**《计算机操作系统》2018-2019学年第二学期期中考试试卷（Ａ卷）**

**一、简答题**

1. **PCB主要存储内容是什么？为什么说PCB是操作系统感知进程存在的唯一依据？（20分）**

PCB 中存储的内容主要包括进程描述信息（含进程标识符）、进程控制和管理信息（含进程调度信息）、资源分配清单和处理机相关信息（含处理机状态信息）。

进程创建时，操作系统为它新建一个 PCB，该结构之后常驻内存。进程结束时，系统收回其 PCB，该进程随之消亡。在进程的整个生命期中，系统总是通过 PCB 对进程进行控制的，亦即系统唯有通过进程的 PCB 才能感知到该进程的存在。所以 PCB 是进程存在的唯一标志。

1. **什么是死锁，死锁的预防与死锁的避免有什么区别。（20分）**

（1）死锁的预防。系统按预定的策略为进程分配资源，这些分配策略能使死锁的四

个必要条件之一不成立，从而使系统不产生死锁。

（2）死锁的避免。系统动态地测试资源分配情况，仅当能确保系统安全时才给进程

分配资源。

（3）死锁的检测。对资源的申请和分配不加限制，只要有剩余的资源就应把资源分

配给申请者，操作系统要定时判断系统是否出现了死锁，当有死锁发生时设法

解除死锁。

1. **分析题**
2. **请用信号量机制给出一种哲学家就餐问题的正确解决方案，并说明该方案为什么可以避免死锁。（30分）**

(1)限制最多N-1个哲学家同时进餐；

(2)偶数序号的先拿右边的筷子，后拿左边的筷子，奇数序号的相反;

(3)加入互斥锁mutex，哲学家要么拿两只筷子要么一只筷子都不拿。

(代码见之前整理的PV操作经典模型，分析省略，可以看王道和PPT)

1. **现有3个并发进程R、M、P，它们共享一个可循环使用的、大小为N的缓冲区B。进程R负责从输入设备读信息，每读一个字符后，把它存入缓冲区B的一个单元中；进程M负责处理读入的字符，若发现读入的字符中有空格符，则把它改成“.”；进程P负责把处理后的字符取出并打印输出。请用P、V操作实现其同步过程。（30分）**

semaphore empty = N;  *// 初始空缓冲区数量*

semaphore full1 = 0;  *// R→M 的待处理数据*

semaphore full2 = 0;  *// M→P 的待输出数据*

semaphore mutex = 1;  *// 互斥锁*

char buffer[N];

int in = 0, out1 = 0, out2 = 0;

cobegin

    procedure R {

        while (1) {

            char x = read\_char();  *// 读入字符*

            P(empty);             *// 申请空缓冲区*

            P(mutex);             *// 进入临界区*

            buffer[in] = x;        *// 写入数据*

            in = (in + 1) % N;

            V(mutex);             *// 离开临界区*

            V(full1);             *// 通知 M 可以处理*

        }

    }

    procedure M {

        while (1) {

            P(full1);             *// 等待 R 写入数据*

            P(mutex);

            char x = buffer[out1];

            if (x == ' ') x = ',';  *// 处理空格*

            buffer[out1] = x;

            out1 = (out1 + 1) % N;

            V(mutex);

            V(full2);             *// 通知 P 可以输出*

        }

    }

    procedure P {

        while (1) {

            P(full2);             *// 等待 M 处理完数据*

            P(mutex);

            char x = buffer[out2];

            out2 = (out2 + 1) % N;

            V(mutex);

            V(empty);             *// 释放缓冲区，通知 R 可写*

            print(x);            *// 输出字符*

        }

    }

coend