



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

“Práctica 4. Administrador de procesos en Linux y Windows (2)”

Grupo: 2CM8

Integrantes:

- Martínez Coronel Brayan Yosafat.
- Monteros Cervantes Miguel Angel.
- Ramírez Olvera Guillermo.
- Sánchez Méndez Edmundo Josue.

Profesor: Cortés Galicia Jorge



Práctica 4. Administrador de procesos en Linux y Windows (2)

Introducción

En la práctica anterior utilizamos procesos para realizar los programas y ahora en esta práctica ocupamos hilos, ¿Cuál es la diferencia?, ¿Cuál usar?, ¿Hay semejanzas? Todo lo anterior y un poco mas lo veremos a continuación.

Para empezar, ¿cómo funcionan los hilos?

Un hilo es la unidad de ejecución dentro de un proceso. Un proceso puede tener desde un solo hilo hasta muchos hilos. Cuando se inicia un proceso, se le asigna memoria y recursos. Cada hilo en el proceso comparte esa memoria y recursos. En los procesos de single-thread (solo un hilo), el proceso contiene un hilo. El proceso y el hilo son uno y el mismo, y solo está sucediendo una cosa. En los procesos de multithread (múltiples hilos=, el proceso contiene más de un hilo, y el proceso está logrando una serie de cosas al mismo tiempo (técnicamente, a veces es *casi* al mismo tiempo)

Hablamos de los dos tipos de memoria disponibles para un proceso o hilo, la pila y el heap (montículo). Es importante distinguir entre estos dos tipos de memoria de proceso porque cada subproceso tendrá su propia pila, pero todos los hilos de un proceso compartirán el montículo.

Los hilos a veces se denominan procesos ligeros porque tienen su propia pila, pero pueden acceder a datos compartidos. Debido a que los hilos comparten el mismo espacio de direcciones que el proceso y otros hilos dentro del proceso, el costo operativo de la comunicación entre los hilos es bajo, lo cual es una ventaja. La desventaja es que un problema con un hilo en un proceso afectará ciertamente a otros hilos y la viabilidad del proceso en sí.

Procesos vs Hilos

PROCESOS	HILOS
Los procesos son operaciones de peso pesado.	Los hilos son operaciones de menor peso.
Cada proceso tiene su propio espacio de memoria.	Los hilos utilizan la memoria del proceso al que pertenecen.
La comunicación entre procesos es lenta ya que los procesos tienen diferentes direcciones de memoria.	La comunicación entre hilos puede ser más rápida que la comunicación entre procesos porque los hilos del mismo proceso comparten memoria con el proceso al que pertenecen.
El cambio de contexto entre procesos es más caro.	El cambio de contexto entre hilos del mismo proceso es menos costoso.

Los procesos no comparten memoria con otros procesos.	Los hilos comparten memoria con otros hilos del mismo proceso.
---	--

¿Qué pasa con la concurrencia y el paralelismo?

Una pregunta que podría hacerse es si los procesos o hilos pueden ejecutarse al mismo tiempo. La respuesta es, depende. En un sistema con múltiples procesadores o núcleos de CPU (como es común con los procesadores modernos), se pueden ejecutar múltiples procesos o hilos en paralelo. Sin embargo, en un solo procesador, no es posible tener procesos o hilos realmente ejecutándose al mismo tiempo. En este caso, la CPU se comparte entre procesos o hilos en ejecución utilizando un algoritmo de programación de procesos que divide el tiempo de la CPU y produce la ilusión de ejecución en paralelo. El tiempo asignado a cada tarea se denomina "intervalo de tiempo". El cambio entre tareas ocurre tan rápido que generalmente no es perceptible. Los términos *paralelismo* (ejecución simultánea genuina) y *concurrencia* (intercalado de procesos en el tiempo para dar la apariencia de ejecución simultánea), distinguir entre los dos tipos de operación simultánea real o aproximada.

¿Por qué debería usar un hilo en lugar de usar un proceso?

Hay dos razones de gran peso para preferir multihilo sobre multiprocesos:

- La comunicación entre hilos (compartir datos, etc.) es significativamente más sencilla de programar que la comunicación entre procesos.
- Los cambios de contexto entre hilos son más rápidos que entre procesos. Es decir, es más rápido para el sistema operativo detener un hilo y comenzar a ejecutar otro que hacer lo mismo con dos procesos.

Ahora un ejemplo práctico:

Las aplicaciones con GUI suelen utilizar un hilo para la GUI y otros para el cálculo en segundo plano. El corrector ortográfico de MS Office, por ejemplo, es un hilo independiente del que ejecuta la interfaz de usuario de Office. En tales aplicaciones, el uso de múltiples procesos daría como resultado un rendimiento más lento y un código difícil de escribir y mantener.

1. Competencias.

El alumno aprende a familiarizarse con el administrador de procesos del sistema operativo Linux y Windows a través de la creación de procesos ligeros (hilos) para el desarrollo de aplicaciones concurrentes sencillas.

2. Desarrollo

2.1. Sección Linux

2.1.1. Información de las funciones

2.1.1.1. pthread_create()

```
01. #include <pthread.h>
02.
03. int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
04. void *(*start_routine) (void *), void *arg);
```

Funcion que inicia un nuevo hilo en el proceso de invocación. Bibliotecas necesarias:

- pthread.h

Valor de retorno:

- 0 = En caso de éxito.
- '-1' = En caso de error.

2.1.1.2. pthread_join()

```
01. #include <pthread.h>
02.
03. int pthread_join(pthread_t thread, void **retval);
```

La funcion *pthread_join()* espera el hilo especificado por hilo para terminar. Si varios subprocesos simultáneamente intentan unirse con el mismo subproceso, los resultados pueden variar. Si *retval* no es NULL, *pthread_join()* copia el estado de salida del hilo objetivo (es decir, el valor que el hilo objetivo suministró a *pthread_exit()* en la ubicación apuntada por *retval* .

Bibliotecas necesarias:

- pthread.h

2.1.1.3. pthread_self()

```
01. #include <pthread.h>
02.
03. pthread_t pthread_self(void);
```

Devuelve el ID del hilo de llamada. Este es el mismo valor que se devuelve en * thread en el pthread_create llamada que creó este hilo.

Bibliotecas necesarias:

- pthread.h

2.1.1.4. pthread_exit()

```
01. #include <pthread.h>
02.
03. void pthread_exit(void *retval);
```

Finaliza el hilo de llamada y regresa un valor vía *retval* que (si el hilo se puede unir) está disponible para otro hilo en el mismo proceso que llama a `pthread_join` .

Bibliotecas necesarias:

- pthread.h

Nota: Para las funciones anteriores es necesario compilar agregando *-l pthread*

2.1.1.5. scandir()

```
01. #include <dirent.h>
02.
03. int scandir(const char *dirp, struct dirent ***namelist,
04.             int (*filter)(const struct dirent *),
05.             int (*compar)(const struct dirent **, const struct dirent **));
```

Escanea el directorio *dirp* , llamando al *filter()* en cada entrada del directorio. Las entradas para las cuales *filter()* devuelve un valor distinto de cero se almacenan en cadenas asignadas mediante *malloc*, ordenadas mediante *qsort()* con la función de comparación *compar()* y recopiladas en la lista de nombres de la matriz asignada a través de *malloc*. Si el filtro es NULL, se seleccionan todas las entradas.

Valor de retorno:

- Devuelve la cantidad de entradas de directorio seleccionadas.
- En caso de error, se devuelve -1.

Bibliotecas necesarias:

- dirent.h

2.1.1.6. stat()

```
01. #include <sys/types.h>
02. #include <sys/stat.h>
03. #include <unistd.h>
04.
05. int stat(const char *pathname, struct stat *statbuf);
```

Devuelven información sobre un archivo. Muestra el archivo al que apunta *pathname* y rellena *statbuf* . Valor de retorno:

- En caso de éxito, se devuelve cero.
- En caso de error, se devuelve -1.

Bibliotecas necesarias:

- sys/types.h
- sys/stat.h
- unistd.h

2.1.2. Ejemplo de hilos en Linux

Código(EjHilo.c)

```
01. #include <stdio.h>
02. #include <pthread.h>
03.
04. void *hilo (void *arg);
05.
06. int main (void)
07. {
08.     pthread_t id_hilo;
09.     pthread_create (&id_hilo, NULL, (void *) hilo, NULL);
10.     pthread_join (id_hilo, NULL);
11.     return 0;
12. }
13.
14. void *hilo (void *arg)
15. {
16.     printf ("Hola mundo desde un hilo en UNIX\n");
17.     return NULL;
18. }
```

Compilación y ejecución del programa:

```
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ gcc EjHilo.c -o EjHilo -l pthread
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ ./EjHilo
Hola mundo desde un hilo en UNIX
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$
```

Creamos un hilo y a través de este mandamos el mensaje 'Hola mundo desde un hilo en UNIX'.

2.1.3. Otro ejemplo de hilos en Linux

Código(EjHilo2.c)

```
01. #include <stdio.h>
02. #include <pthread.h>
03.
04. void *hilo (void *arg);
05. int resultado_hilo;
06.
07. int main (void)
08. {
09.     pthread_t id_hilo;
10.     char *mensaje = "Hola a todos desde el hilo";
11.     int *devolucion_hilo;
12.     char *valor = "hilo";
13.     printf ("%s\n", valor);
14.     pthread_create (&id_hilo, NULL, hilo, (void *) mensaje);
15.     pthread_join (id_hilo, (void **) &devolucion_hilo);
16.     printf ("Valor = %d\n", *devolucion_hilo);
17.     return 0;
18. }
19.
20. void *hilo (void *arg)
21. {
22.     char *men;
23.     men = (char *) arg;
24.     printf ("%s\n", men);
25.     resultado_hilo = 100;
26.     pthread_exit (&resultado_hilo);
27. }
```

Compilación y ejecución del programa:

```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 4$ gcc EjHilo2.c -o EjHilo2 -l pthread
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 4$ ./EjHilo2
hilo
Hola a todos desde el hilo
Valor = 100
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 4$

```

Creamos un hilo y a través de este mandamos el mensaje 'Hola a todos desde el hilo' y también modificamos un valor que tenemos en el main.

