



上海大学

SHANGHAI UNIVERSITY

操作系统（一）研讨指导

一、研讨形式和内容

小班研讨将以学生为主，报告课内外学习收获和体会、提出问题并尝试解答、开展讨论。教师的主要作用是控制研讨流程，参与点评，记录研讨情况，并给积极参与研讨者判分。要求学生主动学习，积极主动发言，课外做好充分准备。研讨题以教学大纲为主线，以主流操作系统为分析实例，与教学进度同步，理论与实践相结合。

二、研讨安排

1、地点：经管 108

2、时间：第 3/5/7/9 周，共 4 次

3、选题：<http://www.elearning.shu.edu.cn> 讨论区

4、4 名同学一组分组开展课外学习和研讨活动。推选一名组长，每次组织 1-2 名小组成员完成一道的题目的主讲，全组同学共同承担思考回答、讨论报告、设计与实现报告等任务。

5、研讨报告前需要准备的材料（PPT，源程序，运行结果，存在的问题或改进方面），规定时间内完成（6-15min）

5、研讨周周日前提交研讨报告材料（ppt，研讨报告，源代码）电子版压缩（按“研讨+周次+组长学号+组长姓名.rar”命名文件名）

六、研讨进度

9/9	不安排（无） 布置 研讨内容：操作系统的硬件接口
9/23	主题一：操作系统的硬件接口 1、多道程序的并发性的硬件支持条件实例分析 2、保护机制的实现（核心态与用户态、特权指令与非特权指令） 3、中断异常机制 4、系统调用

10/7	主题二：进程管理 1、进程信息查看和获取 2、进程同步和互斥的经典模型 3、前趋图中的同步应用实例与分析（实例/Berstein 条件）
10/21	主题二：进程管理 1、进程同步和互斥的现代模型 2、多个父子进程的管道通信的实例与析（无名管道、有名管道） 3、Linux、Unix 的信号机制（软中断）与比较（实例）
11/4	主题三：调度与死锁 1、抢占和非抢占的优先级高者优先调度 2、固定和可变的时间片轮转调度的算法实现与分析 3、实时调度算法的讨论与分析多机调度算法的讨论与分析 4、预防死锁的破坏互斥条件和资源环路链 5、银行家算法的实现 6、死锁检测算法的实现（资源分配图的简化）



上海大学

SHANGHAI UNIVERSITY

操作系统（一）教学研讨参考题

一、研讨形式和内容

研讨采用小班形式，学生分组选择研讨题、开展课外学习、完成研讨准备、参与研讨活动。研讨题以教学大纲为主线、强化教学内容、拓展课外知识、理论与实践相结合。

二、研讨要求

- 1、研讨报告前需要准备的材料（PPT、源代码、演示程序等）
- 2、每个题目的研讨时间无特殊说明，限 ≤ 9 分钟（PPT不超过12页）
- 3、研讨周周日前提交研讨的电子版材料（含PPT、研讨报告和源代码），按“题号+组号.rar”命名压缩包，在截止时间前上传到<http://www.elearning.shu.edu.cn>

三、研讨安排

9/23	主题一：操作系统的硬件接口与Linux系统
10/7	主题二：进程管理
10/21	主题二：进程管理
11/4	主题三：调度与死锁

主题一：操作系统的硬件接口与 Linux 系统（第三周）

- 1、简述微处理器的发展里程，简述 80386 架构的主要寄存器组，举例说明指令执行的一般过程。
- 2、简述中断的基本概念，说明 intel 80386 架构中的中断类型，比较 x86 架构中实模式和保护模式中的中断机制。
- 3、在 PC 体系结构中，起到中断控制器的芯片是 8259a，请描述其工作原理。
- 4、PC 体系结构中，用来提供时钟中断的外部芯片是什么，简介它的工作原理。时钟中断对于操作系统起到什么作用。
- 5、描述 x86 中实模式和保护模式的寻址区别，并说明在这两种模式中逻辑地址、线性地址、和物理地址的关系。IA32 中提供了哪些机制来帮助实现安全可靠的多任务多用户的操作系统。
- 6、简述 Linux 的启动过程的关键步骤，并且和 Windows 系统作比较。
- 7、什么是 Shell？简述 Shell 运行的过程，并举例说明。Shell 脚本与一般高级语言程序有何不同？
- 8、Linux/Unix 的进程家族树的概念介绍？演示 Linux 操作系统下面查看进程树的命令？
- 9、请归纳并给出 Linux 环境以下信息获取的方法：读取指定进程的进程信息、获取总线信息、获取内核所使用设备信息、获取系统所引入的 NFS 文件系统信息、获取 IDE 设备信息、获取 IRQ 信息、获取网络信息、获取 SCSI 设备信息、获取系统信息和获取 TTY 设备信息。
- 10、Linux 环境下进程信息查看和作业控制：虚拟终端 1（root 用户）利用 vi 新建 f1.txt 文件，将 vi 作业挂起；后台启动 find 进程；显示当前所有作业和进程号；显示所有进程含优先级的进程详细信息，并画出进程家族树后，中止 find 进程的后台运行。虚拟终端 2（student 用户）启用 vi 新建 f2.txt 文件，其优先级为 5，后台启动 find 进程，显示含优先级的进程详细信息；再将 student 用户进程的优先数改为 -5，显示含优先级的进程详细信息。写出相应操作步骤。

