

Presentado por:

- Carlos Barón
- Andrés Cocunubo

INFORME-PROYECTO 1- SISTEMAS OPERATIVOS

Descripción del problema.

Construir tres algoritmos que permiten hallar los números de primos entre uno y un número máximo N dado. El primer algoritmo es secuencial, es decir, no usa entidades concurrentes, el segundo y el tercero utilizan entidades concurrentes usando hilos y procesos, respectivamente. Para luego, ser analizados y observar cual tiene mejor desempeño. Adicionalmente, se busca aprender el uso de llamadas al sistema en el lenguaje C.

MECANISMOS DE COMUNICACIÓN UTILIZADOS

- En hilos: Para la creación de un hilo se debe utilizar la función *pthread_create*, en la cual recibe en uno de sus parámetros una estructura de tipo (void*), donde se puede pasar la información que necesite el hilo. Además, los hilos poseen memoria compartida, por lo tanto, pueden compartir variables con otros hilos y el programa principal.
 - MEMORIA COMPARTIDA: Los resultados de los primos obtenidos en cada hilo se van almacenando en una variable compartida en memoria la cual es un arreglo de tipo entero (int *r) y tamaño $N/2$. A cada hilo se le asigna un espacio o rango en dicho arreglo y las posiciones en este se mueven independientemente en cada uno.
 - ESTRUCTURA PARÁMETROS: El trabajo que realiza cada hilo requiere el inicio y el fin de los primos a calcular, por lo tanto, se utiliza una estructura con los siguientes datos:
 - Estructura parámetros {Entero start, Entero end}.
- En Procesos: Para la creación de un proceso se utiliza la función *fork*, la cual crea procesos idénticos al proceso padre llamados hijos, la función retorna cero al hijo y el *id* al padre. También se utiliza la función *wait* que hace que el proceso padre espere a que todos sus hijos terminen de ejecutarse. En este caso se usaron archivos como medio de comunicación.
 - ARCHIVOS: Para cada uno de los procesos hijos se crea un archivo en donde se guardarán los números primos según el rango que le correspondió, y luego de que cada hijo termine de ejecutarse el padre los unifica y borra los archivos creados por los hijos.

TABLA DE DATOS

• HILOS

N	10000			100000			200000			500000			1000000		
Numero de hilos	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)
4	0,075	16	0,001	0,675	33296	0,04	2,514	33492	0,15	14,245	34080	0,81	54,621	35056	2,92
6	0,066	16	0,001	0,679	49688	0,04	2,448	49884	0,15	13,531	50472	0,81	50,799	51448	2,81
12	0,125	98684	0,001	0,737	98864	0,04	2,440	99060	0,15	13,392	99648	0,81	50,508	100624	2,81
20	0,185	164252	0,001	0,861	164432	0,04	2,442	164628	0,15	13,357	165216	0,79	50,205	166192	2,69
Promedio	0,11275	65742	0,001	0,738	86570	0,04	2,461	86766	0,15	13,63125	87354	0,805	51,53325	88330	2,8075

• PROCESOS

N	10000			100000			200000			500000			1000000		
Numero de Procesos	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)
4	0,126	304	0,001	1,718	304	0,03	5,169	304	0,07	22,451	304	0,38	76,318	304	1,09
6	0,133	304	0,001	1,785	304	0,04	5,414	304	0,07	22,168	304	0,32	68,023	304	0,96
12	0,251	304	0,001	1,982	304	0,04	5,719	304	0,08	20,571	304	0,30	67,718	304	0,90
20	0,515	304	0,001	2,178	304	0,04	6,235	304	0,11	20,309	304	0,26	66,806	304	0,84
Promedio	0,25625	304	0,001	1,91575	304	0,0375	5,63425	304	0,0825	21,37475	304	0,315	69,71625	304	0,9475

