

Sin títuloParcial: Uso de Llamadas al Sistema en Linux con C++

Enunciado

El objetivo de este parcial es desarrollar un programa en **C++ para Linux**, utilizando **llamadas al sistema en el contexto del sistema de archivos (file system)**, que implemente uno de los siguientes retos:

1. **Compresión de Archivos:** Implementar un algoritmo de compresión de datos sin pérdida, utilizando llamadas al sistema para la lectura y escritura de archivos. No se permite el uso de librerías de compresión externas.
2. **Encriptación de Archivos:** Implementar un algoritmo de encriptación de datos, utilizando llamadas al sistema para manipular archivos. No se permite el uso de librerías de encriptación externas.

El programa deberá indicar si el archivo a procesar es **binario o de texto**, y el estudiante deberá justificar la elección en su implementación.

Puede usar los codigos vistos en clase:

<https://github.com/evalenciEAFIT/courses/tree/main/SistemasOperativos/LlamadaSistema/ManejoArchivos> El readme.md es el archivo principal.

(<https://github.com/evalenciEAFIT/courses/blob/main/SistemasOperativos/LlamadaSistema/ManejoArchivos/readme.md>)

Requisitos y Consideraciones

- Se debe usar **llamadas al sistema en el contexto del sistema de archivos**, como `open()`, `read()`, `write()`, `close()`, entre otras.
- No se permite el uso de **librerías externas** para compresión o encriptación.
- El programa debe comportarse como una **función de sistema operativo**, aceptando argumentos de línea de comandos.
- Se deben implementar las siguientes **banderas de ejecución**:
 - `-h` o `--help`: Muestra un mensaje con las opciones de uso del programa.
 - `-v` o `--version`: Muestra la versión del programa.
 - `-c <archivo>` o `--compress <archivo>`: Comprime el archivo indicado.
 - `-x <archivo>` o `--descompress <archivo>`: expande el archivo comprimidos indicado en el archivo.
 - `-e <archivo>` o `--encrypt <archivo>`: Encripta el archivo indicado.

- `-d <archivo>` o `--decrypt <archivo>`: Desencripta el archivo indicado.
- Se debe definir claramente:
 - Tipo de archivo a procesar (texto o binario).
 - Algoritmo a utilizar y su justificación.
 - Estrategia de almacenamiento y recuperación del archivo comprimido o encriptado.
- Se debe escribir un **README.md** explicando el algoritmo utilizado y las instrucciones de uso.

Modalidad de Evaluación

- **Trabajo en equipo permitido (sin límite de integrantes).**
- **Evaluación oral:** Se deberá explicar la implementación y justificar el diseño del código.
- Se presentarán preguntas específicas sobre el código, funcionamiento y posibles mejoras.

Criterios de Evaluación

Criterio	Puntos
Uso correcto de llamadas al sistema en Linux (file system)	20
Implementación del algoritmo (compresión o encriptación)	25
Correcta gestión de archivos (apertura, lectura, escritura, cierre)	20
Uso de argumentos y banderas de línea de comandos	10
Justificación de la elección del tipo de archivo (binario o texto)	10
Documentación y README explicativo	10
Evaluación oral y justificación del código	15
BONUS: Implementación de varios métodos de compresión y encriptación	+10
Total	100 + Bonus

Entrega y Presentación

- **Fecha de entrega:** Semana del **17 al 21 de febrero de 2025**, durante la clase.

- **Presentación oral obligatoria** para todo el grupo.
- **Evidencia:** Cada equipo deberá subir el código fuente y README.md en un repositorio de **GitHub**.

Este parcial evaluará no solo la implementación técnica, sino también la capacidad de análisis y justificación de decisiones en el diseño del programa. **Los equipos que implementen más de un método de compresión y encriptación podrán recibir un bonus adicional.**