

Proyecto 1

En este proyecto debe resolver tres problemas relacionados con programación de procesos, hilos e interrupciones en UNIX.

Problema 1

Se requiere que usted desarrolle un programa que recibe como entrada una lista de N enteros positivos n_0, n_1, \dots, n_{N-1} y determina si cada uno de estos enteros n_i es primo o no. El objetivo de este problema es que realice una evaluación de desempeño del programa, donde compare el tiempo de corrida utilizando diferente número de procesos e hilos en Linux.

Llamada

Para correr el programa se utilizará el siguiente formato:

`primos entrada.txt [-t | -p] [-n N]`

`entrada.txt` es el nombre de archivo de entrada

`-p` o `-t` indican que el programa deberá utilizar procesos o hilos, respectivamente.

N es el número de procesos/hilos a crear. Siempre se debe cumplir $1 \leq N \leq 10$

El archivo de entrada es un archivo de texto donde cada línea tendrá un entero positivo. El archivo termina en fin de línea.

Los archivos de salida son archivos de texto donde cada línea tendrá dos enteros a y b separados por un espacio. a es un entero del archivo de entrada y b es 1 si a es primo y 0 en caso contrario. Como el trabajo se va a dividir entre varios procesos/hilos, puede haber más de un archivo de salida.

Arquitectura

A continuación se describe la arquitectura que deberá tener el programa, tanto para la versión de procesos como para la versión de hilos.

El programa tendrá un proceso/hilo maestro y N procesos/hilos trabajadores, que llamaremos t_0, t_1, \dots, t_{N-1} .

Antes de crear los procesos trabajadores, el maestro debe dividir el trabajo entre los N trabajadores. Si el archivo de entrada tiene M líneas, a cada trabajador t_0, t_1, \dots, t_{N-2} le corresponden $M \div N$ líneas. Al trabajador t_{N-1} le corresponden $(M \div N) + (M \bmod N)$ líneas.

El trabajador t_i debe procesar las líneas que le fueron asignadas e imprimir la salida en un archivo llamado $i.txt$, donde i es el número de trabajador. El proceso/hilo maestro, antes de terminar, debe esperar a que terminen todos los trabajadores. Esto es necesario para la correcta medición del tiempo de corrida.

Problema 2

En este problema, se requiere que usted modifique el programa de hilos de la pregunta anterior para que imprima el número total de primos hallados. La entrada tendrá el siguiente formato:

primosn entrada.txt -n N

El hilo maestro deberá imprimir por la salida estándar el número total de primos hallados.

Problema 3

En este problema usted deberá modificar el programa de procesos del problema 1 para que cuando el usuario presione Ctrl-C, el programa imprima:

```
I'm sorry Dave. I'm afraid I can't do that.
```

Cada vez que el usuario pulse Ctrl-C, la frase sólo se debe imprimir una vez (no se debe repetir).

Detalles de implementación

El proyecto debe ser desarrollado en Linux en lenguaje C.

Usted debe hacer un Makefile para compilar su proyecto. El proyecto se debe poder compilar solamente ejecutando la orden `make`.

Para el problema 1, debe entregar un informe con una comparación de desempeño donde reporte:

- Procesos
 - Tiempo de corrida con 1, 2, ..., 10 procesos
- Hilos
 - Tiempo de corrida con 1, 2, ..., 10 hilos

Cada uno de los tiempos de corrida que reporte debe ser el promedio de 10 corridas.

Observe si hay una variación en el tiempo de corrida al aumentar el número de procesos (o hilos). Compare el tiempo de corrida de la versión de procesos con la versión de hilos. Incluya una discusión de estos resultados donde explique el porqué de los resultados obtenidos. Incluya en el informe el diseño de los experimentos, resultados, discusión de los resultados y conclusiones.

Entrega

El código fuente del proyecto debe ser entregado en un archivo tar.gz. Antes de crear el tar.gz, ejecute la orden `make clean`, es decir, elimine los archivos de objeto y el ejecutable de su programa.