

操作系统理发师问题作业

一.源代码

源代码如下：

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <unistd.h>
5 #include <semaphore.h>
6 #include <time.h>
7 #include <stdbool.h>
8 #define barber_chair 1
9 int freechairs= 3;
10 sem_t barber_num,consumer_num;
11 pthread_mutex_t mutex;
12 pthread_mutex_t total_num;
13 pthread_mutex_t all;
14 int total=0;
15 int all_num=0;
16 int p[3]={-1,-1,-1};
17 //int finish=0;
18 typedef struct
19 {
20     int thread_id;
21 } people;
22 void* barber(void* parm)
23 {
24     people* peo = (people*)parm;
25     int id = peo->thread_id;
26     while (1){
27         pthread_mutex_lock(&all);
28         if(all_num>=30){
29             //printf("剪完了! \n");
30             pthread_mutex_unlock(&all);
31             pthread_exit(NULL);
32             return NULL;
33         }
34         pthread_mutex_unlock(&all);
35         sem_wait(&consumer_num); //没有consumer就一直wait不下去
36         printf("编号为%i的理发师准备进入临界区! \n",id);
```

```
37 pthread_mutex_lock(&mutex);
38 printf("编号为%i的理发师已经进入临界区! \n",id);
39 int number=-1;
40 for(int i=0;i<3;i++){
41     if(p[i]!=-1){
42         number=p[i];
43         for(int j=i;j<2;j++){
44             {p[j]=p[j+1];
45             p[j+1]=-1;}}
46         break;}
47     freechairs+=1;
48     printf("轮到顾客%i剪头发, 还剩%i椅子! \n",number,freechairs);
49 pthread_mutex_unlock(&mutex);
50 printf("编号为%i的理发师已经离开临界区! \n",id);
51 printf("编号为%i的理发师开始给第%i个顾客剪头发 \n",id,number);
52 int time1 = rand() % 5 + 1;
53     sleep(time1);
54 sem_post(&barber_num);
55 printf("第%i个顾客剪头发结束! \n",number);
56 pthread_mutex_lock(&all);
57     all_num+=1;
58 pthread_mutex_unlock(&all);
59
60 }
61
62 }
63
64 void* consumer(void* parm)
65 {
66     people* peo = (people*)parm;
67     int id = peo->thread_id;
68     while (1){
69         int time1 = rand() % 5 + 1;
70         sleep(time1);
71         int number=0;
72         pthread_mutex_lock(&total_num);
73         total++;
74         if(total>30){
75             //printf("来了30个顾客! \n");
76             pthread_mutex_unlock(&total_num);
77             pthread_exit(NULL);
78             return NULL;
79         }
80         else{
81             printf("有个顾客来了! 这是第%i个 \n",total);
82             number=total;
83             pthread_mutex_unlock(&total_num);
```

```
84
85     }
86     printf("编号%i的顾客线程准备进入临界区! 这是第%i个! \n",id,number);
87     pthread_mutex_lock(&mutex);
88     printf("编号%i的顾客线程已经进入临界区! 这是第%i个! \n",id,number);
89     if(freechairs>0)
90     {
91         for(int i=0;i<3;i++)
92         if(p[i]==-1)
93         {p[i]=number;
94         break;}
95         freechairs-=1;
96         printf("编号%i的顾客线程已经坐下! 这是第%i个! \n",id,number);
97         printf("还剩%i椅子! \n",freechairs);
98         pthread_mutex_unlock(&mutex);
99         printf("编号%i的顾客线程离开临界区! \n",id);
100    }
101    else
102    {
103        printf("没有椅子了! ");
104        printf("编号%i的顾客线程离开! 这是第%i个! \n",id,number);
105        pthread_mutex_lock(&all);
106        all_num+=1;
107        pthread_mutex_unlock(&all);
108        pthread_mutex_unlock(&mutex);
109        printf("编号%i的顾客线程已经离开临界区! 这是第%i个! \n",id,number);
110        continue;
111    }
112    sem_post(&consumer_num);
113    sem_wait(&barber_num);
114    //可以理发, 等理发师
115    //没有barber就一直wait不下去
116    //等到了
117
118
119    }
120
121 }
122 int main()
123 {
124     srand((unsigned)time(NULL));
125     sem_init(&consumer_num, 0, 0);
126     sem_init(&barber_num, 0, 0);
127     pthread_mutex_init(&mutex, NULL);
128     pthread_mutex_init(&total_num, NULL);
129     pthread_mutex_init(&all, NULL);
130     pthread_t b;
```

