操作系统课程设计

项目文档

刘道畅 1352955

靳鑫 1352831

刘林青 1354361

目录

第一部分

第二部分

第三部分

第一部分

一、项目背景

在多道程序环境下，主存中有着多个进程，其数目往往多于处理机数目，要使这多个进程能够并发地执行，这就要求系统能按某种算法，动态地把处理机分配给就绪队列中的一个进程，使之执行。分配处理机的任务是由处理机调度程序完成的。由于处理机是最重要的计算机资源，提高处理机的利用率及改善系统必（吞吐量、响应时间），在很大程度上取决于处理机调度性能的好坏，因而，处理机调度便成为操作系统设计的中心问题之一。

二、项目需求

用C++编程实现操作模拟操作系统进程调度子系统的基本功能；实现多级反馈队列对进程进行的调度过程。

三、设计方案

3.1设计思路

为了描述和管制进程的运行，系统为每个进程定义了一个数据结构——进程控制块PCB(Process Control Block),PCB中记录了操作系统所需的、用于描述进程的当前情况以及控制进程运行的全部信息，系统总是通过PCB对进程进行控制，即系统是唯一根据进程的PCB来感知进程的存在的，PCB是进程存在的惟一标志。本次课程设计用结构体PROCESS代替PCB的功能。

3.2 设计方法

3.2.1数据结构

设计一种数据结构PROCESS，提供关于进程的一些有效信息：进程优先级，完成进程所需时间，进程进入时间。

```
typedef struct s_proc{
    int arr;
    int remaining;
    int priority;
}PROCESS;
```

3.2.2 算法设计

与其他的调度算法不同，多级反馈队列调度算法允许进程在队列之间移动。其主要思想是根据不同的CPU区间的特点以区分进程。如果进程使用过多CPU时间，那么它会被转移到更低优先级队列。这种方案将I/O约束和交互进程留在更高优先级队列。此外，在较低优先级队列中等待时间过长的进程会被转移到更高优先级队列。这种形式的老化阻止饥饿的发生。

在系统中设置3个就绪队列，每个队列对应一个优先级，第一个队列的优先级最高，第二队列次之，第三个队列的优先级最低。各就绪队列中的进程的运行时间片不同，第一优先级队列的时间片为5，第二优先级队列的时间片为10，第三优先级队列采用先到先服务算法。进程并非总是固定在某一队列中，新进程进入系统后，被存放在第一个队列的末尾。如果某个进程在规定的时间片内没有完成工作，则把它转入到

下一个队列的末尾，直至进入最后一个队列。系统先运行第一个队列中的进程。当第一队列为空时，才运行第二个队列中的进程。依此类推，仅当前面所有的队列都为空时，才运行最后一个队列中的进程。当处理器正在第i个队列中为某个进程服务时，又有新进程进入优先级最高的队列，则此新进程要抢占正在运行进程的处理器，即由调度程序把正在运行的进程放回第i队列的末尾，把处理器分配给新到的高优先级进程。除最低优先权队列外的所有其他队列，均采用相同的进程调度算法即按时间片轮转的FIFO（先进先出）算法。最后一个队列中的进程按FCFS策略进行调度。