2.1.4. Árbol de hilos

Código(EstructuraArbol.h)

```

typedef struct nodo{
    int NoHilos;
    int Generacion;
    pthread_t id;
    struct nodo **subhilos;
} nodo;

nodo * nuevoHilo(){
    nodo *a = calloc(1, sizeof(nodo *));
    a->subhilos = calloc(30, sizeof(nodo *));
    return a;
}

```

Código(ArbolHilos.c)

```

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include "EstructuraArbol.h"

void *hilo(void *arg);
int totalbis=0;
int totalnieto=0;
int totalhijo=0;
int odin=0;

void Espacios(int num){
    for(int i = 0; i < num; i++)
        printf("-");
}

void PrintfGeneracion(int G, pthread_t t, int i, pthread_t aux){
    if(G == 0){
        Espacios(G);
        printf("Soy el proceso odin con Id = %li \n", t);
        odin++;
    }
    if(G == 1){
        Espacios(G);
        printf("Soy el hilo hijo odin (Generacion %i) con Id = %li vengo del proceso odin con Id = %li \n",i, t, aux);
        totalhijo++;
    }
    if(G == 2){
        Espacios(G);
        printf("Soy el hilo nieto odin (Generacion %i) con Id = %li mi padre es %li \n",i, t, aux);
        totalnieto++;
    }
    if(G == 3){
        Espacios(G);
        printf("Soy el bisnieto odin con Id = %li mi padre es %li Practica 2\n", t, aux);
        totalbis++;
    }
}

void GeneraHilos(nodo * h){
    if(!h);
    if(h->NoHilos >= 0){
        pthread_t aux[h->NoHilos];
        for(int i = 0; i < h->NoHilos; i++){
            pthread_create(&aux[i], NULL, hilo, NULL);
            h->subhilos[i]->id = aux[i];
            PrintfGeneracion(h->Generacion, aux[i], i + 1, h->id);
            GeneraHilos(h->subhilos[i]);
        }
    }
}

```

```

int main(void){
    nodo *PrinHilo = nuevoHilo();
    PrinHilo->NoHilos = 1;
    PrinHilo->Generacion = 0;
    PrinHilo->subhilos[0] = nuevoHilo();
    PrinHilo->subhilos[0]->NoHilos = 15;
    PrinHilo->subhilos[0]->Generacion = 1;
    nodo **raiz = PrinHilo->subhilos[0]->subhilos;
    for(int i = 0; i < 15; i++){
        raiz[i] = nuevoHilo();
        nodo *auxHilo = raiz[i];
        auxHilo->NoHilos = 10;
        auxHilo->Generacion = 2;
        for(int j = 0; j < 10; j++){
            auxHilo->subhilos[j] = nuevoHilo();
            auxHilo->subhilos[j]->NoHilos = 5;
            auxHilo->subhilos[j]->Generacion = 3;
            nodo **tempHilo = auxHilo->subhilos[j]->subhilos;
            for(int k = 0; k < 5; k++){
                tempHilo[k] = nuevoHilo();
            }
        }
    }
    GeneraHilos(PrinHilo);
    printf("Total odin %d\n",odin);
    printf("Total hijo odin %d\n",totalhijo);
    printf("Total nieto odin %d\n",totalnieto);
    printf("Total bisnieto odin %d\n",totalbis);
    return 0;
}

void *hilo(void *arg){
    pthread_exit(NULL);
}

```

Compilación y ejecución del código.

```

edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ gcc ArbolHilos.c -o ArbolHilos -l pthread
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ ./ArbolHilos
Soy el proceso odin con Id = 140516281915136
--Soy el hilo hijo odin (Generacion 1) con Id = 140516273522432 vengo del proceso odin con Id = 140516281915136
--Soy el hilo nieto odin (Generacion 1) con Id = 140516265129728 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516256737024 mi padre es 140516265129728 Practica 2
--Soy el bisnieto odin con Id = 140516248344320 mi padre es 140516265129728 Practica 2
--Soy el bisnieto odin con Id = 140516239951616 mi padre es 140516265129728 Practica 2
--Soy el bisnieto odin con Id = 140516231558912 mi padre es 140516265129728 Practica 2
--Soy el bisnieto odin con Id = 140516223166208 mi padre es 140516265129728 Practica 2
--Soy el hilo nieto odin (Generacion 2) con Id = 140516214773504 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516206380800 mi padre es 140516214773504 Practica 2
--Soy el bisnieto odin con Id = 140516197988096 mi padre es 140516214773504 Practica 2
--Soy el bisnieto odin con Id = 140516189595392 mi padre es 140516214773504 Practica 2
--Soy el bisnieto odin con Id = 140516181202688 mi padre es 140516214773504 Practica 2
--Soy el bisnieto odin con Id = 140516172809984 mi padre es 140516214773504 Practica 2
--Soy el hilo nieto odin (Generacion 3) con Id = 140516164417280 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516156024576 mi padre es 140516164417280 Practica 2
--Soy el bisnieto odin con Id = 140516147631872 mi padre es 140516164417280 Practica 2
--Soy el bisnieto odin con Id = 140516139239168 mi padre es 140516164417280 Practica 2
--Soy el bisnieto odin con Id = 140516130846464 mi padre es 140516164417280 Practica 2
--Soy el bisnieto odin con Id = 140516122453760 mi padre es 140516164417280 Practica 2
--Soy el hilo nieto odin (Generacion 4) con Id = 140516114061056 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516105668352 mi padre es 140516114061056 Practica 2
--Soy el bisnieto odin con Id = 140516097275648 mi padre es 140516114061056 Practica 2
--Soy el bisnieto odin con Id = 140516088882944 mi padre es 140516114061056 Practica 2
--Soy el bisnieto odin con Id = 140516080490240 mi padre es 140516114061056 Practica 2
--Soy el bisnieto odin con Id = 140516072097536 mi padre es 140516114061056 Practica 2
--Soy el hilo nieto odin (Generacion 5) con Id = 140516063704832 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516055312128 mi padre es 140516063704832 Practica 2
--Soy el bisnieto odin con Id = 140516046919424 mi padre es 140516063704832 Practica 2
--Soy el bisnieto odin con Id = 140516038526720 mi padre es 140516063704832 Practica 2
--Soy el bisnieto odin con Id = 140516030134016 mi padre es 140516063704832 Practica 2
--Soy el bisnieto odin con Id = 140516021741312 mi padre es 140516063704832 Practica 2
--Soy el hilo nieto odin (Generacion 6) con Id = 140516013348608 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140516004955904 mi padre es 140516013348608 Practica 2
--Soy el bisnieto odin con Id = 140515996563200 mi padre es 140516013348608 Practica 2
--Soy el bisnieto odin con Id = 140515988170496 mi padre es 140516013348608 Practica 2
--Soy el bisnieto odin con Id = 140515979777792 mi padre es 140516013348608 Practica 2
--Soy el bisnieto odin con Id = 140515971385088 mi padre es 140516013348608 Practica 2
--Soy el hilo nieto odin (Generacion 7) con Id = 140515962992384 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140515954599680 mi padre es 140515962992384 Practica 2
--Soy el bisnieto odin con Id = 140515946206976 mi padre es 140515962992384 Practica 2
--Soy el bisnieto odin con Id = 140515937814272 mi padre es 140515962992384 Practica 2
--Soy el bisnieto odin con Id = 140515929421568 mi padre es 140515962992384 Practica 2
--Soy el bisnieto odin con Id = 140515921028864 mi padre es 140515962992384 Practica 2
--Soy el hilo nieto odin (Generacion 8) con Id = 140515912636160 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140515904243456 mi padre es 140515912636160 Practica 2
--Soy el bisnieto odin con Id = 140515895850752 mi padre es 140515912636160 Practica 2
--Soy el bisnieto odin con Id = 140515887458048 mi padre es 140515912636160 Practica 2
--Soy el bisnieto odin con Id = 140515879065344 mi padre es 140515912636160 Practica 2
--Soy el bisnieto odin con Id = 140515870672640 mi padre es 140515912636160 Practica 2
--Soy el hilo nieto odin (Generacion 9) con Id = 140515862279936 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140515853887232 mi padre es 140515862279936 Practica 2
--Soy el bisnieto odin con Id = 140515845494528 mi padre es 140515862279936 Practica 2

