

理论课作业5:第六章练习题

作业内容:

1. 英文版operating system concepts,第7版, P231 6.1, 6.3, 6.9

6.1 第一个著名的正确解决两个进程的临界区问题的软件方法是由 Dekker 设计的。两个进程 P_0 和

习 题 • 199 •

P_1 共享以下变量:

```
boolean flag[2]; /*initially false*/
```

```
int turn;
```

进程 P_i ($i=0$ 或 1) 的结构见图 6.25, 另一个进程为 P_j ($j=0$ 或 1)。试证明这个算法满足临界区问题的所有三个要求。

```
do {  
    flag[i] = TRUE;  
    while (flag[j]) {  
        if (turn == j) {  
            flag[i] = false;  
            while (turn == j)  
                ; // do nothing  
            flag[i] = TRUE;  
        }  
    }  
    // critical section  
    turn = j;  
    flag[i] = FALSE;  
    // remainder section  
} while (TRUE);
```

图 6.25 Dekker 算法中的进程 P_i 结构

- 6.3 术语忙等的含义是什么？操作系统中其他类型的等待有哪些？忙等能否完全避免？为什么？
- 6.4 试解释为什么自旋锁对单处理器系统不合适而对多处理器系统合适。
- 6.5 试解释为什么在单处理器系统上通过禁止中断实现同步原语方法不适用于用户级程序。
- 6.6 试解释为什么通过禁止中断实现同步原语不适合于多处理器系统。
- 6.7 描述如何用 `swap()` 指令来实现互斥并满足有限等待要求。
- 6.8 服务器可设计成限制打开的连接数。例如，某个服务器可能需要在某个时刻只能打开 N 个 `Socket` 连接。一旦有 N 个连接，那么服务器就不再接受新的连接请求，直到有现有连接释放为止。解释如何采用信号量来限制并发连接的数量。
- 6.9 试说明如果 `wait()` 和 `signal()` 操作不是原子化操作，那么互斥可能是不稳定的。

提交要求：

1. 截止时间: 2025.04.28 00:00
2. 作业格式：基于群文件中的作业模板编辑，完成后要求转化为pdf格式提交，命名格式为“作业5-学号-姓名.pdf”
3. 提交邮箱：sysuos25t@163.com