



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

Práctica 3. Hilos en UNIX-LINUX

Parte 1



Objetivo

Aplicar los conocimientos sobre la programación de hilos en situaciones que se pueden presentar en el Sistema Operativo UNIX-LINUX

Introducción

Un hilo tiene muchas similitudes con un proceso, aunque conceptualmente un hilo es un subconjunto de un proceso. Así, un proceso puede contener varios hilos pero un hilo no puede contener varios procesos. La diferencia más importante entre ambos, reside en que un proceso tiene un espacio de memoria único que no puede ser accedido por otro proceso, a menos que se implemente un mecanismo de comunicación entre procesos (IPC). En el caso de los hilos, de manera intrínseca se tiene una región de memoria común a todos los hilos. Así dentro de un proceso puede haber varios hilos de ejecución modificando la misma variable sin necesidad de una programación adicional.

Existen otras diferencias que convierten a los hilos en mecanismos recomendables para la gran mayoría de aplicaciones distribuidas. Por ejemplo un proceso hijo al ser una copia idéntica de todo el proceso padre, ocupa muchos recursos. En contraste un hilo ocupa menos recursos y además se puede asignar prioridades de ejecución a cada hilo por separado. Por otra parte 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).

Funciones de hilos

A continuación se explican algunas de las funciones más importantes que se utilizan en Linux para el manejo de hilos.

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

Esta función permite crear un hilo. 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.



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

Práctica 3. Hilos en UNIX-LINUX

Parte 1



```
pthread_join(pthread th1, void **value)
```

El hilo que llama a esta función espera a que termine el hilo especificado en el primer argumento y que no tiene que ser necesariamente un hilo hijo. Esta función se puede solicitar únicamente sobre hilos dependientes. La función devuelve en el segundo argumento el valor que devuelve el hilo que finaliza su ejecución.

```
pthread_t pthread_self(void)
```

Un hilo puede averiguar su identificador mediante la función anterior.

```
int pthread_exit(void *value)
```

Esta función finaliza un hilo e incluye un apuntador a una estructura que es devuelta al hilo que mande llamar a la función `pthread_join`. Los hilos de un proceso al compartir el espacio de direcciones del proceso, pueden modificar variables globales, acceder a descriptores de archivos abiertos o interferirse de maneras diversas.

Programa3a.c Capture, compile y ejecute el siguiente programa de creación de hilos. Observe su funcionamiento y coméntelo con su equipo.

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

void *funcionHilo(void *argu)
{
    printf("Dentro del hilo...\n");
    pthread_exit(NULL);
}

main ()
{
    pthread_t id_hilo;
    printf("Creacion del hilo...\n");
    pthread_create(&id_hilo, NULL, funcionHilo, NULL);
    printf("Hilo creado. Esperando su finalizacion...\n");
    pthread_join(id_hilo, NULL);
    printf("Hilo finalizado...\n");
}
```

Debido a que se utilizan hilos, se requiere añadir la opción `-lpthread` al compilar.



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

Práctica 3. Hilos en UNIX-LINUX

Parte 1



Ejemplo: \$gcc programa.c -lpthread -o programa

Programa3b.c Capture, compile y ejecute el siguiente programa de creación de hilos. Observe su funcionamiento y coméntelo con su equipo.

```
#include <pthread.h>
#include <stdio.h>
#define MAX_THREADS 10

void func(void)
{
    printf("Hilo %d \n", pthread_self());
    pthread_exit(0);
}

void main()
{
    int j;
    pthread_attr_t attr;
    pthread_t thid[MAX_THREADS];
    /* Se inician los atributos y se marcan como independientes */
    pthread_attr_init(&attr);
    pthread_attr_setdetachstate(&attr,
    PTHREAD_CREATE_DETACHED);
    for(j = 0; j < MAX_THREADS; j++)
        pthread_create(&thid[j], &attr, (void *)func, NULL);
    /* El hilo principal se suspende 5 segundos, para esperar a que los hilos hijos terminen. De lo
    contrario al terminar el proceso principal, también todos sus hijos terminarían */
    sleep(5);
}
```

La función `int pthread_attr_init(pthread_attr_t *attr)` permite iniciar un objeto atributo que se puede utilizar para crear nuevos hilos.

La función `int pthread_attr_setdetachstate(pthread_attr_t *attr, int detachstate)` permite establecer el estado de terminación del hilo, si el segundo argumento es `PTHREAD_CREATE_DETACHED`, el hilo se considera como independiente y liberará sus recursos cuando termine la ejecución. Si el hilo es dependiente entonces no se liberan sus recursos a menos que otro hilo espere su finalización.



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

Práctica 3. Hilos en UNIX-LINUX

Parte 1



Programa3c.c Capture, compile y ejecute el siguiente programa de creación de hilos. Observe su funcionamiento y coméntelo con su equipo.

```
#include <stdio.h>
#include <pthread.h>
void *funcionHilo(void *args);
main ()
{
    pthread_t id_hilo;
    int arg[2]={2,3};
    printf("Creacion del hilo...\n");
    pthread_create(&id_hilo, NULL, funcionHilo, (void *)arg);
    printf("Hilo creado. Esperando su finalizacion...\n");
    pthread_join(id_hilo, NULL);
    printf("Hilo finalizado...\nvalor 1: %i\nvalor 2: %i\n",*arg,*arg+1));
}

void *funcionHilo(void *argu)
{
    int *args=(int *)argu;
    printf("Valor 1: %i\nValor 2: %i\n",*args,*arg+1));
    *args=7;
    *(arg+1)=8;
    pthread_exit(NULL);
}
```



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

Práctica 3. Hilos en UNIX-LINUX

Parte 1



Desarrollo

Practica3a.c Realizar un programa que inicia un hilo principal que crea dos hilos. El hilo principal espera hasta que ambos hilos terminen y después finaliza. Los hilos sólo deben de mostrar algún mensaje en pantalla y terminar. (25% de la calificación de la práctica)

Practica3b.c Realizar un programa con una variable entera global (fuera de main()) con un valor inicial de cero. Crear un hilo que incremente la variable global en a unidades y crear otro hilo que la disminuya en b unidades. Al final el hilo principal imprimirá el valor de la variable global. (25% de la calificación de la práctica)

Practica3c.c Realizar un programa cree un proceso hijo que a su vez creará tres hilos. Cada uno de los tres hilos creará dos hilos más. Cada uno de los hilos creados imprimirá en pantalla sus identificadores. (25% de la calificación de la práctica)

Practica3d.c Realizar un programa que cree tres hilos. El primer hilo se encargará de contabilizar las ocurrencias de una cadena dentro de un archivo específico y devolver el resultado al programa principal; el segundo hilo copiará los archivos de su directorio actual a un subdirectorío que usted elija devolviendo al programa principal el número de archivos copiados; el tercer hilo generará un archivo donde se reportarán los resultados devueltos por los otros dos hilos. (25% de la calificación de la práctica)