第三次操作系统作业

卢雨轩 19071125

2021 年 10 月 5 日

基础作业

1. 考虑下面一组进程，进程占用的 CPU 区间长度以毫秒计算。假设在 0 时刻进程以 P1, P2, P3, P4, P5 的顺序到达。

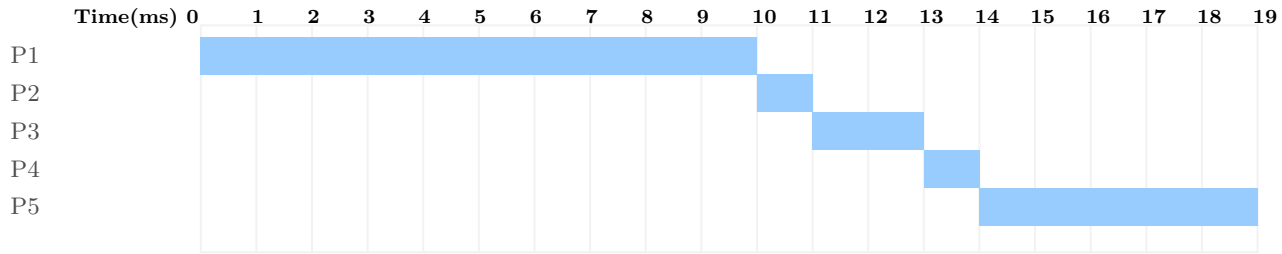
表 1: 进程、区间时间、优先级

进程	区间时间	优先级
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

(a) 画出 4 个 Gantt 图，分别演示使用 FCFS, SJF, 非抢占优先级 (数字越小表示优先级越高) 和 RR(时间片 =1) 算法调度时进程的执行过程。

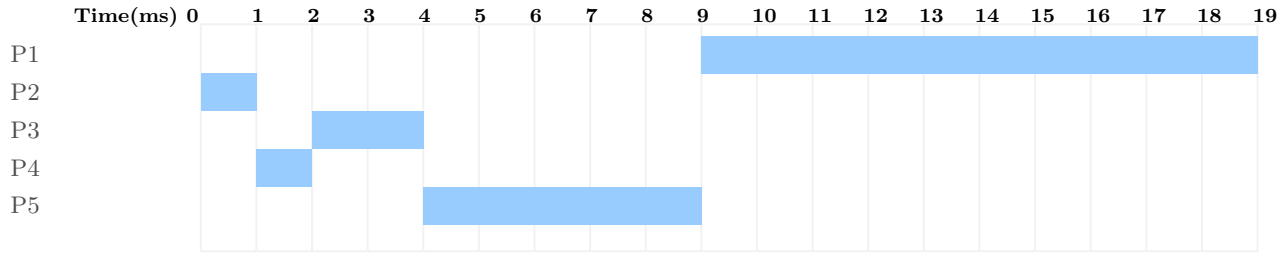
i. FCFS

图 1: 先来先服务算法甘特图



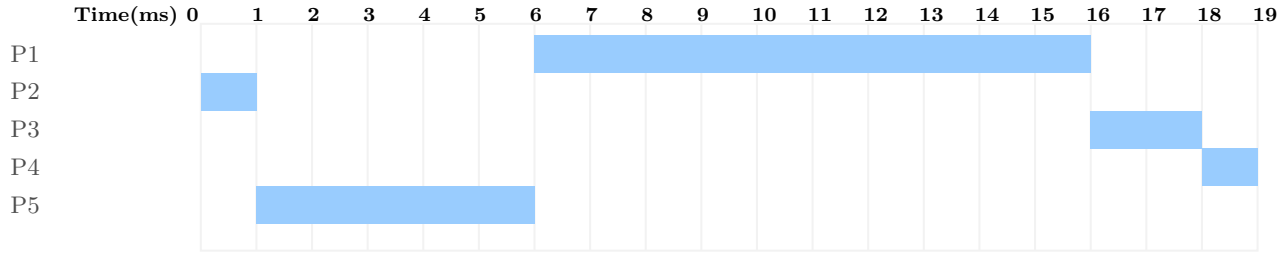
ii. SJF

图 2: 短作业优先算法甘特图



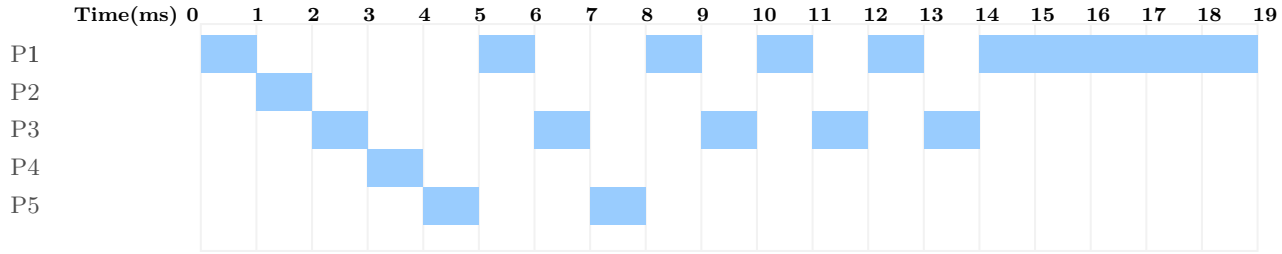
iii. 非抢占优先级调度

图 3: 非抢占优先级调度算法甘特图



iv. RR（时间片 =1）

图 4: 时间片轮转算法甘特图



(b) 每个进程的周转时间是多少?

