```
 1  /***********************
 2   * Учебное задание #6
 3   * *********************/
 4  #include <stdio.h>
 5  #include <stdlib.h>
 6  #include <unistd.h>
 7  #include <sys/ipc.h>
 8  #include <sys/sem.h>
 9  #include <sys/types.h>
10  #include <signal.h>
11  #include <iostream>
12
13
14  void rwlock_create(const char *pathname,int proj_id,int mode){
15      key_t key=ftok(pathname,proj_id);
16      int semid=semget(key,3,IPC_CREAT|mode);
17      unsigned short values[]={0,0,1};
18      semctl(semid,0,SETALL,values);
19  }
20
21  int rwlock_open(const char *pathname,int proj_id){
22      key_t key=ftok(pathname,proj_id);
23      return semget(key,0,0);
24  }
25
26  void rwlock_rlock(int semid){
27      struct sembuf ops[]={
28          {.sem_num=0,.sem_op=0},
29          {.sem_num=1,.sem_op=1}
30      };
31      semop(semid,ops,2);
32  }
33
34  void rwlock_wlock(int semid){
35      struct sembuf ops[]={
36          {.sem_num=0,.sem_op=1},
37          {.sem_num=1,.sem_op=0},
38          {.sem_num=2,.sem_op=-1}
39      };
40      semop(semid,ops,3);
41  }
42
43  void rwlock_runlock(int semid){
44      struct sembuf ops[]={
45          {.sem_num=1,.sem_op=-1}
46      };
47      semop(semid,ops,1);
48  }
49
50  void rwlock_wunlock(int semid){
51      struct sembuf ops[]={
52          {.sem_num=0,.sem_op=-1},
53          {.sem_num=2,.sem_op=1}
54      };
55      semop(semid,ops,2);
56  }
57
58  void rwlock_destroy(const char *pathname,int proj_id){
59      key_t key=ftok(pathname,proj_id);
60      int semid=semget(key,0,0);
61      semctl(semid,0,IPC_RMID);
62  }
63
64
65  int main(){
66      rwlock_create("semkey",1,666);
67      pid_t pid=fork();
68      int n=0;
69      if (pid==-1){
```

```
70          perror("fork");
71          return 1;
72      }else if(!pid){
73        int semid=rwlock_open("semkey",1);
74        std::cout<<"\nchild: pid="<<pid<<"  :  "<<semid<<"\n"<<std::endl;
75        rwlock_wlock(semid);
76        while(n<5){
77            ++n;
78              for(int i=0;i<10;i++){
79                  putchar('C');
80                  fflush(stdout);
81                  usleep(10000);
82              }
83        }
84        putchar('\n');
85        rwlock_wunlock(semid);
86        rwlock_destroy("semkey",1);
87        std::cout<<"child: Семафор удален\n"<<std::endl;
88      }
89
90      if(pid>0){    // the parent
91        sleep(1);
92        int semid=rwlock_open("semkey",1);
93        std::cout<<"\nthe parent: pid="<<pid<<"  :  "<<semid<<"\n"<<std::endl;
94        rwlock_wlock(semid);
95        while(n<5){
96          ++n;
97          for(int i=0;i<10;i++){
98  //          m.lock();
99              putchar('P');
100             fflush(stdout);
101 //          m.unlock();
102             usleep(10000);
103         }
104       }
105       putchar('\n');
106       rwlock_wunlock(semid);
107
108       sleep(3);
109       rwlock_destroy("semkey",1);
110       std::cout<<"the parent: Семафор удален\n"<<std::endl;
111     }
112
113 }
```