

Nov 13, 14 9:59

semtest.txt

Page 1/2

```

1  ===== initsem.c =====
2  #include "semaphore.h"
3
4  void main(void) {
5      sem_init();
6  }
7
8
9
10 ===== semtest1.c =====
11 #include "semaphore.h"
12 #include <stdio.h>
13 int main(void)
14 {
15     int i;
16     int sem[10];
17     int ret;
18
19     errno = 0;
20     ret = create_sem(-4, 1);
21     if (ret == -1)
22         printf("Create_sem failed, errno = %d\n", errno);
23
24     ret = create_sem(111, -3);
25     if (ret == -1)
26         printf("Create_sem failed, errno = %d\n", errno);
27
28     for(i = 0; i < 10; i++) {
29         errno = 0;
30         sem[i] = create_sem(i, 1);
31         printf("create_sem(%d, 1)
32             returns %d\terrno = %d\n", i, sem[i], errno);
33     }
34
35     for(i = 0; i < 10; i++) {
36         errno = 0;
37         ret = delete_sem(sem[i]);
38         printf("delete_sem(%d) returns %d\n", sem[i], ret);
39     }
40 }
41
42
43 ===== semtest2.c =====
44 #include "semaphore.h"
45 #include <unistd.h>
46 #include <stdlib.h>
47 #include <sys/wait.h>
48 #include <stdio.h>
49
50 #define key 111
51
52 int main(void)
53 {
54     int ret;
55     int sem;
56     int i;
57     int status;
58
59     if (fork()) {
60         sem = create_sem(key, 1);
61         printf("Parent process has obtained semaphore %d\n", sem);
62         for(i = 0; i < 3; i++) {
63             sleep(2);
64             printf("Parent tries to enter CS\n");
65             errno = 0;
66             ret = p(sem);
67             printf("Parent has entered CS (ret = %d, errno = %d)\n", ret, errno);
68
69             sleep(3);
70             errno = 0;
71             ret = v(sem);
72             printf("Parent has left CS (ret = %d, errno = %d)\n", ret, errno);

```

Nov 13, 14 9:59

semtest.txt

Page 2/2

```

73     wait(&status);
74     errno = 0;
75     ret = delete_sem(sem);
76     printf("Parent has deleted sem (ret = %d, errno = %d)\n", ret, errno);
77 } else {
78     sem = create_sem(key, 1);
79     printf("Child process has obtained semaphore %d\n", sem);
80     for(i = 0; i < 3; i++) {
81         sleep(1);
82         printf("Child tries to enter CS\n");
83         errno = 0;
84         ret = p(sem);
85         printf("Child has entered CS (ret = %d, errno = %d)\n", ret, errno);
86
87         sleep(4);
88         errno = 0;
89         ret = v(sem);
90         printf("Child has left CS (ret = %d, errno = %d)\n", ret, errno);
91     }
92     exit(0);
93 }
94
95 ===== semtest3.c =====
96 #include "semaphore.h"
97 #include <unistd.h>
98 #include <stdlib.h>
99 #include <sys/wait.h>
100 #include <stdio.h>
101
102 #define key 111
103
104 int body(int id, int sem) {
105     int ret;
106
107     printf("Child(%d) tries to enter CS\n", id);
108     errno = 0;
109     ret = p(sem);
110     printf("Child(%d) has entered CS (ret = %d, errno = %d)\n", id, ret, err
111 no);
112     sleep(5);
113     errno = 0;
114     ret = v(sem);
115     printf("Child(%d) has left CS (ret = %d, errno = %d)\n", id, ret, errno)
116 ;
117     exit(id);
118 }
119
120 int main(void)
121 {
122     int ret;
123     int sem;
124     int i;
125     int status;
126
127     sem = create_sem(key, 2);
128
129     for (i = 0; i < 10; i++) {
130         if (fork() == 0) body(i, sem);
131     }
132
133     for (i = 0; i < 10; i++) {
134         wait(&status);
135         printf("Child(%d) has terminated\n", (status >> 8));

```