```



```
--Soy el bisnieto odin con Id = 140515837101824 mi padre es 140515862279936 Practica 2
--Soy el bisnieto odin con Id = 140515828709120 mi padre es 140515862279936 Practica 2
--Soy el bisnieto odin con Id = 140515820316416 mi padre es 140515862279936 Practica 2
--Soy el hilo nieto odin (Generacion 10) con Id = 140515811923712 mi padre es 140516273522432
--Soy el bisnieto odin con Id = 140515803570928 mi padre es 140515811923712 Practica 2
--Soy el bisnieto odin con Id = 1405157995138384 mi padre es 140515811923712 Practica 2
--Soy el bisnieto odin con Id = 140515786745600 mi padre es 140515811923712 Practica 2
--Soy el bisnieto odin con Id = 140515778352896 mi padre es 140515811923712 Practica 2
--Soy el bisnieto odin con Id = 140515769960192 mi padre es 140515811923712 Practica 2
--Soy el hilo hijo odin (Generacion 2) con Id = 140515761567488 vengo del proceso odin con Id = 140516281915136
--Soy el hilo nieto odin (Generacion 1) con Id = 140515753174784 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515744782080 mi padre es 140515753174784 Practica 2
--Soy el bisnieto odin con Id = 140515736389376 mi padre es 140515753174784 Practica 2
--Soy el bisnieto odin con Id = 140515727996672 mi padre es 140515753174784 Practica 2
--Soy el bisnieto odin con Id = 140515719603968 mi padre es 140515753174784 Practica 2
--Soy el bisnieto odin con Id = 140515711211264 mi padre es 140515753174784 Practica 2
--Soy el hilo nieto odin (Generacion 2) con Id = 140515702818560 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515694425856 mi padre es 140515702818560 Practica 2
--Soy el bisnieto odin con Id = 140515686033152 mi padre es 140515702818560 Practica 2
--Soy el bisnieto odin con Id = 140515677640448 mi padre es 140515702818560 Practica 2
--Soy el bisnieto odin con Id = 140515669247744 mi padre es 140515702818560 Practica 2
--Soy el bisnieto odin con Id = 140515660855040 mi padre es 140515702818560 Practica 2
--Soy el hilo nieto odin (Generacion 3) con Id = 140515652462336 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515644069632 mi padre es 140515652462336 Practica 2
--Soy el bisnieto odin con Id = 140515635678928 mi padre es 140515652462336 Practica 2
--Soy el bisnieto odin con Id = 140515627284224 mi padre es 140515652462336 Practica 2
--Soy el bisnieto odin con Id = 140515618891520 mi padre es 140515652462336 Practica 2
--Soy el bisnieto odin con Id = 140515610498816 mi padre es 140515652462336 Practica 2
--Soy el hilo nieto odin (Generacion 4) con Id = 140515602106112 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515593713488 mi padre es 140515602106112 Practica 2
--Soy el bisnieto odin con Id = 140515585320704 mi padre es 140515602106112 Practica 2
--Soy el bisnieto odin con Id = 140515576928000 mi padre es 140515602106112 Practica 2
--Soy el bisnieto odin con Id = 140515568535296 mi padre es 140515602106112 Practica 2
--Soy el bisnieto odin con Id = 140515560142592 mi padre es 140515602106112 Practica 2
--Soy el hilo nieto odin (Generacion 5) con Id = 140515551749888 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515543357184 mi padre es 140515551749888 Practica 2
--Soy el bisnieto odin con Id = 140515534964480 mi padre es 140515551749888 Practica 2
--Soy el bisnieto odin con Id = 140515526571776 mi padre es 140515551749888 Practica 2
--Soy el bisnieto odin con Id = 140515518179072 mi padre es 140515551749888 Practica 2
--Soy el bisnieto odin con Id = 140515509786368 mi padre es 140515551749888 Practica 2
--Soy el hilo nieto odin (Generacion 6) con Id = 140515501393664 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515493000960 mi padre es 140515501393664 Practica 2
--Soy el bisnieto odin con Id = 140515484608256 mi padre es 140515501393664 Practica 2
--Soy el bisnieto odin con Id = 140515476215552 mi padre es 140515501393664 Practica 2
--Soy el bisnieto odin con Id = 140515467822848 mi padre es 140515501393664 Practica 2
--Soy el bisnieto odin con Id = 140515459430144 mi padre es 140515501393664 Practica 2
--Soy el hilo nieto odin (Generacion 7) con Id = 140515451037440 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515442644736 mi padre es 140515451037440 Practica 2
--Soy el bisnieto odin con Id = 140515434252032 mi padre es 140515451037440 Practica 2
--Soy el bisnieto odin con Id = 140515425859328 mi padre es 140515451037440 Practica 2
--Soy el bisnieto odin con Id = 140515417466624 mi padre es 140515451037440 Practica 2
--Soy el bisnieto odin con Id = 140515409073920 mi padre es 140515451037440 Practica 2
--Soy el hilo nieto odin (Generacion 8) con Id = 140515400681216 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515392288512 mi padre es 140515400681216 Practica 2
--Soy el hilo nieto odin (Generacion 9) con Id = 140515383895808 mi padre es 140515400681216 Practica 2
--Soy el bisnieto odin con Id = 140515375503104 mi padre es 140515400681216 Practica 2
--Soy el bisnieto odin con Id = 140515367110400 mi padre es 140515400681216 Practica 2
--Soy el bisnieto odin con Id = 140515358717696 mi padre es 140515400681216 Practica 2
--Soy el hilo nieto odin (Generacion 9) con Id = 140515350324992 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515341932288 mi padre es 140515350324992 Practica 2
--Soy el bisnieto odin con Id = 140515333539584 mi padre es 140515350324992 Practica 2
--Soy el bisnieto odin con Id = 140515325146880 mi padre es 140515350324992 Practica 2
--Soy el bisnieto odin con Id = 140515316754176 mi padre es 140515350324992 Practica 2
--Soy el bisnieto odin con Id = 140515308361472 mi padre es 140515350324992 Practica 2
--Soy el hilo nieto odin (Generacion 10) con Id = 140515299968768 mi padre es 140515761567488
--Soy el bisnieto odin con Id = 140515291576064 mi padre es 140515299968768 Practica 2
--Soy el bisnieto odin con Id = 140515283183360 mi padre es 140515299968768 Practica 2
--Soy el bisnieto odin con Id = 140515274796656 mi padre es 140515299968768 Practica 2
--Soy el bisnieto odin con Id = 140515266397952 mi padre es 140515299968768 Practica 2
--Soy el bisnieto odin con Id = 140515258005248 mi padre es 140515299968768 Practica 2
--Soy el hilo hijo odin (Generacion 3) con Id = 140515249612544 vengo del proceso odin con Id = 140516281915136
--Soy el hilo nieto odin (Generacion 1) con Id = 140515241219840 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140515232827136 mi padre es 140515241219840 Practica 2
--Soy el bisnieto odin con Id = 140515224434432 mi padre es 140515241219840 Practica 2
--Soy el bisnieto odin con Id = 140515216041728 mi padre es 140515241219840 Practica 2
--Soy el bisnieto odin con Id = 140515207649024 mi padre es 140515241219840 Practica 2
--Soy el bisnieto odin con Id = 140515199256320 mi padre es 140515241219840 Practica 2
--Soy el hilo nieto odin (Generacion 2) con Id = 140515190863616 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140515182470912 mi padre es 140515190863616 Practica 2
--Soy el bisnieto odin con Id = 140515174078208 mi padre es 140515190863616 Practica 2
--Soy el bisnieto odin con Id = 140515165685504 mi padre es 140515190863616 Practica 2
--Soy el bisnieto odin con Id = 140515157292800 mi padre es 140515190863616 Practica 2
--Soy el bisnieto odin con Id = 140515148900096 mi padre es 140515190863616 Practica 2
--Soy el hilo nieto odin (Generacion 3) con Id = 140515140507392 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140515132114688 mi padre es 140515140507392 Practica 2
--Soy el bisnieto odin con Id = 140515123721984 mi padre es 140515140507392 Practica 2
--Soy el bisnieto odin con Id = 140515115329280 mi padre es 140515140507392 Practica 2
--Soy el bisnieto odin con Id = 140515106936576 mi padre es 140515140507392 Practica 2
--Soy el bisnieto odin con Id = 140515098543872 mi padre es 140515140507392 Practica 2
--Soy el hilo nieto odin (Generacion 4) con Id = 140515090151168 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140515081750464 mi padre es 140515090151168 Practica 2
--Soy el bisnieto odin con Id = 140515073365760 mi padre es 140515090151168 Practica 2
--Soy el bisnieto odin con Id = 140515064973056 mi padre es 140515090151168 Practica 2
--Soy el bisnieto odin con Id = 140515056580352 mi padre es 140515090151168 Practica 2
--Soy el bisnieto odin con Id = 140515048187648 mi padre es 140515090151168 Practica 2
--Soy el hilo nieto odin (Generacion 5) con Id = 140515039794944 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140515031402240 mi padre es 140515039794944 Practica 2
--Soy el bisnieto odin con Id = 140515023009536 mi padre es 140515039794944 Practica 2
--Soy el bisnieto odin con Id = 140515014506240 mi padre es 140515039794944 Practica 2
--Soy el bisnieto odin con Id = 140515006113536 mi padre es 140515039794944 Practica 2
--Soy el bisnieto odin con Id = 140514997720832 mi padre es 140515039794944 Practica 2
--Soy el hilo nieto odin (Generacion 6) con Id = 140514989328128 mi padre es 140515249612544
--Soy el bisnieto odin con Id = 140514980935424 mi padre es 140514989328128 Practica 2
--Soy el bisnieto odin con Id = 140514972542720 mi padre es 140514989328128 Practica 2
--Soy el bisnieto odin con Id = 140514964150016 mi padre es 140514989328128 Practica 2
--Soy el bisnieto odin con Id = 140514888251136 mi padre es 140514989328128 Practica 2
--Soy el bisnieto odin con Id = 140514879858432 mi padre es 140514989328128 Practica 2
--Soy el hilo nieto odin (Generacion 7) con Id = 140514871465728 mi padre es 140515249612544
```

```
--Soy el hilo hijo odin (Generacion 15) con Id = 140509038536448 vengo del proceso odin con Id = 140516281915136
--Soy el hilo nieto odin (Generacion 1) con Id = 140509030143744 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140509021751040 mi padre es 140509030143744 Practica 2
--Soy el bisnieto odin con Id = 140509013358336 mi padre es 140509030143744 Practica 2
--Soy el bisnieto odin con Id = 140509004965632 mi padre es 140509030143744 Practica 2
--Soy el bisnieto odin con Id = 140508996572928 mi padre es 140509030143744 Practica 2
--Soy el bisnieto odin con Id = 140508988180224 mi padre es 140509030143744 Practica 2
--Soy el hilo nieto odin (Generacion 2) con Id = 140508979787520 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508971394816 mi padre es 140508979787520 Practica 2
--Soy el bisnieto odin con Id = 140508963002112 mi padre es 140508979787520 Practica 2
--Soy el bisnieto odin con Id = 140508954609408 mi padre es 140508979787520 Practica 2
--Soy el bisnieto odin con Id = 140508946216704 mi padre es 140508979787520 Practica 2
--Soy el bisnieto odin con Id = 140508937824000 mi padre es 140508979787520 Practica 2
--Soy el hilo nieto odin (Generacion 3) con Id = 140508929431296 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508921038592 mi padre es 140508929431296 Practica 2
--Soy el bisnieto odin con Id = 140508912645888 mi padre es 140508929431296 Practica 2
--Soy el bisnieto odin con Id = 140508904253184 mi padre es 140508929431296 Practica 2
--Soy el bisnieto odin con Id = 140508895860480 mi padre es 140508929431296 Practica 2
--Soy el bisnieto odin con Id = 140508887467776 mi padre es 140508929431296 Practica 2
--Soy el hilo nieto odin (Generacion 4) con Id = 140508879075072 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508870682368 mi padre es 140508879075072 Practica 2
--Soy el bisnieto odin con Id = 140508862289664 mi padre es 140508879075072 Practica 2
--Soy el bisnieto odin con Id = 140508853692960 mi padre es 140508879075072 Practica 2
--Soy el bisnieto odin con Id = 140508845504256 mi padre es 140508879075072 Practica 2
--Soy el bisnieto odin con Id = 140508837111552 mi padre es 140508879075072 Practica 2
--Soy el hilo nieto odin (Generacion 5) con Id = 140508828718848 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508820326144 mi padre es 140508828718848 Practica 2
--Soy el bisnieto odin con Id = 140508811933440 mi padre es 140508828718848 Practica 2
--Soy el bisnieto odin con Id = 140508803540736 mi padre es 140508828718848 Practica 2
--Soy el bisnieto odin con Id = 140508795148032 mi padre es 140508828718848 Practica 2
--Soy el bisnieto odin con Id = 140508786755328 mi padre es 140508828718848 Practica 2
--Soy el hilo nieto odin (Generacion 6) con Id = 140508778362624 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508769969920 mi padre es 140508778362624 Practica 2
--Soy el bisnieto odin con Id = 140508761577216 mi padre es 140508778362624 Practica 2
--Soy el bisnieto odin con Id = 140508753184512 mi padre es 140508778362624 Practica 2
--Soy el bisnieto odin con Id = 140508744791808 mi padre es 140508778362624 Practica 2
--Soy el bisnieto odin con Id = 140508736399104 mi padre es 140508778362624 Practica 2
--Soy el hilo nieto odin (Generacion 7) con Id = 140508728006400 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508719613696 mi padre es 140508728006400 Practica 2
--Soy el bisnieto odin con Id = 14050871120992 mi padre es 140508728006400 Practica 2
--Soy el bisnieto odin con Id = 140508702828288 mi padre es 140508728006400 Practica 2
--Soy el bisnieto odin con Id = 140508694435584 mi padre es 140508728006400 Practica 2
--Soy el bisnieto odin con Id = 140508686042880 mi padre es 140508728006400 Practica 2
--Soy el hilo nieto odin (Generacion 8) con Id = 140508677650176 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508669257472 mi padre es 140508677650176 Practica 2
--Soy el bisnieto odin con Id = 140508660864768 mi padre es 140508677650176 Practica 2
--Soy el bisnieto odin con Id = 140508652472064 mi padre es 140508677650176 Practica 2
--Soy el bisnieto odin con Id = 140508644079360 mi padre es 140508677650176 Practica 2
--Soy el bisnieto odin con Id = 140508635686656 mi padre es 140508677650176 Practica 2
--Soy el hilo nieto odin (Generacion 9) con Id = 140508627293952 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508618901248 mi padre es 140508627293952 Practica 2
--Soy el bisnieto odin con Id = 140508610508544 mi padre es 140508627293952 Practica 2
--Soy el bisnieto odin con Id = 140508602115840 mi padre es 140508627293952 Practica 2
```

```
--Soy el bisnieto odin con Id = 140508593723136 mi padre es 140508627293952 Practica 2
--Soy el bisnieto odin con Id = 140508585330432 mi padre es 140508627293952 Practica 2
--Soy el hilo nieto odin (Generacion 10) con Id = 140508576937728 mi padre es 140509038536448
--Soy el bisnieto odin con Id = 140508568545024 mi padre es 140508576937728 Practica 2
--Soy el bisnieto odin con Id = 140508560152320 mi padre es 140508576937728 Practica 2
--Soy el bisnieto odin con Id = 140508551759616 mi padre es 140508576937728 Practica 2
--Soy el bisnieto odin con Id = 140508543366912 mi padre es 140508576937728 Practica 2
--Soy el bisnieto odin con Id = 140508534974208 mi padre es 140508576937728 Practica 2
Total odin 1
Total hijo odin 15
Total nieto odin 150
Total bisnieto odin 750
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 4$
```

