**Example 1:-**

#include<stdio.h>

#include<string.h>

#include<pthread.h>

#include<stdlib.h>

#include<unistd.h>

void\* doSomeThing(void \*arg)

{

printf("\n\tThread says \"Hello World\"\n");

return NULL;

}

int main(void)

{

int err;

pthread\_t tid;

pthread\_create(&tid, NULL, &doSomeThing, NULL);

printf("\n\tThread created successfully\n");

sleep(1);

return 0;

}

**Example 2:-**

#include<stdio.h>

#include<string.h>

#include<pthread.h>

#include<stdlib.h>

#include<unistd.h>

pthread\_t tid[2];

void\* doSomeThing(void \*arg)

{

unsigned long i = 0;

pthread\_t id = pthread\_self();

if(pthread\_equal(id,tid[0]))

{

printf("\n First thread processing\n");

}

else

{

printf("\n Second thread processing\n");

}

return NULL;

}

int main(void)

{

int i = 0;

int err;

while(i < 2)

{

err = pthread\_create(&(tid[i]), NULL, &doSomeThing, NULL);

if (err != 0)

printf("\ncan't create thread :[%s]", strerror(err));

else

printf("\n Thread created successfully\n");

i++;

sleep(1);

}

return 0;

}

**Example 3:-**

#include<stdio.h>

#include<pthread.h>

void\* thread(void \*ptr);

int main()

{

pthread\_t tid;

char\* message = "Thread 1 Message";

int ret = pthread\_create(&tid,NULL,thread,(void\*)message);

pthread\_join(tid,NULL);

printf("\tThread 1 retuns %d \n",ret);

return 0;

}

void\* thread(void \*arg)

{

char\* message = arg;

printf("\tI am thread printing \"%s\" \n",message);

sleep(3);

}

**Example 4:-**

#include<stdio.h>

#include<string.h>

#include<pthread.h>

#include<stdlib.h>

#include<unistd.h>

pthread\_t tid[2];

int ret1,ret2;

void\* doSomeThing(void \*arg)

{

unsigned long i = 0;

pthread\_t id = pthread\_self();

for(i=0; i<(0xFFFFFFFF);i++);

if(pthread\_equal(id,tid[0]))

{

printf("\n First thread processing done\n");

ret1 = 100;

pthread\_exit(&ret1);

}

else

{

printf("\n Second thread processing done\n");

ret2 = 200;

pthread\_exit(&ret2);

}

return NULL;

}

int main(void)

{

int i = 0;

int err;

int \*ptr[2];

while(i < 2)

{

err = pthread\_create(&(tid[i]), NULL, &doSomeThing, NULL);

if (err != 0)

printf("\ncan't create thread :[%s]", strerror(err));

else

printf("\n Thread created successfully\n");

i++;

}

pthread\_join(tid[0], (void\*\*)&(ptr[0]));

pthread\_join(tid[1], (void\*\*)&(ptr[1]));

printf("\n return value from first thread is [%d]\n", \*ptr[0]);

printf("\n return value from second thread is [%d]\n", \*ptr[1]);

return 0;

}