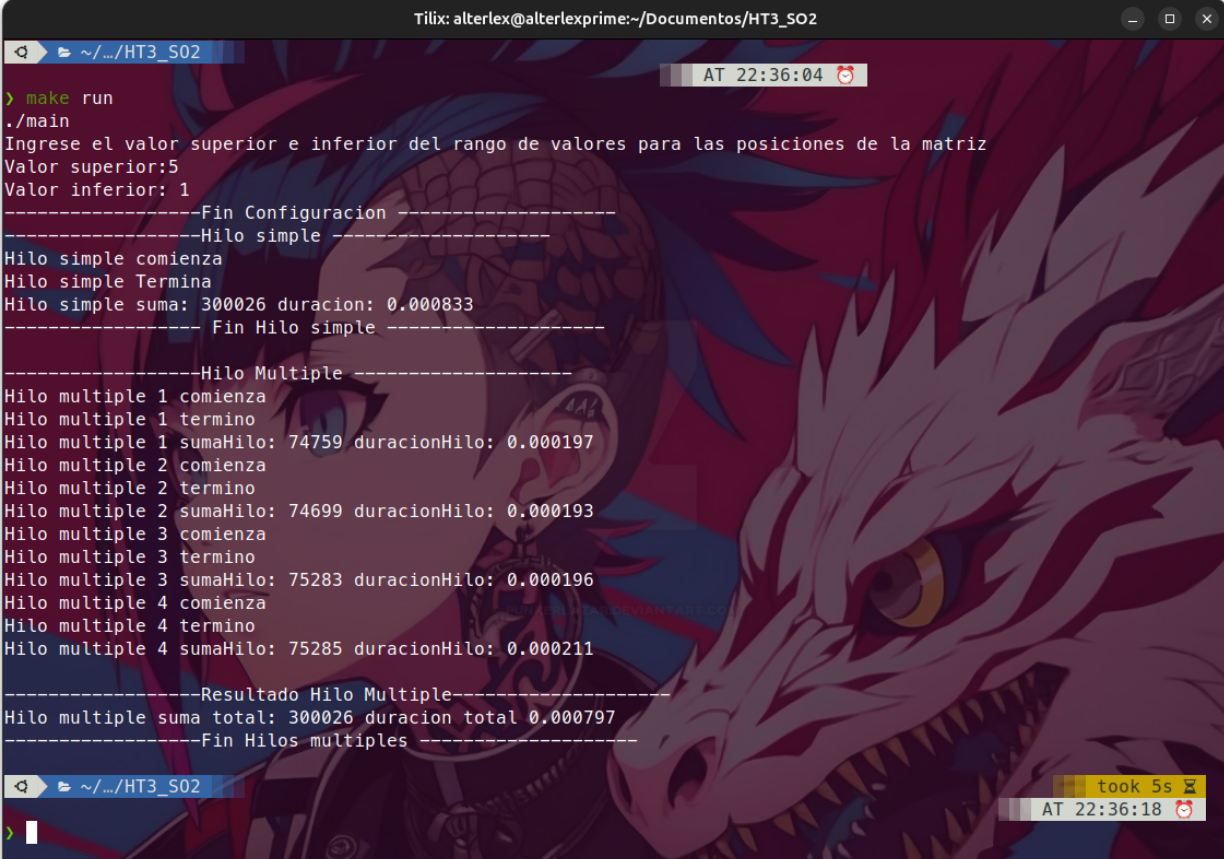


Hoja de trabajo #3

2011-14771 Otoniel Alexander Hernández Arias
Ingeniería en Ciencias y sistemas
Universidad de San Carlos de Guatemala.
Facultad de Ingeniería
Sistemas operativos 2
Primer Semestre 2025
28 de marzo de 2025

Solucion Comparacion de rangos



```
Tilix: alterlex@alterlexprime:~/Documentos/HT3_SO2
~/.../HT3_SO2
> make run
./main
Ingrese el valor superior e inferior del rango de valores para las posiciones de la matriz
Valor superior:5
Valor inferior: 1
-----Fin Configuracion -----
-----Hilo simple -----
Hilo simple comienza
Hilo simple Termina
Hilo simple suma: 300026 duracion: 0.000833
----- Fin Hilo simple -----
-----Hilo Multiple -----
Hilo multiple 1 comienza
Hilo multiple 1 termino
Hilo multiple 1 sumaHilo: 74759 duracionHilo: 0.000197
Hilo multiple 2 comienza
Hilo multiple 2 termino
Hilo multiple 2 sumaHilo: 74699 duracionHilo: 0.000193
Hilo multiple 3 comienza
Hilo multiple 3 termino
Hilo multiple 3 sumaHilo: 75283 duracionHilo: 0.000196
Hilo multiple 4 comienza
Hilo multiple 4 termino
Hilo multiple 4 sumaHilo: 75285 duracionHilo: 0.000211
-----Resultado Hilo Multiple-----
Hilo multiple suma total: 300026 duracion total 0.000797
-----Fin Hilos multiples -----
took 5s
AT 22:36:18
```

