

## Objetivos del Proyecto

1. Resolver un problema concurrente utilizando procesos en Linux (creados a través de fork)
2. Resolver el mismo problema utilizando la librería de hilos de POSIX.
3. Utilizar llamadas al sistema para el manejo de procesos, hilos y archivos.:

## Descripción General

En [álgebra lineal numérica](#) una matriz dispersa, matriz *sparse* o matriz rala es una [matriz](#) de gran tamaño en la que la mayor parte de sus elementos son cero. [Weisstein, Eric W.](#) «Matriz dispersa» (en inglés). [MathWorld. Wolfram Research](#). La idea del proyecto es implementar un programa que, de forma concurrente, determine si una matriz es o no dispersa. El programa se implementará con procesos e hilos. A continuación, se presentan detalles de la implementación.

## Entradas

Los estudiantes realizarán dos programas ejecutables con nombres: **pdispersa** y **hdispersa** que implementarán la concurrencia con procesos e hilos, respectivamente. Cada programa recibirá como argumentos de entrada las dimensiones de la matriz, el archivo donde se encuentra almacenada la matriz (en formato texto), el número de procesos/hilos concurrentes que se deben crear para revisar la matriz y el porcentaje de 0's (del total de elementos), necesario para determinar que la matriz es dispersa. Los comandos se invocarán de la siguiente forma:

### Para el programa implementado con procesos:

**\$/pdispersa -f M -c N -a Narchivo -n Nprocesos -p porcentaje**

Donde:

**M:** número de filas de la matriz que se encuentra almacenada en archivo

**N:** número de columnas de la matriz almacenada en archivo

**Narchivo:** nombre de archivo que contiene, en formato texto, la matriz que se va a examinar. En BS los estudiantes encontrarán al menos dos ejemplos de archivos con matrices. Los estudiantes deben respetar este formato, porque el día de la sustentación los profesores proveerán los archivos de prueba.

**Nprocesos:** número de procesos que se crearán para revisar el contenido de la matriz.

**porcentaje:** número entre 0 y 100 que indica el % de elementos con valor igual a 0 (cero) que debe tener la matriz para ser considerada dispersa. Este valor será un número entero.

*Los elementos -f -c -a -n y -p son selectores que permiten identificar cuál es la función de cada parámetro de entrada del comando. Al usar estos selectores los comandos pueden venir en cualquier orden y su programa debe poder leerlos y ejecutarlos sin importar en qué orden vienen, por ejemplo:*

```
$/pdispersa -f 100 -c 80 -a matriz1.txt -n 4 -p 80
```

*Debe dar los mismos resultados que invocar el programa con:*

```
$/pdispersa -p 80 -n 4 -c 80 -f 100 -a matriz1.txt
```

Para el programa implementado con hilos:

**`$/hdispersa -f M -c N -a Narchivo -n Nhilos -p porcentaje`**

Donde:

**M:** número de filas de la matriz que se encuentra almacenada en archivo

**N:** número de columnas de la matriz almacenada en archivo

**Narchivo:** nombre de archivo que contiene, en formato texto, la matriz que se va a examinar. En BS los estudiantes encontrarán al menos dos ejemplos de archivos con matrices. Los estudiantes deben respetar este formato, porque el día de la sustentación los profesores proveerán los archivos de prueba.

**Nhilos:** número de hilos que se crearán para revisar el contenido de la matriz de forma concurrente.

**porcentaje:** número entre 0 y 100 que indica el % de elementos con valor igual a 0 (cero) que debe tener la matriz para ser considerada dispersa. Este valor será un número entero.

## Salidas

Cada uno de los programas dará como salida el porcentaje (%) de 0's de la matriz, y la conclusión si es o no dispersa.

## Ejemplos

### Invocación #1.

```
$/pdispersa -f 4 -c 4 -a matrizcorta -n 2 -p 80
```

La invocación de pdispersa, recibe como matriz de entrada con 4 filas y 4 columnas. Se crean dos procesos (n=2) para evaluarla. La matriz tiene 16 elementos (4x4), se considera dispersa si el número de ceros es mayor o igual a 13 (80% de 16 es 12.8, pero se redondea al siguiente número entero).

**Salida: La matriz en el archivo matrizcorta tiene un total de 13 ceros (80%), por lo tanto, se considera dispersa.**

### Invocación #2.

```
$/hdispersa -f 10 -c 4 -a matrizA -n 3 -p 60
```

La invocación de hdispersa, recibe como matriz de entrada con 10 filas y 4 columnas. Se crean 3 hilos (h=3) para evaluarla. La matriz tiene 40 elementos (10x4), se considera dispersa si el número de ceros es mayor o igual a 24 (60% de 40 es 24). La idea es repartir de la forma más equitativa posible las filas o columnas entre las entidades concurrentes.

