Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Sistemas Operativos, SIU 4085.
julio-diciembre 2018

Proyecto 1 (20%)

Objetivos del Proyecto

Al finalizar el proyecto el estudiante estará en capacidad de:

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. Comparar el desempeño de hilos y procesos al variar parámetros cómo el tamaño de la carga y los niveles de concurrencia.
4. Establecer comparaciones entre el resultado de la práctica y los conceptos de la clase de teoría.
5. Utilizar llamadas al sistema para el manejo de procesos, hilos y archivos.

Nota: el desarrollo de los puntos 3 y 4 se realizará en una actividad de taller, donde se les suministrarán los programas ya implementados y se les indicará las medidas que deben realizar. **Esta actividad tendrá un valor de 5% y la aplicación concurrente que van a desarrollar y que se describe en este enunciado, tiene un valor del 15% de la calificación total de la asignatura.**

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 de dos formas: con procesos e hilos. A continuación se presentan detalles de la implementación.

Entradas

Se realizarán dos programas ejecutables *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, el número de procesos 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:

```
$ ./pdispersa M N archivo nprocesos porcentaje
```

Donde:

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

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

archivo: archivo que contiene en formato texto la matriz que se va a examinar. El día de la sustentación se probará con archivos generados a través del programa ***matriz.c*** que se colocará como parte de los anexos del proyecto en UVirtual.

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

porcentaje: numero entre 0 y 100 que indica el % de ceros que debe tener la matriz para ser considerada dispersa. Este valor será un número entero.

En el caso del programa implementado con hilos, se invocará de la siguiente forma:

```
$ hdispersa M N archivo nhilos porcentaje
```

Teniendo los parámetros el mismo significado, salvo el parámetro *nhilos* que indica el número de hilos a ser creados para trabajar en forma concurrente.

Salidas

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

Ejemplos

A continuación se colocan ejemplos para la invocación de los programas:

```
$ pdispersa 4 4 matrizcorta 2 80
```

En este caso, la matriz que se recibe como entrada tiene 4 filas y 4 columnas y se crearán dos procesos para evaluarla. Dado que la matriz tiene 16 elementos, se considerará dispersa si el número de ceros es mayor o igual 13. El 80% de 16 es 12,8, pero tomaremos por redondeo el número 13. Cuando se usa redondeo, a cada número real se le asigna el número entero más próximo según su parte decimal. La salida del programa sería algo como:

La matriz en el archivo *matrizcorta* tiene un total de 13 ceros, por tanto, se considera dispersa. Se requieren 13 ceros.

```
$ hdispersa 10 4 matrizA 3 60
```

En este caso se invoca el programa que implementa hilos. La matriz no es cuadrada (10 filas y 4 columnas). Para que la matriz sea dispersa, 24 elementos o más deben ser ceros. Se deben crear tres hilos para trabajar sobre la matriz (en la siguiente sección se explica cómo se puede dividir el trabajo). La idea es repartir, de la forma más equitativa posible, las filas o columnas entre las entidades concurrentes. Suponiendo que la matrizA tiene sólo 10 ceros, la respuesta del programa debe ser:

La matriz en el archivo *matrizA* tiene un total de 10 ceros, por tanto, no se considera dispersa. Se requieren 24 ceros o más para considerarla dispersa.

Funcionamiento de los Programas

El programa principal leerá la matriz del archivo que recibe cómo argumento y una vez almacenada la matriz en memoria, la dividirá entre el número de threads o procesos trabajadores indicados por el usuario. La idea es dividir la matriz en partes más o menos iguales entre los trabajadores 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 proceso. Los estudiantes decidirán cómo hacer la división, siempre que sea lo más equitativa posible. 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 cónsola.

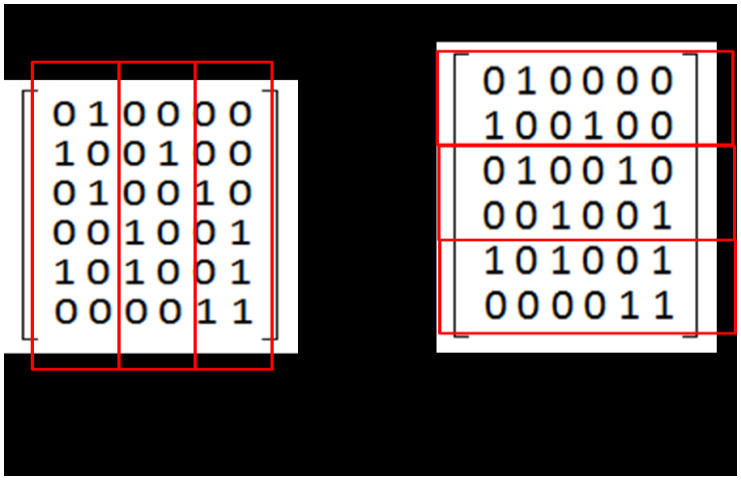

Figura 1

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 y sabe que de 36 elementos 12 son distintos de 0. En función del porcentaje indicado por el usuario decide si la matriz es o no dispersa. Se 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 del hijo al padre a través del exit() (ver en el manual 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 devuelve su valor en una variable compartida entre padre e hijo.

Implementación

La implementación se realizará en lenguaje C (ansi C) siguiendo un modelo maestro-esclavo donde:

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 cónsola. 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 dos estudiantes**.

El día 20 de septiembre para el grupo de los jueves y el 21 de septiembre para el grupo de los viernes, deben subir a UVirtual 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 2 pm. 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.

Utilice la guía de estilo C, que se encuentra en la carpeta Información, archivo ***estiloC.pdf*** para realizar un código más claro, modular, etc. Es obligatorio seguir las recomendaciones de la guía.