Al ejecutar el código llenamos la matriz con valores en el rango de 5 a 1 y podemos observar en la ejecución del hilo simple como en el hilo multiple es la misma suma no difieren en los valores sumados, al ejecutar los hilos multiples con mutex podemos ver que cada vez que termina un hilo imprime lo que sumo el a su variable contador global y el tiempo que duro la ejecución de dicho hilo al terminar los 4 hilos se suman solamente los tiempos de ejecución de cada hilo para la ejecución total.

En esta ejecución podemos ver la siguiente tabla con la duración de los hilos

Duracion hilo simple	Duracion hilo multiple
0.000833 s	0.000797 s

En la mayoría de rangos pequeños casi siempre, tal vez en una proporción de 6 de 10 veces el valor del hilos multiple es menor que la de un hilo simple.

Ahora con un rango mucho mas grande como mostramos en las tres siguiente capturas

```

Tilix: alterlex@alterlexprime:~/Documentos/HT3_SO2
~/.../HT3_SO2
> make run
./main
Ingrese el valor superior e inferior del rango de valores para las posiciones de la matriz
Valor superior:100
Valor inferior: 1
-----Fin Configuracion -----
-----Hilo simple -----
Hilo simple comienza
Hilo simple Termina
Hilo simple suma: 5047402 duracion: 0.000169
----- Fin Hilo simple -----

-----Hilo Multiple -----
Hilo multiple 1 comienza
Hilo multiple 1 termino
Hilo multiple 1 sumaHilo: 1258552 duracionHilo: 0.000039
Hilo multiple 3 comienza
Hilo multiple 3 termino
Hilo multiple 3 sumaHilo: 1260865 duracionHilo: 0.000039
Hilo multiple 4 comienza
Hilo multiple 4 termino
Hilo multiple 4 sumaHilo: 1262957 duracionHilo: 0.000038
Hilo multiple 2 comienza
Hilo multiple 2 termino
Hilo multiple 2 sumaHilo: 1265028 duracionHilo: 0.000151

-----Resultado Hilo Multiple-----
Hilo multiple suma total: 5047402 duracion total 0.000268
-----Fin Hilos multiples -----

```

Duracion hilo simple	Duracion hilo multiple
0.000169 s	0.000268 s

```
Tilix: alterlex@alterlexprime:~/Documentos/HT3_SO2
~/.../HT3_SO2
> make run
./main
Ingrese el valor superior e inferior del rango de valores para las posiciones de la matriz
Valor superior:100
Valor inferior: 1
-----Fin Configuracion -----
-----Hilo simple -----
Hilo simple comienza
Hilo simple Termina
Hilo simple suma: 5039165 duracion: 0.000863
----- Fin Hilo simple -----

-----Hilo Multiple -----
Hilo multiple 1 comienza
Hilo multiple 1 termino
Hilo multiple 1 sumaHilo: 1255290 duracionHilo: 0.000195
Hilo multiple 3 comienza
Hilo multiple 3 termino
Hilo multiple 3 sumaHilo: 1260157 duracionHilo: 0.000196
Hilo multiple 2 comienza
Hilo multiple 2 termino
Hilo multiple 2 sumaHilo: 1261838 duracionHilo: 0.000195
Hilo multiple 4 comienza
Hilo multiple 4 termino
Hilo multiple 4 sumaHilo: 1261880 duracionHilo: 0.000195

-----Resultado Hilo Multiple-----
Hilo multiple suma total: 5039165 duracion total 0.000780
-----Fin Hilos multiples -----

~/.../HT3_SO2
> |
```

