

Teatro de Operaciones para Drones

1. Introducción

El objetivo del proyecto es el diseño de una solución algorítmica para un problema, haciendo uso de hilos o procesos de UNIX, para obtener un aceleramiento de la solución, con respecto a una solución de un solo proceso.

2. Planteamiento del Problema

Se tiene un *teatro de operaciones* que sirve de práctica para drones bombarderos o suicidas. El teatro de operaciones está definido como una cuadrícula de $N \times M$, donde cada cuadrícula tiene un área de 1000 metros cuadrados. En una cuadrículas del teatro de operaciones solo puede haber un objeto, y hay tres tipos de objetos: I objetivos militares (OM), II infraestructura civil (IC), y III tierra baldía (TB) Las cuadrículas están identificadas por las coordenadas (X, Y) , donde la cuadrícula $(0, 0)$, es la que se encuentra en la esquina superior izquierda. Los objetos OM y IC son caracterizados por un número que indica su resistencia al poder explosivo de los drones. Los OM tienen un valor de resistencia negativo, tal que menor sea el número mayor es su resistencia. Es decir, un OM con resistencia -5 es más fuerte que uno con valor -1. Los objetos de IC, tienen la resistencia con valor positivo, siendo la IC con mayor valor, la más resistente. La TB tiene valor de resistencia 0. Los drones caen sobre las cuadrículas. Cada dron esta caracterizado por la cuadrícula (X, Y) donde caen, por su poder explosivo (PE) que viene dado por un número entero, y por su radio de destrucción (RD), que también es un número entero. Cuando un dron cae en una cuadrícula, tiene una onda expansiva que es un área cuadrada que tiene como esquina superior izquierda a la cuadrícula de la posición $(X - RD, Y - RD)$, y como esquina inferior derecha la cuadrícula de la posición $(X + RD, Y + RD)$. Si en un área de la onda expansiva de un dron hay un objeto de tipo OM o IC, a su resistencia se disminuye sumando (a OM) o restando (a IC) el valor de PE. Cuando la resistencia de un objeto OM o IC llega a cero, entonces se dice que *el OM o IC está totalmente destruido*. Si un objeto OM o IC ha sido afectado por una onda expansiva, pero su resistencia no ha llegado a cero, entonces *el OM o IC está parcialmente destruido*. Finalmente, si un objeto OM o IC no ha sido alcanzado por una onda expansiva, se dice que *el OM o IC está intacto*. Suponemos que una TB no es afectada por la ondea expansiva de un dron.

La meta del problema es que, dado un teatro de operaciones, con los diferentes objetos en ella, se le efectúan una serie de ataques de drones, entonces se quiere saber, el número total de objetos OM y IC, que están intactos, parcialmente destruidos y totalmente destruidos.

3. Actividad a Realizar

Debe crear dos aplicaciones que resuelvan el problema descrito anteriormente, a continuación sus nombres y descripción:

teoph: Usa hilos de POSIX (librería pthreads) para acelerar la computación de la solución.

teopp: Usa procesos de Linux para acelerar la computación de la solución.

Estas aplicaciones se ejecutan con la siguiente línea de comando:

```
>./teopX [n] [instancia_a_resolver]
```

Figura 1: Descripción del comando que ejecuta las aplicaciones **teoph** y **teopp**. Observe que **teopX** se debe sustituir por **teoph** y **teopp**.

Donde:

n: Número de hilos o procesos que se deben crear para la solución del problema.

instancia_a_resolver: Archivo con la instancia a resolver.

El formato del archivo con la instancia a resolver, es como sigue. La primera línea tiene el número de filas N y el número de columnas M del teatro de operaciones, separadas por un espacio en blanco. La siguiente línea tiene el número K de objetos OM y IC, que van a estar en el teatro de operaciones. Las K líneas tienen tres números enteros separados por un espacio en blanco. Los dos primeros valores corresponde a las coordenadas X y Y de la cuadrícula donde está el objeto, y luego el valor de la resistencia del objeto. Luego, sigue una línea con el número L de drones. Cada una de las L líneas tiene cuatro números enteros separados por espacio. Los primeros dos números son las coordenadas X y Y de la cuadrícula donde va a explotar el dron. Luego, el tercer número es el radio de destrucción (RD), y el cuarto número es el poder explosivo (PE). A continuación, se presenta un ejemplo de una instancia del problema con el formato de archivo explicado anteriormente:

```
10 10
4
0 0 8
5 5 100
1 1 -2
7 7 -6
5
2 1 2 3
1 1 1 4
7 7 0 3
6 6 4 8
9 9 8 1
```

El otro argumento de la aplicación es el número n de procesos/hilos. Se tiene que n tiene que ser menor que mínimo valor entre el número de cuadrículas y el número de drones. Sea

$nT = \min(N * M, L)$, entonces se tiene que se debe cumplir que, $n \leq nT$. Si un usuario introduce un número de procesos/hilos mayor que nT , entonces la aplicación va a configurar que el número de procesos/hilos es $n = nT$.

La salida de la aplicación debe mostrar, por la salida estándar, el número de objetos OM y IC, sin destruir, parcialmente destruido, y totalmente destruido. El resultado esperado de la instancia anterior es:

```
OM sin destruir: 0
OM parcialmente destruidos: 0
OM totalmente destruido: 2
IC sin destruir: 0
IC parcialmente destruidos: 2
IC totalmente destruido: 0
```

4. Requerimientos de la Implementación

Los archivos fuentes y el Makefile de la aplicación que resuelve el problema con procesos, deben estar contenidos en una carpeta llamada `teopProcesos`, mientras que los que corresponden a la aplicación con hilos, deben estar dentro en la carpeta `teopHilos`. Tiene que diseñar sus aplicaciones, de forma tal, que *haga el mejor uso posible de los procesos/hilos para acelerar la computación de los resultados del teatro de operaciones con drones*. Su código debe compilarse y ejecutarse en un sistema de operación GNU/Linux, usando los flags dados en clase. Su código debe ser modular, legible y estar documentado.

5. Informe del proyecto

Debe realizar un *breve* informe, que tiene como objetivo la descripción de su sistema. El informe debe contener las siguientes secciones:

Portada: con la información de los autores, nombre, apellido y carné de los estudiantes

Descripción del sistema: Descripción de los sistemas para resolver del problema. Debe explicar el diseño de la solución basada en hilos y en procesos. Esta sección no debe ser mayor de dos páginas.

Referencias bibliográficas: En caso de haber utilizado referencias para la elaboración del proyecto.

6. Condiciones de la entrega

Los códigos, el informe y la declaración de autenticidad debidamente firmada, deben estar contenidos en un archivo comprimido `tar.xz`, llamado `Proy2_X_Y.tar.xz`, donde X y Y son los número de carné de los estudiantes. La entrega del archivo `Proy2_X_Y.tar.xz`, debe hacerse por medio de la plataforma *Classroom* antes de las 9:30 A.M. del día jueves 06 de marzo 2025.