#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

#include<pthread.h>

void\* threadFunction(void\* arg)

{

int thread\_id=\*(int\*)arg;

printf("Thread %d is starting\n",thread\_id);

sleep(2);

printf("Thread %d is finished\n",thread\_id);

return NULL;

}

int main()

{

pthread\_t threads[5];

int thread\_ids[5],i;

for(i=0;i<5;i++)

{

thread\_ids[i]=i+1;

if(pthread\_create(&threads[i],NULL,threadFunction,&thread\_ids[i])!=0)

{

printf("Error creating thread %d\n",i+1);

return 1;

}

}

for(i=0;i<5;i++)

{

pthread\_join(threads[i],NULL);

}

printf("All threads have finished execution");

return 0;

}

/\*

gcc threadscheduling.c -pthread

./a.out

Thread 2 is starting

Thread 1 is starting

Thread 3 is starting

Thread 4 is starting

Thread 5 is starting

Thread 2 is finished

Thread 1 is finished

Thread 3 is finished

Thread 4 is finished

Thread 5 is finished

All threads have finished execution

\*/