



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y
CIENCIAS SOCIALES Y ADMINISTRATIVAS



Práctica 8: Hilos

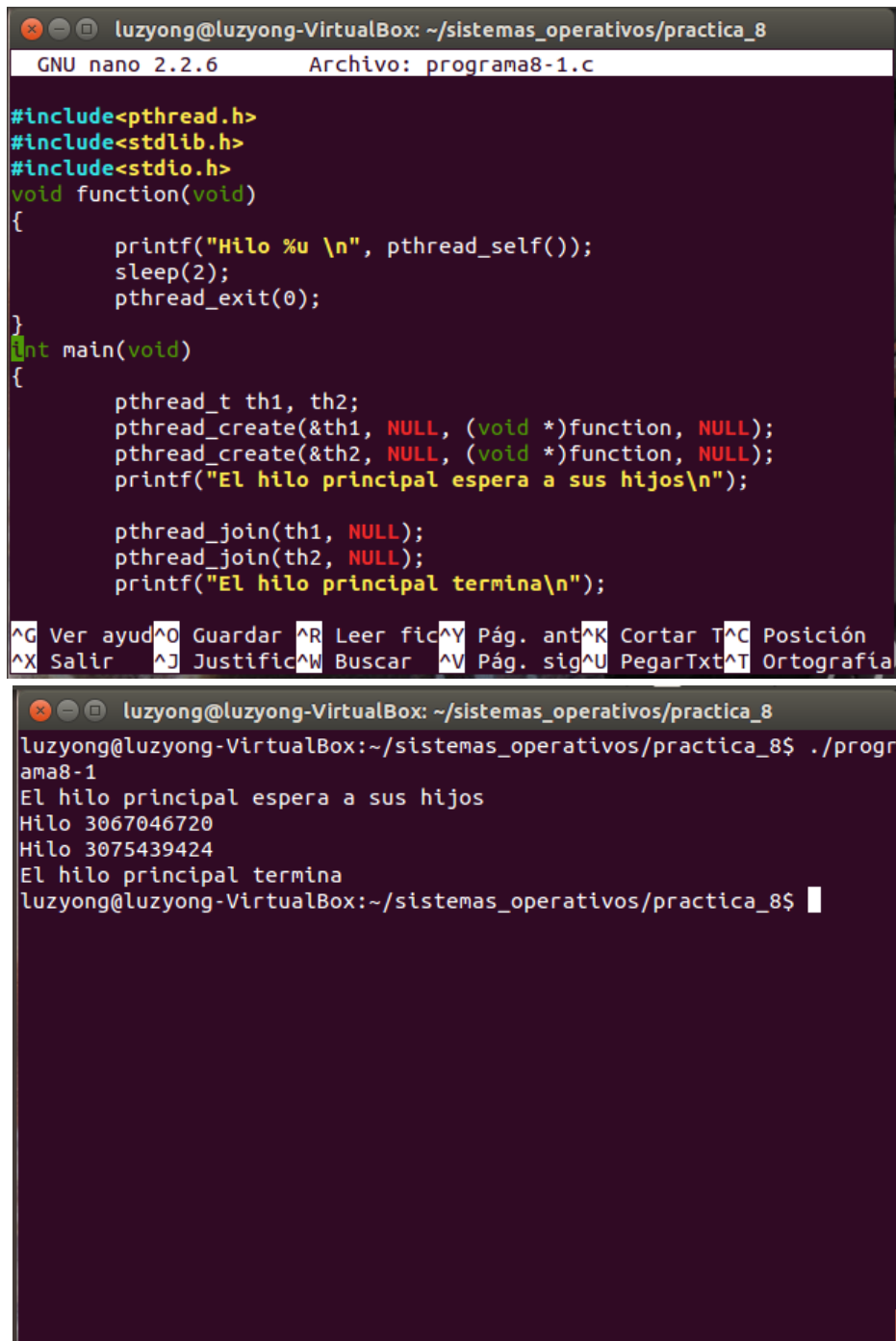
Yong Rodríguez Luz María

Un hilo tiene muchas similitudes con un proceso, aunque un hilo es un subconjunto de un proceso. Un proceso puede contener varios hilos, pero un hilo no puede contener varios procesos. Un proceso tiene un espacio de memoria único que no puede ser accedido por otro proceso que no sea un mecanismo de IPC. En el caso de los hilos, se tiene una región de memoria común entre todos ellos. Es por eso por lo que dentro de un proceso varios hilos pueden modificar una misma variable, sin necesidad de programación adicional.

