P1.c

1 #include<stdio.h>

2 #include<pthread.h>

3 void\* thread1(void \*p)

4 {

5 printf("in thread1...\n");

6 while(1)

7 {

8 printf("hi...\n");

9 sleep(1);

10 }

11 }

12

13 main()

14 {

15 pthread\_t t1;

16 printf("in main thread....\n");

17 pthread\_create(&t1,0,thread1,0);

18 while(1)

19 {

20 printf("hello..\n");

21 sleep(1);

22 }

23 }

P2.c

1 #include<stdio.h>

2 #include<pthread.h>

3 void\* thread1(void \*p)

4 {

5 printf("in thread1...\n");

6 while(1);

7 }

8

9 main()

10 {

11 pthread\_t t1;

12 printf("in main thread....\n");

13 pthread\_create(&t1,0,thread1,0);

14 while(1);

15 }

P3.c

1 // pthread\_self - obtain ID of the calling thread

2 #include<stdio.h>

3 #include<pthread.h>

4 void\* thread1(void \*p)

5 {

6 printf("id:%u\n",pthread\_self());

7 while(1);

8 }

9

10 main()

11 {

12 pthread\_t t1;

13 printf("in main thread....\n");

14 pthread\_create(&t1,0,thread1,0);

15 printf("thread id:%u\n",t1);

16 while(1);

17 }

P4.c

1 // pthread\_self - obtain ID of the calling thread

2 #include<stdio.h>

3 #include<pthread.h>

4 void\* thread1(void \*p)

5 {

6 printf("%s\n",(char\*)p);

7 while(1);

8 }

9 void\* thread2(void \*p)

10 {

11 printf("%s\n",(char\*)p);

12 while(1);

13 }

14 main()

15 {

16 pthread\_t t1,t2;

17 printf("in main thread....\n");

18 pthread\_create(&t1,0,thread1,"hello Thread1");

19 pthread\_create(&t2,0,thread2,"hello Thread2");

20 //while(1);

21 }

P5.c

1 /\*pthread\_join - join with a terminated thread

2 int pthread\_join(pthread\_t thread, void \*\*retval);\*/

3 #include<stdio.h>

4 #include<pthread.h>

5 void\* thread1(void \*p)

6 {

7 printf("%s\n",(char\*)p);

8 sleep(5);

9 printf("thread1 terminated...\n");

10 }

11 void\* thread2(void \*p)

12 {

13 printf("%s\n",(char\*)p);

14 sleep(10);

15 printf("thread2 terminated...\n");

16 }

17 main()

18 {

19 pthread\_t t1,t2;

20 printf("in main thread....\n");

21 pthread\_create(&t1,0,thread1,"hello Thread1");

22 pthread\_create(&t2,0,thread2,"hello Thread2");

23 printf("before pthread\_jooin...\n");

24 //pthread\_join(t1,0);

25 printf("after pthread\_join...\n");

26 }

P6.c

1 /\* pthread\_exit - terminate calling thread

2 void pthread\_exit(void \*retval);

3 The pthread\_exit() function terminates the calling thread and returns a

4 value via retval \*/

5

6 #include<stdio.h>

7 #include<pthread.h>

8 void\* thread1(void \*p)

9 {

10 printf("%s\n",(char\*)p);

11 sleep(5);

12 printf("thread1 terminated...\n");

13 pthread\_exit("by by thread1"); //exit the thread

14 }

15 void\* thread2(void \*p)

16 {

17 printf("%s\n",(char\*)p);

18 sleep(10);

19 printf("thread2 terminated...\n");

20 }

21 main()

22 {

23 pthread\_t t1,t2;

24 char \*p;

25 printf("in main thread....\n");

26 pthread\_create(&t1,0,thread1,"hello Thread1");

27 pthread\_create(&t2,0,thread2,"hello Thread2");

28 printf("before pthread\_jooin...\n");

29 //pthread\_join(t1,0);

30 pthread\_join(t1,&p); //waiting for specific thread

31 printf("%s\n",p);

32 printf("after pthread\_join...\n");

33 }