

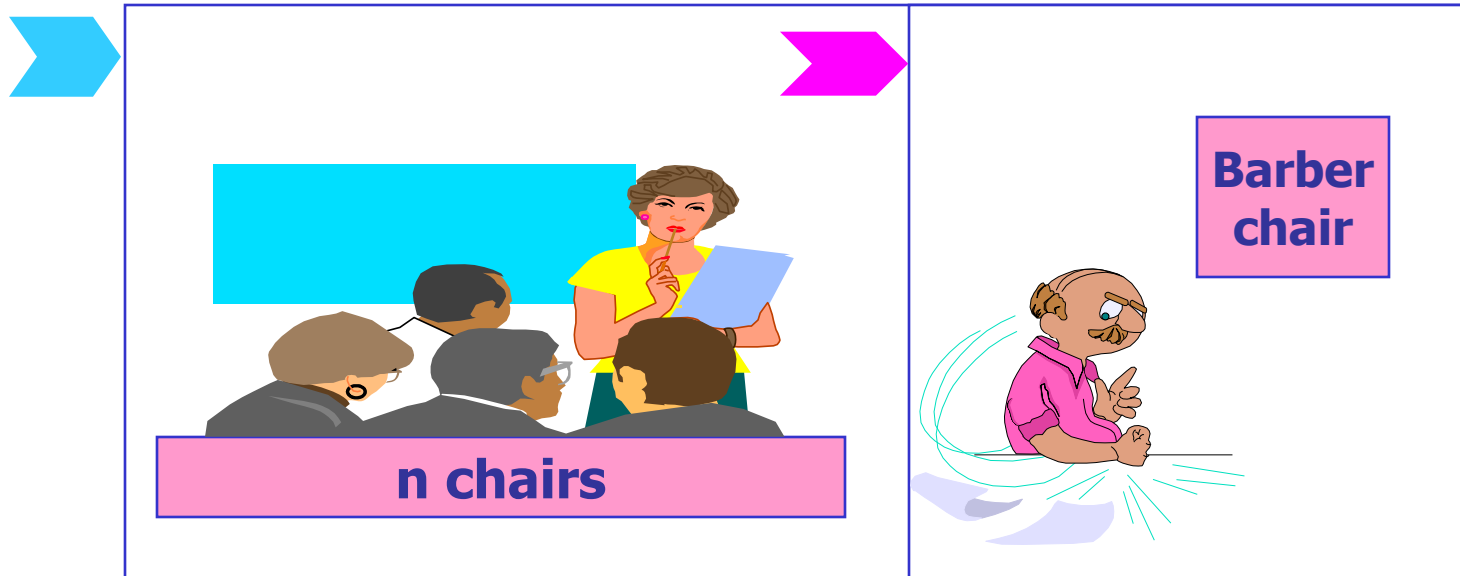
APPLIED OPERATING SYSTEM CONCEPTS



**Homework
and Experiments**

The Sleeping-Barber Problem

- Barbershop consists of a waiting room with **n** chairs, and a barber room with **one** barber chair
- No customer, barber goes to sleep
- If barber is asleep, new customer wakes up him





```
Var empty, full, mutex: semaphore:=n,0,1;
```

```
Begin
```

```
  parbegin
```

```
    Customer:
```

```
      begin
```

```
        repeat
```

```
          wait(empty);
```

```
          wait(mutex);
```

```
            find a seat;
```

```
          signal(mutex);
```

```
          signal(full);
```

```
        until false;
```

```
      end
```

```
    parend
```

```
end
```

```
  Barber:
```

```
    begin
```

```
      repeat
```

```
        wait(full);
```

```
        signal(empty);
```

```
          cutting;
```

```
      until false;
```

```
    end
```



```
Var int waiting=0;
```

```
semaphore customers, barber, mutex;
```

```
customers=0; barber=0; mutex=1;
```

```
{parbegin
```

```
  Customer:
```

```
    {wait(mutex);  
      if (waiting<n)  
        { waiting++  
          signal(customers);  
          signal(mutex);  
          wait(barber) ; }  
    }  
  else  
    singal(mutex);  
}
```

```
parend
```

```
}
```

```
  Barber:
```

```
    {repeat  
      wait(customers);  
      wait(mutex);  
      waiting--;  
      signal(mutex);  
      cutting;  
      signal(barber);  
    }  
  until false;  
}
```



