

## Proyecto 1: Búsqueda Concurrente de Números Primos

20%

### ***Objetivos General del Proyecto***

El objetivo de este proyecto es que el estudiante resuelva un problema en forma concurrente utilizando procesos e hilos. Una vez realizada la implementación con ambos tipos de entidades concurrentes, el estudiante comparará el desempeño de las soluciones concurrentes con la solución secuencial. A través de este proyecto el estudiante también se familiarizará con llamadas al sistema para el manejo de archivos utilizando el lenguaje de programación C.

### ***El problema a Resolver***

El estudiante desarrollará tres programas para el cálculo de los números primos en el rango de **1 a N**. El primer programa realizará el cálculo de forma secuencial (sin entidades concurrentes), el segundo de forma concurrente usando hilos y el tercero de forma concurrente usando procesos Unix. Los nombres de los programas serán ***primsec***, ***primhilos*** y ***primproc***

### ***Entradas y Salidas de los Programas***

El programa secuencial se invocará de la siguiente forma:

```
$ ./primsec N fileoutput
```

Donde:

- **N**: define el límite superior del rango donde se buscarán los números primos.
- **fileoutput**: nombre del archivo de texto donde se escribirán los números primos resultantes. Deben estar escritos en orden ascendente.

```
$ ./primhilos N numhilos fileoutput
```

Los parámetros **N** y **fileoutput** permanecen similares. El parámetro **numhilos** indica el número de hilos que serán creados para colaborar en la búsqueda de números primos; debe ser un número menor o igual a N.

```
$ primproc N numproc fileoutput
```

Los parámetros **N** y **fileoutput** permanecen similares. El parámetro **numproc** indica el número de hilos que serán creados para colaborar en la búsqueda de números

Ejemplos de invocación de los comandos son:

**\$ ./primproc** //invocación errónea, faltan los argumentos.

**\$ ./primproc -h** // Imprime informacion de que hace el programa y que parameros deben usarse para su correcta ejecución

**\$ ./primsec 10000 salida** //El programa busca e imprime en el archivo **salida** los primos que se encuentran entre 1 y 10000.

**\$ ./primhilos 5000000 4 salida.out** //El programa busca e imprime en el archivo **salida.out** los primos que se encuentran entre 1 y 5000000. Debe dividir el dominio de búsqueda entre 4 hilos.

**\$ ./primproc 899 10 output** //El programa busca e imprime en el archivo **output** los primos que se encuentran entre 1 y 899. Debe dividir el dominio de búsqueda entre 10 procesos.

**\$ ./primproc 20 2 archivosal** // El programa busca e imprime en el archivo **output** los primos que se encuentran entre 1 y 100. Debe dividir el dominio de búsqueda entre 2 procesos.

En las busquedas no se cuenta ni el primer valor (1), ni el ultimo del rango especificado

En este ultimo caso, el archivo **archivosal** debe contener los siguientes números:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 y 97

Estos son todos los **primos** menores que 100.

## ***Características de cada uno de los programas.***

### **Primsec**

Este es un programa en C donde Ud. empleará la búsqueda de primos existentes en un rango de números enteros comprendidos del 2 al N-1. Los números resultantes los colocará en el archivo cuyo nombre se recibe como argumento de entrada.

## Primhilos

Este programa realizará el cálculo de números primos en forma concurrente. Para ello dividirá el rango entre el número de hilos a crear ( $N/numhilos$ ) y a cada hilo le informa del subrango que le ha correspondido. Ejemplo: si el rango es de 0 a 100 y se ha solicitado la creación de 2 hilos, el primero buscará números primos entre 2 y 49 y el segundo explorará el rango de números de 50 al 99.

El proceso padre (el que crea todos los hilos), espera que ellos terminen y escribe los resultados en el archivo cuyo nombre recibió como parámetro. En la implementación de esta solución los estudiantes deben aprovechar el hecho de que los hilos se comunican a través de memoria compartida; de esta forma la solución será más eficiente.

## Primproc

Este programa también realizará el cálculo de números primos en forma concurrente. Para ello dividirá el rango entre el número de procesos a crear ( $N/numproc$ ) y a cada proceso le informa del subrango que le ha correspondido.

El proceso padre espera que los hijos terminen y escribe los resultados en el archivo que recibió como parámetro. Existen muchos mecanismos de comunicación entre procesos (pipes, memoria compartida, etc), pero en este proyecto usaremos archivos. Existen varias opciones para hacer esta implementación:

1. Cada proceso hijo deja los números primos encontrados en un archivo y el proceso padre, procesa todos estos archivos para producir una salida única en el archivo cuyo nombre se recibe como argumento de entrada.
2. Todos los procesos escriben su salida a un mismo archivo. Esto puede generar contención y el archivo no queda ordenado. El padre debe ordenar la salida, ya que es un requisito del proyecto.

## Detalles de la Implementación

La implementación se realizará en lenguaje C (ansi C). Para la creación de los hilos se utilizará la librería POSIX. Deben seguir en su código las recomendaciones de la Guía de Programación en C publicada en UVirtual (**archivo estiloC.pdf, en la carpeta de información sobre el curso**). Se recomienda el uso del compilador gcc, si Ud. usa otro compilador asegúrese de que su código esté en ansi C siguiendo las recomendaciones de la guía de estilo.

Si alguno de los programas crea archivos temporales debe eliminarlos.

