

Naming the file: ANSHGOEL_20BCE1798_EX-7_THREADS

RegNo:20BCE1798

Name: ANSH GOEL

Course Code: CSE2005

Course Name: Operating Systems (Embedded Lab)

Slot: L27+L28

Ex No 7: Basic Commands

Date: 26-02-22

Ex No 7
Date: 26-02-22

Threads

1. Create two threads thread1 and thread2 and call functions fun1 and fun2 respectively. Initialize the value of 'a' and return it to thread1 object and initialize the value of 'b' and return it to thread2 object. Add the values of 'a' and 'b' and print the same in the main function.
2. Create two threads thread1 and thread2 and call functions fun1 and fun2 respectively. Compute and print factorial of a number in fun1, square of a number in fun2.

Q.1.

Create two threads thread1 and thread2 and call functions fun1 and fun2 respectively. Initialize the value of 'a' and return it to thread1 object and initialize the value of 'b' and return it to thread2 object. Add the values of 'a' and 'b' and print the same in the main function.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
void *print_message_function();
int sum=0;
int main()
{
    pthread_t thread1,thread2;
    int iret1, iret2;
    //Create independent threads each of which will execute function
    iret1 = pthread_create(&thread1,NULL,print_message_function,NULL);
    iret2 = pthread_create(&thread2,NULL,print_message_function,NULL);
    pthread_join(thread1,NULL);
    pthread_join(thread2,NULL);
    printf("Thread 1 returns: %d\n",iret1);
    printf("Thread 2 returns %d\n",iret2);
    printf("Sum = %d\n",sum);
    exit(0);
}
void *print_message_function()
{
    int n;
    scanf("%d",&n);
    sum+=n;
    pthread_exit(0);
}
```

Activities Text Editor

Mar 3 16:43

thread11.c

Save

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<pthread.h>
4 void *print_message_function();
5 int sum=0;
6 int main()
7 {
8     pthread_t thread1,thread2;
9     int iret1, iret2;
10    //Create independent threads each of which will execute function
11    iret1 = pthread_create(&thread1,NULL,print_message_function,NULL);
12    iret2 = pthread_create(&thread2,NULL,print_message_function,NULL);
13    pthread_join(thread1,NULL);
14    pthread_join(thread2,NULL);
15    printf("Thread 1 returns: %d\n",iret1);
16    printf("Thread 2 returns %d\n",iret2);
17    printf("Sum = %d\n",sum);
18    exit(0);
19 }
20 void *print_message_function()
21 {
22     int n;
23     scanf("%d",&n);
24     sum+=n;
25     pthread_exit(0);
26 }
```

C Tab Width: 8 Ln 26, Col 2 INS

OUTPUT:

```
ansh@ansh:~
```

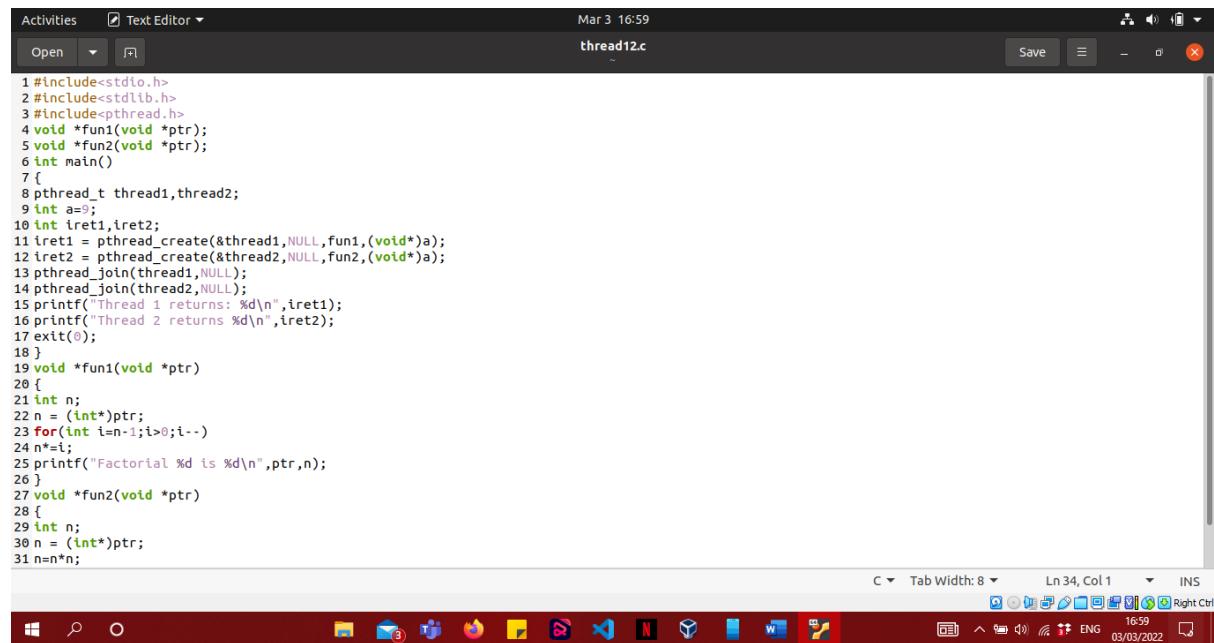
```
ansh@ansh:~$ gedit thread11.c
ansh@ansh:~$ gcc thread11.c -o thread11 -lpthread
ansh@ansh:~$ ./thread11
90
89
Thread 1 returns: 0
Thread 2 returns 0
Sum = 179
ansh@ansh:~$
```


