

# 5

## 5.1

### 5.1.1

-> -> -> -> ->

### 5.1.2

### 5.1.3 pthread\_t

PID " PID " tid pthread\_t

```
#include <pthread.h>
pthread_t pthread_self(void);
```

pthread\_self 1 tid

## 1 Phtread\_txex1.c

```
1#include <pthread.h>
2#include <stdio.h>
3
4int main()
5{
6    pthread_t tid = pthread_self();//    tid
7    printf("tid = %lu\n",(unsigned long)tid);
8    return 0;
9}
```

POSIX

pthread

gcc

gcc xxx.c -lpthread

## MultiThread\_Image001

Image not found or type unknown

# 5.1.4

```
#include <pthread.h>
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,void *(*start_routine) (void
*), void *arg);
0
```

pthread\_create

pthread\_t

NULL

void \* v

## 2 Phtread\_txex2.c

```
1#include <pthread.h>
2#include <stdio.h>
3#include <unistd.h>
4#include <errno.h>
5
6void *fun(void *arg)
7{
```

```

8  printf("pthread_New = %lu\n", (unsigned long)pthread_self()); //    tid
9
10
11 int main()
12 {
13
14 pthread_t tid1;
15 int ret = pthread_create(&tid1, NULL, fun, NULL); //
16 if (ret != 0) {
17     perror("pthread_create");
18     return -1;
19 }
20
21 /*tid_main    pthread_self    ID tid_new    pthread_create    tid    */
22 printf("tid_main = %lu tid_new = %lu \n", (unsigned long)pthread_self(), (unsigned long)tid1);
23
24 /*          sleep          */
25 sleep(1);
26
27 return 0;
28
29

```

## MultiThread\_Image002

Image not found or type unknown

```

pthread_create          pthread_create    tid          pthread_self
                                sleep
25

```

## MultiThread\_Image003

Image not found or type unknown

```

3      2

```

# 5.1.5

## 3 Phtread\_txex3.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <errno.h>
5
6 void *fun1(void *arg)
7 {
8     printf("%s: arg = %d Addr = %p\n", __FUNCTION__, *(int *)arg, arg);
9 }
10
11 void *fun2(void *arg)
12 {
13     printf("%s: arg = %d Addr = %p\n", __FUNCTION__, (int)(long)arg, arg);
14 }
15
16 int main()
17 {
18
19     pthread_t tid1, tid2;
20     int a = 50;
21     int ret = pthread_create(&tid1, NULL, fun1, (void *)&a); // a
22     if(ret != 0){
23         perror("pthread_create");
24         return -1;
25     }
27     ret = pthread_create(&tid2, NULL, fun2, (void*)(long)a); // a
28     if(ret != 0){
29         perror("pthread_create");
30         return -1;
31     }
32     sleep(1);
33     printf("%s: a = %d Add = %p \n", __FUNCTION__, a, &a);
34     return 0;
35 }
36
```

Image not found or type unknown

21

a

void

#### 4 Phtread\_txex4.c

```
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <unistd.h>
4  #include <errno.h>
5
6  void *fun1(void *arg)
7  {
8      while(1){
9          printf("%s: arg = %d Addr = %p\n", __FUNCTION__, *(int *)arg, arg);
10         sleep(1);
11     }
12 }
13
14
15 void *fun2(void *arg)
16 {
17     while(1){
18         printf("%s: arg = %d Addr = %p\n", __FUNCTION__, (int)(long)arg, arg);
19         sleep(1);
20     }
21 }
22
23
24 int main()
25 {
26
27     pthread_t tid1, tid2;
28     int a = 50;
29     int ret = pthread_create(&tid1, NULL, fun1, (void *)&a);
30     if(ret != 0){
31         perror("pthread_create");
```

```

32    return -1;
33}
34sleep(1);
35ret = pthread_create(&tid2, NULL, fun2, (void *) (long) a);
36if(ret != 0){
37    perror("pthread_create");
38    return -1;
39}
40while(1){
41    a++;
42    sleep(1);
43    printf("%s: a = %d Add = %p \n", __FUNCTION__, a, &a);
44}
45return 0;
46}
47

```

## MultiThread\_Image005

Image not found or type unknown

### 5 Phtread\_txex5.c