主题二：进程管理（第五周+第七周）

1、Linux 环境下用户间非实时通信

(1) root 用户向 student 用户发送一封邮件，内容为“请尽快上交实验一：Linux 的桌面环境及基本 shell 命令的实验报告。”，主题是“催交实验报告”，并请转发给 hellen 用户。student 用户阅读邮件后，回复“知道”，并将发来的信件保存为文件 vipletter 后删除该邮件。最后退出 mail.，检查 mbox 文件，并分析说明检查结果。写出相应操作步骤。

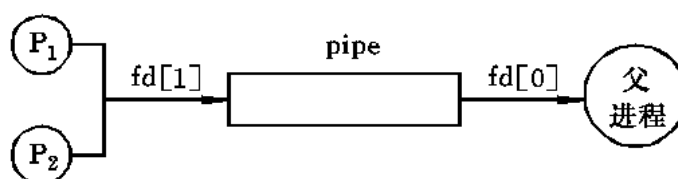
(2) 假定在进程通信中我们使用一个类似邮箱的机制。当一个进程试图往一个满箱放东西的时候，或者从一个空箱拿东西的时候，该进程并不阻塞。而是得到一个错误返回码。该进程立即重复上述过程（往邮箱里放东西或者从邮箱里拿东西），直到成功为止。请问这种机制会造成竞争吗？为什么？

2、无名管道通信的应用

(1) 利用一条管道命令实现统计指定目录中子目录和文件的总数，并将统计结果保存到指定的文件中。

(2) 利用一条管道命令实现将一封来自键盘的邮件发给用户 USER1、USER2，邮件内容自定。

(3) 管道通信的编程题：无名管道通信的实例分析。编写一算法，建立一个管道。同时，父进程生成子进程 P1、P2，这两个子进程分别向管道中写入各自的字符串，父进程读出它们，其原理如下图所示。



下面题干供第 3-5 题参考：

所谓“读者-写者问题”是指保证一个 Writer 进程必须与其他进程互斥地访问共享对象的同步问题。为实现 Reader 与 Writer 进程间在读或写时的互斥而设置了一个互斥信号量 Wmutex。另外，再设置一个整型变量 Readcount 表示正

在读的进程数目。由于只要有一个 Reader 进程在读，便不允许 Writer 进程去写。因此，仅当 Readcount=0，表示尚无 Reader 进程在读时，Reader 进程才需要执行 P(Wmutex) 操作。若 P(Wmutex) 操作成功，Reader 进程便可去读，相应地，做 Readcount+1 操作。同理，仅当 Reader 进程在执行了 Readcount 减 1 操作后其值为 0 时，才须执行 V(Wmutex) 操作，以便让 Writer 进程写。又因为 Readcount 是一个可被多个 Reader 进程访问的临界资源，因此，应该为它设置一个互斥信号量 rmutex。

3、“读者-写者问题”扩展一：利用记录型信号量集解决读者-写者问题。

4、“读者-写者问题”扩展二：利用记录型信号量分别给出“读写无优先、写者优先”问题的同步算法；

5、“读者-写者问题”扩展三：利用记录型信号量集分别给出“读写无优先、写者优先”问题的同步算法。

下面题干供第 6-7 题参考：

睡觉理发师问题（如图所示）：一个理发店有一个理发师，一张理发椅子，多张等待椅子（可以假定为 $n > 1$ 张）。当没有顾客的时候，理发师就睡觉。当一个顾客来到店里是，如果理发师在睡觉，则叫醒理发师；如果理发师正在给人理发，则坐在椅子上等待。



6、回答：题干所述安排有发生竞争的可能吗？如果没有，请予以证明。如果有，什么情况下会发生竞争？利用记录型信号量给出同步算法：如果等待的椅子都满了，则顾客仍然在理发店等待理发；

7、回答：题干所述安排有发生竞争的可能吗？如果没有，请予以证明。如果有，什么情况下会发生竞争？利用记录型信号量给出同步算法：如果等待的椅子都满

了，则顾客就离开理发店而不理发了。

8、前趋图中的同步应用

(1) 设有三个进程，input 进程、compute 进程和 output 进程；它们通过共享一个缓冲区 buf 的合作关系如下：

(a) input 进程每次输入数据后，把数据送到 buf, 供 compute 进程计算和 output 进程打印；

(b) compute 进程每次从 buf 取出已输入的可计算的数据进行计算，并当 output 进程把输入数据打印完成后，把计算结果送入 buf 供 output 进程打印；

