Experiment : 10

10. Write a C program that illustrates two processes communicating using shared memory

Sourcecode:

```
include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
        int i;
        void *shared_memory;
        char buff[100];
        int shmid;
        shmid=shmget((key_t)2345,1024,0666|IPC_CREAT);
        printf("key of shared memory is %d\n",shmid);
        shared_memory=shmat(shmid,NULL,0);
        printf("process attached at %p\n",shared_memory);
         printf("Enter some data to write to shared memory\n");
         read(0,buff,100);
        strcpy(shared_memory,buff);
        printf("you wrote : %s\n",(char *)shared_memory);
}
```

Output:

```
[exam66@Linux ~]$ vi os.c
[exam66@Linux ~]$ cc os.c
[exam66@Linux ~]$ ./a.out
key of shared memory is 163845
process attached at 0xb77b0000
Enter some data to write to shared memory
WELOCOME TO OS LAB
you wrote : WELOCOME TO OS LAB
```

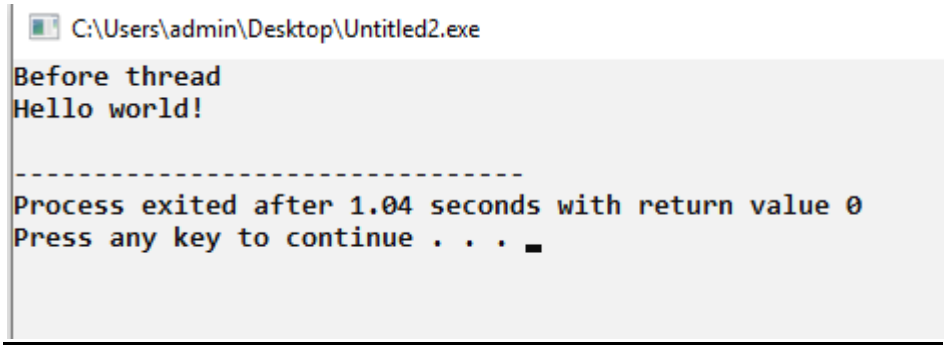
Experiment : 11

11. Write C program to create a thread using pthreads library and let it run its function.

Sourcecode:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *mythread(void *vargp)
{
        sleep(1);
        printf("Hello world! \n");
        return NULL;
}
int main()
{
        pthread_t tid;
        printf("Before thread\n");
        pthread_create(&tid,NULL,mythread,NULL);
        pthread_join(tid,NULL);
        exit(0);
}
```

Output:



Experiment : 12

12. Write a C program to illustrate concurrent execution of threads using pthreads library

Sourcecode:

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
void *mythread1(void *vargp)
{
   int i;
   printf("thread1\n");
   for(i=1;i<=5;i++)
   printf("i=%d\n",i);
   printf("Exit from thread1\n");
   return NULL;
}
void *mythread2(void *vargp)
{
   int j;
   printf("thread2 \n");
   for(j=1;j<=5;j++)
   printf("j=%d\n",j);
   printf("Exit from thread2\n");
   return NULL;
}
int main()
{
  pthread_t tid;
  printf("Before thread\n");
  pthread_create(&tid,NULL,mythread2,NULL);
  pthread_join(tid,NULL);
  pthread_create(&tid,NULL,mythread1,NULL);
  pthread_join(tid,NULL);
```

```
  exit(0);
}
```

Output: