- 其中说明你改动的部分和思路,并将各个椅子数目下的实验结果截图插 入

本作业基于Oranges第七章O代码修改

const.h
100: 修改 #define NR_SYS_CALL 5
proc.h
39 行：给 process 结构增加了成员 sleepticks 来实现不分配时间片
53 行：增加结构 semaphore 结构
修改 NR_PROCS，增加四个进程，删除 testb、testc
proto.h
47：修改 outchar 声明，增加颜色的变量
24：增加新建进程的声明
63：增加新建系统调用 p.v.sleep 的声明
console.c：
修改 outchar 方法实现，使之可以打印多种颜色
global.c：
增加进程的声明
proc.c：
增加 p\v\sleep 实现
其中 sleep 模仿 clock.c 中 tick 的计算，将传进来的毫秒转化为 sleeptick 数，每次时钟中断时将所有进程的 sleeptick 递减，若为 0 则让该进程醒过来，若不为 0 则进程继续睡眠
p\v 按照定义实现
tty.c：
修改 sys_write，对于不同的进程使用不同的颜色，配合 console.c 中对 outchar 方法的修改，使得可以传入不同的颜色进行显示
vsprintf.c：
修改，增加对 “%d”，使得可以显示十进制数
main.c:
增加新进程
其中理发师进程为
while(1){
sys_tem_p(0,0,&customer,&proc_table[2]) ;//消耗一个顾客，如果顾客数为负，表示没有可用顾客，理发师休眠

```
printf("now waiting customer num is %d\n", customer.value);
if(proc_table[2].sleepTicks != 0){
    printf("barber is going to sleep\n");

}
    milli_delay(100);
    printf("starting barbering for custmer%d\n", barberCustomerID);//若有顾客,
```

理发师开始理发

```
        milli_delay(barberTime);//理发中
        printf("barbering over , customerID%d leaves\n" , barberCustomerID);//理发结束，顾客离开
        sys_tem_v(0 , 0 , &barber , &proc_table[2] ) ;//可用理发师的数量增加

        if(customerID%10 == 0){//每有十名顾客时清空屏幕

            clearConsole = 1;


        }
        }
```

顾客进程为：

```
        tempCustomerID = ++customerID;  //当前顾客 ID
        printf("customer%d come in\n" , tempCustomerID);
        if( customer.value >= chair) {//如果有空的椅子，则顾客坐下来，否则顾客离开
            printf("not enough chair for customer to wait , customer%d leave\n" , tempCustomerID);
            milli_delay(customer1ComeTime);
            continue;
        }


        sys_tem_v( 0 , 0 , &customer , &proc_table[3]);//顾客到来并增加一个顾客



        // printf("now the custom value is %d\n", customer.value);
        sys_tem_p( 0 , 0 , &barber , &proc_table[3]);//消耗理发师资源，如果资源可用则继续，否则睡眠
        // printf("now barber available is %d\n", barber.value);
        if( proc_table[3].sleepTicks != 0){
            printf("customer%d has to sleep,waiting number is %d\n" , tempCustomerID , customer.value);
            milli_delay(100);
        }
        printf("customer%d get service\n" , tempCustomerID);//顾客获得服务
```

```
barberCustomerID = tempCustomerID;
// printf("cust %d\n", customer.value);
// milli_delay(100);

// printf("customer%d leave\n" , tempCustomerID);


milli_delay(customer1ComeTime);//顾客下一次到来的时间
```