

```

1  #include<iostream>
2  #include<stdio.h>
3  #include<stdlib.h>
4  #include<unistd.h>
5  #include<fstream>
6  #include<string.h>
7  using namespace std;
8  #define RUN "运行"
9  #define WAIT "等待"
10 #define DELETE "销毁"
11 #define READY "就绪"
12 #define WAITCREATE "待创建"
13 typedef struct pcb
14 {
15     int name;
16     char state[20];
17     struct pcb*next;
18
19 }pcb;
20 pcb *head=NULL; //正常队列
21 pcb *headl=NULL; //未到达队列
22 pcb pcbs[5];
23
24 class mainPcb{
25
26 public:
27     void runProcess(); //运行进程，以实现操作
28     void createProcess();
29     void deleteProcess();
30     void hangProcess();
31     void activeProcess();
32     void timeTo();
33     void initialize(); //初始化 所有进程均为待创建
34     //void processStste();
35 };
36 //运行进程
37 void mainPcb::runProcess() {
38     bool flag=false;
39     for(int i=0; i<5;i++){
40         if(strcmp(pcbs[i].state,RUN)==0) {
41             flag =true;
42             break;
43         }
44     }
45     if(flag==false){
46         for(int i=0;i<5;i++){
47             if(strcmp(pcbs[i].state,READY)==0) {
48                 strcpy(pcbs[i].state,RUN);
49                 break;
50             }
51         }
52     }
53 }
54
55 void mainPcb::initialize()
56 {
57     cout<<"进程序号"<<"\t"<<"进程初始状态"<<"\n";
58     for(int i=0;i<5;i++)
59     {
60         pcbs[i].name=i+1;
61         strcpy(pcbs[i].state,WAITCREATE);
62         cout<<pcbs[i].name<<"\t\t"<<pcbs[i].state<<"\n";
63     }
64 }
65
66 /*void mainPcb::processStste()
67 {
68     for(int i=0;i<5;i++)
69     {
70         cout<<pcbs[i].name<<"\t"<<pcbs[i].state<<"\n";
71     }
72 }
73

```

```

74  */
75  void mainPcb::createProcess() //创建进程
76  {
77      cout<<"创建进程\n";
78      cout<<"=====\n";
79
80      for(int i=0;i<5;i++) {
81
82          pcbs[i].name=i+1;
83          if(strcmp(pcbs[i].state,WAITCREATE)==0) {
84              strcpy(pcbs[i].state,READY);
85              break;
86          }
87          cout<<"进程序号\t"<<"进程状态\n";
88          for(int j=0;j<5;j++)
89              cout<<pcbs[j].name<<"\t\t"<<pcbs[j].state<<"\n";
90      }
91      cout<<"\n";
92
93  }
94
95  void mainPcb::deleteProcess() //销毁进程
96  {
97      cout<<"销毁进程";
98      cout<<"=====\n";
99      for(int i=0;i<5;i++) {
100          pcbs[i].name=i+1;
101          if(strcmp(pcbs[i].state,RUN)==0) {
102              strcpy(pcbs[i].state,DELETE);
103              break;
104          }
105          cout<<"进程序号\t"<<"进程状态\n";
106          for(int j=0;j<5;j++)
107              cout<<pcbs[j].name<<"\t\t"<<pcbs[j].state<<"\n";
108      }
109      cout<<"\n";
110  }
111  void mainPcb::hangProcess() //挂起进程
112  {
113      cout<<"挂起进程\n";
114      cout<<"=====\n";
115
116      for(int i=0;i<5;i++) {
117
118          pcbs[i].name=i+1;
119          if(strcmp(pcbs[i].state,RUN)==0) {
120              strcpy(pcbs[i].state,WAIT);
121              break;
122          }
123
124          }
125          cout<<"进程序号\t"<<"进程状态\n";
126          for(int j=0;j<5;j++) {
127              cout<<pcbs[j].name<<"\t\t"<<pcbs[j].state<<"\n";
128
129          }
130      }
131      cout<<"\n";
132  }
133  void mainPcb::activeProcess() //激活进程
134  {
135      cout<<"激活进程";
136      cout<<"=====\n";
137      for(int i=0;i<5;i++) {
138
139          pcbs[i].name=i+1;
140          if(strcmp(pcbs[i].state,WAIT)==0) {
141              strcpy(pcbs[i].state,READY);
142              break;
143          }
144
145      }
146      cout<<"进程序号\t"<<"进程状态\n";

```

```

147         for(int j=0;j<5;j++)
148             cout<<pcbs[j].name<<"\t\t"<<pcbs[j].state<<"\n";
149         cout<<"\n";
150     }
151     void mainPcb::timeTo()    //时间片到
152     {
153         cout<<"时间片到";
154         cout<<"=====\n";
155         for(int i=0;i<5;i++){
156
157             pcbs[i].name=i+1;
158             if(strcmp(pcbs[i].state,RUN)==0){
159                 strcpy(pcbs[i].state,READY);
160                 break;
161             }
162         }
163         cout<<"进程序号\t"<<"进程状态\n";
164         for(int j=0;j<5;j++)
165             cout<<pcbs[j].name<<"\t\t"<<pcbs[j].state<<"\n";
166         cout<<"\n";
167     }
168
169
170     int main()
171     {
172
173
174         mainPcb textpcb;
175         textpcb.initialize();
176
177         cout<<"=====操作系统第二次实验=====\n";
178         cout<<"陈子恒 2016210889\n";
179         cout<<"0退出实验\n";
180         cout<<"1创建进程\n";
181         cout<<"2销毁进程\n";
182         cout<<"3挂起进程\n";
183         cout<<"4激活进程\n";
184         cout<<"5时间片到\n";
185
186         int choice;
187         cout<<"请输入你的选择（数字）\n";
188         cin>>choice;
189
190         while(choice!=0){
191             switch(choice)
192             {
193
194                 case 1:
195                     textpcb.createProcess();    //创建进程
196                     //textpcb.processStste();
197                     break;
198                 case 2:
199                     textpcb.deleteProcess();    //销毁进程
200                     break;
201                 case 3:
202                     textpcb.hangProcess();    //挂起进程
203                     break;
204                 case 4:
205                     textpcb.activeProcess();    //激活进程
206                     break;
207                 case 5:
208                     textpcb.timeTo();    //时间片到
209                     break;
210             }
211
212             cin>>choice;
213             cout<<"\n";
214             textpcb.runProcess();    //无论做出什么选择，都要有一个运行进程才行
215         }
216
217         cout<<"您以退出本次实验"<<"\n";
218
219     }

```