Creamos un hilo principal y después en ese hilo generamos 15 hilos con 10 hilos en cada uno, y en los 10 hilos creamos 5 hilos en cada hilo con la llamada al sistema `pthread_create()` y terminamos cada hilo `pthread_exit()`. En cada hilo se manda su respectivo `pthread_t` que es su id, el id hilo de su padre, la generación de cada hilo para el caso de los hijos y nietos de Odín, sin embargo, para los bisnietos de Odín también se muestra 'Practica 4'. Al final se colocan los valores de cuantos Odín, hijos de Odín, nietos de Odín y bisnietos de Odín se crearon para comprobar que el árbol de hilos se creo correctamente.

Nota: Debido a la cantidad de datos imprimidos en pantalla se decidió no poner todo el resultado ya que se imprimieron 919 líneas las cuales son muchas y más de la mitad del documento seria de puras capturas de pantalla.

2.1.5. Aplicación con 5 hilos (Punto 5 practica 3 pero ahora son hilos)

Código(EstructuraMatriz.h)

```
#include <stdbool.h>
#include <math.h>
typedef double* Vector;
typedef Vector* Matriz;
#define N 7

Matriz NuevaMatriz(){
    int M = N;
    if(N%1)
        M+=1;
    Matriz A = calloc(M, sizeof(Vector));
    for(int i = 0; i < N; ++i)
        A[i] = calloc(M, sizeof(double));
    return A;
}

bool esCero(double x){
    return fabs(x) < 1e-8;
}

Matriz Suma(Matriz A, Matriz B){
    Matriz C = NuevaMatriz();
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            C[i][j] = A[i][j] + B[i][j];
    return C;
}

Matriz Resta(Matriz A, Matriz B){
    Matriz C = NuevaMatriz();
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            C[i][j] = A[i][j] - B[i][j];
    return C;
}

Matriz Multiplicacion(Matriz A, Matriz B){
    Matriz C = NuevaMatriz();
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            for(int k = 0; k < N; ++k)
                C[i][j] += A[i][k] * B[k][j];
    return C;
}

Matriz Transpuesta(Matriz A){
    Matriz C = NuevaMatriz();
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            C[i][j] = A[j][i];
    return C;
}

Matriz GeneradorMatriz(){
    Matriz A = NuevaMatriz();
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            A[i][j] = rand() % 10;
    return A;
}

void EscribirMatriz(Matriz A, char* nombre){
    FILE * fp = fopen(nombre, "w");
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            fprintf(fp, "%0.3lf ", A[i][j]);
        fprintf(fp, "\n");
    fclose(fp);
}

Matriz LeerMatriz(char* nombre){
    Matriz A = NuevaMatriz();
    FILE * fp = fopen(nombre, "r");
    for(int i = 0; i < N; ++i)
        for(int j = 0; j < N; ++j)
            fscanf(fp, "%lf", &A[i][j]);
    fclose(fp);
    return A;
}

void printMatriz(Matriz A){
    for(int i = 0; i < N; ++i){
        for(int j = 0; j < N; ++j)
            printf("%0.3lf\t", A[i][j]);
        printf("\n");
    }
}
```

Código(Matriz5Hilos.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include "EstructuraMatriz.h"

void *hilo(void *arg);
Matriz A;
Matriz B;

int main(void){
    pthread_t id_hilos[6];
    srand(time(NULL));
    clock_t tInicio = clock();
    A = GeneradorMatriz();
    B = GeneradorMatriz();
    EscribirMatriz(A, "MatrizA.txt");
    EscribirMatriz(B, "MatrizB.txt");
    for(int i = 1; i <= 4; i++){
        int *opcion = malloc(sizeof(int));
        *opcion = i;
        pthread_create(&id_hilos[i], NULL, hilo, (void*)opcion);
    }
    for(int i = 1; i <= 4; i++){
        pthread_join(id_hilos[i], NULL);
    }
    for(int i = 5; i <= 5; i++){
        int *opcion = malloc(sizeof(int));
        *opcion = i;
        pthread_create(&id_hilos[i], NULL, hilo, (void*)opcion);
        pthread_join(id_hilos[i], NULL);
    }
    clock_t tFin = clock();
    printf("\nTiempo: %0.7fs\n", (double)(tFin - tInicio) / CLOCKS_PER_SEC);
    return 0;
}

void * hilo(void *arg) {
    int opcion = *((int*)arg);
    A = LeerMatriz("MatrizA.txt");
    B = LeerMatriz("MatrizB.txt");
    if(opcion == 1){
        EscribirMatriz(Suma(A, B), "Suma.txt");
    }else if(opcion == 2){
        EscribirMatriz(Resta(A, B), "Resta.txt");
    }else if(opcion == 3) {
        EscribirMatriz(Multiplicacion(A, B), "Multiplicacion.txt");
    }else if(opcion == 4){
        EscribirMatriz(Transpuesta(A), "TranspuestaA.txt");
        EscribirMatriz(Transpuesta(B), "TranspuestaB.txt");
    }else if(opcion == 5){
        printf("Matriz A:\n");
        printMatriz(LeerMatriz("MatrizA.txt"));
        printf("\nMatriz B:\n");
        printMatriz(LeerMatriz("MatrizB.txt"));
        printf("\nSuma de matrices:\n");
        printMatriz(LeerMatriz("Suma.txt"));
        printf("\nResta de matrices:\n");
        printMatriz(LeerMatriz("Resta.txt"));
        printf("\nMultiplicacion de matrices:\n");
        printMatriz(LeerMatriz("Multiplicacion.txt"));
        printf("\nTranspuesta de A:\n");
        printMatriz(LeerMatriz("TranspuestaA.txt"));
        printf("\nTranspuesta de B:\n");
        printMatriz(LeerMatriz("TranspuestaB.txt"));
    }
}
```


Compilación y ejecución del código.

