

2025春《操作系统原理课程设计》

实验题目要求及评分标准

目录

2025 春《操作系统原理课程设计》	1
题目一：进程调度：多级队列调度算法	1
题目二：牙医就诊模拟程序（线程的同步/互斥）	3
题目三：自定义 Linux 系统调用	4
附录：题目二伪代码示例（基于 Linux）（仅供参考！）	5
验收规则与评分标准	7

题目一：进程调度：多级队列调度算法

实验内容：

设进程优先级RQ分为RQ1、RQ2和RQ3；

RQ1>RQ2>RQ3 （优先级：1>2>3）；

RQ1队列：采用轮转法，时间片q=4；每个进程最多可连续轮2次；

RQ2队列：采用轮转法，时间片q=8；每个进程只能轮转1次；

RQ3队列：采用短进程(还有多久结束)优先调度，每个进程运行直至结束。

测试数据如下：RQ1: P1-P7, RQ2: P8-P10

进程	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
需运行时间	16	21	14	30	7	21	18	10	27	14
已等待时间	6	5	4	3	2	1	2	3	4	5

RQ3队列初始为空。

实现描述（参考）：

```
typedef struct tag_pcb
{
    char name[8];
    int need;//需运行的时间
    int turn;//周转时间
    struct tag_pcb *next;
} PCB;
PCB * RQ1,*RQ2, *RQ3, *Finish;
int clock=0; //时钟
```

```

main ( )
{
    输入RQ1;
    输入RQ2; (最好从文件读入)
    while (条件? 每个进程最多轮转2次)
    {
        从RQ1中选取一进程Pi准备运行;
        记录其本次实际运行的时间t(不超过时间片);
        clock+=t; //表示Pi运行t;
        if (Pi完成) 计算其turn;
        否则
            下一次轮转 或 Pi加入到RQ2队尾? ;
    }
    while (条件? )
    {
        从RQ2中选取一进程Pi准备运行;
        记录其本次实际运行的时间t(不超过时间片)
        clock+=t;
        if (Pi完成) 计算其turn;
        否则
            Pi加入到RQ3队尾?
    }

    while (条件? )
    {
        从RQ3中选取一进程Pi准备运行;
        记录其本次实际运行的时间t;
        clock+=t;
        计算Pi的turn;
    }

    输出每个进程的周转时间;
}

```

其他要求:

- ✓ 在Linux下用C/C++/STL C++实现。
- ✓ 一些相对独立的功能，如链表的操作，应该根据逻辑关系在不同的源代码里面进行必要的封装；用makefile组建工程。

题目二：牙医就诊模拟程序（线程的同步/互斥）

题目场景：

某医院有一名牙医，某一天，他在诊室对10名患者进行患牙医治。医生的诊室内部每次只能接待1名患者。在医生诊室部的走廊里，有6把椅子，有1名护士负责对坐在椅子上的患者进行叫号。

规则是：

1. 患者需要首先坐在走廊里的椅子上按照顺序等待，如果没有空椅子，只能等其他患者腾出椅子。
2. 护士负责从坐在椅子上排队等候的患者中顺序引领一名患者进入诊室；如果没有人等待，护士等候；如果诊室中已有患者就诊，则须等就诊患者结束就诊后才能引领下一位患者。
3. 当医生诊治完一名患者后，患者方可离开；同时，医生等待护士引领下一位患者进入；如果没有患者进入，医生等待。

实验内容：

- 在Linux操作系统上，利用pthread API提供的信号量机制，编写应用程序实现该问题的解决方案。
- 可在一个进程里完成：从1个主线程开始，为每位患者生成1个线程（代表患者的动作），为医生、护士分别生成由1个线程（分别代表医生、护士的动作）。
- 通过在程序中合适的位置设置打印信息，打印到屏幕上或文本文件中，以显示程序的正确执行过程。
- 题目中的就诊操作：可用函数模拟（例如：**sleep**一段时间，3~5秒钟为宜）。
- 要求：输出内容清晰明了，输出格式简单统一，能够很直观地看出是哪个线程输出的什么内容。

技术思路（仅供参考）：

- a) 3把椅子构成一个有序缓冲区（队列），患者需要对该队列进行竞争。
- b) 每一位患者，在排队时，需要和护士线程进行同步，此时患者线程和护士线程之间构成一个可变共享缓冲区问题。
- c) 护士叫号时，需要和医生线程同步，此时护士线程和医生线程之间构成一个单一缓冲区共享问题。
- d) 就诊患者必须等待医生诊治完毕以后方可离开，此时二者之间构成一个合作进程的执行次序问题。

其他要求：

- ✓ 本题的编程可在Linux/Windows下实现，编程示例可参见附录。
编程语言要求采C/C++/C++STL，推荐（不强迫）使用C++STL，可以很方便地实现缓冲区的各种操作。

