

KHUSHBU PATEL (150034)

ASSIGNMENT : 3 MULTI-THREADING

QUE : 1) 1. Write a multithreading program, where threads runs same shared global variable of the process between them.

```
#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

int sharevar=5;

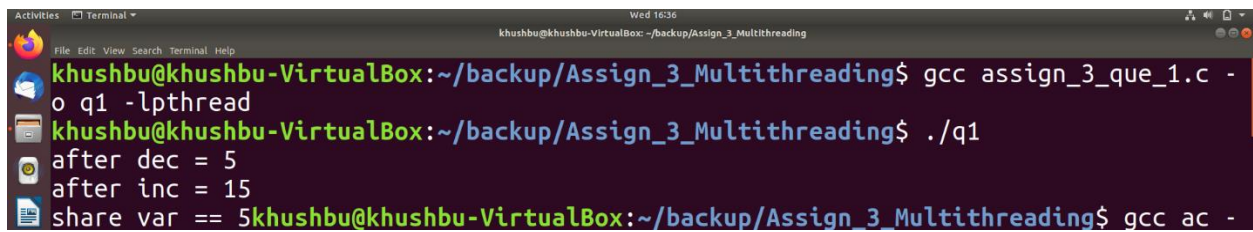
void *thread_sum(void *arg)
{
    sharevar+=10;
    printf("after inc = %d \n",sharevar);
}

void *thread_subtract(void *arg)
{
    sharevar-=10;
    printf("after dec = %d \n",sharevar);
}

int main()
{
    pthread_t thread1,thread2;
    pthread_create(&thread1 , NULL ,thread_sum,NULL );
    pthread_create(&thread2 , NULL ,thread_subtract,NULL );
    pthread_join(thread1,NULL);
    pthread_join(thread2,NULL);
    printf("share var == %d",sharevar);
```

```
return 0;
```

```
}
```



```
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ gcc assign_3_que_1.c -  
o q1 -lpthread  
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ ./q1  
after dec = 5  
after inc = 15  
share var == 5khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ gcc ac -
```

QUE : 2) 2. Write a program where thread cancel itself.(use pthread_cancel())

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<errno.h>
```

```
#include<unistd.h>
```

```
#include<pthread.h>
```

```
#include<sys/types.h>
```

```
#include<bits/types.h>
```

```
void *process(void *arg)
```

```
{
```

```
    printf("sleep 2 sec\n");
```

```
    sleep(2);
```

```
    printf("enter into thread....\n\n");
```

```
}
```

```
int main()
```

```
{
```

```
    pthread_t t_id;
```

```
    pthread_create(&t_id,NULL,process , NULL);
```

```
    if(errno) perror("pthread created..");
```

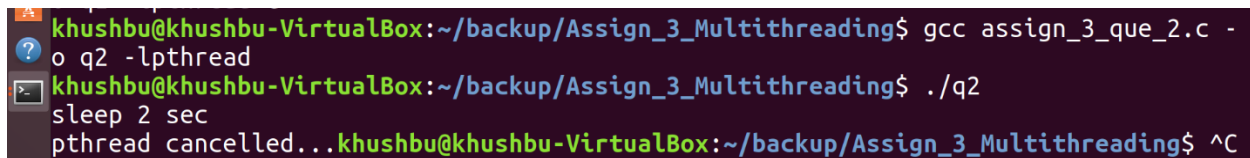
```

    int a;

    a=pthread_cancel(t_id);

    if(a==0)
{
    printf("pthread cancelled...");
}
}

```



```

khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ gcc assign_3_que_2.c -
o q2 -lpthread
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ ./q2
sleep 2 sec
pthread cancelled...khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ ^C

```

QUE : 3) Write a program that changes the default properties of newly created posix threads.(ex: to change default pthread stack size)

```

#include<stdio.h>

#include<pthread.h>

#include<unistd.h>

#include<stdlib.h>

void * proc(void * param)
{
    sleep(2);
    return 0;
}

int main()
{
    pthread_attr_t Attr;
    pthread_t Id;
    void *stk;
    size_t siz;

```