Ejercicios prácticos.

Programa 8-1.

Un proceso crea dos hilos hijos independientes y espera a que terminen.



The image consists of two terminal screenshots. The top screenshot shows the source code of a C program named 'programa8-1.c' being edited in the nano text editor. The code includes headers for pthread.h, stdlib.h, and stdio.h. It defines a function 'function(void)' that prints the thread ID, sleeps for 2 seconds, and then exits. The 'main(void)' function creates two threads, 'th1' and 'th2', using 'pthread_create', prints a message 'El hilo principal espera a sus hijos\n', joins both threads using 'pthread_join', and finally prints 'El hilo principal termina\n'.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
GNU nano 2.2.6 Archivo: programa8-1.c

#include<pthread.h>
#include<stdlib.h>
#include<stdio.h>
void function(void)
{
    printf("Hilo %u \n", pthread_self());
    sleep(2);
    pthread_exit(0);
}
int main(void)
{
    pthread_t th1, th2;
    pthread_create(&th1, NULL, (void *)function, NULL);
    pthread_create(&th2, NULL, (void *)function, NULL);
    printf("El hilo principal espera a sus hijos\n");

    pthread_join(th1, NULL);
    pthread_join(th2, NULL);
    printf("El hilo principal termina\n");
}
^G Ver ayud^O Guardar ^R Leer fic^Y Pág. ant^K Cortar T^C Posición
^X Salir ^J Justific^W Buscar ^V Pág. sig^U PegarTxt^T Ortografía
```

The bottom screenshot shows the execution of the program. The user runs './programa8-1' in the terminal. The output shows the main thread printing 'El hilo principal espera a sus hijos', followed by two child threads printing their IDs: 'Hilo 3067046720' and 'Hilo 3075439424'. Finally, the main thread prints 'El hilo principal termina'.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$ ./programa8-1
El hilo principal espera a sus hijos
Hilo 3067046720
Hilo 3075439424
El hilo principal termina
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$
```

- Pregunta 8-1: ¿Qué sucede si el proceso principal termina un instante después de haber creado los hilos?

No se ejecutan las funciones dentro de los hilos hijos.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$ ./pregu
nta8-1
El hilo principal espera a sus hijos
El hilo principal termina
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$
```

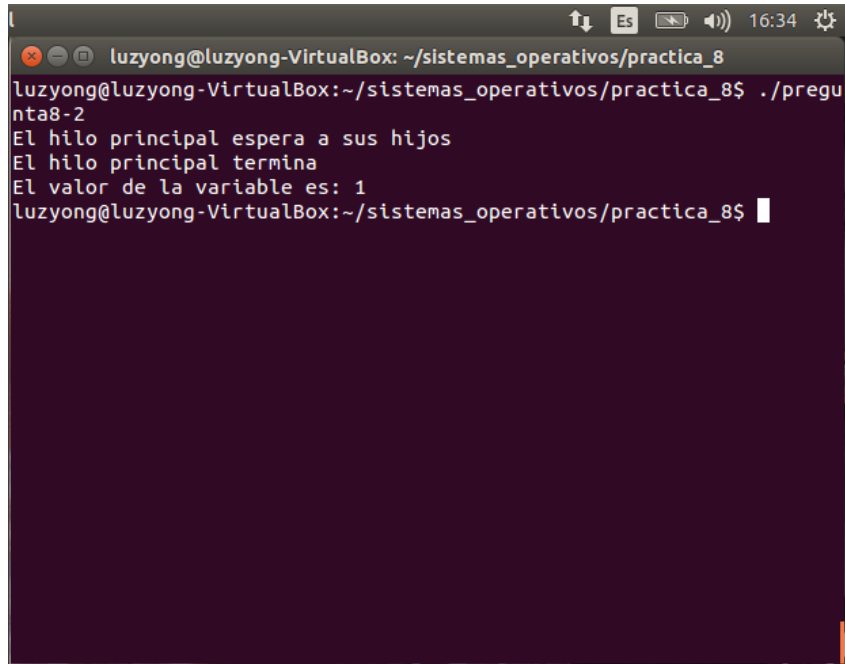
¿Puede un hilo hijo crear otro hilo y esperar a que este nieto termine?

Sí

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$ ./pregu
nta8-1-1
El hilo principal espera a sus hijos
El hilo hijo tiene un hilo nieto
El hilo hijo tiene un hilo nieto
Hilo nieto 3058850624
Hilo nieto 3049257792
Hilo 3075636032
Hilo 3067243328
El hilo principal termina
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$
```

