

**Práctica número 3: Llamadas al Sistema**

**Méndez Gutiérrez Néstor Javier**

**Profesor – M.I. Marcela Ortiz Hernández**

**03 de septiembre de 2019**

**Semestre - 2019-2020/I**

### 1. Llamadas al Sistema

### 2. Introducción

En esta practica se compila y ejecuta un programa en tres lenguajes diferentes; C, C++ y Java, cuando estos tres programas se ejecutan se hace uso del comando **strace**, el cual es una utilidad de GNU/Linux para comprobación de errores el cual permite monitorear las llamadas al sistema que realiza un determinado programa cuando este es ejecutado.

### 3. Datos

No Aplica

### 4. Figuras, imágenes o gráficos

Salida de strace con el programa en C.

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	1		read
0.00	0.000000	0	1		write
0.00	0.000000	0	2		open
0.00	0.000000	0	2		close
0.00	0.000000	0	1		execve
0.00	0.000000	0	3	3	access
0.00	0.000000	0	1		brk
0.00	0.000000	0	1		munmap
0.00	0.000000	0	3		mprotect
0.00	0.000000	0	7		mmap2
0.00	0.000000	0	3		fstat64
0.00	0.000000	0	1		set_thread_area
100.00	0.000000		26	3	total

Salida de strace con el programa en C++.

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	4		read
0.00	0.000000	0	2		write
0.00	0.000000	0	5		open
0.00	0.000000	0	5		close
0.00	0.000000	0	1		execve
0.00	0.000000	0	6	6	access
0.00	0.000000	0	1		brk
0.00	0.000000	0	1		munmap
0.00	0.000000	0	5		mprotect
0.00	0.000000	0	16		mmap2
0.00	0.000000	0	6		fstat64
0.00	0.000000	0	1		set_thread_area
100.00	0.000000		53	6	total

Salida de strace con el programa en Java.

% time	seconds	usecs/call	calls	errors	syscall
99.83	0.032000	10667	3	1	futex
0.17	0.000055	0	140	127	open
0.00	0.000000	0	12		read
0.00	0.000000	0	14		close
0.00	0.000000	0	1		execve
0.00	0.000000	0	10	9	access
0.00	0.000000	0	3		brk
0.00	0.000000	0	2		readlink
0.00	0.000000	0	4		munmap
0.00	0.000000	0	1		clone
0.00	0.000000	0	1		uname
0.00	0.000000	0	12		mprotect
0.00	0.000000	0	1		prctl
0.00	0.000000	0	2		rt_sigaction
0.00	0.000000	0	1		rt_sigprocmask
0.00	0.000000	0	1		getrlimit
0.00	0.000000	0	31		mmap2
0.00	0.000000	0	112	110	stat64
0.00	0.000000	0	13		fstat64
0.00	0.000000	0	2		getdents64
0.00	0.000000	0	1		set_thread_area
0.00	0.000000	0	1		set_tid_address
0.00	0.000000	0	1		openat
0.00	0.000000	0	1		set_robust_list
100.00	0.032055		370	247	total

## 5. Preguntas

Llamadas al sistema:

1. UTIME(2): cambia el último acceso al archivo y los tiempos de modificación.
2. Unshare: disocia partes del contexto de ejecución del proceso.
3. Truncate: trunca un archivo a una longitud especificada.
4. Tkill: Envía una señal a un hilo.
5. Sync: comprometer cachés del sistema de archivos en el disco.

¿Qué programa realiza más llamadas al sistema: C, C++ ó Java?

El programa que realiza mas llamadas al sistema es el que está escrito en Java

¿Porqué?

Esto es por que es un lenguaje Java es un lenguaje de alto nivel, esto quiere decir que el lenguaje no es tan cercano a la arquitectura de hardware, es por ello que realiza mas llamadas al sistema.

## 6. Conclusiones

En esta práctica se realizó un seguimiento con la ayuda de el comando **strace** de la ejecución de un programa escrito en tres lenguajes de programación cada uno con un nivel de acercamiento al hardware diferente, estos lenguajes son C el cual es considerado un lenguaje de bajo nivel, C++ el cual es considerado de nivel medio y por último Java que es considerado de alto nivel.

Lo observado en la practica es que cuando se ejecuta un programa que fue escrito en un lenguaje de alto nivel, este realiza más llamadas al sistema que los programas que fueron escritos en lenguajes de bajo nivel.