```

int err;

size_t my_stksize=0x30000000;

void *my_stack;

pthread_attr_init(&Attr);

pthread_attr_getstacksize(&Attr,&siz);

pthread_attr_getstackaddr(&Attr,&stk);

printf("Default:Addr =%08x default size=%d\n",stk ,siz);

my_stack=(void*)malloc(my_stksize);

pthread_attr_setstack(&Attr,my_stack,my_stksize);

pthread_create(&ld,&Attr,proc,NULL);

pthread_attr_getstack(&Attr,&stk,&siz);

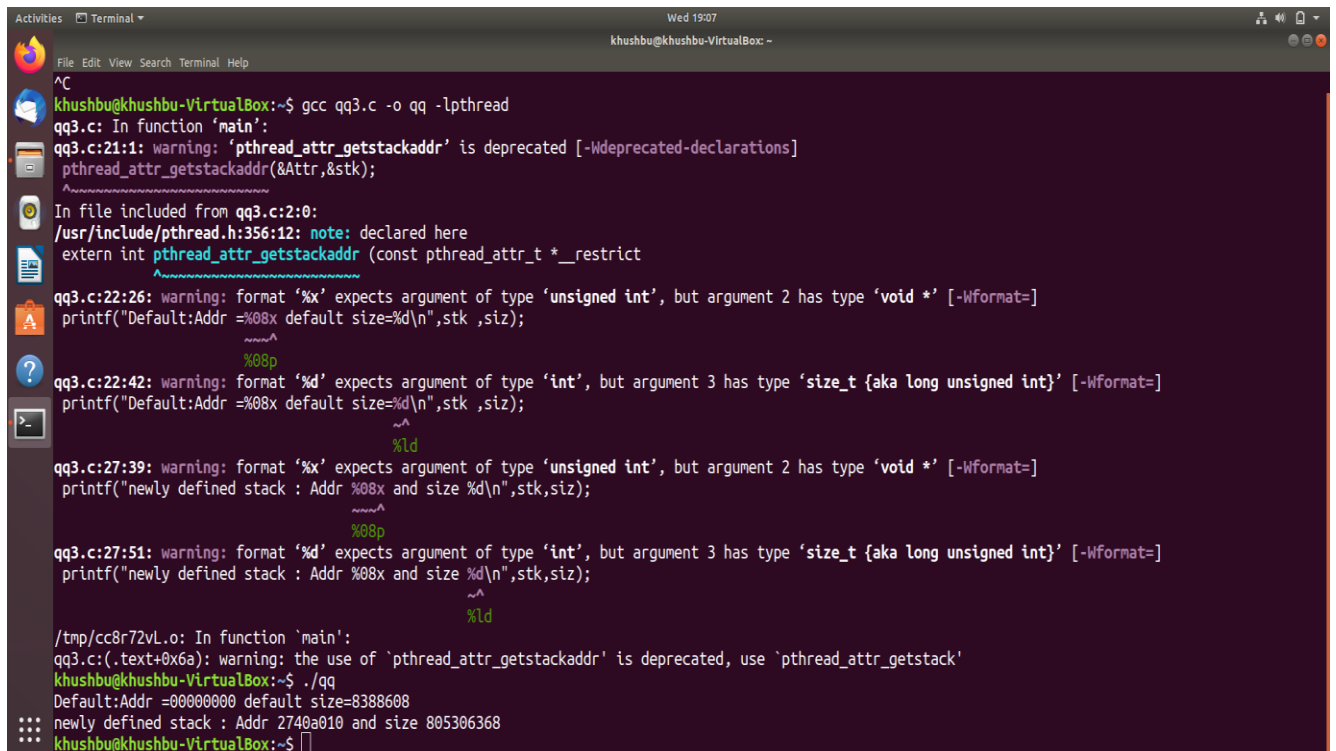
printf("newly defined stack : Addr %08x and size %d\n",stk,siz);

sleep(3);

return 0;

