

Universidad De San Carlos De Guatemala  
Facultad De Ingeniería  
Escuela De Ciencias Y Sistemas  
Sistemas Operativos 2 Sección A  
Ing. Edgar René Ornelis Hoil  
Aux. Derek Esquivel Díaz  
Vacaciones Junio 2024



# Practica 1

## Llamadas de Sistema

### Objetivos

- Comprender como funcionan las llamadas para la gestión de procesos en Linux.
- Comprender como funcionan las llamadas para la gestión de archivos en Linux.
- Aprender a interceptar las llamadas al sistema realizadas por un proceso.

### Descripción

En los sistemas de computación, se conoce como un proceso hijo a un proceso creado por otro proceso (el proceso padre), con la finalidad de realizar sub tareas. Este proceso padre debe de ser capaz de comunicarse y poder monitorear las acciones que realiza el proceso hijo para asegurarse este ejecutándose correctamente.

La practica consiste en escribir un programa que sea capaz de monitorear y almacenar en un log todas las llamadas de sistema realizadas por los procesos hijos que ha creado.

### Especificaciones

#### Proceso padre

Este será el proceso principal. Utilizando la llamada de sistema `fork()` creará dos procesos hijo, que deberá de monitorear escribiendo las llamadas de sistema que realicen en un archivo de texto llamado *"syscalls.log"*, el cual deberá de ser vaciado al iniciar una nueva ejecución del programa o ser creado en caso de que no exista.

Para cada llamada que se intercepte, en el archivo de texto se deberá de escribir:

- **Proceso <<pid>> : <<llamada realizada>> (<<fecha y hora de la llamada>>)**

Este proceso finalizará al enviar la señal SIGINT (Ctrl + C) desde la terminal, el proceso debe de ser capaz de capturar esta señal e imprimir los siguientes datos antes de terminar su ejecución:

- Número total de llamadas al sistema que realizaron los procesos hijo durante la ejecución.
- Numero de llamadas al sistema que realizo por tipo (Eje: Read: 2, Write: 5).

## Procesos hijo

Cada uno de los procesos hijo tendrá un ciclo donde realizaran llamadas de sistema para el manejo de archivos sobre un archivo llamado “practica1.txt”, el cual deberá de ser vaciado al iniciar una nueva ejecución del programa o ser creado en caso de que no exista.

Una vez abierto el archivo los procesos procederán a realizar llamadas de sistema seleccionadas de manera aleatoria, estas serán ser:

- Open: Llamada cuando se abre el archivo.
- Write: Escribirá una línea de texto con 8 caracteres alfanuméricos aleatorios.
- Read: Leerá 8 caracteres.

Estas llamadas serán realizadas con un lapso aleatorio de tiempo de 1 a 3 segundos.

La finalidad de estas llamadas es únicamente las de simular el funcionamiento que tendría un subproceso realizando llamadas de sistema, por lo que no se deberá escribir nada en la terminal cuando se realicen.

## SystemTap

Esta es una herramienta diseñada para la recolección de información del kernel con la finalidad de monitorear el funcionamiento del sistema operativo.

Los estudiantes deberán de realizar un script con SystemTap para interceptar únicamente las llamadas al sistema realizadas por los procesos hijo. Este script luego será ejecutado desde el proceso padre para realizar el monitoreo.

## Observaciones

- La práctica se realizará en parejas.
- El lenguaje de programación a utilizar para el proceso padre e hijo será C.
- El código de los procesos hijo deberá de estar en un archivo .BIN separado al código del proceso padre.
- La herramienta para interceptar las llamadas al sistema será SystemTap.
- Cualquier copia parcial o total será reportada a la Escuela de Ciencias y Sistemas para que proceda como indica el reglamento.

## Entregables

- Código fuente del proceso padre.
- Código fuente de los procesos hijo.
- Script de SystemTap.
- Manual técnico con explicación de lo realizado en la práctica. Este debe ser un archivo .md (Markdown) y será colocado como el README de la carpeta.

## Forma de entrega

- Utilizar un repositorio de GitHub, el cual debe ser privado con el nombre:  
**SO2\_GRUPO<<no. grupo>>.**
- Crear la carpeta **“Practica 1”**
- Agregar al auxiliar al repositorio: **Desquivel501**
- La entrega se realizará por medio de UEDI en el apartado correspondiente, donde el estudiante subirá un archivo de texto con el link a su repositorio.

**La entrega se debe realizar el 9 de junio de 2024 antes de las 23:59.**