

SISTEMAS OPERATIVOS

TRABAJO PRÁCTICO 2:

"Llamadas al Sistema"

Objetivos del práctico

Al terminar este trabajo Ud. habrá aprendido a:

1. Utilizar convenientemente algunas de las principales llamadas a sistemas del SO LINUX.
2. Configurar dispositivos de almacenamiento secundario (disketteras/imágenes)
3. Conocer editores y familiarizarse con el que mejor se adapte a su requerimiento.
4. Generar archivos ejecutables (archivos interpretados-scripts).

Herramientas necesarias:

Para resolver los ejercicios propuestos necesitará:

1. Una PC con SO Windows con el Software de Virtualización VMWARE.
2. El material de Campus de la Cátedra.

Fuentes de Información sugeridas

Encontrará información útil en:

- Páginas de manual de LINUX
- El material provisto por la Cátedra.
- Guía de clases de laboratorio: Uso de Comandos Básicos de Linux - El Shell -
- <http://www.debian.org/doc/manuals/reference/ch-tutorial.es.html>
- <http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>
- Kernighan, Brian W.; Pike, Rob. (1984). The Unix Programming Enviroment, Prentice Hall.
- Tutorial BASH de MacProgramadores
- Newham, Cameron. (2005). Learning the Bash Shell, Third Edition, O'Reilly Media
- Shotts, William E., Jr - The Linux Command Line <http://linuxcommand.org/tlcl.php>
- <http://tldp.org/LDP/abs/html/tests.html>

Requisitos de Entrega

Lugar y Fecha de entrega:

1. La fecha de entrega para este práctico será informada por el CAMPUS en el momento de publicar el TP.
2. Los trabajos deben ser entregados vía CAMPUS EN UN UNICO ARCHIVO con el nombre formateado de la siguiente manera: **"TP02-2024-GRUPO XX.zip"** (XX es el número que identifica al grupo). (Archivo UNICO comprimido conteniendo 2 archivos; uno **.img** y otro **.pdf**)
3. No se aceptarán trabajos incompletos.

Formato de Entrega.

Deberá enviar dos archivos con la resolución del trabajo:

- A- La imagen de un diskette "en formato ext2" conteniendo los programas fuentes y compilados cuyo nombre debes ser: **"TP02-2024-GRUPO XX.img"**
- B- El segundo, es un archivo **en formato PDF** cuyo nombre debes ser: **"TP02-2024-GRUPO XX.pdf"** Deberá reunir las siguientes características:

1. Secciones del documento (Todas obligatorias):
 1. **Carátula de presentación:** Debe incluir OBLIGATORIAMENTE:
 1. Asignatura
 2. Número y Descripción del trabajo práctico
 3. Año y Cuatrimestre de Cursado
 4. Identificación del Grupo
 5. Nombres, Apellidos y direcciones de correo electrónico de TODOS los Integrantes del grupo
 - 2- **Sección Principal:** Aquí debe incluirse la resolución de cada uno de los problemas planteados. **El código de los problemas deberá estar en formato de TEXTO (no captura de imagen).** Para cada respuesta debe indicarse OBLIGATORIAMENTE, el número y título del problema al que corresponde tal como aparece en el enunciado.
 - 3- **Sección de Descargos:** Aquí debe incluirse cualquier comentario que deba tenerse en cuenta para la corrección del práctico. Use esta sección para indicar cosas como:
 - Qué no pudo resolver alguno de los problemas
 - Qué no pudo resolver COMPLETAMENTE alguno de los problemas.
 - Qué no está seguro si el problema está resuelto correctamente.

Comentar los problemas en esta sección es la única forma de obtener puntaje parcial para un ítem que no está bien resuelto. Si se encuentra un problema no resuelto o resuelto de manera INCOMPLETA y eso no está comentado en esta sección, perderá puntos adicionales (no sólo le descontaremos puntos por el error sino también por no avisarnos). Si no tiene ningún comentario, deje esta sección en blanco.

- C- **Todos los PROGRAMAS Compilados deberán mostrar por PANTALLA ("al inicio") los datos del Grupo al que pertenece como así también los datos de todos los integrantes del mismo (Nombre / Apellido / EMAIL).**

Penalizaciones.

Los prácticos entregados en fechas posteriores al límite fijado tendrán una quita de puntos. Para ver el método empleado para restar puntos consulte en el Campus.

Cambios al enunciado del práctico, fechas de entrega, etc.

Cualquier cambio en los enunciados, fechas de entrega, etc. será informado utilizando dos métodos:

1. El campus virtual.
2. La lista de correos.

El alumno no puede alegar que no estaba al tanto de los cambios si esos cambios fueron anunciados utilizando alguno de los dos métodos.

SUGERENCIA: Consulte frecuentemente las novedades del Curso en el Campus Virtual y asegúrese de que ha sido incorporado a la lista de correos.

Honestidad académica:

Está bien hablar entre los grupos acerca de cómo resolver problemas, pero los grupos son de hasta 3 integrantes.