```
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ gcc Matriz5Hilos.c -o Matriz5Hilos -l pthread
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$ ./Matriz5Hilos

Matriz A:
3.000  9.000  6.000  6.000  8.000  2.000  9.000
3.000  5.000  5.000  4.000  1.000  7.000  0.000
4.000  0.000  9.000  8.000  8.000  8.000  8.000
8.000  2.000  8.000  7.000  9.000  3.000  9.000
7.000  2.000  9.000  0.000  1.000  7.000  8.000
2.000  0.000  7.000  7.000  5.000  4.000  1.000
8.000  4.000  1.000  2.000  6.000  2.000  0.000

Matriz B:
4.000  0.000  0.000  5.000  3.000  0.000  2.000
4.000  3.000  2.000  1.000  7.000  3.000  3.000
9.000  0.000  1.000  3.000  0.000  0.000  0.000
7.000  6.000  1.000  5.000  0.000  4.000  7.000
6.000  6.000  9.000  3.000  9.000  9.000  8.000
2.000  0.000  2.000  8.000  5.000  6.000  9.000
3.000  9.000  4.000  4.000  2.000  7.000  7.000

Suma de matrices:
7.000  9.000  6.000  11.000  11.000  2.000  11.000
7.000  8.000  7.000  5.000  8.000  10.000  3.000
13.000  0.000  10.000  11.000  8.000  8.000  8.000
15.000  8.000  9.000  12.000  9.000  7.000  16.000
13.000  8.000  18.000  3.000  10.000  16.000  16.000
4.000  0.000  9.000  15.000  10.000  10.000  10.000
11.000  13.000  5.000  6.000  8.000  9.000  7.000

Resta de matrices:
-1.000  9.000  6.000  1.000  5.000  2.000  7.000
-1.000  2.000  3.000  3.000  -6.000  4.000  -3.000
-5.000  0.000  8.000  5.000  8.000  8.000  8.000
1.000  -4.000  7.000  2.000  9.000  -1.000  2.000
1.000  -4.000  0.000  -3.000  -8.000  -2.000  0.000
0.000  0.000  5.000  -1.000  0.000  -2.000  -8.000
5.000  -5.000  -3.000  -2.000  4.000  -5.000  -7.000

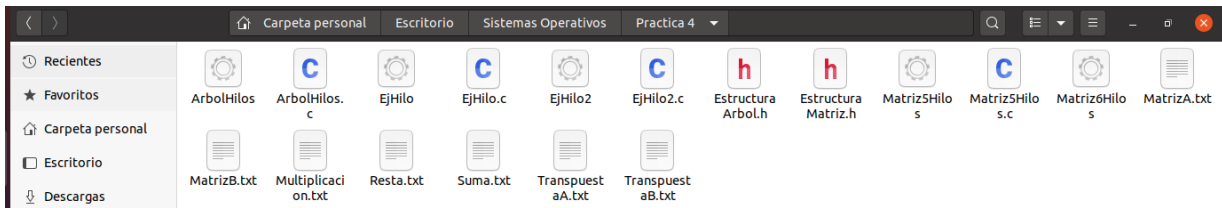
Multiplicacion de matrices:
223.000  192.000  142.000  148.000  172.000  198.000  220.000
125.000  45.000  42.000  114.000  88.000  82.000  120.000
241.000  168.000  137.000  207.000  140.000  208.000  256.000
248.000  183.000  142.000  188.000  152.000  196.000  233.000
161.000  84.000  68.000  155.000  95.000  113.000  147.000
161.000  81.000  71.000  117.000  73.000  104.000  136.000
111.000  60.000  69.000  91.000  116.000  86.000  108.000

Transpuesta de A:
3.000  3.000  4.000  8.000  7.000  2.000  8.000
9.000  5.000  0.000  2.000  2.000  0.000  4.000
6.000  5.000  9.000  8.000  9.000  7.000  1.000
6.000  4.000  8.000  7.000  0.000  7.000  2.000
8.000  1.000  8.000  9.000  1.000  5.000  6.000
2.000  7.000  8.000  3.000  7.000  4.000  2.000
9.000  0.000  8.000  9.000  8.000  1.000  0.000

Transpuesta de B:
4.000  4.000  9.000  7.000  6.000  2.000  3.000
0.000  3.000  0.000  6.000  6.000  0.000  9.000
0.000  2.000  1.000  1.000  9.000  2.000  4.000
5.000  1.000  3.000  5.000  3.000  8.000  4.000
3.000  7.000  0.000  0.000  9.000  5.000  2.000
0.000  3.000  0.000  4.000  9.000  6.000  7.000
2.000  3.000  0.000  7.000  8.000  9.000  7.000

Tiempo: 0.0018630s
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 4$
```

Carpeta donde se crearon los respectivos archivos operación generada por el respectivo hilo (aquí también se hicieron las pruebas de los programas anteriores que se realizaron):



Creamos 5 hilos diferentes con la llamada al sistema `pthread_create()` los cuales realizarán distintas operaciones con matrices en cada hilo mencionar que se tuvieron unos problemas al momento de crear y leer las matrices creadas ya que al no estar en una misma función no se puede usar de forma global ya que las matrices se arman de forma dinámica, pero el problema fue solucionado al momento de que el hilo ejecuta su tarea correspondiente. Y a diferencia de los procesos, los hilos son más tardados en el momento de ejecución.

2.2. Sección Windows

2.2.1. Ejemplo de hilos en Windows

Código(EjHiloW.c)

```
01. #include <windows.h>
02. #include <stdio.h>
03. DWORD WINAPI funcionHilo(LPVOID IpParam);
04.
05. typedef struct Informacion{
06.     int val_1;
07.     int val_2;
08. } info;
09.
10. int main(){
11.     DWORD idHilo; /*Identificador del Hilo*/
12.     HANDLE manHilo; /*Manejador del Hilo*/
13.     info argumentos;
14.     argumentos.val_1 = 10;
15.     argumentos.val_2 = 100;
16.     //Creacion Hilo
17.     manHilo = CreateThread(NULL, 0, funcionHilo, &argumentos, 0, &idHilo);
18.     //Espera la fin del hilo
19.     WaitForSingleObject(manHilo, INFINITE);
20.     printf("Valores al salir del Hilo:%i\t%i\n", argumentos.val_1, argumentos.val_2);
21.     //Cierre del manejador del hilo creado
22.     CloseHandle(manHilo);
23.     return 0;
24. }
25.
26. DWORD WINAPI funcionHilo(LPVOID IpParam){
27.     info *datos = (info *)IpParam;
28.     printf("Valores al entrar al Hilo:%i\t%i\n", datos->val_1, datos->val_2);
29.     datos->val_1 *= 2;
30.     datos->val_2 *= 2;
31.     return 0;
32. }
```

Compilación y ejecución del código.

```
C:\Users\Edmundo J Sanchez M\Desktop\SO\P4>gcc EjHiloW.c -o EjHiloW
C:\Users\Edmundo J Sanchez M\Desktop\SO\P4>EjHiloW.exe
Valores al entrar al Hilo:10    100
Valores al salir del Hilo:20    200
C:\Users\Edmundo J Sanchez M\Desktop\SO\P4>
```

Creamos un hilo y a través de este al entrar al hilo nos muestra el valor inicial de los valores después antes de salir duplicamos esos valores para que al final al salir del hilo se vea esa acción.

2.2.2. Árbol de hilos

Código(ArbolHilosW.c)

```
#include <stdio.h>
#include <windows.h>
char hiloNivel[3] = {15, 10, 5};
int hijoOdin = 0;
int nietoOdin = 0;
int bisnietoOdin = 0;
typedef struct informacion{
    DWORD Id;
    int numeroHilos, nivel;
} datosHilo;

char *spaces(int n){
    char *s = calloc(n + 1, sizeof(char));
    memset(s, '.', n);
    s[n] = '\0';
    return s;
}

datosHilo copiar(datosHilo *original){
    datosHilo temp;
    temp.Id = original->Id;
    temp.numeroHilos = original->numeroHilos;
    temp.nivel = original->nivel;
    return temp;
}

DWORD WINAPI creaHilos(LPVOID data){
    datosHilo copia = copiar((datosHilo *)data);
    copia.Id = GetCurrentThreadId();
    if (copia.nivel < 6){
        int i;
        datosHilo nuevo;
        nuevo.numeroHilos = hiloNivel[copia.nivel - 2];
        nuevo.nivel = copia.nivel + 1;
        nuevo.Id = GetCurrentThreadId();
        DWORD *idHilo = calloc(nuevo.numeroHilos, sizeof(DWORD)); //Id's Hilo
        HANDLE *manHilo = calloc(nuevo.numeroHilos, sizeof(HANDLE)); //Id's manHilo
        for (i = 0; i < nuevo.numeroHilos; i++){
            manHilo[i] = CreateThread(NULL, 0, creaHilos, &nuevo, 0, &idHilo[i]);
            if (nuevo.nivel < 5){
                printf("%sCreacion del hilo padre(Generacion %d): %d desde el proceso: %d\n", spaces(nuevo.nivel + 1), nuevo.nivel - 2, idHilo[i], copia.Id);
                nietoOdin++;
            }else{
                printf("%sCreacion de hilo final: %d desde hilo padre: %d \tPractica 2\n", spaces(nuevo.nivel + 1), idHilo[i], nuevo.Id);
                bisnietoOdin++;
            }
        }