Duracion hilo simple	Duracion hilo multiple
0.000863 s	0.000780 s

```
Tilix: alterlex@alterlexprime:~/Documentos/HT3_SO2
~/.../HT3_SO2
> make run
./main
Ingrese el valor superior e inferior del rango de valores para las posiciones de la matriz
Valor superior:100
Valor inferior: 1
-----Fin Configuracion -----
-----Hilo simple -----
Hilo simple comienza
Hilo simple Termina
Hilo simple suma: 5048065 duracion: 0.000175
----- Fin Hilo simple -----

-----Hilo Multiple -----
Hilo multiple 2 comienza
Hilo multiple 2 termino
Hilo multiple 2 sumaHilo: 1264185 duracionHilo: 0.000176
Hilo multiple 4 comienza
Hilo multiple 4 termino
Hilo multiple 4 sumaHilo: 1262741 duracionHilo: 0.000215
Hilo multiple 3 comienza
Hilo multiple 3 termino
Hilo multiple 3 sumaHilo: 1264343 duracionHilo: 0.000193
Hilo multiple 1 comienza
Hilo multiple 1 termino
Hilo multiple 1 sumaHilo: 1256796 duracionHilo: 0.000193

-----Resultado Hilo Multiple-----
Hilo multiple suma total: 5048065 duracion total 0.000776
-----Fin Hilos multiples -----
```

Duracion hilo simple	Duracion hilo multiple
0.000175 s	0.000776 s

Podemos observar que solo una vez el hilo multiple tiene una menor duracion que el simple y esto pasa un poco mas seguido que con rangos pequeños.

Observacion de rendimientos

Podemos ver que con rangos pequeños los hilos multiples duran en realizar el trabajo menos que un hilo simple pero ya en valores mas variados y mayores no pasa eso lo que si podemos observar cuando los valores son mas grandes dentro de los elementos de la matriz puede que la operacion de la suma tarde mas y eso hace una diferencia significativa en la tardanza entre la de un hilo simple y los multiples hilos.

Otro elemento que puede estar afectan en la duracion mucho mayor de los hilos multiples que el de hilo simple es que al usar mutex los hilos se bloquean los recursos compartidos por lo que solo uno puede realizar la tarea a la vez, creando un cuello de botella, y que al momento de crear un hilo el sistema operativo es el que asigna al procesador para realizar la tarea, y dependiendo de prioridad y la posicion en las lista de ejecucion del planificador este se le asigna tiempo para la ejecucion entonces esa tardanza entre realizar la tarea y pedir tiempo de ejecucion en el planificador del sistema operativo al separar la tarea se tarda mas si no hay recursos disponibles, esto podria ser el amplia diferencia que tiene la ultima captura.

Discusion sobre multihilo

Beneficios

- Los hilos multiples pueden ser beneficiosos para tareas pequeñas y rapidas, donde el paralelismo puede compensar la sobrecarga de la crecion y gestion de hilos
- El rendimiento de los hilos múltiples depende del tamaño de la tarea, la carga del sistema operativo y la disponibilidad de recursos
- Si una tarea se puede dividirse en subtares independientes se pueden aprovechar los recursos
- Pueden mejorar la eficiencia general de un programa y la capacidad de respuesta
- Si la tareas son mutuamente excluyentes los multihilos son mucho mejor que una hilo simple porque tiene una mayor eficiencia

Inconvenientes

- Al usar multihilos que acceden a recursos compartidos es necesario un mecanismo como mutex para evitar condiciones de carrera pero esto introduce un carga mayor al sistemas debido a que los hilos deben esperar a que se terminen otros
- Si se usa mucho mecanismos de sincronizacion como mutex se corre el riesgo de que en vez de que sea paralelismo se convierte en selializacion.
- Si no usaramos un metodo de sincronizacion como mutex o los otros que existen tenemos el riesgo que los recursos compartidos se corrompan y no se realice bien la tarea
- Las tareas multihilos son dificiles de depurar ya que los errores no son los mismos porque son dependientes de la ejecucion
- El diseño para realizar una tarea en multihilo puede agregar complejidad porque no es lo mismo que una tarea se realice por una unidad de trabajo que por varias unidades de trabajo.
- Los hilos multiples cuando la tareas son grandes y complejas, con sincronizacion de mutex pueden superar los beneficios del paralelismo y empeorar el rendimiento

