

# 操作系统第三次作业

BobAnkh

March 2021

**1. 某 Web 服务器，有 1/3 的可能网页不在高速缓存中，如果所需的网页在高速缓存中，处理时间为 15ms；如果所需的网页不在高速缓存中，需要花费额外的 75ms 读取磁盘。如果 Web 服务器是单线程的，平均每秒可处理多少个请求？如果 Web 服务器是多线程的，平均每秒可处理多少个请求？**

答：

如果是单线程，则处理一个请求平均花费的时间为： $15 + \frac{1}{3} \times 75 = 40\text{ms}$ ，那么平均每秒可以处理的请求数为： $1000/40 = 25$  个；

如果是多线程，则读磁盘的操作可以与线程执行并行。每个线程高速缓存处理时间为 15ms，平均磁盘操作时间为  $\frac{1}{3} \times 75 = 25\text{ms}$ 。因为读磁盘的操作可以与线程执行并行，所以最大效率下平均处理一个请求需要花费 25ms。那么平均每秒可以处理的请求数为： $1000/25 = 40$  个。

**2. 什么是临界资源？什么是临界区？**

答：临界资源是指系统中的某些一次只允许一个进程使用的共享资源；临界区是指进程中访问临界资源的代码片断。

**3. 什么是信号量？信号量 s 的计数值  $s.\text{count} > 0$ ， $s.\text{count} = 0$  和  $s.\text{count} < 0$  的意义分别是什么？**

答：信号量是一个地位高于进程的管理者，用以解决公有资源的使用问题，使 OS 可从进程管理者的角度来处理互斥和同步的问题。每个信号量 s 包括一个整数值  $s.\text{count}$ (计数) 以及一个进程等待队列  $s.\text{queue}$ ，队列中是阻塞在该信号量上的各个进程的标识。信号量只能通过初始化和两个标准的原语 (P、V) 来访问，其作为 OS 核心代码执行，不受进程调度的打断。

信号量 s 的计数值  $s.\text{count}$  意义分别如下：

- $s.\text{count} > 0$  表示有 count 个资源可用；
- $s.\text{count} = 0$  表示无资源可用；
- $s.\text{count} < 0$  则  $|count|$  表示 s 等待队列中的进程个数。

**4. 当低优先级进程正处在临界区之中时，如果高优先级进程变为就绪并被调度，若采用忙等则产生优先级反转问题。优先级反转问题是否也适用于线程？分别考虑 ULT 和 KLT。**

答：对于用户级线程，线程在用户态，内核不知道线程的存在，线程优先级与内核调度无关，不可能发生低优先级线程突然被剥夺而允许高优先级线程运行，因为是不可剥夺的，所以不会出现优先级反转问题。而内核级线程，线程在内核实现，依然会出现这个问题。

**5. 假使 A、B 两个火车站之间是单线铁路，许多列车可以同时到达 A 站，然后经 A 站到 B 站，又列车从 A 到 B 的行驶时间是  $t$ ，列车到 B 站后的停留时间是  $t/2$ 。在该问题模型中，哪些是临界资源？**

答：因为许多列车可以同时到达 A 站，所以 A 站不是互斥资源，B 站只允许一辆车停，显然是临界资源，而 A、B 之间的单铁线路每次只能允许一辆列车发出以后另一辆才能发出，因为列车行驶时间为  $t$ ，而 B 站的停留时间为  $t/2$ ，所以只要下一辆列车在前一辆列车走完该单线铁路前  $1/2$  路程后再发车就可以保证到达 B 站时前一辆车已经离开 B 站，因此 A、B 间的单线铁路的前半段、A、B 间的单线铁路的后半段和 B 站是临界资源。

**6. 苹果-桔子问题：桌子上有一只盘子，每次只能向其中放入一个水果，要求：爸爸专向盘子里放苹果，女儿专等吃盘子中的苹果；妈妈专向盘子里放桔子，儿子专等吃盘子中的桔子；只有盘子为空时，爸爸或妈妈才可向盘子中放入一个水果；仅当盘子中有自己需要的水果时，儿子或女儿可以从盘子中取出。使用信号量和 P、V 操作使爸爸、妈妈、儿子和女儿正确同步工作。**

答：设置信号量 empty, mutex, apple, orange。

由题可知，因为只能放一个水果，所以 empty 初值为 1；信号量 mutex 控制对盘子的互斥访问，初值为 1；apple 和 orange 分别表示盘中苹果和橘子的个数，初值为 0。爸爸妈妈都需要先等待盘子为空时，才获取盘子资源（即获取对盘子的操作），操作完毕后释放对盘子的占用并且分别更改苹果或橘子的信息量（即通知女儿或者儿子可以来盘中获取水果）。儿子女儿都需要先到等待盘子中出现自己想要的对应的水果，才获取水果资源，然后再获取对盘子的操作，操作完毕后，释放对盘子的占用，并且更改盘子为空的信息量（即通知爸爸或者妈妈来放水果）。

设计各自正确工作的伪码如下：

```
typedef int semaphore;
semaphore empty=1;
semaphore mutex=1;
semaphore apple=0;
semaphore orange=0;

void father() {
    while(TRUE) {
        P(empty);
        P(mutex);
        put_apple(); //放苹果
    }
}
```

```

        V(mutex);
        V(apple);
    }
}

void mather() {
    while(TRUE) {
        P(empty);
        P(mutex);
        put_orange(); //放橘子
        V(mutex);
        V(orange);
    }
}

void son() {
    while(TRUE) {
        P(orange);
        P(mutex);
        eat_orange(); //吃橘子
        V(mutex);
        V(empty);
    }
}

void daugther() {
    while(TRUE) {
        P(apple);
        P(mutex);
        eat_apple(); //吃苹果
        V(mutex);
        V(empty);
    }
}

void main() {
    // 四个进程并发进行
    father(); mather(); son(); daugther();
}

```