```

131 pthread_t c[6];
132 people param;
133 people param2[6];
134 for (int i = 0; i < 3; i++)
135 p[i] = -1;
136 param.thread_id = 0;
137 pthread_create(&b, NULL, barber, (void*)&param);
138 for (int t = 0; t < 5; t++) {
139 param2[t].thread_id = t;
140 pthread_create(&c[t], NULL, consumer, (void*)&param2[t]);
141 }
142
143 pthread_join(b, NULL);
144 //
145 for (int t = 0; t < 5; t++) {
146 pthread_join(c[t], NULL);
147 }
148 sem_destroy(&consumer_num);
149 sem_destroy(&barber_num);
150 pthread_mutex_destroy(&mutex);
151 pthread_mutex_destroy(&total_num);
152 pthread_mutex_destroy(&all);
153 return 0;
154
155
156 }
157

```

二.代码解析

本次实验一共有1个理发师线程，和5个顾客线程(线程编号与顾客顺序编号不同，一个线程可以有多个顾客顺序编号)，设置5个是为了制造椅子不够的情况。函数的具体介绍如下：

对于理发师线程函数void* barber(void* parm):

```

1 void* barber(void* parm)
2 {
3 people* peo = (people*)parm;
4 int id = peo->thread_id;
5 while (1){
6 pthread_mutex_lock(&all);
7 if(all_num>=30){
8 //printf("剪完了! \n");
9 pthread_mutex_unlock(&all);
10 pthread_exit(NULL);

```

```

11     return NULL;
12 }
13 pthread_mutex_unlock(&all);
14 sem_wait(&consumer_num); //没有consumer就一直wait不下去
15 printf("编号为%i的理发师准备进入临界区! \n", id);
16 pthread_mutex_lock(&mutex);
17 printf("编号为%i的理发师已经进入临界区! \n", id);
18 int number=-1;
19 for(int i=0; i<3; i++)
20     if(p[i]!=-1){
21         number=p[i];
22         for(int j=i; j<2; j++)
23             {p[j]=p[j+1];
24              p[j+1]=-1;}
25         break;
26         freechairs+=1;
27         printf("轮到顾客%i剪头发, 还剩%i椅子! \n", number, freechairs);
28 pthread_mutex_unlock(&mutex);
29 printf("编号为%i的理发师已经离开临界区! \n", id);
30 printf("编号为%i的理发师开始给第%i个顾客剪头发 \n", id, number);
31 int time1 = rand() % 5 + 1;
32     sleep(time1);
33 sem_post(&barber_num);
34 printf("第%i个顾客剪头发结束! \n", number);
35 pthread_mutex_lock(&all);
36     all_num+=1;
37 pthread_mutex_unlock(&all);
38
39 }
40
41 }

```

首先会先检查临界值，即已经剪头发的人和不剪就走的人加起来是否为30个，如果是则线程函数返回。

接着开始等待sem_wait(&consumer_num)，没有consumer就一直wait不下去，如果有顾客理发师就准备进入临界区。输出相关的信息。临界区为3把椅子。将轮到的顾客取出并开始剪头发。这里采取随机时间来模拟剪头发的时间，剪完后便sem_post(&barber_num)，通知理发师有空了。

对于顾客线程函数void* consumer(void* parm):

```

1 void* consumer(void* parm)
2 {
3     people* peo = (people*)parm;
4     int id = peo->thread_id;
5     while (1){

```

```
6     int time1 = rand() % 5 + 1;
7     sleep(time1);
8     int number=0;
9     pthread_mutex_lock(&total_num);
10    total++;
11    if(total>30){
12        //printf("来了30个顾客! \n");
13        pthread_mutex_unlock(&total_num);
14        pthread_exit(NULL);
15        return NULL;
16    }
17    else{
18        printf("有个顾客来了! 这是第%i个 \n",total);
19        number=total;
20        pthread_mutex_unlock(&total_num);
21    }
22    printf("编号%i的顾客线程准备进入临界区! 这是第%i个! \n",id,number);
23    pthread_mutex_lock(&mutex);
24    printf("编号%i的顾客线程已经进入临界区! 这是第%i个! \n",id,number);
25    if(freechairs>0)
26    {
27        for(int i=0;i<3;i++)
28        if(p[i]==-1)
29        {p[i]=number;
30        break;}
31        freechairs-=1;
32        printf("编号%i的顾客线程已经坐下! 这是第%i个! \n",id,number);
33        printf("还剩%i椅子! \n",freechairs);
34        pthread_mutex_unlock(&mutex);
35        printf("编号%i的顾客线程离开临界区! \n",id);
36    }
37    else
38    {
39        printf("没有椅子了! ");
40        printf("编号%i的顾客线程离开! 这是第%i个! \n",id,number);
41        pthread_mutex_lock(&all);
42        all_num+=1;
43        pthread_mutex_unlock(&all);
44        pthread_mutex_unlock(&mutex);
45        printf("编号%i的顾客线程已经离开临界区! 这是第%i个! \n",id,number);
46        continue;
47    }
48    sem_post(&consumer_num);
49    sem_wait(&barber_num);
50    //可以理发,等理发师
51    //没有barber就一直wait不下去
```