        WaitForSingleObject(manHilo[i], INFINITE);
        if (nuevo.nivel < 5)
            printf("%sFin del hilo padre(Generacion %d): %d\n", spaces(nuevo.nivel + 1), nuevo.nivel - 2, idHilo[i]);
        CloseHandle(manHilo[i]);
    }
    return 0;
}
```

```

int main(int argc, char *argv[]){
    int nivel, i;
    char argp[30];
    if (argc < 2)
        nivel = 0;
    else
        sscanf(argv[1], "%d", &nivel);
    if (nivel == 0)
        printf("Inicio de main: %d\n", GetCurrentProcessId());
    else
        printf("%sInicio de proceso hijo: %d\n", spaces(nivel), GetCurrentProcessId());
    if (nivel < 2){
        HANDLE thread, process;
        DWORD pid;
        STARTUPINFO si;
        PROCESS_INFORMATION pi;
        ZeroMemory(&si, sizeof(si));
        si.cb = sizeof(si);
        ZeroMemory(&pi, sizeof(pi));
        sprintf(argp, "%s %d", argv[0], nivel + 1);
        CreateProcess(NULL, argp, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        process = pi.hProcess;
        thread = pi.hThread;
        pid = pi.dwProcessId;
        printf("%sCreando proceso hijo(GENERACION %d): %d desde: %d\n", spaces(nivel), nivel + 1, pid, GetCurrentProcessId());
        WaitForSingleObject(process, INFINITE);
        CloseHandle(process);
        CloseHandle(thread);
        printf("%sFin proceso hijo(GENERACION %d): %d\n", spaces(nivel + 1), nivel + 1, pid);
    }
    else{
        datosHilo info;
        info.numeroHilos = hiloNivel[0];
        info.nivel = nivel + 1;
        info.Id = GetCurrentProcessId();
        DWORD *idHilo = calloc(info.numeroHilos, sizeof(DWORD)); //Id's Hilo
        HANDLE *manHilo = calloc(info.numeroHilos, sizeof(HANDLE)); //Id's manHilo
        for (i = 0; i < info.numeroHilos; i++){
            manHilo[i] = CreateThread(NULL, 0, creaHilos, &info, 0, &idHilo[i]);
            hijoOdin++;
            printf("%sCreacion del hilo padre(GENERACION %d): %d desde el proceso: %d\n", spaces(nivel + 1), info.nivel - 2, idHilo[i], GetCurrentProcessId());
            WaitForSingleObject(manHilo[i], INFINITE);
            printf("%sFin del hilo padre(GENERACION %d): %d\n", spaces(nivel + 1), info.nivel - 2, idHilo[i]);
            CloseHandle(manHilo[i]);
        }
        printf("Hijo Odin %d\n", hijoOdin);
        printf("Nieto Odin %d\n", nietoOdin);
        printf("Bisnieto Odin %d\n", bisnietoOdin);
    }
    return 0;
}

```

Compilación y ejecución del código.

```

C:\Users\Edmundo J Sanchez\Desktop\SO\P4>gcc ArbolHilosW.c -o ArbolHilosW.exe
C:\Users\Edmundo J Sanchez\Desktop\SO\P4>ArbolHilosW.exe
Inicio de main: 3424
Creando proceso hijo(GENERACION 1): 11952 desde: 3424
..Inicio de proceso hijo: 11952
..Creando proceso hijo(GENERACION 2): 4052 desde: 11952
..Inicio de proceso hijo: 4052
....Creacion del hilo padre(GENERACION 1): 22512 desde el proceso: 4052
.....Creacion del hilo padre(GENERACION 2): 17636 desde el proceso: 22512
.....Creacion de hilo final: 22080 desde hilo padre: 17636 Practica 2
.....Creacion de hilo final: 9420 desde hilo padre: 17636 Practica 2
.....Creacion de hilo final: 20156 desde hilo padre: 17636 Practica 2
.....Creacion de hilo final: 22160 desde hilo padre: 17636 Practica 2
.....Creacion de hilo final: 16420 desde hilo padre: 17636 Practica 2
.....Fin del hilo padre(GENERACION 2): 17636
.....Creacion del hilo padre(GENERACION 2): 21476 desde el proceso: 22512
.....Creacion de hilo final: 19228 desde hilo padre: 21476 Practica 2
.....Creacion de hilo final: 18336 desde hilo padre: 21476 Practica 2
.....Creacion de hilo final: 19308 desde hilo padre: 21476 Practica 2
.....Creacion de hilo final: 19616 desde hilo padre: 21476 Practica 2
.....Creacion de hilo final: 17632 desde hilo padre: 21476 Practica 2
.....Fin del hilo padre(GENERACION 2): 21476
.....Creacion del hilo padre(GENERACION 2): 4716 desde el proceso: 22512
.....Creacion de hilo final: 2680 desde hilo padre: 4716 Practica 2
.....Creacion de hilo final: 22848 desde hilo padre: 4716 Practica 2
.....Creacion de hilo final: 14020 desde hilo padre: 4716 Practica 2
.....Creacion de hilo final: 3208 desde hilo padre: 4716 Practica 2
.....Creacion de hilo final: 6760 desde hilo padre: 4716 Practica 2
.....Fin del hilo padre(GENERACION 2): 4716
.....Creacion del hilo padre(GENERACION 2): 4860 desde el proceso: 22512
.....Creacion de hilo final: 19728 desde hilo padre: 4860 Practica 2
.....Creacion de hilo final: 11348 desde hilo padre: 4860 Practica 2
.....Creacion de hilo final: 2328 desde hilo padre: 4860 Practica 2
.....Creacion de hilo final: 528 desde hilo padre: 4860 Practica 2
.....Creacion de hilo final: 19320 desde hilo padre: 4860 Practica 2
.....Fin del hilo padre(GENERACION 2): 4860
.....Creacion del hilo padre(GENERACION 2): 19744 desde el proceso: 22512
.....Creacion de hilo final: 11828 desde hilo padre: 19744 Practica 2
.....Creacion de hilo final: 3332 desde hilo padre: 19744 Practica 2
.....Creacion de hilo final: 8292 desde hilo padre: 19744 Practica 2
.....Creacion de hilo final: 13108 desde hilo padre: 19744 Practica 2
.....Creacion de hilo final: 3628 desde hilo padre: 19744 Practica 2
.....Fin del hilo padre(GENERACION 2): 19744
.....Creacion del hilo padre(GENERACION 2): 18124 desde el proceso: 22512
.....Creacion de hilo final: 14820 desde hilo padre: 18124 Practica 2
.....Creacion de hilo final: 2568 desde hilo padre: 18124 Practica 2
.....Creacion de hilo final: 15684 desde hilo padre: 18124 Practica 2
.....Creacion de hilo final: 6504 desde hilo padre: 18124 Practica 2
.....Creacion de hilo final: 10260 desde hilo padre: 18124 Practica 2
.....Fin del hilo padre(GENERACION 2): 18124
.....Creacion del hilo padre(GENERACION 2): 10608 desde el proceso: 22512
.....Creacion de hilo final: 3064 desde hilo padre: 10608 Practica 2
.....Creacion de hilo final: 19044 desde hilo padre: 10608 Practica 2
.....Creacion de hilo final: 10384 desde hilo padre: 10608 Practica 2
.....Creacion de hilo final: 15596 desde hilo padre: 10608 Practica 2
.....Creacion de hilo final: 20364 desde hilo padre: 10608 Practica 2
.....Fin del hilo padre(GENERACION 2): 10608
.....Creacion del hilo padre(GENERACION 2): 10156 desde el proceso: 22512
.....Creacion de hilo final: 20248 desde hilo padre: 10156 Practica 2
.....Creacion de hilo final: 20988 desde hilo padre: 10156 Practica 2
.....Creacion de hilo final: 13708 desde hilo padre: 10156 Practica 2

```