```

1  #include <pthread.h>
2  #include <stdio.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <errno.h>
6
7  struct Stu{
8      int Id;
9      char Name[32];
10     float Mark;
11 };
12
13 void *fun1(void *arg)
14 {
15     struct Stu *tmp = (struct Stu *) arg;

```

```

16    printf("%s: Id = %d Name = %s Mark = %.2f\n", __FUNCTION__, tmp->Id, tmp->Name, tmp->Mark);
17
18}
19
20int main()
21{
22
23    pthread_t tid1, tid2;
24    struct Stu stu;
25    stu.Id = 10000;
26    strcpy(stu.Name, "ZhangSan");
27    stu.Mark = 94.6;
28
29    int ret = pthread_create(&tid1, NULL, fun1, (void *)&stu);
30    if(ret != 0){
31        perror("pthread_create");
32        return -1;
33    }
34    printf("%s: Id = %d Name = %s Mark = %.2f\n", __FUNCTION__, stu.Id, stu.Name, stu.Mark);
35    sleep(1);
36    return 0;
37}
38

```

## MultiThread\_Image006

Image not found or type unknown

# 5.1.6

pthread\_exit

pthread\_cancel

```

#include <pthread.h>
void pthread_exit(void *retval);

```

void\*

NULL

```
#include <pthread.h>
int pthread_join(pthread_t thread, void **retval);
    0
```

tid

```
#define _GNU_SOURCE
#include <pthread.h>
int pthread_tryjoin_np(pthread_t thread, void **retval);
    0
```

0 pthread\_join

```
#include <pthread.h>
int pthread_cancel(pthread_t thread);
    0
```

tid tid 0

API API

## 6 Phtread\_txex6.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <errno.h>
5
6 void *fun1(void *arg)
7 {
8     static int tmp = 0; // static pthread_join
9     //int tmp = 0;
10    tmp = *(int *)arg;
11    tmp+=100;
12    printf("%s: Addr = %p tmp = %d\n", __FUNCTION__, &tmp, tmp);
13    pthread_exit((void *)&tmp); // tmp void*
14 }
15
```



```

16
17int main()
18{
19
20pthread_t tid1;
21int a = 50;
22void *Tmp = NULL; // pthread_join      void**
23int ret = pthread_create(&tid1, NULL, fun1, (void *) &a);
24if(ret != 0){
25perror("pthread_create");
26return -1;
27}
28pthread_join(tid1, &Tmp);
29printf("%s: Addr = %p Val = %d\n", __FUNCTION__, Tmp, *(int *) Tmp);
30return 0;
31}
32

```

## MultiThread\_Image007

Image not found or type unknown

23

100

void\*

pthread\_join

slee

### 7 Phtread\_txex7.c

```

1 #define _GNU_SOURCE
2 #include <pthread.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <errno.h>
6
7 void *fun(void *arg)
8 {
9     printf("Pthread: %d Come !\n", (int)(long) arg+1);
10    pthread_exit(arg);
11}
12
13
14int main()

```

```

15{
16    int ret,i,flag = 0;
17    void *Tmp = NULL;
18    pthread_t tid[3];
19    for(i = 0;i < 3;i++){
20        ret = pthread_create(&tid[i],NULL,fun,(void*)(long)i);
21        if(ret != 0){
22            perror("pthread_create");
23            return -1;
24        }
25    }
26    while(1){//          3
27        for(i = 0;i <3;i++){
28            if(pthread_tryjoin_np(tid[i],&Tmp) == 0){
29                printf("Pthread : %d exit !\n",(int)(long)Tmp+1);
30                flag++;
31            }
32        }
33        if(flag >= 3) break;
34    }
35    return 0;
36 }
37

```

## MultiThread\_Image008

Image not found or type unknown

7

6

pthread\_tryjoin\_np

8 Phtread\_txex8.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6

```

```

7 void *fun1(void *arg)
8 {
9     printf("Pthread:1 come! \n");
10    while(1){
11        sleep(1);
12    }
13 }
14
15 void *fun2(void *arg)
16 {
17     printf("Pthread:2 come! \n");
18     pthread_cancel((pthread_t)(long)arg); // 1
19     pthread_exit(NULL);
20 }
21
22 int main()
23 {
24     int ret,i,flag = 0;
25     void *Tmp = NULL;
26     pthread_t tid[2];
27     ret = pthread_create(&tid[0],NULL,fun1,NULL);
28     if(ret != 0){
29         perror("pthread_create");
30         return -1;
31     }
32     sleep(1);
33     ret = pthread_create(&tid[1],NULL,fun2,(void *)tid[0]); // 1
34     if(ret != 0){
35         perror("pthread_create");
36         return -1;
37     }
38     while(1){ // 2
39         for(i = 0;i <2;i++){
40             if(pthread_tryjoin_np(tid[i],NULL) == 0){
41                 printf("Pthread : %d exit !\n",i+1);
42                 flag++;
43             }
44         }
45         if(flag >= 2) break;

```

