

SDEP Optimización

Ejercicio 1

A) Tiempo en que se procesan los 10,000 votos con el servidor de la práctica pasada.

Para la práctica pasada, omitiendo las impresiones de pantalla, con 10,000 votos, obtuvimos un tiempo de 0.503 segundos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor$ ./server 10mil
Servidor iniciado...

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente$ time ./client 127.0.0.1 prueba 10000

real    0m0.503s
user    0m0.070s
sys     0m0.206s

```

Figura 1. Impresión de pantalla de la practica pasada con 10,000 votos.

B) Tiempo en que se procesan los 10,000 votos, sin validar en el servidor si el número de teléfono celular está repetido (práctica antepasada).

Para la práctica antepasada, con 10,000 votos, el tiempo fue de 0.473 segundos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Servidor$ ./server 10mil
Servidor iniciado...

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Cliente$ time ./client 127.0.0.1 prueba 10000

real    0m0.473s
user    0m0.080s
sys     0m0.231s
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Cliente$

```

Figura 2. Impresión de pantalla de la practica antepasada con 10,000 votos.

Como se puede ver, en la práctica antepasada (la que no requería validación del celular), demora menos tiempo que la práctica que si hace la validación.

Ahora dé una respuesta bien justificada y consensada con su equipo a cada una de las siguientes preguntas. No olvide anotar todas sus respuestas en la bitácora.

- 1) ¿Es posible en los casos A y B atender setenta millones de votos en el periodo de doce horas que duran las votaciones? ¿Por qué?
Debido al tiempo que tardaron las ejecuciones, para la prueba 1, con 10,000 votos, el tiempo necesario fue de 0.503 segundos, así que multiplicado eso por 7,000 para que sean 70 millones, y dividiendolo entre 3600 segundos que son los que tiene una hora, nos da aproximadamente 0.9781 horas.
Del mismo modo para la prueba sin validación, con 10,000 votos, el tiempo fue de 0.473 segundo, siguiendo el mismo procedimiento anterior, entoces, el tiempo que se tardaría en procesar los 70 millones de votos, sería de 0.9197 horas.
- 2) ¿En qué proporción irá creciendo el tiempo de respuesta en el servidor conforme aumenta el número de votos (pruebe con 5,000 y 20,000 votos e infiera)?

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor$ ./server 5mil
Servidor iniciado....
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente$ time ./client 127.0.0.1 prueba 5000
real    0m0.293s
user    0m0.041s
sys     0m0.120s

```

Figura 3. Impresión de pantalla de la practica pasada con 5,000 votos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor$ ./server 20mil
Servidor iniciado....
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente$ time ./client 127.0.0.1 prueba 20000
real    0m0.958s
user    0m0.130s
sys     0m0.398s

```

Figura 4. Impresión de pantalla de la practica pasada con 20,000 votos.

Como se puede apreciar, para la práctica pasada, del tiempo con de la prueba con 5,000 votos al tiempo de la prueba con 20,000 votos, el aumento en proporción fue de 3.2696245. Podemos notar que el aumento de 10,000 a 20,000 votos de tiempo es de casi el doble.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Servidor$ ./server 5mil
Servidor iniciado....
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Cliente$ time ./client 127.0.0.1 prueba 5000
real    0m0.229s
user    0m0.024s
sys     0m0.127s

```

Figura 5. Impresión de pantalla de la practica antepasada con 20,000 votos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Servidor$ ./server 20mil
Servidor iniciado....
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_confiable/Cliente$ time ./client 127.0.0.1 prueba 20000
real    0m0.872s
user    0m0.168s
sys     0m0.404s

```

Figura 6. Impresión de pantalla de la practica antepasada con 20,000 votos.

Como se puede apreciar, para la práctica antepasada, del tiempo con de la prueba con 5,000 votos al tiempo de la prueba con 20,000 votos, el aumento en proporción fue de 3.807860262. Casi como si la proporción de aumento fuera de 1 a 4.

Ejercicio 2

Reutilizamos el código de la página que nos brindó, y realizamos distintas pruebas como lo solicita la práctica.

Primero probamos con 10,000 votos.

```
luis@luis-HP-Laptop-15-da0xxx: ~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Servidor
Archivo Editar Ver Buscar Terminal Ayuda
^C
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Servidor$ ./server 10mil
Servidor iniciado....
^C
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Servidor$ ./server 10mil
Servidor iniciado....
^C
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Servidor$ 
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Cliente$ time ./client 127.0.0.1 prueba 10000
real    0m0.493s
user    0m0.069s
sys     0m0.208s
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP_Optimizacion/Cliente$
```

Figura 7. Impresión de pantalla de la ejecución del servidor y cliente con el árbol implementado, con 10,000 votos.