```

.....Creacion de hilo final: 21096 desde hilo padre: 10156      Practica 2
.....Creacion de hilo final: 7268 desde hilo padre: 10156      Practica 2
.....Fin del hilo padre(GENERACION 2): 10156
.....Creacion del hilo padre(GENERACION 2): 13224 desde el proceso: 22512
.....Creacion de hilo final: 4756 desde hilo padre: 13224      Practica 2
.....Creacion de hilo final: 19960 desde hilo padre: 13224      Practica 2
.....Creacion de hilo final: 21880 desde hilo padre: 13224      Practica 2
.....Creacion de hilo final: 8128 desde hilo padre: 13224      Practica 2
.....Creacion de hilo final: 18356 desde hilo padre: 13224      Practica 2
.....Fin del hilo padre(GENERACION 2): 13224
.....Creacion del hilo padre(GENERACION 2): 15188 desde el proceso: 22512
.....Creacion de hilo final: 20740 desde hilo padre: 15188      Practica 2
.....Creacion de hilo final: 10540 desde hilo padre: 15188      Practica 2
.....Creacion de hilo final: 18644 desde hilo padre: 15188      Practica 2
.....Creacion de hilo final: 17924 desde hilo padre: 15188      Practica 2
.....Creacion de hilo final: 13748 desde hilo padre: 15188      Practica 2
.....Fin del hilo padre(GENERACION 2): 15188
.....Fin del hilo padre(GENERACION 1): 22512
.....Creacion del hilo padre(GENERACION 1): 21024 desde el proceso: 4052
.....Creacion del hilo padre(GENERACION 2): 14216 desde el proceso: 21024
.....Creacion de hilo final: 18476 desde hilo padre: 14216      Practica 2
.....Creacion de hilo final: 4608 desde hilo padre: 14216      Practica 2
.....Creacion de hilo final: 14252 desde hilo padre: 14216      Practica 2
.....Creacion de hilo final: 19196 desde hilo padre: 14216      Practica 2
.....Creacion de hilo final: 22280 desde hilo padre: 14216      Practica 2
.....Fin del hilo padre(GENERACION 2): 14216
.....Creacion del hilo padre(GENERACION 2): 20084 desde el proceso: 21024
.....Creacion de hilo final: 20080 desde hilo padre: 20084      Practica 2
.....Creacion de hilo final: 8280 desde hilo padre: 20084      Practica 2
.....Creacion de hilo final: 21368 desde hilo padre: 20084      Practica 2
.....Creacion de hilo final: 1132 desde hilo padre: 20084      Practica 2
.....Creacion de hilo final: 13792 desde hilo padre: 20084      Practica 2
.....Fin del hilo padre(GENERACION 2): 20084
.....Creacion del hilo padre(GENERACION 2): 21424 desde el proceso: 21024
.....Creacion de hilo final: 5516 desde hilo padre: 21424      Practica 2
.....Creacion de hilo final: 8316 desde hilo padre: 21424      Practica 2
.....Creacion de hilo final: 5552 desde hilo padre: 21424      Practica 2
.....Creacion de hilo final: 2932 desde hilo padre: 21424      Practica 2
.....Creacion de hilo final: 2016 desde hilo padre: 21424      Practica 2
.....Fin del hilo padre(GENERACION 2): 21424
.....Creacion del hilo padre(GENERACION 2): 20848 desde el proceso: 21024
.....Creacion de hilo final: 1528 desde hilo padre: 20848      Practica 2
.....Creacion de hilo final: 19056 desde hilo padre: 20848      Practica 2
.....Creacion de hilo final: 13840 desde hilo padre: 20848      Practica 2
.....Creacion de hilo final: 22460 desde hilo padre: 20848      Practica 2
.....Creacion de hilo final: 7332 desde hilo padre: 20848      Practica 2
.....Fin del hilo padre(GENERACION 2): 20848
.....Creacion del hilo padre(GENERACION 2): 6076 desde el proceso: 21024
.....Creacion de hilo final: 2132 desde hilo padre: 6076      Practica 2
.....Creacion de hilo final: 128 desde hilo padre: 6076      Practica 2
.....Creacion de hilo final: 21176 desde hilo padre: 6076      Practica 2
.....Creacion de hilo final: 17628 desde hilo padre: 6076      Practica 2
.....Creacion de hilo final: 11160 desde hilo padre: 6076      Practica 2
.....Fin del hilo padre(GENERACION 2): 6076
.....Creacion del hilo padre(GENERACION 2): 4932 desde el proceso: 21024
.....Creacion de hilo final: 9068 desde hilo padre: 4932      Practica 2
.....Creacion de hilo final: 12772 desde hilo padre: 4932      Practica 2
.....Creacion de hilo final: 11100 desde hilo padre: 4932      Practica 2
.....Creacion de hilo final: 3032 desde hilo padre: 4932      Practica 2
.....Creacion de hilo final: 1112 desde hilo padre: 4932      Practica 2
.....Fin del hilo padre(GENERACION 2): 4932
.....Creacion del hilo padre(GENERACION 2): 13916 desde el proceso: 8204
.....Creacion de hilo final: 3444 desde hilo padre: 13916      Practica 2
.....Creacion de hilo final: 21336 desde hilo padre: 13916      Practica 2
.....Creacion de hilo final: 16444 desde hilo padre: 13916      Practica 2
.....Creacion de hilo final: 476 desde hilo padre: 13916      Practica 2
.....Creacion de hilo final: 4392 desde hilo padre: 13916      Practica 2
.....Fin del hilo padre(GENERACION 2): 13916
.....Fin del hilo padre(GENERACION 1): 8204
Hijo Odin 15
Nieto Odin 150
Bisnieto Odin 750
.....Fin proceso hijo(GENERACION 2): 4052
.....Fin proceso hijo(GENERACION 1): 11952
C:\Users\Edmundo J Sanchez M\Desktop\S0\P4>

```

Primero se crea un proceso padre a partir del main, después desde el proceso padre se crea un proceso hijo. Desde el proceso hijo se crean 15 hilos, cada uno de estos crea 10 hilos y a su vez cada uno crea 5 hilos; estos imprimirán 'Practica 4'. Para crear proceso se usa la llama al sistema: *CreateProcess()* antes estudiada y para la creación de hilos se usa: *CreateThread()*. Al final se colocan los valores de cuantos hijos de Odín, nietos de Odín y bisnietos de Odín se crearon para comprobar que el árbol de hilos se creó correctamente.

Nota: Debido a la cantidad de datos imprimidos en pantalla se decidió no poner todo el resultado ya que se imprimieron poco más de 910 líneas las cuales son muchas y más de la mitad del documento sería de puras capturas de pantalla.

2.2.3. Aplicación con 5 hilos (Punto 5 practica 3 pero ahora son hilos)

Se hace uso del mismo archivo EstructuraMatriz.h para las operaciones correspondientes con las matrices, por lo que no se vuelva a agregar para no alargar sin necesidad este documento.

Código(Matriz5HilosW.c)

```
#include <stdio.h>
#include <windows.h>
#include <time.h>
#include "EstructuraMatriz.h"

void menuOperaciones(int op){
    Matriz A = LeerMatriz("A.txt");
    Matriz B = LeerMatriz("B.txt");
    if(op == 1)
        EscribirMatriz(Suma(A,B), "Suma.txt");
    else if(op == 2)
        EscribirMatriz(Resta(A,B), "Resta.txt");
    else if(op == 3)
        EscribirMatriz(Multiplicacion(A,B), "Multiplicacion.txt");
    else if(op == 4){
        EscribirMatriz(Transpuesta(A), "TranspuestaA.txt");
        EscribirMatriz(Transpuesta(B), "TranspuestaB.txt");
    }else if(op == 5){
        printf("Matriz A:\n");
        printMatriz(LeerMatriz("A.txt"));
        printf("\nMatriz B:\n");
        printMatriz(LeerMatriz("B.txt"));
        printf("\nSuma de matrices:\n");
        printMatriz(LeerMatriz("Suma.txt"));
        printf("\nResta de matrices:\n");
        printMatriz(LeerMatriz("Resta.txt"));
        printf("\nMultiplicacion de matrices:\n");
        printMatriz(LeerMatriz("Multiplicacion.txt"));
        printf("\nTranspuesta de A:\n");
        printMatriz(LeerMatriz("TranspuestaA.txt"));
        printf("\nTranspuesta de B:\n");
        printMatriz(LeerMatriz("TranspuestaB.txt"));
    }
}

DWORD WINAPI ejecutar(LPVOID dato)
{
    menuOperaciones(*(int *)dato);
    return 0;
}