实验一 CPU Scheduling

一、实验内容

选择或者自行设计一个调度算法，实现处理机调度

二、实验目的

多道系统中，当就绪进程数大于处理机数时，须按照某种策略决定哪些进程优先占用处理机。本实验模拟实现处理机调度，加深了解处理机调度的工作过程

三、实验题目

- 1、设计一个按优先权调度算法实现处理机调度的程序
- 2、设计一个按时间片轮转实现处理机调度的程序

要求与提示

构建 PCB，内容至少涵盖：

进程名 /PID；

要求运行时间（单位时间）；


优先权；

状态：

PCB 指针；

- 1、可随机输入若干进程，并按优先权排序
- 2、采用动态优先权调度，从就绪队首选进程运行：
优先权 -1/ 要求运行时间 -1
要求运行时间为 0 时，撤销该进程
- 3、重新排序，进行下轮调度

要求与提示

- 
- 最好采用图形界面
 - 可动态增加进程
 - 规定道数，设置后备队列和挂起状态
 - 如果内存中进程数少于规定道数，可自动从后备队列通过作业调度选择一作业进入，作业调度算法可自行选择
 - 被挂起进程入挂起队列，设置解挂功能用于将指定挂起进程解挂并入就绪队列
 - 每次调度后，显示各进程状态。



实验二 Allocation & Reclaim

一、实验内容

主存储器空间的分配和回收

二、实验目的

帮助了解在不同的存储管理方式下，应怎样实现主存空间的分配和回收

三、实验题目

在可变分区管理方式下，采用最先适应算法实现主存空间的分配和回收



要求与提示

1、自行假设主存空间大小，预设操作系统所占大小并构造未分分区表；

表目内容：起址、长度、状态（未分 / 空表目）

2、结合实验一，PCB 增加为：

{PID，要求运行时间，优先权，状态，所需主存大小，主存起始位置，PCB 指针}

3、采用最先适应算法分配主存空间；

4、进程完成后，回收主存，并与相邻空闲分区合并。



Report

- 程序中使用的数据结构及符号说明
- 提交一份源程序并附注释
- 针对输入的数据，给出程序执行结果



实验三 Virtual Memory

一、实验内容

模拟分页虚存管理中硬件的地址转换和缺页中断，以及选择页面调度算法处理缺页中断

二、实验目的

帮助理解在分页式存储管理中，如何实现主存空间的分配和回收

三、实验题目

模拟分页式存储管理中硬件的地址转换和产生缺页中断，并利用 **LRU** 页面调度算法处理缺页中断



要求与提示

1、内存分页管理，设每个页框

128B

2、每个作业创建一个页表，页表格式：

页号，标志，页框号，磁盘位置

3、页表初始化（见右表）

4、定义一个数组，保存作业已在主存的页号

5、作业执行时，输入一组逻辑地址

页号	标志	页框号	外存地址
0	1	5	011
1	1	8	012
2	1	9	013
3	1	1	021
4	0		022
5	0		023
6	0		121



要求与提示

6、将逻辑地址转换为 **(P,d)** 格式，并判断地址的合法性；
非法：提示越界；**合法**：转下一步

7、按 **(P,d)** 访问页表

- **对应标志为 0：**

按 **LRU** 算法置换，修改页表，重新执行指令

- **对应标志为 1：**

按下列算式转换为绝对地址，并输出

绝对地址 = 页框号 * 页长

+d

8、进程完成后 回收主存



Report

- 程序中使用的数据结构及符号说明
- 提交一份源程序并附注释
- 针对输入的数据，给出程序执行结果