

INF-343 SISTEMAS DISTRIBUIDOS

TAREA N°3: “Aplicación e implementación del algoritmo de Ricart y Agrawala para asegurar exclusión mutua distribuida en un sistema de turnos para rellenar una matriz”

31 de octubre de 2023

ANTECEDENTES GENERALES

Objetivos:

- Aplicar un algoritmo de exclusión mutua distribuida para coordinar la actualización de datos compartidos por parte de procesos concurrentes.

Fechas importantes:

Ayudantía por Zoom:

Jueves, 02 de noviembre de 2023

Fecha de entrega de la tarea:

Viernes, 17 de noviembre de 2023

ESPECIFICACIÓN DE LA TAREA

Descripción

Un grupo de niños suele reunirse para colorear dibujos que son presentados como matrices de $N \times M$, donde cada elemento de la matriz es un carácter o un punto perteneciente al alfabeto inglés. El siguiente ejemplo ilustra el rostro de un robot dentro de una matriz de 11×20 . Nótese que los elementos de la matriz se separan exactamente por un espacio.

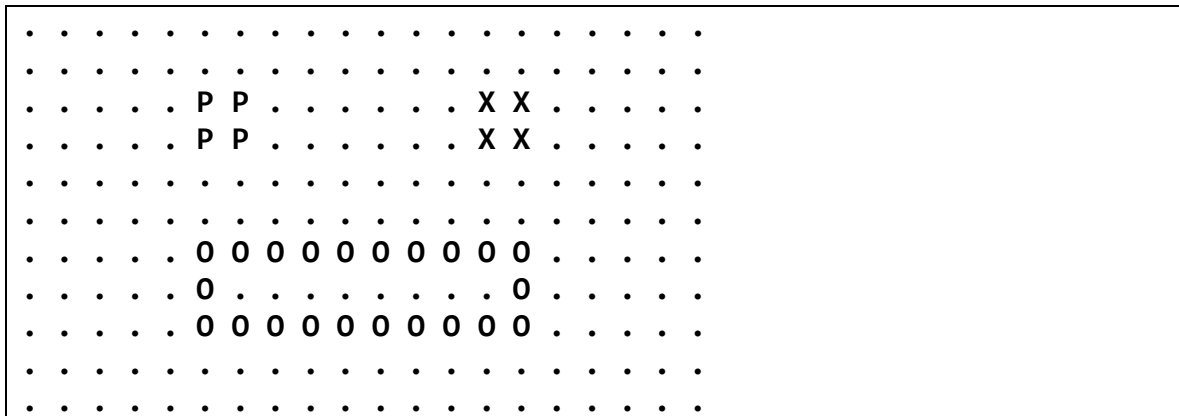
.
.
.	X	X	X	X
.	X	X	X	X
.
.
.	0	0	0	0	0	0	0	0	0	0	0
.	0	0
.	0	0	0	0	0	0	0	0	0	0	0
.
.

Los niños se turnan para pintar ciertas áreas del dibujo, pero muchas veces ocurre que dos o más niños pintan la misma área. Para coordinar a los niños, se le pide desarrollar un programa que a través del algoritmo Ricart y Agrawala permita una mejor gestión de los turnos.

Un turno se define por tres parámetros:

1. Coordenada X: Ubicación en la fila X de la matriz.
2. Coordenada Y: Ubicación en la columna Y de la matriz.
3. Color: Puede ser cualquier letra del alfabeto inglés.

La manera de pintar consiste en cambiar el color del área seleccionada a través de la coordenada considerada. Por ejemplo, si consideramos la coordenada (2, 5) y el color “P”, obtendremos el siguiente resultado:



Nótese que a pesar de que hay otra área con el mismo color “X”, esta no se pinta, ya que no está relacionada con el área seleccionada a través de la coordenada (2, 5).

Para trabajar sobre la matriz, se considerará un archivo TXT llamado "matriz.txt". Este archivo inicialmente tendrá una matriz de $N \times M$ con distintas áreas de ciertos colores. Cada vez que la matriz será pintada, este archivo tiene que ser sobrescrito.

Por otro lado, los turnos estarán listados en un archivo TXT llamado “turnos.txt”. Cada turno es una línea que contiene los siguientes valores separados por un espacio:

1. Coordenada X.
2. Coordenada Y.
3. Color.

Un ejemplo del archivo “turnos.txt” se ilustra a continuación:

2	5	P
7	5	Q
3	15	A

Debido a que es necesario saber qué niño pintó en cada turno, se deberá imprimir el siguiente mensaje por cada ocasión:

```
Niño i pintó desde (X, Y) con color C.
```

Donde “i” corresponde al número identificativo del niño, (X, Y) la respectiva coordenada y “C” el color utilizado. Para el archivo de turnos antes mencionado, una posible salida considerando 3 niños sería la siguiente:

```
Niño 1 pintó desde (2, 5) con color P.  
Niño 3 pintó desde (7, 5) con color Q.  
Niño 2 pintó desde (3, 15) con color A.
```

Las siguientes alternativas también son válidas:

```
Niño 1 pintó desde (2, 5) con color P.  
Niño 1 pintó desde (7, 5) con color Q.  
Niño 2 pintó desde (3, 15) con color A.
```

```
Niño 2 pintó desde (2, 5) con color P.  
Niño 3 pintó desde (7, 5) con color Q.  
Niño 1 pintó desde (3, 15) con color A.
```

La cantidad de niños debe ser vista como la cantidad de procesos, por lo que, si consideramos 3 niños, tendremos 3 procesos para pintar. Cada niño se enumera desde el 1 hasta la cantidad de procesos.

Desarrollo:

El programa contendrá el desarrollo en un único archivo llamado main.go, donde el ejecutable considerará los siguientes parámetros:

- Cantidad de procesos a considerar.
- Cantidad de filas de la matriz.
- Cantidad de columnas de la matriz.
- Nombre del archivo de la matriz.
- Nombre del archivo de los turnos.

La instrucción para construir el ejecutable será la siguiente:

```
go build main.go
```

Luego, para ejecutar:

```
./main 3 11 20 matriz.txt turnos.txt
```

La ejecución debe generar en la salida estándar una nueva línea cada vez que un proceso sea capaz de pintar desde las coordenadas seleccionadas.

Adicionalmente, el archivo TXT que contiene la matriz debe quedar actualizado hasta el último pintado.

Nota:

- No es necesario crear un algoritmo para pintar las áreas de la matriz, se recomienda utilizar un algoritmo ya desarrollado. Se sugiere investigar acerca de “*Flood Fill*”: https://en.wikipedia.org/wiki/Flood_fill

Consideraciones:

- No es necesario realizar ninguna validación para los datos de entrada ni tampoco para los archivos TXT.
- La matriz no superará las dimensiones 100 x 100.
- El número de procesos será entre 1 y 3.
- Solo se considerarán letras del alfabeto inglés y el carácter de punto.
- Dos o más procesos **no deben** pintar la misma área dada la coordenada de entrada. Solo un único proceso pinta el área considerada.
- La versión de Go es la 1.19.X.
- Para consultas, escriba a hugo.leyton@usm.cl o comuníquese a través del servidor de Discord.

Reglas de la tarea:

- La tarea se entrega en grupos de 2 personas.
- La fecha de entrega es el día **17 de noviembre a las 23:55 horas**.
- La tarea debe ser subida al mismo repositorio de las tareas anteriores en <https://gitlab.inf.utfsm.cl/>. Debe crear una nueva rama llamada tarea_3.
- En el README del repositorio, incluya el nombre y apellido de cada integrante. Así como también las instrucciones que estime conveniente para la ejecución de la tarea.