```
53      //等到了
54
55
56      }
57
58 }
```

首先采取随机时间来模拟顾客的随机到来，然后检查临界值，看是否来了30个人，如果来狗=够了则线程函数返回。

接着进入临界区，看有没有椅子坐，有就坐下，freechair减1，没有则直接离开，并输出相应结果。然后离开临界区。

接着sem_post(&consumer_num)通知理发师有客人了，然后sem_wait(&barber_num)，如果理发师没空则继续等待。

本次实验实现了以下信息的输出：

- (a) 进入临界区前，输出某某编号（理发师/顾客）线程准备进入临界区。
- (b) 进入临界区后，输出某某编号（理发师/顾客）线程已进入临界区。
- (c) 离开临界区后，输出某某编号（理发师/顾客）线程已离开临界区。
- (d) 理发师给一个顾客理发开始和结束时，输出相关信息
- (e) 顾客进店、暂坐、理发、离店时，输出相关信息。

三.实现结果

本次实验实现了

当顾客线程开始运行时，一个完整的流程如下：

有个顾客来了！这是第1个

编号4的顾客线程准备进入临界区！这是第1个！

编号4的顾客线程已经进入临界区！这是第1个！

编号4的顾客线程已经坐下！这是第1个！

还剩2椅子！

编号4的顾客线程离开临界区！

如果顾客线程发现没有椅子则是这种情况：

编号4的顾客线程准备进入临界区！这是第11个！

编号4的顾客线程已经进入临界区！这是第11个！

没有椅子了！编号4的顾客线程离开！这是第11个！

编号4的顾客线程已经离开临界区！这是第11个！

对于理发师线程，他的一个完整流程如下：

- 编号为0的理发师准备进入临界区！
- 编号为0的理发师已经进入临界区！
- 轮到顾客1剪头发，还剩3椅子！
- 编号为0的理发师已经离开临界区！
- 编号为0的理发师开始给第1个顾客剪头发
- 第1个顾客剪头发结束！

在开始剪头发到结束时不占用临界区资源。

四.实验心得

编写理发师问题让我理解了线程安全的重要性，特别是在多个线程需要访问和修改共享资源时。我思考了如何去构建一个线程的问题，更深入地理解了信号量，临界区等知识。这让我受益匪浅。

五.附输出结果

- 有个顾客来了！ 这是第1个
- 编号4的顾客线程准备进入临界区！ 这是第1个！
- 编号4的顾客线程已经进入临界区！ 这是第1个！
- 编号4的顾客线程已经坐下！ 这是第1个！
- 还剩2椅子！
- 编号4的顾客线程离开临界区！
- 编号为0的理发师准备进入临界区！
- 编号为0的理发师已经进入临界区！
- 轮到顾客1剪头发，还剩3椅子！
- 编号为0的理发师已经离开临界区！
- 编号为0的理发师开始给第1个顾客剪头发
- 有个顾客来了！ 这是第2个
- 编号1的顾客线程准备进入临界区！ 这是第2个！
- 编号1的顾客线程已经进入临界区！ 这是第2个！
- 编号1的顾客线程已经坐下！ 这是第2个！
- 还剩2椅子！
- 编号1的顾客线程离开临界区！

有个顾客来了！这是第3个

编号3的顾客线程准备进入临界区！这是第3个！

编号3的顾客线程已经进入临界区！这是第3个！

编号3的顾客线程已经坐下！这是第3个！

还剩1椅子！

编号3的顾客线程离开临界区！

有个顾客来了！这是第4个

编号2的顾客线程准备进入临界区！这是第4个！

编号2的顾客线程已经进入临界区！这是第4个！

编号2的顾客线程已经坐下！这是第4个！

还剩0椅子！

编号2的顾客线程离开临界区！

有个顾客来了！这是第5个

编号0的顾客线程准备进入临界区！这是第5个！

编号0的顾客线程已经进入临界区！这是第5个！

没有椅子了！编号0的顾客线程离开！这是第5个！

编号0的顾客线程已经离开临界区！这是第5个！

第1个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客2剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第2个顾客剪头发

第2个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客3剪头发，还剩2椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第3个顾客剪头发

有个顾客来了！这是第6个

编号0的顾客线程准备进入临界区！这是第6个！

编号0的顾客线程已经进入临界区！这是第6个！

编号0的顾客线程已经坐下！这是第6个！

还剩1椅子！

编号0的顾客线程离开临界区！

有个顾客来了！这是第7个

编号1的顾客线程准备进入临界区！这是第7个！

编号1的顾客线程已经进入临界区！这是第7个！

编号1的顾客线程已经坐下！这是第7个！

还剩0椅子！

编号1的顾客线程离开临界区！

有个顾客来了！这是第8个

编号4的顾客线程准备进入临界区！这是第8个！

编号4的顾客线程已经进入临界区！这是第8个！

没有椅子了！编号4的顾客线程离开！这是第8个！