int main(){
    int i;
    srand(time(NULL));
    clock_t tInicio = clock();
    Matriz A = GeneradorMatriz();
    Matriz B = GeneradorMatriz();
    EscribirMatriz(A, "A.txt");
    EscribirMatriz(B, "B.txt");
    HANDLE manHilo[6];
    DWORD hiloID[6];
    for (i = 1; i <= 4; ++i){
        int *opcion = malloc(sizeof(int));
        *opcion = i;
        manHilo[i] = CreateThread(NULL, 0, ejecutar, opcion, 0, &hiloID[i]);
    }
    for (i = 1; i <= 4; ++i){
        WaitForSingleObject(manHilo[i], INFINITE);
        CloseHandle(manHilo[i]);
    }
    for (i = 5; i <= 5; ++i){
        int *opcion = malloc(sizeof(int));
        *opcion = i;
        manHilo[i] = CreateThread(NULL, 0, ejecutar, opcion, 0, &hiloID[i]);
        WaitForSingleObject(manHilo[i], INFINITE);
        CloseHandle(manHilo[i]);
    }
    clock_t tFin = clock();
    printf("\nTiempo: %0.7fs\n", (double)(tFin - tInicio) / CLOCKS_PER_SEC);
    return 0;
}
```

Compilación y ejecución del código.

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P4>gcc Matriz5HilosW.c -o Matriz5HilosW

C:\Users\Edmundo J Sanchez M\Desktop\S0\P4>Matriz5HilosW.exe

Matriz A:
5.000  7.000  1.000  5.000  1.000  8.000  2.000
8.000  3.000  5.000  0.000  0.000  7.000  2.000
6.000  7.000  6.000  4.000  5.000  6.000  2.000
9.000  6.000  1.000  1.000  7.000  4.000  9.000
4.000  1.000  9.000  0.000  7.000  3.000  0.000
9.000  3.000  8.000  2.000  9.000  3.000  0.000
0.000  4.000  0.000  5.000  2.000  7.000  1.000

Matriz B:
2.000  9.000  8.000  7.000  0.000  7.000  8.000
3.000  2.000  7.000  0.000  5.000  9.000  4.000
9.000  7.000  9.000  5.000  2.000  9.000  1.000
0.000  5.000  9.000  8.000  3.000  6.000  2.000
5.000  0.000  0.000  5.000  0.000  0.000  3.000
7.000  1.000  8.000  3.000  0.000  4.000  9.000
8.000  4.000  3.000  2.000  1.000  8.000  9.000

Suma de matrices:
7.000  16.000  9.000  12.000  1.000  15.000  10.000
11.000  5.000  12.000  0.000  5.000  16.000  6.000
15.000  14.000  15.000  9.000  7.000  15.000  3.000
9.000  11.000  10.000  9.000  10.000  10.000  11.000
9.000  1.000  9.000  5.000  7.000  3.000  3.000
16.000  4.000  16.000  5.000  9.000  7.000  9.000
8.000  8.000  3.000  7.000  3.000  15.000  10.000

Resta de matrices:
3.000  -2.000  -7.000  -2.000  1.000  1.000  -6.000
5.000  1.000  -2.000  0.000  -5.000  -2.000  -2.000
-3.000  0.000  -3.000  -1.000  3.000  -3.000  1.000
9.000  1.000  -8.000  -7.000  4.000  -2.000  7.000
-1.000  1.000  9.000  -5.000  7.000  3.000  -3.000
2.000  2.000  0.000  -1.000  9.000  -1.000  -9.000
-8.000  0.000  -3.000  3.000  1.000  -1.000  -8.000

Multiplicacion de matrices:
117.000  107.000  213.000  113.000  54.000  185.000  172.000
135.000  128.000  192.000  106.000  27.000  172.000  162.000
170.000  144.000  241.000  151.000  61.000  223.000  177.000
180.000  145.000  191.000  141.000  44.000  220.000  237.000
148.000  104.000  144.000  117.000  23.000  130.000  93.000
165.000  156.000  207.000  173.000  37.000  186.000  150.000
79.000  44.000  132.000  73.000  36.000  102.000  104.000

Transpuesta de A:
5.000  8.000  6.000  9.000  4.000  9.000  0.000
7.000  3.000  7.000  6.000  1.000  3.000  4.000
1.000  5.000  6.000  1.000  9.000  8.000  0.000
5.000  0.000  4.000  1.000  0.000  2.000  5.000
1.000  0.000  5.000  7.000  7.000  9.000  2.000
8.000  7.000  6.000  4.000  3.000  3.000  7.000
2.000  2.000  2.000  9.000  0.000  0.000  1.000

Transpuesta de B:
2.000  3.000  9.000  0.000  5.000  7.000  8.000
9.000  2.000  7.000  5.000  0.000  1.000  4.000
8.000  7.000  9.000  9.000  0.000  8.000  3.000
7.000  0.000  5.000  8.000  5.000  3.000  2.000
0.000  5.000  2.000  3.000  0.000  0.000  1.000
7.000  9.000  9.000  6.000  0.000  4.000  8.000
8.000  4.000  1.000  2.000  3.000  9.000  9.000

Tiempo: 0.0400000s

C:\Users\Edmundo J Sanchez M\Desktop\S0\P4>
```

Carpeta donde se crearon los respectivos archivos operación generada por el respectivo hilo (aquí también se hicieron las pruebas de los programas anteriores que se realizaron):

te equipo > Escritorio > SO > P4

Nombre	Fecha de modificación	Tipo	Tamaño
A	11/11/2020 12:39 p. m.	Documento de texto	1 KB
ArbolHilosW	11/11/2020 01:04 p. m.	C Source File	4 KB
ArbolHilosW	11/11/2020 01:07 p. m.	Aplicación	45 KB
B	11/11/2020 12:39 p. m.	Documento de texto	1 KB
EjHiloW	08/11/2020 03:12 a. m.	C Source File	1 KB
EjHiloW	08/11/2020 03:12 a. m.	Aplicación	42 KB
EstructuraMatriz	11/11/2020 11:56 a. m.	C Header File	2 KB
Matriz5HilosW	11/11/2020 12:39 p. m.	C Source File	2 KB
Matriz5HilosW	11/11/2020 12:39 p. m.	Aplicación	45 KB
Multiplificacion	11/11/2020 12:39 p. m.	Documento de texto	1 KB
Resta	11/11/2020 12:39 p. m.	Documento de texto	1 KB
Suma	11/11/2020 12:39 p. m.	Documento de texto	1 KB
TranspuestaA	11/11/2020 12:39 p. m.	Documento de texto	1 KB
TranspuestaB	11/11/2020 12:39 p. m.	Documento de texto	1 KB

Como vemos solo se adaptó la aplicación para que se ejecute con hilos. Se puede observar en el tiempo de ejecución del programa que es más bajo que el de procesos, sin embargo, creo recordar que la vez que realice el de procesos tenía más programas abiertos de los que actualmente tengo, algo que sin duda alguna afecta a la ejecución del problema.

3. Análisis de la práctica

3.1 Linux

En esta práctica se hizo uso de llamadas al sistema para la creación y manipulación de hilos los cuales se ejecutan de manera paralela y concurrente, se pueden crear uno tras otro o varios en el mismo momento. Las funciones de las llamadas al sistema de los hilos son:

- Crear un hilo, varias veces el mismo o crear distintos con diferente *pthread_t*.
- Identificar un hilo con su *pthread_t*.
- Finalizar un hilo.
- Esperar a que termine un hilo.

Son diferentes la forma de crear un hilo tras otro al crear varios hilos a la vez, su manejo de los *pthread_t* son distintos. Los hilos pueden esperar a que finalice otro hilo o finalizar en un momento establecido.

3.2 Windows

Al igual que Windows se deben usar llamadas al sistema para crear hilos. Lo mínimo para crear hilos, mediante llamadas al sistema, son:

- HANDLE para manejar los hilos creados.
- Crear el hilo: *CreatThread()*.
- Función en donde se va a utilizar el hilo.

- Parámetros para la función donde se va a utilizar el hilo; que después, dentro la función se debe castear al tipo de puntero original.

4. Observaciones

4.1. Linux

- Al crear un hilo este tiene un *pthread_t* que es un identificador el cual se puede conocer con la llamada al sistema *pthread_self()*.
- Se pueden crear distintos hilos creando un arreglo de *pthread_t* con el número de hilos que se desea crear y mandar cada *pthread_t* a la llamada al sistema *pthread_create()*.
- Se debe tener cuidado al querer crear distintos hilos, ya que no funcionan de la misma manera que en la creación de procesos con la llamada al sistema *fork()* el cual genera un *id_proc* diferente si se invoca de la misma manera en diferentes líneas del código.
- Las llamadas al sistema *pthread_join()* y *pthread_exit()* para hilos son similares con las llamadas al sistema para la creación de procesos que son *wait()* y *exit()*.
- Se debe tener cuidado con la llamada al sistema *pthread_join()* si se quiere crear varios hilos diferentes porque, aunque tengamos un arreglo de *pthread_t* puede ocupar el mismo *pthread_t* en la llamada al sistema *pthread_create()*. También se debe tener cuidado con la llamada al sistema *pthread_exit()* por que puede dejar de crear los hilos antes de la cantidad solicitada.
- La mayoría de las llamadas al sistema para la manipulación de hilos se encuentran en la biblioteca "pthread.h".
- Si un hilo padre finaliza también finalizara el hilo hijo.

4.2. Windows

- Así mismo como la función *CreateProcess()*, para usar *CreateThread()* se usa la biblioteca "windows.h".
- El orden "jerárquico" de espera es que primero debe finalizar el hilo hijo antes del hilo padre, de lo contrario el hilo hijo se destruirá sin importar en que se esté ocupando.
- Es relativamente más fácil usar hilos debido a la simplicidad de la función *CreateThread()* en comparación con *CreateProcess()*, a parte se necesitan menos argumentos para *CreateThread()*.
- Es posible pasar de argumentos de una forma "indirecta" para que los trabaje el hilo.
- Se le puede especificar al hilo en que función se quiere que esté trabajando.
- La función *CreateThread* devuelve un HANDLE.
- Con los hilos solo se debe cerrar un HANDLE, que es el que devuelve la función *CreateThread*. Con procesos se debían cerrar 2 HANDLES.

- La función donde entra el hilo que se crea recibe como argumento un LPVOID que es un tipo de puntero "genérico", o sea que es un puntero a cualquier tipo de dato. Es por eso por lo que se debe castear dentro de la función al tipo de dato original que se la pasó a dicha función.
- Para que se lleve a cabo la tarea asignada al hilo en cuestión se usa la función *WaitForSingleObject()*, la misma usada para procesos.

4.3. Generales

- Los hilos reciben como argumento principal la función que se ejecutará en él. Vemos que en la creación de procesos por sustitución de código requeríamos la ruta al ejecutable, de ahí el porqué de que los hilos son "procesos ligeros".
- Si finaliza el programa principal, también lo harán todos los hilos que creó, pues son dependientes a él.
- La mejor forma de pasarle argumentos a una función dentro de un hilo es por medio de una estructura.
- Si creamos hilos dentro de un bucle *for* y a cada uno le pasamos como argumento una referencia al iterador (por ejemplo, en el programa de las matrices) no funcionará, pues la variable estará cambiando de valor y puede que cada uno de los hilos no reciba el valor que le tocaba. Para solucionarlo hay que copiar la variable en otra y pasarle la referencia de esta nueva variable al hilo.
- Para hacer que realmente los hilos se ejecuten concurrentemente, primero se crean todos los hilos necesarios y después esperamos a cada uno de ellos. Hacerlo de forma anidada es incorrecto.

5. Conclusiones

En ambos sistemas operativos Linux y Windows tenemos diversas llamadas al sistema para la creación y manipulación de hilos, ambos sistemas crean, finalizan y esperan a los hilos. Las llamadas al sistema tienen un valor de retorno el cual ayuda al control de los hilos.

Es apropiado usar hilos cuando se quieren tareas simples y cortas, por el mismo hecho que es más fácil manejar hilos que procesos, aunque es importante recalcar que los hilos son más lentos que los procesos por lo menos esto sucedió en la parte de Linux, se esperaba que ocurriera lo mismo en Windows, pero vaya sorpresa nos dio los hilos fueron más rápidos.