四、详细设计

4.1数据结构

定义结构体PROCESS（代替进程控制块PCB）

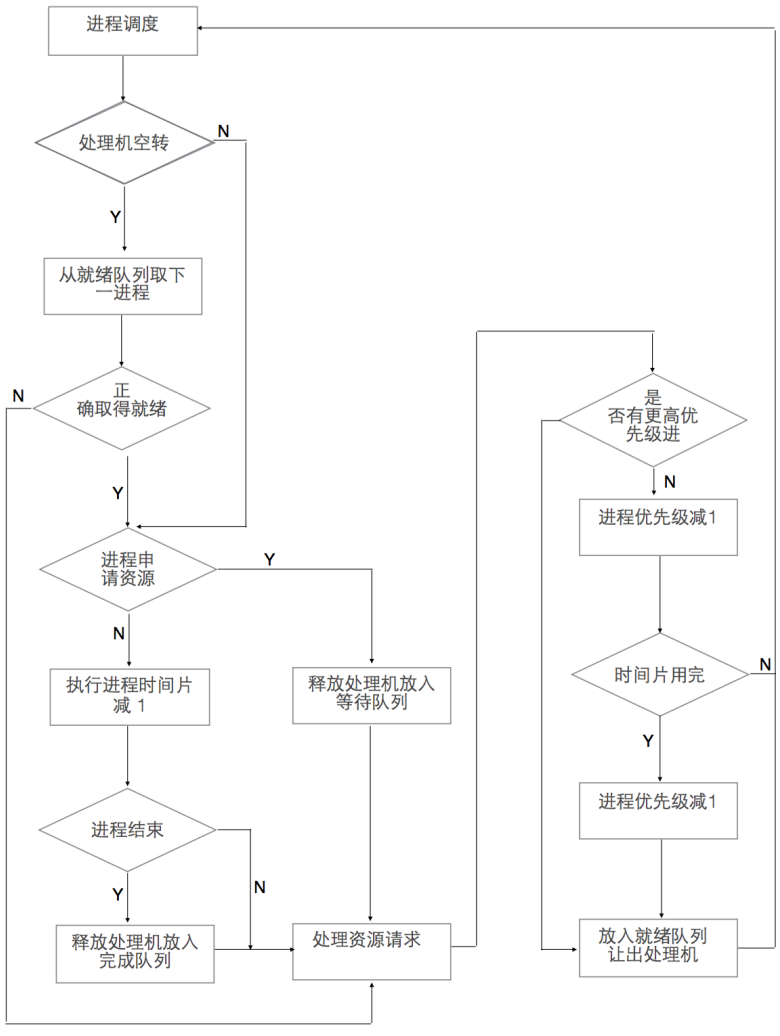
```
typedef struct s_proc{
     int arr;//进程进入时间
     int remaining;//完成进程所需时间
     int priority;//进程优先级
}PROCESS;
```