(c) 每个进程在每种调度算法下的等待时间是多少?

表 2: 进程的周转时间与等待时间

进程	FCFS		SJF		优先级		RR	
	周转时间	等待时间	周转时间	等待时间	周转时间	等待时间	周转时间	等待时间
P1	10ms	0ms	19ms	9ms	16ms	6ms	19ms	0ms
P2	11ms	10ms	1ms	0ms	1ms	0ms	2ms	1ms
P3	13ms	11ms	4ms	2ms	18ms	16ms	14ms	2ms
P4	14ms	13ms	2ms	1ms	19ms	18ms	4ms	3ms
P5	19ms	14ms	9ms	4ms	6ms	1ms	8ms	4ms

2. 什么是忙等待? 指通过死循环等待时间发生，与暂停进程执行直到时间发生相对。

3. 吸烟者问题：有 3 个吸烟者和一个供应者。第一个吸烟者有自己的烟草；第二个吸烟者有自己的纸；第三个吸烟者有自己的火柴。供应者每次随机放两样东西到桌子上提供给 3 个吸烟者之中的一个以完成吸烟。请用信号量为吸烟者和供应者进程编写程序。

   (a) 解法 1:

```
semaphore tobacco_and_paper = 0;
semaphore tobacco_and_lighter = 0;
semaphore lighter_and_paper = 0;
semaphore smoked = 0;
void smoker(){
    while(true) {
        if(has tobacco){
            wait(lighter_and_paper);
        } else if(has lighter) {
            wait(tobacco_and_paper);
        } else {
            wait(tobacco_and_lighter);
        }
        smoke();
        signal(smoked);
    }
}

void supplier(){
    while(true){
        item_to_supply = generate_random_item();
        if(item_to_supply == (tobacco, paper)){
            signal(tobacco_and_paper);
        } else if (item_to_supply == (tobacco, lighter)){
            signal(tobacco_and_lighter);
        } else {
            signal(lighter_and_paper);
        }
        wait(smoked);
```

```
    }
}
```

(b) 解法 2

```
semaphore first_tobacco = 0, second_tobacco = 0;
semaphore first_lighter = 0, second_lighter = 0;
semaphore first_paper = 0, second_paper = 0;
semaphore smoked = 0;
void smoker(){
    while(true) {
        if(has tobacco){
            wait(first_lighter);
            wait(second_paper);
        } else if(has lighter) {
            wait(first_paper);
            wait(second_tobacco);
        } else {
            wait(first_tobacco);
            wait(second_lighter);
        }

        smoke();
        signal(smoked);
    }
}

void supplier(){
    while(true){
        item_to_supply = generate_random_item();
        if(item_to_supply == (tobacco, paper)){
            // we need to serve smoker with lighter.
            signal(first_paper);
            signal(second_tobacco);
        } else if (item_to_supply == (tobacco, lighter)){
            // we need to serve smoker with paper.
            signal(first_tobacco);
            signal(second_lighter);
        } else {
            // we need to serve smoker with tobacco.
            signal(first_lighter);
            signal(second_paper);
        }
        wait(smoked);
    }
}
```

补充作业

1. 假设有三个进程 R、W1、W2 共享缓冲区 B。B 中只能存放一个数。R 每次从输入设备中读一个整数放入 B 中。如果这个整数是奇数，由 W1 取出打印。如果这个整数是偶数，则由 W2 取出打印。规定仅当 B 中没有数据或数据已经被打印才会启动 R 去读数。W1、W2 对 B 中的数据不能重复打印，当 B 中没有数据时也不能打印。要求用信号量操作写出 R、W1、W2 三个进程的程序。（请详细描述所使用变量的含义）

```
buffer B; // Shared buffer.
semaphore buffer_read = 1; // initially, there is no data in buffer

semaphore read_lock_w1 = 0;
semaphore read_lock_w2 = 0;
void R(){
    while(true){
        // wait until two worker finished processing one data.
        // if we only signal(buffer_read) in the worker printing
        // that data, the other worker may not be scheduled
        // then run twice, causing read from empty buffer.
        wait(buffer_read);
        wait(buffer_read);
        data = read();
        write_to_buffer(B, data);
        signal(read_lock_w1);
        signal(read_lock_w2);
    }
}
void W1(){
    while(true){
        wait(read_lock_w1);
        if(read_from(B) % 2 == 1){ // is odd number
            print(read_from(B));
            clear(B);
        }
        signal(buffer_read); // we finished process this data.
    }
}
void W2(){
    while(true){
        wait(read_lock_w2);
        if(read_from(B) % 2 == 0){ // is even number
            print(read_from(B));
            clear(B);
        }
        signal(buffer_read); // we finished process this data.
    }
}
```