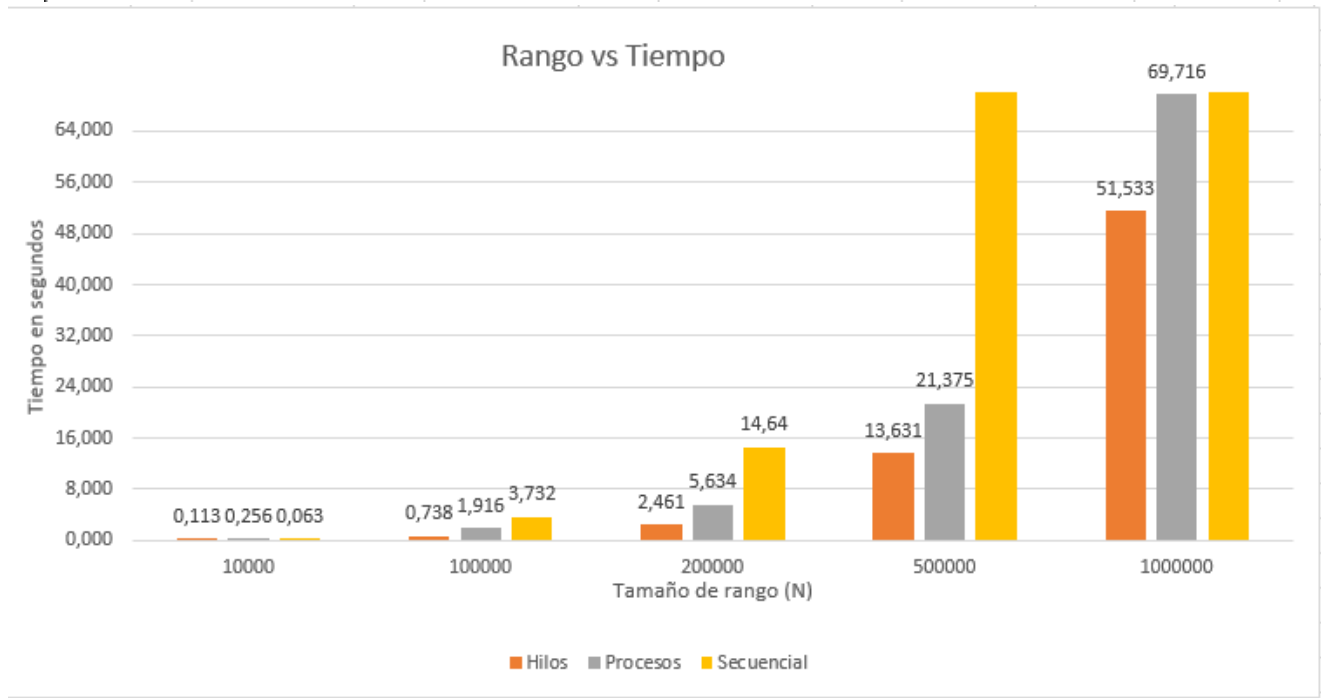
• SECUENCIAL

10000			100000			200000			500000			1000000		
Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)	Tiempo Utilizado(s)	Uso de memoria(KB)	Uso CPU(%)
0,063	16	0,001	3,732	304	0,06	14,64	304	0,25	93,082	304	1,54	373,338	304	5,75