编号4的顾客线程已经离开临界区！这是第8个！

第3个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客4剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第4个顾客剪头发

有个顾客来了！这是第9个

编号3的顾客线程准备进入临界区！这是第9个！

编号3的顾客线程已经进入临界区！这是第9个！

编号3的顾客线程已经坐下！这是第9个！

还剩0椅子！

编号3的顾客线程离开临界区！

有个顾客来了！这是第10个

编号4的顾客线程准备进入临界区！这是第10个！

编号4的顾客线程已经进入临界区！这是第10个！

没有椅子了！编号4的顾客线程离开！这是第10个！

编号4的顾客线程已经离开临界区！ 这是第10个！

有个顾客来了！ 这是第11个

编号4的顾客线程准备进入临界区！ 这是第11个！

编号4的顾客线程已经进入临界区！ 这是第11个！

没有椅子了！ 编号4的顾客线程离开！ 这是第11个！

编号4的顾客线程已经离开临界区！ 这是第11个！

第4个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客6剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第6个顾客剪头发

有个顾客来了！ 这是第12个

编号4的顾客线程准备进入临界区！ 这是第12个！

编号4的顾客线程已经进入临界区！ 这是第12个！

编号4的顾客线程已经坐下！ 这是第12个！

还剩0椅子！

编号4的顾客线程离开临界区！

有个顾客来了！ 这是第13个

编号2的顾客线程准备进入临界区！ 这是第13个！

编号2的顾客线程已经进入临界区！ 这是第13个！

没有椅子了！ 编号2的顾客线程离开！ 这是第13个！

编号2的顾客线程已经离开临界区！ 这是第13个！

第6个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客7剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第7个顾客剪头发

有个顾客来了！ 这是第14个

编号0的顾客线程准备进入临界区！ 这是第14个！

编号0的顾客线程已经进入临界区！这是第14个！

编号0的顾客线程已经坐下！这是第14个！

还剩0椅子！

编号0的顾客线程离开临界区！

第7个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客9剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第9个顾客剪头发

有个顾客来了！这是第15个

编号2的顾客线程准备进入临界区！这是第15个！

编号2的顾客线程已经进入临界区！这是第15个！

编号2的顾客线程已经坐下！这是第15个！

还剩0椅子！

编号2的顾客线程离开临界区！

有个顾客来了！这是第16个

编号1的顾客线程准备进入临界区！这是第16个！

编号1的顾客线程已经进入临界区！这是第16个！

没有椅子了！编号1的顾客线程离开！这是第16个！

编号1的顾客线程已经离开临界区！这是第16个！

有个顾客来了！这是第17个

编号1的顾客线程准备进入临界区！这是第17个！

编号1的顾客线程已经进入临界区！这是第17个！

没有椅子了！编号1的顾客线程离开！这是第17个！

编号1的顾客线程已经离开临界区！这是第17个！

第9个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客12剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第12个顾客剪头发

有个顾客来了！这是第18个

编号1的顾客线程准备进入临界区！这是第18个！

编号1的顾客线程已经进入临界区！这是第18个！

编号1的顾客线程已经坐下！这是第18个！

还剩0椅子！

编号1的顾客线程离开临界区！

有个顾客来了！这是第19个

编号3的顾客线程准备进入临界区！这是第19个！

编号3的顾客线程已经进入临界区！这是第19个！

没有椅子了！编号3的顾客线程离开！这是第19个！

编号3的顾客线程已经离开临界区！这是第19个！

第12个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客14剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第14个顾客剪头发

有个顾客来了！这是第20个

编号4的顾客线程准备进入临界区！这是第20个！

编号4的顾客线程已经进入临界区！这是第20个！

编号4的顾客线程已经坐下！这是第20个！

还剩0椅子！

编号4的顾客线程离开临界区！

有个顾客来了！这是第21个