Codigo en C

```
#define _POSIX_C_SOURCE 199309L
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>

pthread_t Hilo1;
pthread_t HiloM[4];
unsigned long counterSimple=0;
unsigned long counterMulti=0;
int matrix[100][1000];
pthread_mutex_t lock;

typedef struct
{
    int id;
    double duracionMultiple;
    unsigned long sumaHilo;
} ParmHilo;

ParmHilo *argumentos[4];

void llenarMatrix(int low,int up){
    int lower = low;
    int upper = up;
    srand(time(NULL));
    for(int i=0; i<100; i++){
        for(int j=0; j<1000; j++){
            matrix[i][j] = (rand() % (upper - lower + 1)) + lower;
        }
    }
}

void* HiloUnico(void* arg)
{
    printf("Hilo simple comienza\n");
```

```

    struct timespec Inicio, Fin;
    double duracion;

    clock_gettime(CLOCK_MONOTONIC, &Inicio);

    for(int i=0; i<100; i++){
        for(int j=0; j<1000; j++){
            counterSimple+=matrix[i][j];
        }
    }

    clock_gettime(CLOCK_MONOTONIC, &Fin);

    duracion = (Fin.tv_sec - Inicio.tv_sec) + (Fin.tv_nsec - Inicio.tv_nsec) /
1000000000.0;

    printf("Hilo simple Termina\n");

    printf("Hilo simple suma: %lu duracion: %f \n", counterSimple,duracion);

    return NULL;
}

void* HiloMutex(void* arg)
{
    pthread_mutex_lock(&lock);
    ParmHilo *p=(ParmHilo *)arg;
    printf("Hilo multiple %d comienza\n",p->id);

    struct timespec Inicio, Fin;
    double duracion;

    clock_gettime(CLOCK_MONOTONIC, &Inicio);
    int temp=(p->id-1)*25;

    for(int i=temp; i<temp+25; i++){
        for(int j=0; j<1000; j++){
            p->sumaHilo+=matrix[i][j];
            counterMulti+=matrix[i][j];
        }
    }

    clock_gettime(CLOCK_MONOTONIC, &Fin);

    duracion = (Fin.tv_sec - Inicio.tv_sec) + (Fin.tv_nsec - Inicio.tv_nsec) /
1000000000.0;

    p->duracionMultiple=duracion;

    printf("Hilo multiple %d termino\n",p->id);

    printf("Hilo multiple %d sumaHilo: %lu
duracionHilo: %f \n", p->id,p->sumaHilo,duracion);

    pthread_mutex_unlock(&lock);
    return NULL;
}

int main(void)
{
    printf("Ingrese el valor superior e inferior del rango
de valores para las posiciones de la matriz\n");
    printf("Valor superior:");
    int up=100;
    if (scanf("%d", &up)!=1){
        printf("Se pone valor default 100\n");
        up=100;
        while (getchar() != '\n');
    }
    printf("Valor inferior: ");
    int low=100;

```

```

if (scanf("%d", &low)!=1){
    printf("Se pone valor default 1\n");
    low=1;
    while (getchar() != '\n');
}
printf("-----Fin Configuracion -----\\n");

llenarMatrix(low,up);

// printf("Matriz generada\\n");
// for(int i=0; i<100; i++){
//     for(int j=0; j<1000; j++){
//         printf("%d ", matrix[i][j]);
//     }
//     printf("\\n");
// }

// -----Hilo simple -----

printf("-----Hilo simple -----\\n");

int resultado = pthread_create(&Hilo1, NULL, HiloUnico, NULL);
if (resultado != 0) {
    perror("Error al crear el hilo simple \\n");
    return EXIT_FAILURE;
}

resultado = pthread_join(Hilo1, NULL);
if (resultado != 0) {
    perror("Error al esperar al hilo simple");
    return EXIT_FAILURE;
}

printf("----- Fin Hilo simple -----\\n");
// -----Hilo multiple -----

printf("\\n-----Hilo Multiple -----\\n");
for(int i=0; i<4; i++){
    argumentos[i]=malloc(sizeof(ParmHilo));
    argumentos[i]->id=i+1;
    argumentos[i]->sumaHilo=0;
    argumentos[i]->duracionMultiple=0;
}

if (pthread_mutex_init(&lock, NULL) != 0) {
    printf("Error al crear el muter\\n");
    return EXIT_FAILURE;
}

int error[4];

for (int i = 0; i < 4; i++)
{
    error[i]=pthread_create(&HiloM[i], NULL, HiloMutex, argumentos[i]);
    if (error[i] != 0) {
        printf("Error al crear el hilo multiple %d\\n",i+1);
    }
}

for (int i = 0; i < 4; i++)
{
    error[i]=pthread_join(HiloM[i], NULL);
    if (error[i] != 0) {
        printf("Error al esperar al hilo multiple %d\\n",i+1);
    }
}

float durationMultiple=0;
for (int i = 0; i < 4; i++)
{
    durationMultiple+=argumentos[i]->duracionMultiple;
}

```

```
printf("\n-----Resultado Hilo Multiple-----\n");

printf("Hilo multiple suma total: %lu duracion total %f \n",
counterMulti,durationMultiple);
printf("-----Fin Hilos multiples -----\n");
pthread_mutex_destroy(&lock);

return 0;
}
```