GRÁFICOS

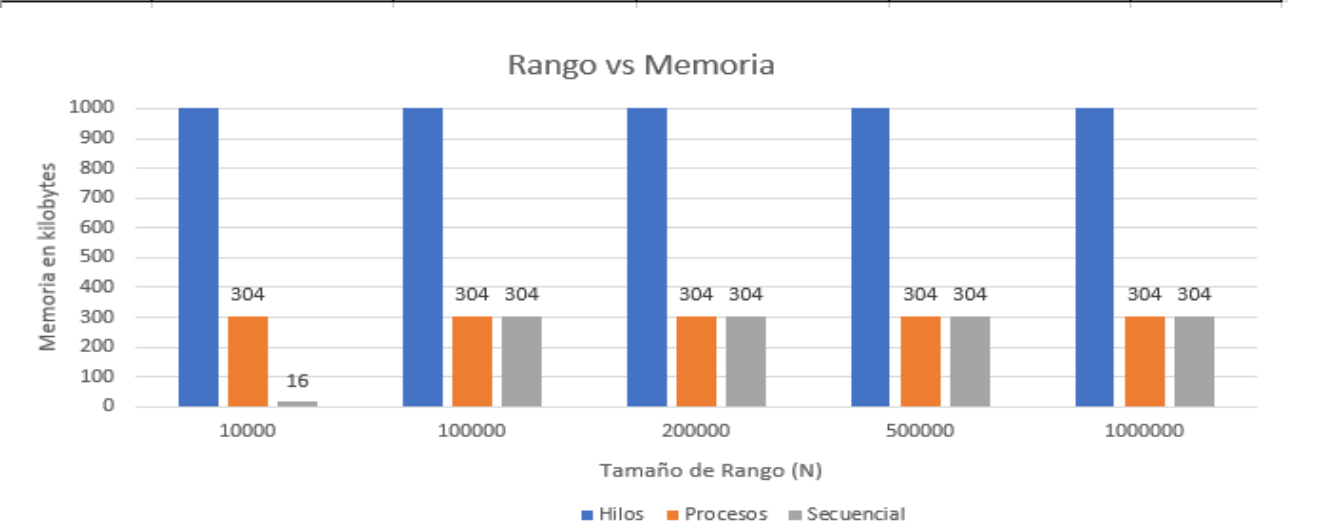
• TIEMPO UTILIZADO

Tiempo Utilizado(s)					
Rango	10000	100000	200000	500000	1000000
Hilos	0,113	0,738	2,461	13,631	51,533
Procesos	0,256	1,916	5,634	21,375	69,716
Secuencial	0,063	3,732	14,64	93,082	373,388



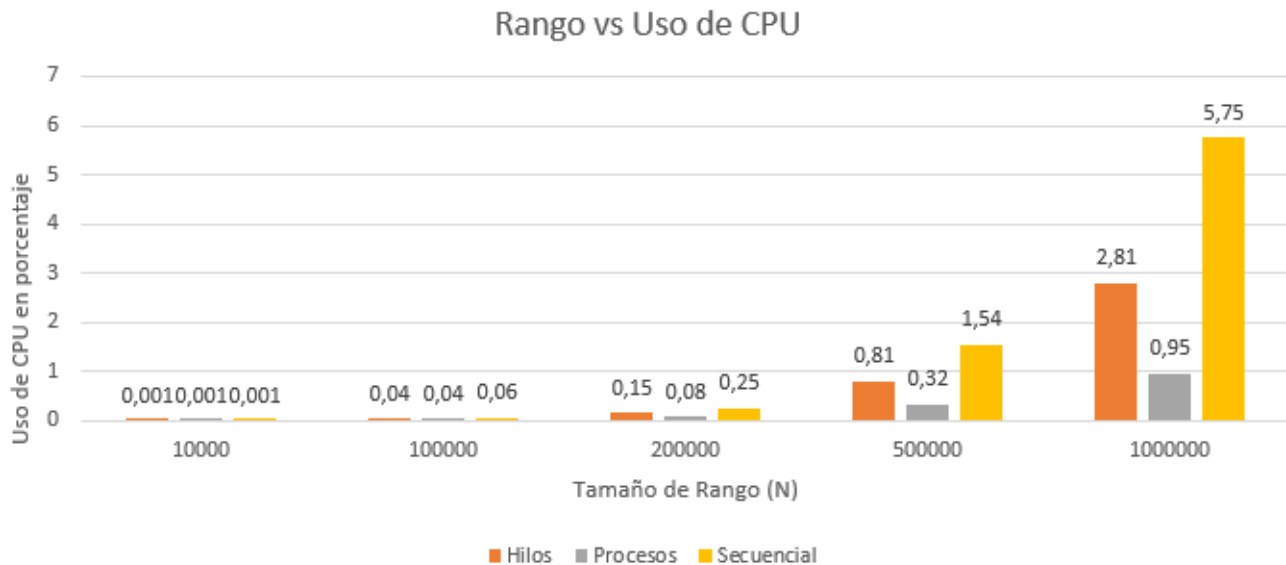
• USO DE MEMORIA

Uso Memoria(KB)					
Rango	10000	100000	200000	500000	1000000
Hilos	65742	86570	86766	87354	88330
Procesos	304	304	304	304	304
Secuencial	16	304	304	304	304