4.2 功能划分

模拟进程调度有以下基本功能

- 创建进程：系统为每一个进程定义有效信息
- 进程调度：采用多级反馈队列调度方式
- 输出状态：输出显示进程进入、进程切换、进程运行时间等各种信息

4.3 算法流程图

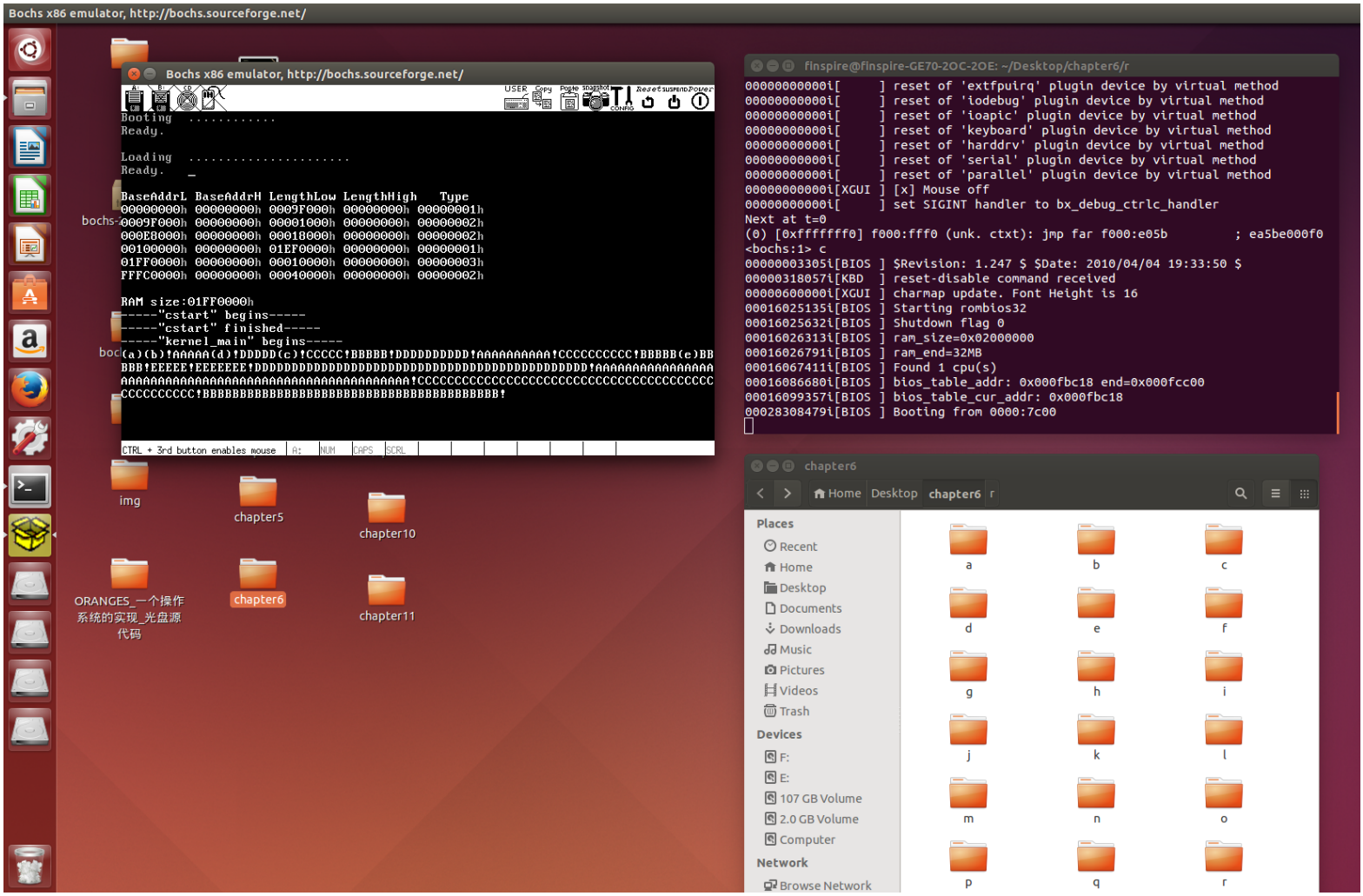


4.4函数功能定义

void schedual() 调度算法
void checkArr() 将新到来的进程添加进就绪队列
void priorNormalize()将就绪队列中的进程按优先级链接

五、项目实现及说明

5.1 实现截图



5.2 说明

- （a）代表a 进程进入
- ！代表进程切换

第二部分

一、项目背景

计算机主要有两个主要任务：I/O操作与计算机处理。在许多情况下，主要任务是I/O操作，而计算机处理只是附带的。

二、项目需求

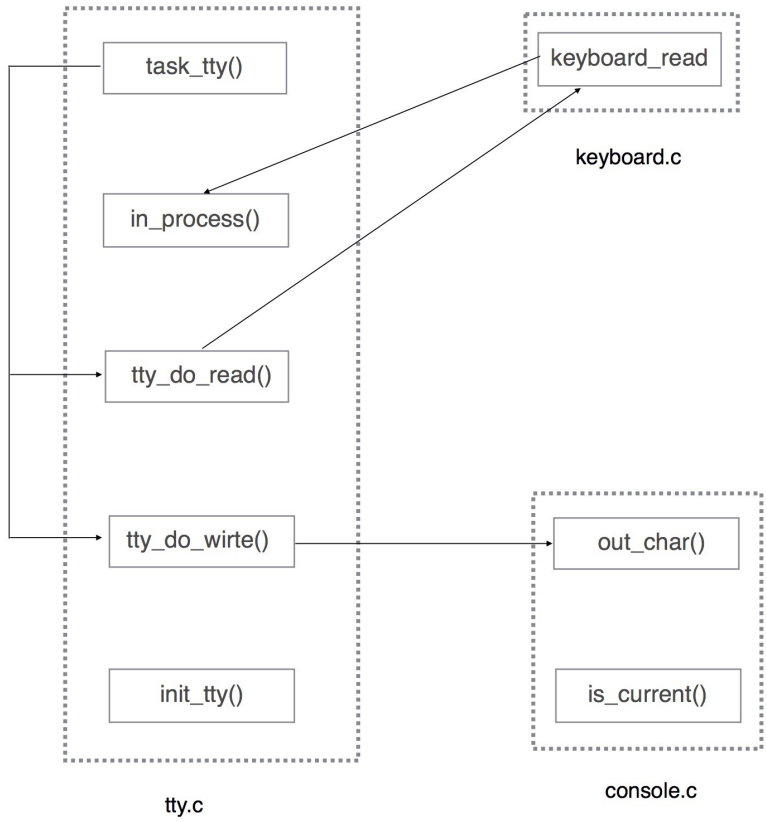
用C++编程实现操作模拟操作系统I/O系统的基本功能；按键之后会返回相应的返回值。

三、设计方案

3.1程序流程

在task_tty()中，通过循环来处理每一个TTY的读和写操作，读和写操作全部放在tty_do_read()和tty_do_write()两个函数中，这样使得task_tty()很简洁，而且逻辑清晰。读操作会调用keyboard_read()，此时会多一个参数；写操作会调用out_char(),他会将字符写入指定的console。