```

46    }
47    return 0;
48 }
49

```

## MultiThread\_Image009

Image not found or type unknown

8 pthread\_cancel 27 33

# 5.2

## 5.2.1

“ ” 1 2 “ ”

9 Phtread\_txex9.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7
8  int Num = 0;
9
10 void *fun1(void *arg)
11 {
12     while(Num < 3){
13         Num++;
14         printf("%s: Num = %d\n", __FUNCTION__, Num);
15         sleep(1);
16     }
17     pthread_exit(NULL);
18 }

```

```

19
20 void *fun2( void *arg)
21 {
22     while( Num > -3){
23         Num--;
24         printf( "%s: Num = %d\n", __FUNCTION__, Num);
25         sleep(1);
26     }
27     pthread_exit(NULL);
28 }
29
30 int main()
31 {
32     int ret;
33     pthread_t tid1, tid2;
34     ret = pthread_create( &tid1, NULL, fun1, NULL);
35     if( ret != 0){
36         perror("pthread_create");
37         return -1;
38     }
39     ret = pthread_create( &tid2, NULL, fun2, NULL);
40     if( ret != 0){
41         perror("pthread_create");
42         return -1;
43     }
44     pthread_join( tid1, NULL);
45     pthread_join( tid2, NULL);
46     return 0;
47 }
48

```

## MultiThread\_Image010

Image not found or type unknown

pthread

## 5.2.2 API

```

#include <pthread.h>
int pthread_mutex_init(pthread_mutex_t *mutex,
const pthread_mutexattr_t *restrict attr);
0

```

pthread\_mutex\_t

1

pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;

```

/
#include <pthread.h>
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);

```

lock	unlock	pthread_mutex_t	0	lock
	lock	unlock		pthread_cancel

```

#include <pthread.h>
int pthread_mutex_trylock(pthread_mutex_t *mutex);
0

```

```

#include <pthread.h>
int pthread_mutex_destory(pthread_mutex_t *mutex);
0

```

0

## 10 Phtread\_txex10.c

```

1 #define _GNU_SOURCE
2 #include <pthread.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <errno.h>
6
7 pthread_mutex_t mutex; //

```

```

8
9  int Num = 0; //
10
11 void *fun1( void *arg)
12 {
13     pthread_mutex_lock( &mutex); //
14     while( Num < 3){
15         Num++;
16         printf( "%s: Num = %d\n", __FUNCTION__, Num);
17         sleep( 1);
18     }
19     pthread_mutex_unlock( &mutex); //
20     pthread_exit( NULL); //      pthread_join
21 }
22
23 void *fun2( void *arg)
24 {
25     pthread_mutex_lock( &mutex); //
26     while( Num > -3){
27         Num--;
28         printf( "%s: Num = %d\n", __FUNCTION__, Num);
29         sleep( 1);
30     }
31     pthread_mutex_unlock( &mutex); //
32     pthread_exit( NULL); //      pthread_join
33 }
34
35 int main()
36 {
37     int ret;
38     pthread_t tid1, tid2;
39     ret = pthread_mutex_init( &mutex, NULL); //
40     if( ret != 0){
41         perror( "pthread_mutex_init");
42         return -1;
43     }
44     ret = pthread_create( &tid1, NULL, fun1, NULL); //      1
45     if( ret != 0){
46         perror( "pthread_create");
47         return -1;

```

```

48    }
49    ret = pthread_create( &tid2, NULL, fun2, NULL); //    2
50    if(ret != 0){
51        perror("pthread_create");
52        return -1;
53    }
54    pthread_join(tid1, NULL); //    1
55    pthread_join(tid2, NULL); //    2
56    pthread_mutex_destroy( &mutex); //
57    return 0;
58 }
59

```

## MultiThread\_Image011

Image not found or type unknown

while

## 5.2.3

sleep

11

11 Phtread\_txex11.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7  void *fun1(void *arg)
8  {
9      printf("%s: Pthread Come! \n", __FUNCTION__);
10     pthread_exit(NULL);
11 }
12

```