- **USO DE CPU**

Uso de CPU(%)					
Rango	10000	100000	200000	500000	1000000
Hilos	0,001	0,04	0,15	0,81	2,81
Procesos	0,001	0,04	0,08	0,32	0,95
Secuencial	0,001	0,06	0,25	1,54	5,75



ANÁLISIS DE LOS RESULTADOS

Herramientas e instrucciones adicionales:

- **Cálculo de tiempo de ejecución por un programa:** Para calcular que tardan cada uno de los programas se utilizó el comando *time* en la terminal de Linux, el cual recibe como argumento el programa al cual se le calculará el tiempo de ejecución. Al finalizar se muestra como resultado tres datos: el tiempo real, el tiempo de usuario y el tiempo de sistema, en minutos y segundos. Ejemplo de uso: `time ./primhilos`.
- **Cálculo de memoria utilizada por un programa:** Para calcular la memoria que consume un programa, se provee de un programa escrito en C++ (Por: Ginés G.M, basado en el código de la función *SafeExec*), el cual hace una estimación (aproximada) de la máxima cantidad de memoria utilizada por un programa cualquiera. Ejemplo de uso: `./memory ./primhilos`.
- **Cálculo del uso de la CPU:** Para calcular el uso de CPU por cada programa se hace uso del comando *pidstat*, en puntos clave del programa (al terminar cada hilo, proceso o instantes antes de terminar la ejecución del programa en el caso secuencial). Para hacer este llamado al sistema se hace uso de *-G* el cual permite buscar las estadísticas de la ejecución de un proceso mediante su nombre, obtenido mediante la compilación del programa. Ejemplo de uso: `pidstat -G primhilos`.

Análisis Teórico vs Experimental:

La creación de un proceso es más costosa de lanzar, debido a que los procesos no comparten memoria entre ellos por lo tanto se necesita crear una copia de la memoria en el programa, mientras que los hilos al poseer memoria compartida mucho más ligeros en cuanto a su creación. Para los procesos realizar cambios de contexto es una operación compleja, por lo tanto, cada proceso debe realizar sus propias llamadas al sistema, mientras que en los hilos esta tarea es mucho más fácil porque pueden conmutar entre ellos rápidamente puesto que los recursos no cambian. Finalmente se puede decir que la creación, comunicación y uso del sistema en procesos es mucho más compleja y lenta que en hilos.

En cuanto a los datos obtenidos acerca del tiempo se puede observar que para cálculos pequeños el uso de algoritmos secuenciales es más eficiente que el uso de algoritmos por concurrencia. Esto se da ya que un algoritmo secuencial solo se encarga de realizar el cálculo, mientras que un algoritmo concurrente se encarga de realizar el cálculo y crear sus respectivas entidades, lo cual consume más tiempo de ejecución. Sin embargo, para realizar operaciones más grandes el tiempo de creación de entidades es despreciable y es más eficiente utilizar algoritmos concurrentes que pueden ir realizando cálculos a la par para una respuesta más rápida. Como se mencionó anteriormente un proceso es mucho más pesado de crear que un hilo por lo que los hilos siempre dan una respuesta más rápida independientemente del cálculo a realizar.

Acerca del uso de memoria, es importante recalcar que los hilos poseen una comunicación rápida entre ellos, por lo que utilizan memoria compartida. En la práctica se pudo observar que tanto procesos, como el algoritmo secuencial utilizan una cantidad de memoria mínima ya que solo se encargan de realizar sus respectivas operaciones, lo cual produce una respuesta más lenta, mientras, que los hilos utilizan una cantidad de memoria mucho mayor debido a que los resultados de sus operaciones se van almacenando en variables compartidas, para brindar una respuesta más rápida.

Finalmente, sobre el uso de la CPU, se evidencia que en cálculos pequeños e intermedios el uso de la CPU es muy similar, aunque, siempre se observa que los procesos consumen menos, la diferencia se puede apreciar en operaciones muy grandes que requieren bastante uso del CPU, puesto que a medida que un proceso se termina se va desocupando espacio en el CPU. Mientras que, en los hilos a pesar de que se dé la sensación de que existan múltiples programas en realidad solo son etapas del mismo programa. En el algoritmo secuencial se realiza todo en un mismo programa y se da un consumo elevado de CPU.