- Pregunta 8-2: ¿Los hilos comparten la variable global?

Sí, al tener una región de memoria en común, pueden compartir variables globales, pues todas las funciones tienen acceso a la dirección de memoria de esas variables.

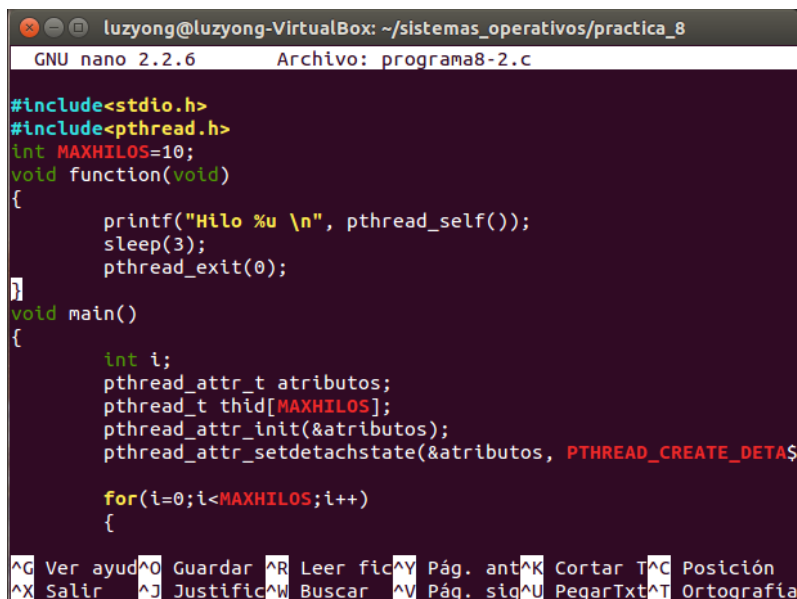
A terminal window titled 'luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8'. The prompt is 'luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8\$./pregunta8-2'. The output shows: 'El hilo principal espera a sus hijos', 'El hilo principal termina', and 'El valor de la variable es: 1'. The prompt returns to 'luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8\$'.

¿Y qué sucede si la variable se declara dentro de la función main()?

Solo lo que está dentro del main puede utilizarlas, a menos que las variables se pasen a través de llamadas a funciones.

Programa 8-2.

Crea diez hilos independientes que imprimen su id.

A code editor window titled 'luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8' showing the source code for 'programa8-2.c' using GNU nano 2.2.6. The code includes <stdio.h> and <pthread.h>, defines MAXHILOS as 10, and declares a function 'function(void)' that prints the thread ID and sleeps for 3 seconds. The 'main()' function creates an array of 10 pthread_t threads, initializes attributes, and starts them in a loop. The code is as follows:

```
#include<stdio.h>
#include<pthread.h>
int MAXHILOS=10;
void function(void)
{
    printf("Hilo %u \n", pthread_self());
    sleep(3);
    pthread_exit(0);
}
void main()
{
    int i;
    pthread_attr_t atributos;
    pthread_t thid[MAXHILOS];
    pthread_attr_init(&atributos);
    pthread_attr_setdetachstate(&atributos, PTHREAD_CREATE_DETACHED);

    for(i=0;i<MAXHILOS;i++)
    {
```

Ejercicio 8-1.