```
13 void *fun2( void *arg)
14 {
15     printf("s: Pthread Come! \n", __FUNCTION__);
16     pthread_exit(NULL);
17 }
18
19 void *fun3( void *arg)
20 {
21     printf("s: Pthread Come! \n", __FUNCTION__);
22     pthread_exit(NULL);
23 }
24
25 int main()
26 {
27     int ret;
28     pthread_t tid1, tid2, tid3;
29     ret = pthread_create( &tid1, NULL, fun1, NULL);
30     if( ret != 0) {
31         perror("pthread_create");
32         return -1;
33     }
34     ret = pthread_create( &tid2, NULL, fun2, NULL);
35     if( ret != 0) {
36         perror("pthread_create");
37         return -1;
38     }
39     ret = pthread_create( &tid3, NULL, fun3, NULL);
40     if( ret != 0) {
41         perror("pthread_create");
42         return -1;
43     }
44     pthread_join( tid1, NULL);
45     pthread_join( tid2, NULL);
46     pthread_join( tid3, NULL);
47     return 0;
48 }
49
```

Image not found or type unknown

## 5.2.4 API

```
#include <semaphore.h>
int sem_init(sem_t *sem, int pshared, unsigned int value);
0
```

sem\_t                      0                      0                      1

```
PV
#include <pthread.h>
int sem_wait(sem_t *sem);
int sem_post(sem_t *sem);
0
```

sem\_wait                      “sem-1”                      “sem-1”

sem\_post                      “sem+1”

2                      PV

```
#include <pthread.h>
int sem_trywait(sem_t *sem);
0
```

sem\_wait

```
#include <pthread.h>
int sem_destroy(sem_t *sem);
0
```

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6  #include <semaphore.h>
7
8  sem_t sem1, sem2, sem3; //
9
10 void *fun1( void *arg)
11 {
12     sem_wait( &sem1); // sem1          sem1-1
13     printf( "%s: Pthread Come! \n", __FUNCTION__ );
14     sem_post( &sem2); //   sem2
15     pthread_exit( NULL);
16 }
17
18 void *fun2( void *arg)
19 {
20     sem_wait( &sem2); // sem2          14          sem2-1
21     printf( "%s: Pthread Come! \n", __FUNCTION__ );
22     sem_post( &sem3); //   sem3
23     pthread_exit( NULL);
24 }
25
26 void *fun3( void *arg)
27 {
28     sem_wait( &sem3); // sem3          22          sem3-1
29     printf( "%s: Pthread Come! \n", __FUNCTION__ );
30     sem_post( &sem1); //   sem1
31     pthread_exit( NULL);
32 }
33
34 int main()
35 {
36     int ret;
37     pthread_t tid1, tid2, tid3;
38     ret = sem_init( &sem1, 0, 1); //      1
39     if( ret < 0) {
40         perror( "sem_init");

```

```

41    return -1;
42}
43ret = sem_init(&sem2, 0, 0); //    2
44if(ret < 0){
45    perror("sem_init");
46    return -1;
47}
48ret = sem_init(&sem3, 0, 0); //    3
49if(ret < 0){
50    perror("sem_init");
51    return -1;
52}
53ret = pthread_create(&tid1, NULL, fun1, NULL); //    1
54if(ret != 0){
55    perror("pthread_create");
56    return -1;
57}
58ret = pthread_create(&tid2, NULL, fun2, NULL); //    2
59if(ret != 0){
60    perror("pthread_create");
61    return -1;
62}
63ret = pthread_create(&tid3, NULL, fun3, NULL); //    3
64if(ret != 0){
65    perror("pthread_create");
66    return -1;
67}
68/*    */
69pthread_join(tid1, NULL);
70pthread_join(tid2, NULL);
71pthread_join(tid3, NULL);
72
73/*    */
74sem_destroy(&sem1);
75sem_destroy(&sem2);
76sem_destroy(&sem3);
77
78return 0;
79}
80

```

MultiThread\_Image013

Image not found or type unknown

				1	sem_wait	sem_pos
38	1	43	48	2	3	sem_wait

5.3

MultiThread\_Image014

Image not found or type unknown

MultiThread\_Image015

Image not found or type unknown

PV

MultiThread\_Image016

Image not found or type unknown