

# 哈工大操作系统-L9多进程图像

## 哈工大操作系统-L9多进程图像

- 1.多进程图像概述
- 2.引出：操作系统要实现多进程应该解决什么问题
- 3.多个进程执行会互相影响--需要进行内存管理
- 4.总结
  - 4.1实现多进程需要什么？(五部分)
  - 4.2进程管理+内存管理=多进程图像

操作系统的多进程图像是什么样子的？

## 1.多进程图像概述

- 操作系统通过管理进程来管理用户对计算机的使用。
- 用户启动计算机就是启动了一堆进程。用户管理计算机也是在管理这一堆进程。

### 多个进程使用CPU的图像

#### ■ 如何使用CPU呢？

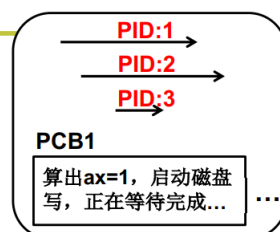
- 让程序执行起来

#### ■ 如何充分利用CPU呢？

- 启动多个程序，交替执行

#### ■ 启动了的程序就是进程，所以是多个进程推进

- 操作系统只需要把这些进程记录好、要按照合理的次序推进(分配资源、进行调度)
- 这就是多进程图像...



### 多进程图像从启动开始到关机结束

#### ■ main中的fork()创建了第1个进程

- init执行了shell(Windows桌面)

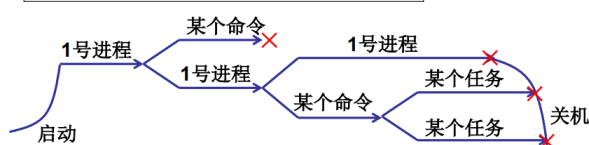
#### ■ shell再启动其他进程

```
int main(int argc, char * argv[])
{ while(1) { scanf("%s", cmd);
  if(!fork()) { exec(cmd);} wait(); } }
```

一命令启动一个进程，返回shell再启动其他进程...

```
if(!fork()) {init();}
```

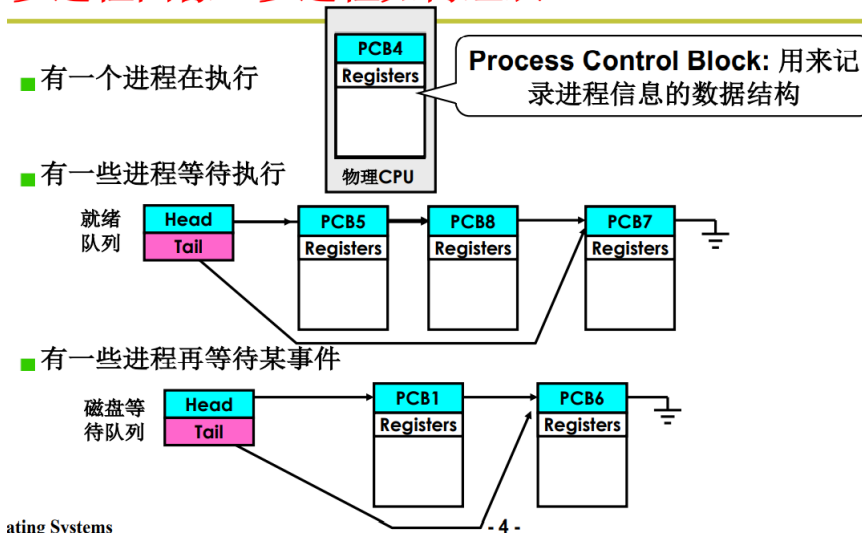
```
login as: log
login192.168.1.1's password:
last login: Mon Mar 12 22:47:58 2006 from 192.168.1.2
[log@localhost ~]$ ls
work.txt
[log@localhost ~]$
```



## 2.引出：操作系统要实现多进程应该解决什么问题

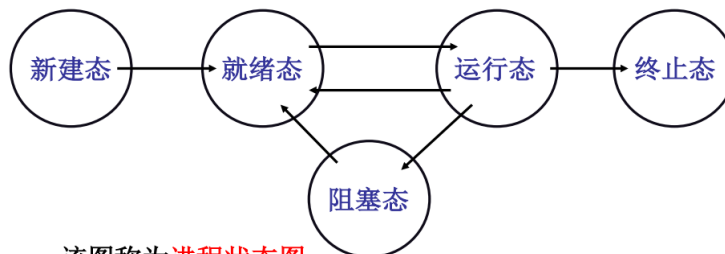
- 多个进程的信息如何存放--使用PCB(Process Control Block)
- 多个进程如何组织--PCB+队列+状态
  - 不同状态的进程在对应的不同的队列
  - 把PCB放在不同的队列中，状态转换推进进程的执行
- 多个进程如何交替--调度+切换
  - 调度需要调度策略保证多个进程的调度公平性和执行的效率
  - 切换需要将当前进程的信息保存在PCB结构体中(比如在看书突然要去开门，就用书签记录我们看到哪)，然后把另一个进程的PCB中的东西载入到CPU中恢复另一个进程的状态。(需要汇编代码)

## 多进程图像：多进程如何组织？



## 多进程的组织：PCB+状态+队列

■ 运行 → 等待; 运行→就绪; 就绪→运行.....



- 该图称为**进程状态图**
- 它能给出进程生存期的清晰描述
- 它是认识操作系统进程管理的一个窗口

## 交替的三个部分：队列操作+调度+切换

■ 就是进程调度，一个很深刻的话题

### ■ FIFO?

- FIFO显然是公平的策略
- FIFO显然没有考虑进程执行的任务的区别

### ■ Priority?

- 优先级该怎么设定?可能会使某些进程饥饿

### 3.多个进程执行会互相影响--需要进行内存管理

- 多个进程交替执行，因此需要多个进程都在内存中。
  - 某进程访问的内存可能会影响到别的进程的东西。
- 因此为了不相互影响，需要进行**内存管理**，将多进程的地址空间分离
  - 所以**进程管理**连带**内存管理**形成了操作系统的两大部分的其中之一--**多进程图像**
- 多进程也需要**合作**，合作就需要共享数据，这些**共享的数据也需要进行访问管理**
  - 典型问题：生产者消费者问题
  - **进程应该同步**，以**合理的顺序**推进不同的进程交替执行(什么时候可以切换进程，什么时候切换是不允许的)

#### 两个合作的进程都要修改counter



### 4.总结

本课就是构建起操作系统两部分(多进程图像，文件系统图像)其中的多进程图像的框架：

如何形成多进程图像？

- 读写PCB，OS中最重要的结构，贯穿始终
- 要操作寄存器完成切换 (L10, L11, L12)
- 要写调度程序 (L13, L14)
- 要有进程同步与合作 (L16, L17)
- 要有地址映射 (L20)

#### 4.1实现多进程需要什么? (五部分)

- 保存不同进程执行时的信息--需要PCB结构体
- 微观切换：从一个进程切换到另一个需要保存什么信息，如何保存--操作寄存器
- 宏观切换：如何选择下一个切换的程序，如何设计程序状态--调度程序
- 多进程需要同步与合作--同步与合作的实现
- 多进程需要地址的分隔--地址映射

## 4.2进程管理+内存管理=多进程图像