题目三：自定义 Linux 系统调用

实验内容：

- 将题目二中所用到的用于模拟就诊操作的随机sleep函数改写成一个新的Linux系统调用，并在题目二中使用；
- 假设该系统调用最终可供用户使用的函数是diag(int number)，要求输入参数number是自己的学号(2023*****)，diag函数的功能定义为：以number为随机数种子，产生一个1000到5000之间的随机数字X；然后让程序等待X毫秒。

技术思路（仅供参考）：

- a) 增加一个新的 Linux 系统调用的方法参见课本 3.5.2~3.5.4 节（P70-73）或上课的课件（或网上的参考资料）。
- b) 随机数的产生、程序等待函数的调用，参见Linux C语言函数库（自行学习）。

因为大家的Linux版本不尽相同，所以做法的大致步骤是：

- 1、修改Linux内核某文件，将你实现的函数名进行注册。
- 2、重新编译内核，使得上述修改生效。
- 3、编写程序，调用你的内核函数，即系统调用。

注意：本题可能涉及Linux内核编译，做该题目时，可对自己的关键步骤进行截屏，以证实自己的过程。

附录：题目二伪代码示例（基于 Linux）（仅供参考!）

（该伪代码仅供同学们理解思路的参考，并不代表唯一正确的解法；请同学们认真理解题意，然后设计自己的算法求解；**算法不唯一!**）

初始化操作：

信号量数据类型：sem_t

sem_init(&s,1,1);//初始化信号量 s 为 1

sem_init(&s,0, SIZE);//初始化计数信号量 s 为 SIZE

sem_init(&s,0,0);//初始信号量 s 为 0

P 操作： p(s): sem_wait(&s)

V 操作： v(s): sem_post(&s)

获取当前线程 id 号：pthread_self();

创建线程：pthread_create()

必要的头文件：

#include <pthread.h>

#include <semaphore.h>

#include <stdio.h>

int main()//主函数示例

{

可能的变量初始化及信号量初始化;;

初始化缓冲区;

创建 1 个医生线程、1 个护士线程和 10 个患者线程;

让每个线程在主线程 main 执行前全部执行完毕(用 pthread_join 实现线程阻塞);

}

void *pat()//患者函数示例

{

P1 操作询问是否有空椅子;

坐椅子排队; //队列互斥

V2 操作产生一名等待者;

P5 操作等待就诊完成;

患者离开;

return;

}

void *nur()//护士函数示例

```
{
    while (true)
    {
        P3 操作询问医生是否有空;
        P2 操作查询是否有患者等待;
        顺序叫起一名患者，腾出一把椅子; //队列互斥
        V1 操作产生一把空椅子;
        V4 操作通知医生一名患者进入;
    }
    return;
}
```

void *doc()//医生函数示例

```
{
    while (true)
    {
        P4 操作询问是否有患者进入;
        为患者进行诊治（随机 sleep 函数代替）;
        V5 信号通知患者诊治结束;
        V3 操作通知护士此时无患者就诊;
    }
    return;
}
```

问题：

- 1、如何改进程序，实现对 10 名患者的计数？ 当 10 名患者全部就诊完毕后，程序如何结束？
- 2、如果没有 P5/V5 操作，会有什么状况发生？如何在打印信息中体现出来？

验收规则与评分标准

1. 题目检查

(1) 遵循《课程设计报告模板》里的说明，要有足够的实验过程、痕迹。

(2) 不得私自修改题目。

(3) 每个人都必须单独进行验收，每人有且仅有1次验收机会。

在用C/C++/STL C++实现，一些相对独立的功能，应该根据逻辑关系在不同的源代码里面进行必要的封装。

检查过程认定为非自主完成（如抄袭、雷同、不能解释功能意义等）者，本课程成绩直接不及格。

2. 报告撰写

格式：封面、正文、字体等格式遵守《课程设计报告模板》；

打印：A4纸，双面打印；（指导教师：董理君）

3. 报告提交

2025年6月5日(含)之前，集体提交纸质版。

以班级为单位，将每位同学的全部资料（程序代码、课设报告等）进行电子版归档，每人以自己的“班号-姓名”（如“192231-霍元甲”）建立一个目录，目录中存放自己的资料。以压缩包的形式发送到 homework.dlj@qq.com。

如果个别同学需要单独提交，也请遵循以上命名规范，将全部资料（程序代码、课设报告等）做一个压缩包，班号-姓名，发送至homework.dlj@qq.com，并在邮件主题上注明班级姓名。

特别注意：每位同学资料不得超过8M！（不需要代码编译产生的中间文件，也不要文档中大量使用截屏！）

评分标准

按照满分100分计算，其中：

过程评分50分：（三个题目各占1/3的比例）

实验基本操作：20分

实验功能正确：20分

实验设计优化（改进、扩展）：10分

报告考核50分：

报告内容充实：20分

实验结果分析：15分

报告格式规范：15分