



**UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA
FACULTAD DE INGENIERÍA
INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN
GRUPO 5HW1**



**SISTEMAS OPERATIVOS
Sistemas Operativos I - Proyecto**



Alumnos:

Irlanda Andrea Burciaga Herrera – 358353

Danna Maribel Corral Salcedo – 358147

Adrian Alejandro González Domínguez - 359834

Luis Dahrey Moncayo Beltran - 363641

Asesor: Iván Miguel Chavero Jurado

Fecha de entrega: 15 de mayo de 2024

Funcionamiento del código:

La función principal inicia abriendo el directorio /proc, que es un sistema de archivos especial que contiene información sobre los procesos en ejecución. Proporciona acceso al estado de cada proceso activo y hebra del sistema.

Luego, se itera a través de los elementos en este directorio. Para cada entrada, se verifica si su nombre comienza con un dígito, ya que los procesos siempre están representados por un número de proceso (PID), y los directorios que representan los procesos se llaman igual que el pid del proceso.

Dentro de cada iteración se construyen las rutas a los archivos /proc/[pid]/cmdline y /proc/[pid]/status donde [pid] es el id del proceso actual. Estos archivos contienen información relevante del proceso, el primero, cmdline, contendrá el comando ejecutado en la línea de comandos (ya sea por el usuario, o por un programa) para iniciar el proceso en cuestión; si el archivo no contiene ninguna información, este se tratará de un proceso de kernel. El segundo, status, tiene información como el nombre del programa, el estado en el que se encuentra (Nuevo, listo, ejecución, bloqueado, terminado), el pid, información sobre el uso de la memoria física y virtual, entre otras cosas.

Se utilizan dos funciones auxiliares, `get_process_cmdline()` y `get_process_name()`, para extraer esta información.

Para obtener el contenido de estos 2 archivos, utilizamos 2 funciones. `get_process_cmdline()` se describe a sí misma y `get_process_name()` que accede al archivo status para obtener el nombre del proceso.

Luego de realizar estas 2 acciones, comprobamos si el proceso fue un proceso de kernel o no preguntando por el contenido que recuperamos del archivo cmdline, y finalmente, mostramos por la salida estándar el ID del proceso, el nombre del proceso, y su línea de comando asociada.

¿Cómo interacciona el programa realizado con los subsistemas del Kernel?

- El subsistema de usuarios participa ya que es el que valida que el programa y el usuario tienen los permisos de lectura para poder revisar el contenido de las carpetas y de los archivos.
- Hace uso del subsistema de archivos ya que estamos realizando operaciones de lectura de archivos. La función `opendir()` y `readdir()` interactúan con el sistema de archivos del kernel para abrir y leer el directorio /proc y sus entradas. Así mismo, las funciones `fopen()`, `fclose()`, y `getline()` también

interactúan con el sistema de archivos para abrir, cerrar y leer los archivos que contienen información sobre los procesos.

- El subsistema de procesos dispone la información de los procesos del sistema para que sean accesibles en espacio de usuarios mediante archivos y directorios, en otras palabras, concede un directorio virtual que contiene la información de los procesos que están en ejecución.
- Consecuente a la apertura del archivo, hacemos uso del subsistema de memoria. Está involucrado en la gestión de la memoria necesaria para leer y almacenar el contenido de estos archivos en las variables locales del programa, así como también en la liberación de la memoria después de que se hayan utilizado estas variables.