



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



**Asignatura**      **Sistemas operativos**  
**Prof**            **Ana Belem Juárez Méndez**  
**Práctica 5**      **Hilos**

### Objetivo

Utilizar los conocimientos de hilos en aplicaciones de sistemas operativos.

### Introducción

Un hilo es un subconjunto de un proceso. Un proceso puede contener varios hilos pero un hilo no puede contener varios procesos. Los hilos de manera intrínseca tienen una región de memoria común. En un sistema multiprocesador una aplicación multihilos se ejecutará más rápidamente que una aplicación implementada mediante múltiples procesos. Aunque existen varios paquetes de hilos los recomendables son los que se apegan a POSIX (Portable Operative System – IX).

La siguiente función permite crear un hilo:

```
int pthread_create(pthread_t *thread, pthread_attr_t *attr, void  
*(*start_routine)(void *), void *arg)
```

- ◆ El primer argumento apunta al identificador del hilo.
- ◆ El segundo argumento especifica los atributos asociados al nuevo hilo.
  - Si es NULL se utilizan los atributos predeterminados lo cual incluye ser un hilo dependiente.
- ◆ El tercer argumento indica el nombre de la función a ejecutar por el hilo.
- ◆ El cuarto argumento es un único parámetro que puede pasársele a la función.

El hilo que llama a la función `pthread_join` espera a que termine el hilo especificado en el primer argumento. La sintaxis de dicha función es:

```
int pthread_join(pthread_t th1, void **value)
```

- ◆ Esta función se puede solicitar únicamente sobre hilos dependientes.
- ◆ La función regresa en el segundo argumento el valor que devuelve el hilo que finaliza su ejecución con `int pthread_exit(void *value)`





# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



Realiza las siguientes pruebas :

\* E1=-200 E2=9000000 N=1

\* E1=-200 E2=9000000 N=2

\* E1=-200 E2=9000000 N=4

\* E1= N1 E2= N2 N=1

\* E1= N1 E2= N2 N=2

\* E1= N1 E2= N2 N=4

N1 Y N2 = valores de  
tu elección

### Ejercicio 3.

Realizar un programa con una variable entera global con un valor inicial de cero. Crear un hilo que incremente la variable global en A unidades. Crear otro hilo que la disminuya en B unidades. Al final el hilo principal(main) imprimirá el valor de la variable global.

### Para el ejercicio 4

**NOTA** Si el segundo argumento de la función `pthread_create` es `NULL` se utilizan los atributos predeterminados, lo cual implica ser un hilo *dependiente*. Si un hilo es dependiente no se liberarán sus recursos a menos que otro hilo espere su finalización, esto mediante `pthread_join`. Se pueden crear hilos sin necesidad de esperar a que dichos hilos terminen, de ser así deben tener el atributo de independientes (`PTHREAD_CREATE_DETACHED`).

La función `pthread_attr_setdetachstate(pthread_attr_t *attr, int detachstate)` permite establecer el estado de terminación de un hilo, si el segundo argumento es `PTHREAD_CREATE_DETACHED`, el hilo se considera como independiente.



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



**Ejercicio 4.** El siguiente programa muestra cómo se crean 10 hilos independientes. Compíllalo y ejecútalo.

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

#define NHILOS 10

void funcion_hilos(void)
{
    printf("\nHilo %u \n",pthread_self()); // pthread_self devuelve el identificador del hilo
    pthread_exit(0);
}

void main()
{
    int i;
    pthread_attr_t atributos;
    pthread_t hilos[NHILOS];

    /* Se inicializan los atributos como independientes */
    pthread_attr_init(&atributos);
    pthread_attr_setdetachstate(&atributos, PTHREAD_CREATE_DETACHED);

    for(i=0;i<NHILOS;i++)
    {
        pthread_create(&hilos[i],&atributos,(void *)funcion_hilos,NULL);
    }
    /* El hilo principal se suspende 4 segundos, para esperar a que los hilos terminen, de no ser así
    al terminar el hilo principal, todos los hilos terminarían */
    sleep(4);
}
```

## Entrega

- ◆ La fecha máxima de entrega es para el 27 de marzo del 2020.
- ◆ Esta práctica la pueden realizar en equipo de dos personas o individual.
- ◆ Enviar en un archivo .zip los programas .c y su reporte en formato PDF. Su reporte debe contener: Portada, Objetivo, Introducción, Desarrollo( Código y Pantallas de ejecución de sus pruebas), Conclusiones y Referencias.
- ◆ Al enviar su práctica deben colocar en el asunto:

Practica5 ApellidoPaternoIntegrante1 ApellidoPaternoIntegrante2