Q.2.

Create two threads thread1 and thread2 and call functions fun1 and fun2 respectively. Compute and print factorial of a number in fun1, square of a number in fun2.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
void *fun1(void *ptr);
void *fun2(void *ptr);
int main()
{
pthread_t thread1,thread2;
int a=9;
int iret1,iret2;
iret1 = pthread_create(&thread1,NULL,fun1,(void*)a);
iret2 = pthread_create(&thread2,NULL,fun2,(void*)a);
pthread_join(thread1,NULL);
pthread_join(thread2,NULL);
printf("Thread 1 returns: %d\n",iret1);
printf("Thread 2 returns %d\n",iret2);
exit(0);
}
void *fun1(void *ptr)
{
int n;
n = (int*)ptr;
for(int i=n-1;i>0;i--)
n*=i;
printf("Factorial %d is %d\n",ptr,n);
}
void *fun2(void *ptr)
{
int n;
n = (int*)ptr;
n=n*n;
printf("Square of %d is %d\n",ptr,n);
}
```



```
Activities Text Editor Mar 3 16:59
Open thread12.c Save
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<pthread.h>
4 void *fun1(void *ptr);
5 void *fun2(void *ptr);
6 int main()
7 {
8 pthread_t thread1,thread2;
9 int a=9;
10 int iret1,iret2;
11 iret1 = pthread_create(&thread1,NULL,fun1,(void*)a);
12 iret2 = pthread_create(&thread2,NULL,fun2,(void*)a);
13 pthread_join(thread1,NULL);
14 pthread_join(thread2,NULL);
15 printf("Thread 1 returns: %d\n",iret1);
16 printf("Thread 2 returns %d\n",iret2);
17 exit(0);
18 }
19 void *fun1(void *ptr)
20 {
21 int n;
22 n = (int*)ptr;
23 for(int i=n-1;i>0;i--)
24 n*=i;
25 printf("Factorial %d is %d\n",ptr,n);
26 }
27 void *fun2(void *ptr)
28 {
29 int n;
30 n = (int*)ptr;
31 n=n*n;
```

OUTPUT:

```
ansh@ansh:~$ gedit thread12.c
ansh@ansh:~$ gcc thread12.c -o thread12 -lpthread
thread12.c: In function ‘main’:
thread12.c:11:43: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
  11 |     iret1 = pthread_create(&thread1,NULL,fun1,(void*)a);
                 ^
thread12.c:12:43: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
  12 |     iret2 = pthread_create(&thread2,NULL,fun2,(void*)a);
                 ^
thread12.c: In function ‘fun1’:
thread12.c:22:3: warning: assignment to ‘int’ from ‘int *’ makes integer from pointer without a cast [-Wint-conversion]
  22 |     n = (int*)ptr;
                 ^
thread12.c:25:20: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘void *’ [-Wformat=]
  25 |     printf("Factorial %d is %d\n",ptr,n);
                 ~^          ~~
                 |          |
                 int           void *
thread12.c: In function ‘fun2’:

```



```
ent 2 has type ‘void *’ [-Wformat=]
  25 |     printf("Factorial %d is %d\n",ptr,n);
                 ~^          ~~
                 |          |
                 int           void *
thread12.c: In function ‘fun2’:
thread12.c:30:3: warning: assignment to ‘int’ from ‘int *’ makes integer from pointer without a cast [-Wint-conversion]
  30 |     n = (int*)ptr;
                 ^
thread12.c:32:20: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘void *’ [-Wformat=]
  32 |     printf("Square of %d is %d\n",ptr,n);
                 ~^          ~~
                 |          |
                 int           void *
ansh@ansh:~$ ./thread12
Factorial 9 is 362880
Square of 9 is 81
Thread 1 returns: 0
Thread 2 returns 0
ansh@ansh:~$
```