No entregue el trabajo de otras personas como propio. Tampoco entregue trabajos publicados en Internet como propios sin citar las fuentes. Cualquier trabajo, porción de trabajo o texto sin la cita correspondiente es plagio.

Cada grupo debe mantener su código para sí mismo, si su proyecto es copiado, puede ser difícil determinar quién es el verdadero autor.

Cualquier ayuda que reciba deberá documentarla como un comentario al inicio del programa. Por ejemplo, si encuentra una solución a un ejercicio en un texto o manual, debería citar la fuente. Una razonable ayuda, no afectará la aprobación de los trabajos, pero fallas al citar las fuentes o la ausencia de las mismas es fraude.

Queda debidamente aclarado, que los trabajos son de autoría, desarrollo y elaboración propia y no de un tercero.

Por último, la Cátedra podrá solicitar revisar el Trabajo Práctico con coloquio presencial con los alumnos del grupo en caso de considerarlo.

ARGUMENTOS POR LÍNEA DE COMANDO (ARGC – ARGV – OPCIONES)

EJERCICIO 1 (30 Pts)

Implementación de un programa para la copia de archivos con monitoreo en tiempo real.

El objetivo de este ejercicio es implementar un programa en C que copie un archivo de origen a un archivo de destino utilizando llamadas al sistema (system calls) de Linux. Además de la copia, el programa deberá mostrar el progreso en tiempo real, es decir, el porcentaje de bytes copiados en cada momento.

El programa deberá recibir dos argumentos en la línea de comandos: la ruta del archivo de origen y la ruta del archivo de destino. Si no se especifican los argumentos, deberá imprimir por pantalla un mensaje con la sintaxis del programa: *ejercicio1 ruta_archivo_origen ruta_archivo_destino*.

El programa deberá manejar los errores de forma adecuada, como cuando el archivo de origen no existe o cuando no se puede crear el archivo de destino.

Implementar el cálculo del porcentaje de bytes copiados y mostrarlo en la salida estándar en tiempo real.

NOTA:

Para resolver el ejercicio deben utilizar las llamadas a sistema: `open()`, `read()`, `write()`, `lseek()`, `stat()` y `close()`, entre otras.

GESTIÓN DE PROCESOS

EJERCICIO 2: (30 Pts)

Implementación de un programa para monitorear procesos.

El objetivo de este ejercicio es implementar un programa en C que utilice llamadas al sistema (system calls) de Linux para monitorear el estado de un proceso dado. El programa deberá recibir el ID de un proceso (PID) como argumento y reportar la siguiente información:

1. Nombre del proceso: Deberás obtener el nombre del proceso a partir del PID.
2. Estado del proceso: El estado actual del proceso (Ejecutando, Durmiendo, Detenido, etc.).
3. Uso de memoria: La cantidad de memoria (en KB) que el proceso está utilizando.
4. Uso de CPU: El porcentaje de CPU que el proceso ha utilizado.
5. Prioridad del proceso: La prioridad con la que el proceso está siendo ejecutado.

El programa deberá manejar errores de forma adecuada, como cuando el proceso no existe o el PID proporcionado es inválido.

Si no se proporciona un PID, el programa deberá mostrar un mensaje de error indicando la sintaxis y salir. Sintaxis: *ejercicio2 PID*.

NOTA:

Llamadas al sistema que podrán ser de utilidad para resolver el ejercicio: `kill()`, `open()`, `read()`, `close()`, `stat()`, `getrusage()`, `waitpid()`, entre otras.

COMUNICACIÓN ENTRE PROCESOS

EJERCICIO 3: (40 Pts)

En un sistema operativo Linux, los procesos pueden comunicarse entre sí utilizando diversas técnicas, y a su vez pueden interactuar con el sistema de archivos mediante operaciones de entrada/salida. Este ejercicio tiene como objetivo trabajar con múltiples procesos, comunicación interprocesos (IPC) y operaciones con archivos.

Escribe un programa en C que realice lo siguiente:

- El programa debe crear dos procesos hijos utilizando la llamada `fork()`.
- Cada proceso hijo deberá realizar una operación matemática diferente sobre un conjunto de datos (por ejemplo, un array de números).
- El proceso hijo 1 deberá calcular la suma de los números en el array, y el proceso hijo 2 deberá calcular el producto de los números.
- El proceso padre debe recibir los resultados de ambos hijos utilizando pipes como mecanismo de comunicación interprocesos.
- El proceso padre debe escribir ambos resultados (suma y producto) en un archivo de texto llamado `resultados.txt`, y luego leer el archivo para mostrar el contenido en pantalla.

Velar de que todos los procesos manejen posibles errores en las llamadas a sistemas.

Asegurar de manejar correctamente los extremos de lectura y escritura de los pipes para evitar bloqueos.

El programa debe estar estructurado de forma clara, y los resultados de las operaciones deben imprimirse en pantalla desde el proceso padre después de leer del archivo.

NOTA:

Para resolver el ejercicio deben utilizar las llamadas a sistema: `fork()`, `pipe()`, `write()`, `read()`, `open()`, `close()`, entre otras.