**Salida: La matriz en el archivo matrizA tiene un total de 10 ceros (25%), por lo tanto, no se considera dispersa.**

## Funcionamiento de los programas

El programa principal leerá la matriz del archivo que recibe como argumento y una vez almacenada la matriz en la memoria, la dividirá entre el número de hilos (threads) o procesos trabajadores indicados por el usuario. La idea es dividir la matriz entre los trabajadores (hilos/procesos), de la forma más equitativa posible, para que ellos busquen la cantidad de elementos distintos de cero en un determinado sector de la matriz. En la figura 1, se muestran dos posibilidades de división de una matriz entre 3 procesos trabajadores. En la **Figura 1-A** se le reparten 3 columnas a cada proceso trabajador y en

la **Figura 1-B** se le reparten 3 filas a cada trabajador. Los estudiantes decidirán cómo hacer la división. Cada proceso tomará sus filas o columnas, contará el número de elementos distintos de cero y está información la devolverá al proceso padre. El padre, que tiene las dimensiones de la matriz, el porcentaje y la respuesta de cada hijo, decide si la matriz es dispersa o no, e imprime una respuesta por la consola.

p1	p2	p3
0 1	0 0	0 0
1 0	0 1	0 0
0 1	0 0	1 0
0 0	1 0	0 1
1 0	1 0	0 1
0 0	0 0	1 1

A

p1	0 1 0 0 0 0
	1 0 0 1 0 0
p2	0 1 0 0 1 0
	0 0 1 0 0 1
p3	1 0 1 0 0 1
	0 0 0 0 1 1

B

**Figura 1**

En el caso de la **Figura 1A** el proceso **P1** devuelve al padre el valor 4 (número de elementos distintos de 0), el proceso **P2** devuelve 3 y el proceso **P3** devuelve 5. El padre totaliza la cantidad de elementos distintos de 0 y conoce el total de elementos de la matriz (36). Por tanto, en función del porcentaje indicado por el usuario decide si la matriz es o no dispersa. Los hijos (hilos o procesos) devuelven los elementos distintos de 0, porque se supone que este número será menor en las matrices dispersas; si es un número menor a 254 se podrá devolver el valor del hijo al padre a través del `exit()` (ver en el manual de la llamada al sistema el mayor entero positivo que se puede devolver).

### Comunicación Padres-Hijos

El padre le debe pasar a los hijos las filas o columnas correspondientes. Para el caso de la Figura 1A, serían:

P1 = 0, 1  
P2 = 1, 2  
P3 = 2, 3

Puede pasar también la fila o columna de inicio y cuántas filas o columnas le corresponden a partir de ese valor.

### Comunicación Hijos-Padres

Cada hijo devolverá al padre el número de elementos distintos de ceros, en la parte de la matriz que le correspondió revisar según la división realizada. Para el caso de los procesos, este número se devolverá en el `exit`, solo si es menor o igual a 254. En caso de que sea 255 o mayor, se devolverá 255 al padre para indicar que la información se encuentra en un archivo. Padres e hijos deben conocer el nombre de los archivos donde compartirán la información. En el caso de los hilos, cada hijo puede devolver el valor en un parámetro o a través de variables globales.

### Implementación

La implementación se realizará en lenguaje C (ansi C). Aquellos estudiantes que implementen el proyecto en C++ deben usar las mismas llamadas al sistema vistas en clase, así como la misma librería de hilos. En el modelo a implementar, el proceso maestro (padre) lee los argumentos que recibe por la línea de comandos y crea tantos procesos hijos como se le



hubiera indicado. A cada proceso le envía los parámetros necesarios y luego espera por las respuestas de todos los hijos. Una vez que tiene las respuestas, totaliza y escribe la salida correspondiente por la consola. El mismo modelo aplica para la programación usando hilos.

### Observaciones Adicionales

Deben validarse los parámetros de entrada. Revise las llamadas al sistema: **fork**, **wait**, **waitpid**, **read**, **write**, **open**, **close**, **fread**, **fwrite**, **perror**.

El proyecto lo deben realizar en grupos de cómo máximo tres estudiantes.

El día 27 de mayo deben cargar en Bright Space los archivos fuente del proyecto (.c y .h) y el archivo Makefile que permita automatizar la compilación del proyecto. Todos estos archivos deben colocarse en la plataforma en un archivo formato targz antes de las 11:59pm, en su defecto se recomienda que se entregue el enlace al repositorio. Este archivo también debe contener un pequeño informe de no más de 3 páginas que explique: cómo realizó la división de la matriz, qué información se pasa del padre a los hijos y de los hijos al padre y cómo se pasan la información: por medio de estructuras de datos, archivos, etc. En todos los casos, mencione el nombre de las estructuras de datos o archivos utilizados para la comunicación. Puede realizar figuras que aclaren la explicación. En el informe debe respetar aspectos de presentación, ortografía y redacción. Cualquier duda consulte las recomendaciones a los monitores o los profesores. En cada entrega se debe hacer individual (enlace a su repositorio personal), aún cuando se haya trabajado en equipo.