## Evaluación del Rendimiento de los Programas

Una vez desarrollados los tres programas, los estudiantes deberán comparar el desempeño de las tres implementaciones para distintos valores de N y un número determinado de hilos y procesos (en el caso concurrente). Las métricas a comparar serán los tiempos de respuesta de los programas, pero también pueden observar cómo es el uso del CPU y de la memoria en cada implementación. Para medir el tiempo pueden usar el comando **time**. También se le proveerá de un programa que permite medir el uso de la memoria. **La comparación con la solución secuencial sólo la realizará con 4 entidades concurrentes (hilos o procesos).**

	N			
Numero de Hilos	100	1K	10K	100K
4	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria
6	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria
12	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria	Tiempo total de ejecución, Uso CPU, uso de Memoria

Una vez realizadas las medidas debe colocar los resultados en un informe que contenga no más de 5 páginas y los siguientes puntos:

- Identificación, título, etc.
- Descripción del problema planteado.
- Para los programas concurrentes debe describir las estructuras de datos y/o mecanismos de comunicación utilizados. ¿Cómo los procesos hijos transmiten información a los padres?
- Tablas de datos donde se observen los valores de cada medición realizada.
- Gráficos donde se pueda observar el comportamiento de las soluciones (la figura 1 es un ejemplo de gráfico con datos ficticios).
- Análisis de los resultados obtenidos. ¿Los resultados son consistentes con los conceptos de la clase teórica? ¿Cuál de las tres soluciones es más eficiente? Proporcione una explicación para los resultados obtenidos. Indique si utilizó alguna herramienta o instrucción adicional para indagar o explorar más detenidamente sobre los comportamientos obtenidos.

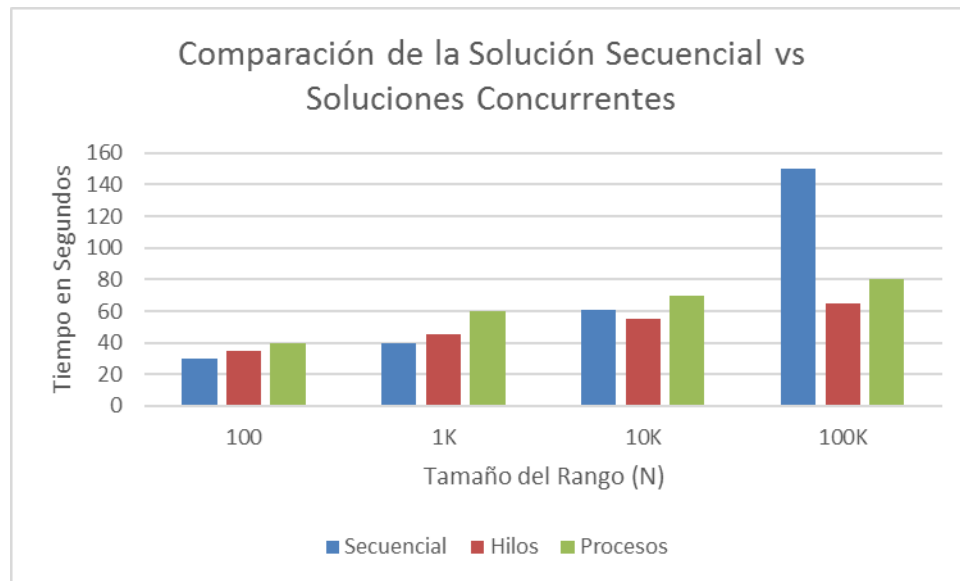


Figura 1: Ejemplo de gráfico donde se muestran tiempos de los tres programas para diferentes tamaños de N.

### ¿Qué debe entregar y cómo debe hacerlo?

- El proyecto lo deben entregar el miércoles de la semana 8 (miércoles 8 de septiembre) antes de las 12 mm, por UVirtual. La sustentación se realizará en la siguiente hora de práctica. Si el proyecto no está en UVirtual a las 12 mm no se corregirá. Deberán colocar en UVirtual:
  - Un Makefile que permita producir los ejecutables a partir de sus archivos fuentes.
  - Los archivos fuente de su programa.
  - El informe en formato pdf.

**Coloque estos programas y documentos en un archivo .tar.gz que tenga el nombre de los integrantes del equipo.**
- Cada archivo fuente debe estar debidamente identificado y cada función debidamente documentada. Deben validarse todas y cada una de las llamadas al sistema y los argumentos de la entrada.

### Observaciones Adicionales

- Revise las llamadas al sistema: *fork*, *wait*, *waitpid*, *fread*, *fwrite*, *fopen*, *fclose*, *unlink()* y aquellas relacionadas con la manipulación de hilos.
- El proyecto lo deben realizar en grupos de **como máximo dos** estudiantes.
- Recuerde que, si se descubre copia, el proyecto, será calificado con la nota de 1pto y el caso será elevado a la decanatura de la facultad.
- El día de la sustentación se evaluará también la interacción con el sistema operativo Linux. Todos los estudiantes deben estar en la capacidad de correr su proyecto incluso de forma individual. Si un estudiante no maneja el sistema adecuadamente se considerará que no maneja el proyecto y que su

participación fue sólo marginal. Esto es independiente de las preguntas que realice el profesor sobre el contenido del proyecto.

**Nota:** Cualquier duda sobre el enunciado del proyecto debe consultarla con los profesores en forma oportuna. La comprensión del problema y su correcta implementación, según lo indica el enunciado, es parte de lo que se está evaluando.

Suerte

Elaborado por los Profesores  
Ricardo González y Mariela Curiel