Elabore un programa donde se comparta una variable con un valor inicial de cero y dos hilos independientes deben aumentar y decrementar la variable en uno. Después de n segundos el proceso debe imprimir el valor final de la variable y terminar. El valor de n debe pasar en la línea de comandos.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
GNU nano 2.2.6 Archivo: ejercicio8-1.c

{
    for(;;)
        variable-=1;
    pthread_exit(0);
}
void main()
{
    int i, n;
    printf("Ingresa el tiempo del proceso:\n");
    scanf("%d",&n);
    pthread_attr_t atributos;
    pthread_t thid[2];
    pthread_attr_init(&atributos);
    pthread_attr_setdetachstate(&atributos, PTHREAD_CREATE_DETACHED);
    pthread_create(&thid[0], &atributos, (void *)aumento, NULL);
    pthread_create(&thid[1], &atributos, (void *)decremento, NULL);

    sleep(n);
    printf("El valor de la variable es:%d\n",variable);
}

^G Ver ayuda ^O Guardar ^R Leer fichero ^Y Pág. anterior ^K Cortar texto ^C Posición
^X Salir ^J Justificar ^W Buscar ^V Pág. siguiente ^U Pegar texto ^T Ortografía

luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$ ./ejercicio8-1
Ingresa el tiempo del proceso:
4
El valor de la variable es:9342693
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$
```

Programa 8-3.

Uso de semáforos para sincronizar dos hilos.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
GNU nano 2.2.6 Archivo: programa8-3.c

int main(void)
{
    pthread_t th1, th2;
    sem_init(&mutex1, 0, 1);
    sem_init(&mutex2, 0, 0);
    pthread_create(&th1, NULL, (void *)Hilo1, NULL);
    pthread_create(&th2, NULL, (void *)Hilo2, NULL);
    pthread_join(th1, NULL);
    pthread_join(th2, NULL);

    exit(0);
}

void Hilo1(void)
{
    while(i>0)
    {
        sem_wait(&mutex1);
        printf("Soy el hilo 1 (%d) y ésta es la impresion %d \n", $
        44 líneas leídas
^G Ver ayuda^O Guardar ^R Leer fich^Y Pág. ant^K Cortar T^C Posición
^X Salir ^J Justific^W Buscar ^V Pág. sig^U PegarTxt^T Ortografía
```

Ejercicio 8-3.

Modifique el programa 8-3 para que se sincronicen las impresiones consecutivas de tres hilos en vez de dos.

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$ ./ejerc
icio8-3
Soy el hilo 3 (-1236317376) y esta es la impresion 12
Soy el hilo 1 (-1219531968) y ésta es la impresion 11
Soy el hilo 2 (-1227924672) y esta es la impresion 10
Soy el hilo 1 (-1219531968) y ésta es la impresion 9
Soy el hilo 2 (-1227924672) y esta es la impresion 8
Soy el hilo 1 (-1219531968) y ésta es la impresion 7
Soy el hilo 3 (-1236317376) y esta es la impresion 6
Soy el hilo 2 (-1227924672) y esta es la impresion 5
Soy el hilo 3 (-1236317376) y esta es la impresion 4
Soy el hilo 3 (-1236317376) y esta es la impresion 3
Soy el hilo 1 (-1219531968) y ésta es la impresion 2
Soy el hilo 2 (-1227924672) y esta es la impresion 1
Soy el hilo 3 (-1236317376) y esta es la impresion 0
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_8$
```

```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_8
GNU nano 2.2.6 Archivo: ejercicio8-3.c

void Hilo3(void);
int main(void)
{
    pthread_t th1, th2, th3;
    sem_init(&mutex1, 0, 1);
    sem_init(&mutex2, 0, 0);
    sem_init(&mutex3, 0, 2);
    pthread_create(&th1, NULL, (void *)Hilo1, NULL);
    pthread_create(&th2, NULL, (void *)Hilo2, NULL);
    pthread_create(&th3, NULL, (void *)Hilo3, NULL);
    pthread_join(th1, NULL);
    pthread_join(th2, NULL);
    pthread_join(th3, NULL);
    exit(0);
}

void Hilo1(void)
{
    [ 58 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer fichero ^Y Pág. anterior ^K Cortar texto ^C Posición
^X Salir ^J Justificar ^W Buscar ^V Pág. siguiente ^U Pegar texto ^T Ortografía
```

Conclusiones.

En esta práctica aprendí sobre el manejo de hilos como parte de un proceso. Aprendí que actúan de forma parecida a un proceso y que se pueden implementar en muchas aplicaciones.