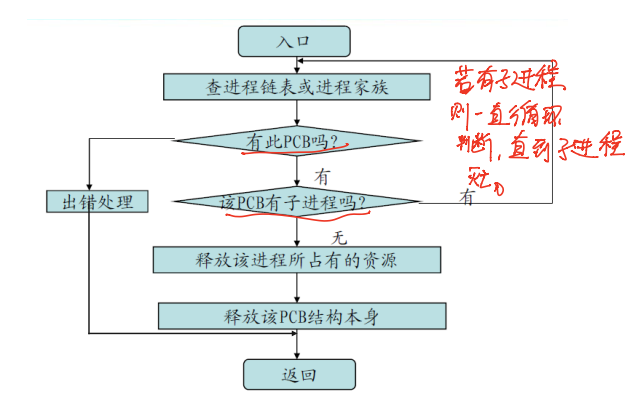
**8、执行一个新程序—exec族函数**

Fork：子进程继承了父进程的地址空间，父子进程都继续执行处于fork()之后的指令

Exec：加载并执行.exe文件，以新程序取代子进程的地址空间

### 3.3.2 进程终止

**1、进程中止**



**2、僵尸进程和孤儿进程**：

①僵尸进程（过渡状态）：

②孤儿进程：

**3、exit()系统调用**

①正常终止：返回值（）；其余均为异常终止

②在exit之后，操作系统会释放其进程资源；但是还存在于进程表的条目中。直到其父进程调用wait()，父进程释放其pid和在**进程表中的条目**。

4、