}

```



```

khushbu@khushbu-VirtualBox:~$ gcc qq3.c -o qq -lpthread
qq3.c: In function 'main':
qq3.c:21:1: warning: 'pthread_attr_getstackaddr' is deprecated [-Wdeprecated-declarations]
pthread_attr_getstackaddr(&Attr,&stk);
^
In file included from qq3.c:2:0:
/usr/include/pthread.h:356:12: note: declared here
extern int pthread_attr_getstackaddr (const pthread_attr_t *__restrict
^
qq3.c:22:26: warning: format '%x' expects argument of type 'unsigned int', but argument 2 has type 'void *' [-Wformat=]
printf("Default:Addr =%08x default size=%d\n",stk ,siz);
^
qq3.c:22:42: warning: format '%d' expects argument of type 'int', but argument 3 has type 'size_t {aka long unsigned int}' [-Wformat=]
printf("Default:Addr =%08x default size=%d\n",stk ,siz);
^
qq3.c:27:39: warning: format '%x' expects argument of type 'unsigned int', but argument 2 has type 'void *' [-Wformat=]
printf("newly defined stack : Addr %08x and size %d\n",stk,siz);
^
qq3.c:27:51: warning: format '%d' expects argument of type 'int', but argument 3 has type 'size_t {aka long unsigned int}' [-Wformat=]
printf("newly defined stack : Addr %08x and size %d\n",stk,siz);
^
/tmp/cc8r72vL.o: In function 'main':
qq3.c:(.text+0x6a): warning: the use of 'pthread_attr_getstackaddr' is deprecated, use 'pthread_attr_getstack'
khushbu@khushbu-VirtualBox:~$ ./qq
Default:Addr =00000000 default size=8388608
newly defined stack : Addr 2740a010 and size 805306368
khushbu@khushbu-VirtualBox:~$

```

QUE : 4) 4. Write a program where pthread task displays the thread id and also prints the calling

process pid.

```
#include<pthread.h>
```

```
pthread_t tid;
```

```
void *thr_fn(void *arg)
```

```
{
```

```
    pid_t pid;
```

```
    pthread_t tid;
```

```
    pid=getpid();
```

```
    tid=pthread_self();
```

```
    printf("process id== %u thread id== %u \n", (unsigned int)pid, (unsigned int)tid);
```

```
    return 0;
```

```
}
```

```
int main(void)
```

```
{
```

```
    int err;
```

```
    err=pthread_create(&tid, NULL, thr_fn, NULL);
```

```
    if(err!=0)
```

```
{
```

```
        printf("can't create...%s\n", strerror(err));
```

```
}
```

```
    while(1);
```

```
    exit(0);
```

```
}
```

```
assign_3_que_4.c:25:2: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]
    exit(0);
    ^~~~~
assign_3_que_4.c:25:2: warning: incompatible implicit declaration of built-in function 'exit'
assign_3_que_4.c:25:2: note: include '<stdlib.h>' or provide a declaration of 'exit'
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ ./q4
process id== 3087  thread id== 3487794944
^C
```

QUE : 5) Write a program that implements threads synchronization using mutex techniques.

```
#include<stdio.h>

#include<stdlib.h>

#include<semaphore.h>

#include<pthread.h>

int sharevar=5;

pthread_mutex_t my_mutex;

void *thread_sum(void *arg)
{
    pthread_mutex_lock(&my_mutex);
    sharevar+=10;
    pthread_mutex_unlock(&my_mutex);
    printf("after summation = %d \n",sharevar);
}

void *thread_subtract(void *arg)
{
    pthread_mutex_lock(&my_mutex);
    sharevar-=10;
    pthread_mutex_unlock(&my_mutex);
    printf("after subtraction = %d \n",sharevar);
}
```

```
int main()
{
    pthread_t thread1,thread2;
    pthread_mutex_init(&my_mutex , NULL);
    pthread_create(&thread1 , NULL, thread_sum,NULL);
    pthread_create(&thread2 , NULL, thread_subtract,NULL);
    pthread_join(thread1,NULL);
    pthread_join(thread2,NULL);
    printf("share var == %d",sharevar);
    return 0;
}
```

```
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ gcc assign_3_que_5.c -o q5 -lpthread
khushbu@khushbu-VirtualBox:~/backup/Assign_3_Multithreading$ ./q5
after summation = 15
after subtraction = 5
```