2. 有一个铁笼子，猎手放入老虎，农民放入猪，动物园等待取走老虎，饭店等待取走猪。笼子中只能放入一个动物。请使用信号量方法为猎手、农民、动物园、饭店进程编写程序。

```
semaphore cage_available = 1;
semaphore tiger_in_cage = 0;
semaphore pig_in_cage = 0;
void hunter(){
    while(true){
        tiger = hunt();
        wait(cage_available);
        put_tiger_in_cage();
        signal(tiger_in_cage);
    }
}
void farmer(){
    while(true){
        pig = farm();
        wait(cage_available);
        put_pig_in_cage();
        signal(pig_in_cage);
    }
}
void zoo(){
    while(true){
        wait(tiger_in_cage);
        remove_tiger_from_cage();
        signal(cage_available);
    }
}
void restaurant(){
    while(true){
        wait(pig_in_cage);
        remove_pig_from_cage();
        signal(cage_available);
    }
}
```

3. 某寺庙，有小、老和尚若干。有一个水缸，由小和尚提水入缸供老和尚饮用。水缸可容 10 桶水。水取自一个井中，水井窄，每次只能容一个水桶。水桶总数为 3。水缸每次进出也仅 1 桶水，不可以同时进行。请设置合适的信号量描述小和尚、老和尚取水、入水的算法。

```
semaphore water = 0;
semaphore water_remain = 10; // remaining space of gang
semaphore well = 1;
semaphore gang = 1;
semaphore bucket = 3;
void small_monk(){
    while(true){
```

```
        wait(gang_remain);
        wait(bucket);

        wait(well);
        get_water();
        signal(well);

        wait(gang);
        put_water_in_gang();
        signal(gang);

        signal(water);
        signal(bucket);
    }
}
void big_monk(){
    while(true){
        wait(water); // wait for water
        wait(gang);
        drink_water();
        signal(gang);
        signal(water_remain); // ask small monk to get waker.
    }
}
```

4. 判断对错

   (a) 在 RR 调度中，上下文切换的时间应该小于时间片的长度。

   正确。

   (b) SJF 调度算法是最适合分时系统的调度算法。

   不正确。

   (c) FCFS 调度算法只能是非抢占式的

   正确。

   (d) 一个系统中进程之间可能是独立的也可能是合作的。

   正确。

   (e) 如果用锁来保护临界区可以防止竞争条件。

   正确。

   (f) 一个计数信号量的值只能取 0 或者 1.

   错误。

   (g) 在管程中本地变量只能由本地过程来访问。

   正确。

5. 选择题

   (a) 关于竞争条件哪句话是对的?

   A. 几个线程要并发读同样的数据

B. 几个线程要并发读写同样的数据
C. 只有在执行结果与执行顺序无关的时候发生

B

(b) 关于原子指令哪句话是对的?
A. 原子指令只能由一条机器指令组成
B. 作为一个单独的，不可以中断的单元执行
C. 不能用于解决临界区问题

B

(c) 一个临界区的解决方案不需要实现下面的哪一条?
A. 互斥
B. 有空让进
C. 原子性
D. 有限等待

C

一、 附加题

1. 独木桥问题：某条河上只有一座独木桥，两边都有人要过河，为保证安全，一个方向有人过河另一个方向的人就要等待，并且允许一个方向上的人连续过河。请使用信号量实现正确的管理。

```
int flag = 0;
semaphore lock = 1;
semaphore x = 1;
void direction1(){
    while(true){
        wait(x);
        if(flag > 0){
            // do nothing, we have lock
            flag++;
        } else {
            // people currently using bridge is another direction.
            // wait for lock.
            wait(lock);
            // after got lock, flag should be zero.
            flag = 1;
        }
        signal(x);

        go_across_bridge();

        wait(x);
        flag--;
        if(flag == 0){
            signal(lock);
        }
        signal(x);
```

```
    }
}

void direction2(){
    while(true){
        wait(x);
        if(flag < 0){
            // do nothing, we have lock
            flag--;
        } else {
            // people currently using bridge is another direction.
            // wait for lock.
            wait(lock);
            // after got lock, flag should be zero.
            flag = -1;
        }
        signal(x);

        go_across_bridge();

        wait(x);
        flag++;
        if(flag == 0){
            signal(lock);
        }
        signal(x);
    }
}
```