Figura 8. Impresión de pantalla del uso de memoria RAM al ejecutar el servidor y el cliente con 10,000 votos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$ ./server 70mil
Servidor iniciado....
^C
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Ciente$ time ./client 127.0.0.1 prueba 70000
real    0m3.754s
user    0m0.489s
sys     0m1.594s

```

Figura 9. Impresión de pantalla de la ejecución del servidor y cliente con el árbol implementado, con 20,000 votos.

En un principio, no se notó un aumento en el uso de la memoria RAM, ni con 10,000 votos ni con 20,000, así que buscando en internet una solución para saber como saber el consumo de memoria, encontramos una página que no brindó la siguiente información, junto con un programa que se debe ejecutar, al final del reporte se anexa el link de donde lo obtuvimos.

Medida de la memoria máxima usada por un programa

En teoría, el comando **time** de Linux debería permitir ver información sobre la memoria usada por un proceso, tal y como lo fija el estándar GNU .7. Pero en la práctica, la mayoría de las versiones actuales de Linux no lo hacen. Por ello, hemos implementado un sencillo programa para paliar esta carencia, denominado **memory**. [Descargar](#) y compilar con:

```
g++ memory.cpp -o memory
```

La sintaxis es muy sencilla: **memory** seguido del comando cuya ejecución queremos medir (con los parámetros correspondientes). El programa **memory** evalúa el uso de memoria unas 16 veces por segundo, y se queda con el valor máximo. Los resultados están dados en kilobytes, e igualmente **time** se escriben en la salida de error (**stderr**).

Debido a la forma de medir el uso de memoria mediante un muestreo, puede haber cierta imprecisión en los valores devueltos por **memory**. Por ello, puede ser conveniente contrastar sus resultados con los de la [medida usando mem_count](#).

Figura 10. Impresión de pantalla de un sitio web, acerca de como pedir la memoria usada al ejecutar un programa[1].

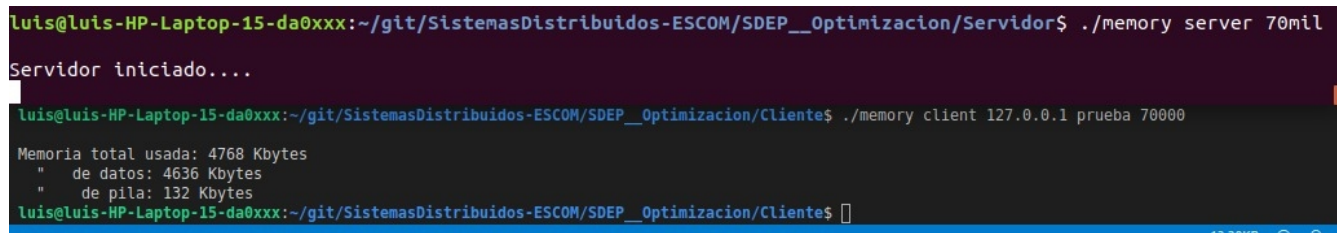
Entonces, con ayuda de la información que esta nos brinda, pudimos ver el total de memoria que utilizó cada ejecución del servidor para 10,000 y 70,000 votos.

```

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$ ./memory server 10mil
Servidor iniciado....
^C
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Ciente$ ./memory client 127.0.0.1 prueba 10000
Memoria total usada: 1012 Kbytes
"   de datos: 880 Kbytes
"   de pila: 132 Kbytes
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Ciente$

```

Figura 11. Impresión de pantalla de la ejecución del servidor y cliente con el árbol implementado, con 10,000 votos, mostrando el uso de memoria RAM.



```
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Servidor$ ./memory server 70mil
Servidor iniciado....

luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Cliente$ ./memory client 127.0.0.1 prueba 70000
Memoria total usada: 4768 Kbytes
"   de datos: 4636 Kbytes
"   de pila: 132 Kbytes
luis@luis-HP-Laptop-15-da0xxx:~/git/SistemasDistribuidos-ESCOM/SDEP__Optimizacion/Cliente$
```

Figura 12. Impresión de pantalla de la ejecución del servidor y cliente con el árbol implementado, con 20,000 votos, mostrando el uso de memoria RAM.

¿Cuánto tardaría en procesar los 70 millones de votos?

El tiempo que tomo para procesar 70,000 votos fue de 3.754 segundos, así que si fuera para 70 millones, tardaría en procesarse en aproximadamente 1.042777778 horas.

¿Cuánto espacio de memoria en RAM ocupa con 10,000 votos y cuanto ocupará con los 70 millones de votos?

Por 70,000 votos, se ocupó 4768 Kbytes, así que para 70 millones se necesitarían aproximadamente 4768000 Kbytes, o también 4.768 Gigabytes.

Referencias:

[1] MEDIDA DEL TIEMPO Y A MEMORIA DE UN PROGRAMA. [En línea]. Disponible en:
<http://dis.um.es/~ginesgm/medidas.html>