

# Laboratorio 3 Sistemas Operativos

Autor: David Calistro.

Fecha de entrega: 9 de diciembre de 2016

## 1 Resultados

Table 1: Resultados Obtenidos - Programa1

Nº de Iteración	0	1	2	3	4
Tiempo de ejecución	5.441168	5.147659	5.146964	5.141036	5.141567
Nº de Iteración	5	6	7	8	9
Tiempo de ejecución	5.140778	5.140804	5.145092	5.145050	5.142780
Nº de Iteración	10	11	12	13	14
Tiempo de ejecución	5.143871	5.143509	5.146477	5.150536	5.150391
Nº de Iteración	15	16	17	18	19
Tiempo de ejecución	5.158188	5.143585	5.143661	5.142914	5.145729

Table 2: Cálculos Obtenidos - Programa1

Promedio: 5.160088

Desviación Estádar: 0.064610

Table 3: Resultados Obtenidos - Programa2

Nº de Iteración	0	1	2	3	4
Tiempo de ejecución	13.277753	12.719424	12.688293	12.683666	12.669308
Nº de Iteración	5	6	7	8	9
Tiempo de ejecución	12.648834	12.662170	12.655571	12.681618	12.654671
Nº de Iteración	10	11	12	13	14
Tiempo de ejecución	12.656113	12.684616	12.678726	12.674103	12.667709
Nº de Iteración	15	16	17	18	19
Tiempo de ejecución	12.650085	12.656952	12.654602	12.836929	12.895020

Table 4: Cálculos Obtenidos - Programa2

Promedio:	12.719809
Desviación Estádar:	0.142057

El laboratorio 3 de Sistemas Operativos se realizo utilizando la mitad del arreglo solicitado, es decir, 536870912/2 ints, debido a que el computador utilizado no posee tal cantidad de memoria.

## 2 Conclusiones

Tal como se observa en los resultados obtenidos y mostrados en las tablas 1, 2, 3 y 4 podemos decir que en la 1era ejecución se demora más ya que tiene que traer los recursos desde memoria secundaria a la memoria principal, es decir, hay muchos miss iniciales, posteriormente al proseguir y tener precargada en memoria los recursos contiguos se demora menos, este tiempo de ejecución hace que la desviación estándar se dispare y sea más grande, lo mencionado se refleja en ambos programas.

Ahora en el programa1 debido a que explota mucho mejor la localidad espacial y temporal los tiempos de ejecución son mucho menores que en el programa2 donde cada vez realiza un salto random de una parte del arreglo a otro obteniendo así un nivel de miss en la memoria principal superior por lo que esta largo tiempo realizando swaps de los datos en memoria, aumentando drásticamente su tiempo de ejecución, sobrepasando en más del doble de tiempo de ejecución que el programa1.

Dado los resultados es obviamente mejor programar pensando en explotar los principios de localidad espacial y temporal, dentro de lo posible, así poder optimizar el uso de recursos que ya están en memoria principal realizando swap cuando sea necesario y reduciendo el tiempo de ejecución.

Se aprecia claramente lo visto en clases con respecto a el swap y que aunque los procesadores y los tiempos de ejecución hoy en día sean prácticamente imperceptibles los detalles como la paginación, segmentación, y el tamaño de páginas o segmentos en memoria principal es algo importante para el tiempo de ejecución final de un programa y es algo que se debe tener en cuenta para reducir tiempos y optimizar el uso de recursos.