编号3的顾客线程准备进入临界区！这是第21个！

编号3的顾客线程已经进入临界区！这是第21个！

没有椅子了！编号3的顾客线程离开！这是第21个！

编号3的顾客线程已经离开临界区！这是第21个！

有个顾客来了！这是第22个

编号3的顾客线程准备进入临界区！这是第22个！

编号3的顾客线程已经进入临界区！ 这是第22个！

没有椅子了！ 编号3的顾客线程离开！ 这是第22个！

编号3的顾客线程已经离开临界区！ 这是第22个！

第14个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客15剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第15个顾客剪头发

第15个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客18剪头发，还剩2椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第18个顾客剪头发

有个顾客来了！ 这是第23个

编号0的顾客线程准备进入临界区！ 这是第23个！

编号0的顾客线程已经进入临界区！ 这是第23个！

编号0的顾客线程已经坐下！ 这是第23个！

还剩1椅子！

编号0的顾客线程离开临界区！

有个顾客来了！ 这是第24个

编号3的顾客线程准备进入临界区！ 这是第24个！

编号3的顾客线程已经进入临界区！ 这是第24个！

编号3的顾客线程已经坐下！ 这是第24个！

还剩0椅子！

编号3的顾客线程离开临界区！

有个顾客来了！ 这是第25个

编号0的顾客线程准备进入临界区！ 这是第25个！

编号0的顾客线程已经进入临界区！ 这是第25个！

没有椅子了！ 编号0的顾客线程离开！ 这是第25个！

编号0的顾客线程已经离开临界区！ 这是第25个！

第18个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客20剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第20个顾客剪头发

有个顾客来了！ 这是第26个

编号0的顾客线程准备进入临界区！ 这是第26个！

编号0的顾客线程已经进入临界区！ 这是第26个！

编号0的顾客线程已经坐下！ 这是第26个！

还剩0椅子！

编号0的顾客线程离开临界区！

有个顾客来了！ 这是第27个

编号1的顾客线程准备进入临界区！ 这是第27个！

编号1的顾客线程已经进入临界区！ 这是第27个！

没有椅子了！ 编号1的顾客线程离开！ 这是第27个！

编号1的顾客线程已经离开临界区！ 这是第27个！

有个顾客来了！ 这是第28个

编号1的顾客线程准备进入临界区！ 这是第28个！

编号1的顾客线程已经进入临界区！ 这是第28个！

没有椅子了！ 编号1的顾客线程离开！ 这是第28个！

编号1的顾客线程已经离开临界区！ 这是第28个！

第20个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客23剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第23个顾客剪头发

有个顾客来了！ 这是第29个

编号1的顾客线程准备进入临界区！ 这是第29个！

编号1的顾客线程已经进入临界区！这是第29个！

编号1的顾客线程已经坐下！这是第29个！

还剩0椅子！

编号1的顾客线程离开临界区！

第23个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客24剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第24个顾客剪头发

有个顾客来了！这是第30个

编号4的顾客线程准备进入临界区！这是第30个！

编号4的顾客线程已经进入临界区！这是第30个！

编号4的顾客线程已经坐下！这是第30个！

还剩0椅子！

编号4的顾客线程离开临界区！

第24个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客26剪头发，还剩1椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第26个顾客剪头发

第26个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客29剪头发，还剩2椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第29个顾客剪头发

第29个顾客剪头发结束！

编号为0的理发师准备进入临界区！

编号为0的理发师已经进入临界区！

轮到顾客30剪头发，还剩3椅子！

编号为0的理发师已经离开临界区！

编号为0的理发师开始给第30个顾客剪头发

第30个顾客剪头发结束！