

Se desean simular determinadas características de un sistema operativo Unix. Para ello, se debe modelar una estructura de archivos *a-la* Unix, junto con ciertos comandos del sistema operativo que permiten manipular a los archivos.

El sistema de archivos debe permitir tener archivos y directorios. Existe un directorio destacado que es el raíz.

Además existen usuarios, que tienen un nombre y una password. Existe un usuario administrador conocido como `root`.

El sistema debe contemplar:

- Iniciar sesión como `root` o como cualquier usuario. Se debe lanzar un error si el usuario no existe, o la contraseña es incorrecta.
- Cada vez que se inicia sesión, el usuario debe comenzar en el directorio raíz.
- Agregar o remover usuarios (sólo `root` puede hacerlo)
- Crear y borrar archivos y directorios en el directorio actual.
- El archivo tiene un contenido textual que puede ser editado en cualquier momento. Nos interesa saber si un usuario puede editar un archivo o no.
- Navegar entre los diferentes directorios (tal como lo hace el comando `cd`)
- Listar el contenido de un directorio (tal como lo hace el comando `ls`)
- Administrar los permisos de archivos y directorios:
 - Tenemos permisos de lectura, escritura y ejecución (para los archivos) y de lectura y escritura (para los directorios)
 - A cada archivo se le asignan permisos para el usuario *owner* (o sea, el usuario logueado que creó el archivo), y para los demás.
 - Los permisos deben ser chequeados al momento de una operación.
 - El usuario `root` puede ejecutar cualquier operación con los archivos.
 - El *owner* de un archivo o directorio siempre tiene permisos de lectura y escritura
 - Ejemplo: si el archivo `tesis.txt` es propiedad de Juan, y tiene permiso sólo de lectura para los demás, entonces Juan (y `root`) pueden borrar el archivo (porque ambos tienen permiso de escritura), pero el usuario Pedro sólo puede leer su contenido (porque es de los “demás”)
- Nos interesa copiar un archivo a la carpeta de Cuis, utilizando streams.
- Se pide además que se tenga la posibilidad de enviar el resultado del comando `ls` a un archivo en el disco real (como lo hace el operador `>`)
 - La salida esperada tiene la siguiente forma:

miDirectorio:

miSubDirectorio	lectura	juan
miArchivo.txt	lectura/escritura	root
miArchivo2.txt	lectura/escritura/ejecucion	pepe