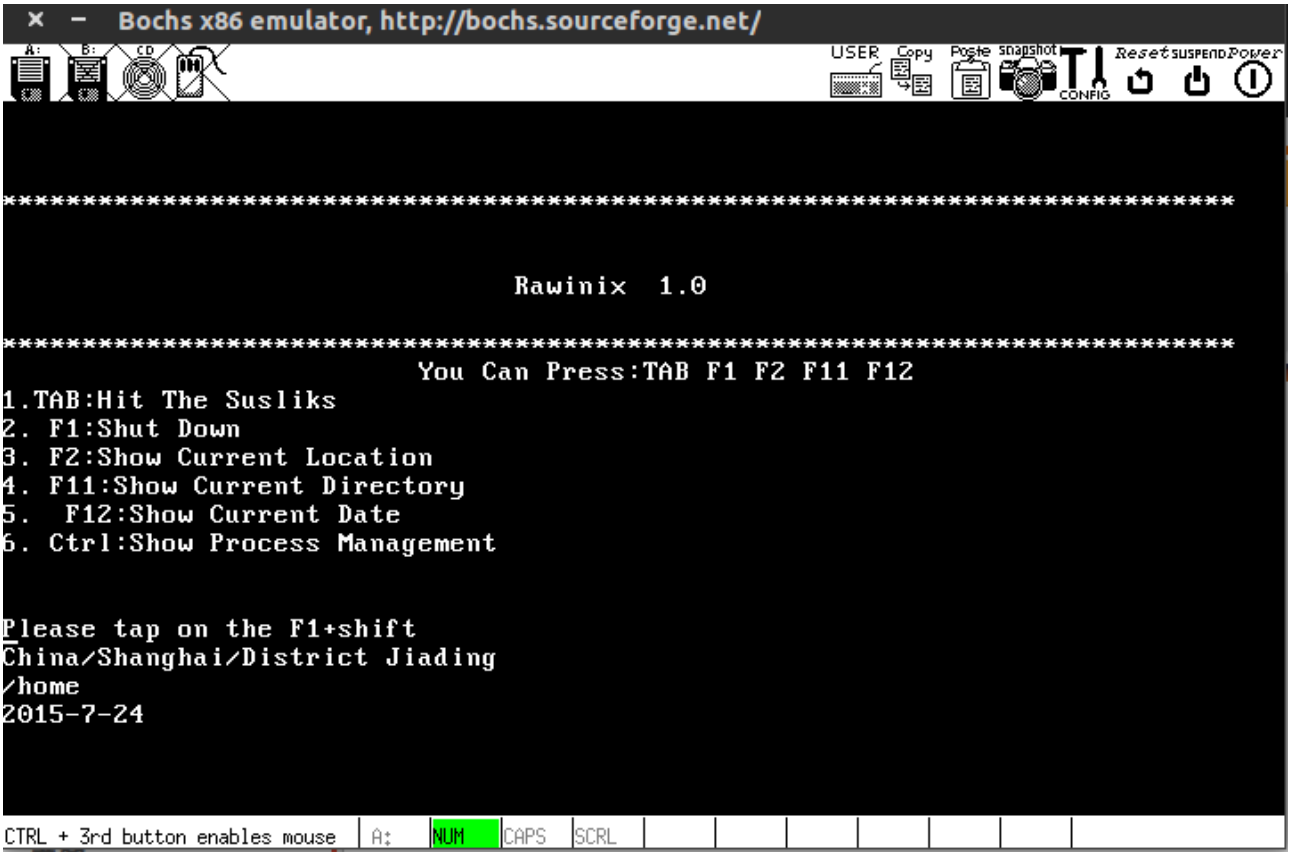
3.2 任务代码示意



3.3 函数功能定义

PRIVATE void init_tty(TTY* p_tty);
PRIVATE void tty_do_read(TTY* p_tty);
PRIVATE void tty_do_write(TTY* p_tty);
PRIVATE void put_key(TTY* p_tty, u32 key);
PUBLIC void task_tty();
PUBLIC void in_process(TTY* p_tty, u32 key);

四、项目实现



第三部分

一、项目需求

用C++编程实现一个用户级应用—小游戏

二、设计方案

2.1 游戏说明（打地鼠）

地鼠随机出现在键盘的某个位置，用其左右的字母来表示，玩家需要尽可能快地击打10只地鼠，结束时告知玩家分数。

三、项目实现