



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Tarea 3

INF246 - 2022-2

Sistemas Operativos

4 de noviembre de 2022

Profesor: Viktor Tapia

Ayudante De Cátedra: Juan Pablo Varas

Ayudantes De Tarea: Javiera Cárdenas y José Runín

1. Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje Java. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de sus programas junto a cualquier indicación que sea necesaria y un archivo MAKE para poder ejecutar el programa

2. Enunciado

Una embarcación se ha hundido esta mañana en las cercanías de puerto de Valparaíso. Los tripulantes fueron encontrados luego de deambular toda la noche en un bote de rescate y al ser interrogados respecto al lugar del naufragio, solo se pudo estimar una zona probable. Usted es el responsable de encontrar la embarcación utilizando el aproximado dado por la tripulación de la manera más rápida posible, antes de que el cargamento se estropee.

Se le solicita realizar una búsqueda en base a cuadrantes utilizando **Threads** de manera recursiva como principal herramienta.

2.1. Instrucciones

El mapa de la zona será una matriz cuadrada de largo variable (10x10, 12x12, etc) que se les entregará con solo 2 símbolos en ella: 0 para indicar agua y 1 para indicar la embarcación, ilustrado de la siguiente forma:

```

0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0

```

Ejemplo de Mapa 12x12

El mapa vendrá dentro de un archivo .txt llamado *informe-nautico.txt*, el cual estará organizado de la siguiente manera:

1. Primera línea. **N**: Dimensión de la Matriz (NxN)
2. Segunda línea. **M**: Largo de la embarcación
3. Tercera línea. **Matrix**: Mapa de la zona

Deberán crear un programa en Java que utilice la clase Multithreading de manera recursiva para encontrar la embarcación. Para ello, se debe dividir el mapa en 4 cuadrantes y crear una hebra para cada uno de ellos. Luego, cada hebra debe repetir el mismo proceso hasta quedar con cuadrantes cuadrados del largo de la embarcación (**M**). Llegado a ese punto, cada hebra deberá buscar si la embarcación se encuentra en su cuadrante y la hebra que lo encuentre debe indicar su posición por consola. Esto último puede ser desde donde hasta donde se encuentra o indicando la posición del comienzo de la embarcación.

Por ejemplo, si se tiene un mapa de 12x12 y una embarcación de largo $M = 3$ (como el caso de la Figura 1), debiesen quedar 16 hebras distintas, cada una analizando un cuadrante de 3x3. Luego, la hebra que lo encuentre puede indicar la posición en cualquiera de los 2 formatos siguientes

- fila 12, columna [7,9]
- fila 12, columna 7

Finalmente, dentro del archivo excel entregado se debe realizar un breve análisis sobre si es eficiente o no la utilización de hebras para este ejercicio probando múltiples escenarios, tales como medir el tiempo de ejecución con hebras y compararlo al tiempo de como sería sin las hebras, buscando casilla por casilla. Junto con el tiempo que demora el programa, tanto con hebras como sin hebras, ejecutando solo el laboratorio o múltiples programas a la vez aparte de este. Para esto, se le solicita incluir la marca, el modelo y frecuencia del procesador de su computador junto con funcionalidades extras que este tenga (Ejemplos: Hyperthreading, Turbo boost o Overclocking)

2.2. Consideraciones

Se deben tener en cuenta las siguientes consideraciones a la hora de hacer la tarea

1. Las hebras deben ser creadas recursivamente, siempre dividiendo el mapa en 4 cuadrantes. No se aceptarán tareas que creen cada hebra *a mano*
2. Las embarcaciones siempre estarán orientadas horizontalmente, nunca verticales o diagonales
3. Las embarcaciones siempre estarán completas dentro de los cuadrantes, no quedarán mitades o tercios en otras hebras
4. De la mano con la primera consideración, las hebras deben crearse dentro de otras hebras, no en paralelo

3. Archivo de prueba

Se adjunta con el enunciado un archivo de prueba. Sin embargo, considere que al momento de que su tarea sea corregida, el o los archivos serán distintos (en contenido, no en formato).

4. Presentación Aleatoria

Para cada tarea, se seleccionarán grupos al azar para presentar su tarea frente a ayudantes y eventualmente profesor, recibiendo una ponderación del 75 % y 25 % entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente. Se comunicará días antes que grupos presentarán.

5. README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los nombres de los archivos. (Cual corresponde a cada sección de la tarea)
- Instrucciones generales de compilación y uso.

6. Consideraciones Generales

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 18 de Noviembre de 2022 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en Java. Se asume que usted sabe programar en este lenguaje, ha tenido vivencias con el, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- Las tareas serán ejecutadas en **Linux**, cualquier tarea que no se pueda ejecutar en dicho sistema operativo, partirá de nota máxima 60.
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato **TAREA3_ROL1_ROL2**.
- Las preguntas deben ser hechas por Aula o por correo, **se responderán consultas hasta 48 hrs. antes de la fecha y hora de entrega**.

- Si no se entrega README o MAKE, o si su programa no funciona, la nota es 0 hasta la corrección.
- Se descontarán 50 puntos por:
 - Mala implementación del Makefile.
 - No respetar el formato de entrega.