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

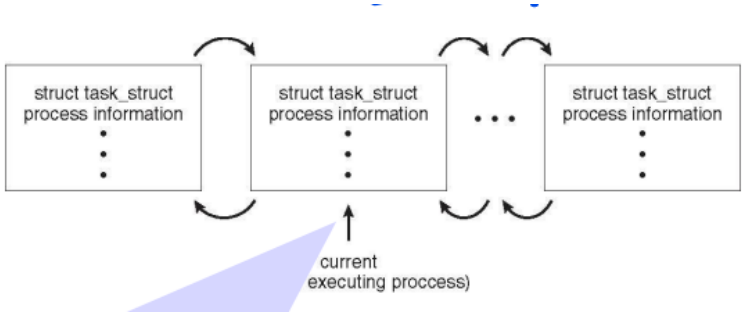
②

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:

```
[tony@localhost tony]$ ps -a
  PID TTY          TIME CMD
 2038 pts/0    00:00:00 ps
```

PID:

TTY:

TIME:

CMD:

②ps -axu|more

```
[tony@localhost tony]$ ps -axu |more
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.8  0.1   1368   468 ?        S    15:44   0:03 init
root         2  0.0  0.0      0     0 ?        SW   15:44   0:00 [keventd]
root         3  0.0  0.0      0     0 ?        SW   15:44   0:00 [kapmd]
tony      2011  0.1  4.2  31212 10784 ?        S    15:46   0:00 gnome-terminal
tony      2012  0.0  0.2   1848   568 ?        S    15:46   0:00 [gnome-pty-helpe]
tony      2013  0.0  0.5   5552  1392 pts/0    S    15:46   0:00 bash
tony      2042  0.0  0.2   2648   704 pts/0    R    15:52   0:00 ps -axu
tony      2043  0.0  0.1   4516   472 pts/0    S    15:52   0:00 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：()、()

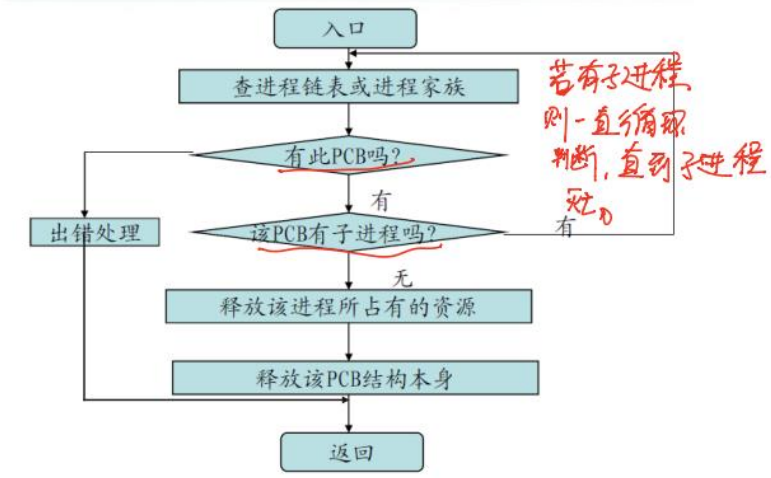
③数据段的内容
子进程刚创建时，复制父进程的（）
父子进程可以修改各自的（）而互不影响

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 、