(c) output 进程每次把 buf 中要打印的数据在打印机上输出。

请给出能正确实现这三个进程同步关系的的进程同步算法。

(2) 求一元二次方程 $ax^2+bx+c=0$ 的根： $x_1 = (-b+\sqrt{b^2 - 4ac}) / (2a)$;
 $x_2 = (-b-\sqrt{b^2 - 4ac}) / (2a)$ 。现把计算分解为如下 6 个可并发执行的过程：请分析这 6 个过程的同步关系，画出前趋图，并用 P、V 操作实现它们的同步。

P1: $d_1 = 4ac$;
P2: $d_2 = b^2$;
P3: $d_3 = 2a$;
P4: $d_4 = \sqrt{d_2 - d_1}$;
P5: $x_1 = (-b+d_4) / d_3$;
P6: $x_2 = (-b-d_4) / d_3$ 。

9、Linux/Unix 的信号机制（软中断）

阅读下面的程序：（1）给出程序执行的结果，并指出程序执行所对应的进程数。

（2）在 Linux 环境下，给出输入编辑源程序（设文件名为 prg2.c）、编译和执行此目标程序的 shell 命令。

```
#include <stdio.h>
#include <signal.h>
func( )
{
    printf( "A signal has been received\n" );
}
```

```

main()
{
    int i ;
    signal(18,func);
    printf( 'Processing signal function has been set\n' );
    i = fork( );
    if (i == 0)
    { printf( "Child process is now executing!\n" );
      kill(getppid( ),18);
      sleep(10);
      printf( "child finished\n" );
      exit();
    }
    sleep(10);
    printf( "This is the parent\n" );
    kill(i,18);
    wait( );
    printf( 'parent finished\n' );
    exit( );
}

```

10、Linux/Unix 的信号机制（软中断）。

阅读下面的程序：（1）给出程序执行的结果；若删除含有“第二次设置处理方式语句”标记的语句，给出程序执行的结果，请比较并说明程序执行的结果。（2）指出程序执行所对应的进程数。

```

#include <signal.h>
void func( )
{ printf( "This is func\n" ); }
main( )
{ int i , j ;
  signal(SIGUSR1,func); /* 第一次设置处理方式语句 */
  while (i=fork( )== -1) ;
  if (i == 0)
  { sleep(5);
    signal(SIGUSR1,func); /* 第二次设置处理方式语句 */
    printf( "This is the first child\n" );
    sleep(5);
    exit( );
  }
  while ( j = fork( ) == -1 ) ;
  if ( j ==0)
  { sleep(3);
    printf( "This is the second child\n" );
  }
}

```



```
    kill(i , SIGUSR1);  
    sleep(3);  
    exit( );  
}  
printf( "This is the parent\n" );  
kill(j , SIGUSR1);  
wait( );  
kill( i , SIGUSR1);  
wait( );  
exit( );  
}
```

主题三：调度与死锁（第九周）

1、列举 Linux 和 Windows 中使用的调度算法。

2、设有 5 个进程 P1, P2, P3, P4, P5，在 0 时刻到达，它们需要的 CPU 时间和优先级如下：

进程	CPU 时间	优先级
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

- 分别画出使用 FCFS、SJF、非抢占式优先级（小优先级数值代表大优先级）以及 RR（时间片为 1）调度算法的 4 个 Gantt 图，以说明这些算法的执行时间。
- 计算 a. 中每个进程的周转时间。
- 计算 a. 中每个进程的等待时间。
- a. 中哪种调度算法的平均等待时间最少？

3、设有 3 个进程，到达时间和所需 CPU 时间见下表，采用非抢占式调度。

进程	到达时间	所需 CPU 时间
P1	0.0	8
P2	0.4	4
P3	1.0	1

- 采用 FCFS 算法时这组进程的平均周转时间。
 - 采用 SJF 算法时这组进程的平均周转时间。
 - 如果调度在 CPU 空闲 1 个时间单位后才开始，b 的结果会否有所改善？
- 4、如何评价调度算法的性能？谈谈下面每对指标间的关系。
- CPU 利用率和响应时间
 - 平均周转时间和最大等待时间
 - I/O 设备利用率和 CPU 利用率
- 5、分析比较普通调度和实时调度，哪些常用的调度算法可用于实时调度？
- 6、什么是活锁？如何产生的？是否可以解决？
- 7、以现有常用操作系统或者应用为例谈谈如何预防死锁？
- 8、死锁恢复的方法有哪些？如何提高恢复的效率？
- 9、银行家算法的实质是什么？如何理解安全状态和非安全状态？
- 10、从资源管理的角度看，系统怎样才能保证不死锁？