



Proyecto 1

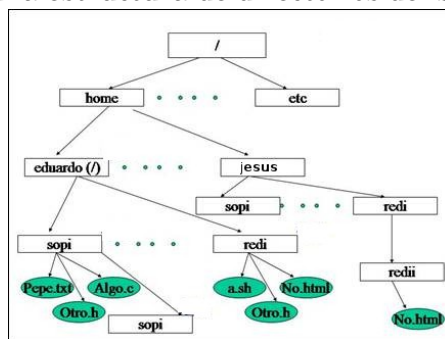
Objetivo

Familiarizarse con las llamadas al sistema para el acceso a las estructuras del sistema de archivos.

Introducción al Problema

En Linux hay diversas herramientas para empaquetar, es decir, “juntar” dos o más archivos en un solo archivo (paquete). Una herramienta muy usada en Unix es **tar**. Otra operación es la de cifrar un archivo para proteger su contenido. Una herramienta muy popular para cifrar es **gpg**.

La siguiente figura muestra una estructura de directorios del sistema Unix:



donde:

- los cuadros representan los directorios.
- Y los óvalos en verde representan los archivos no directorios

Ejecutar el siguiente comando:

```
tar -czf homes.tar.gz /home
```

Crea un paquete llamado `homes.tar.gz` con el contenido de toda la estructura del directorio `/home`. Esto incluye los directorios `eduardo` y `jesus`, con todos los archivos que contienen: directorios, regulares, etc. En otras palabras, `tar` es **recursivo**.

Descripción general del problema

Para este proyecto Ud. debe implementar un empaquetador llamado mytar. Éste debe preservar la información sobre el archivo, en la medida de lo posible: permisos, tipo, dueño, grupo, número de bloques. Y por supuesto, tamaño y contenido exacto. Datos como fechas y número de inodos no pueden ser manejados a este nivel.

Este comando deberá reconocer las siguientes opciones:

Mytar <listaOpciones>

1. -f <nombreTar>: especifica el archivo tar a usar.
2. -c: permite crear el archivo nombreTar. Si ya existe, lo sobrescribe.
 - Debe preservar los tipos de archivos: directorios, regulares, pipes, y enlaces tanto simbólicos y fuertes.
3. -n: ignora todo archivo que no sea regular o directorio. Aplica a las opciones -c, -x y -t.
4. -t: Muestra el contenido del archivo tar. Usando un formato similar al del comando 'ls -l'.
5. -o <nombreDirectorio>: Directorio donde colocará la salida de una extracción. Por defecto es el directorio actual.
6. -z <desp>: cifra el contenido del archivo. Se usa sólo con la opción -c. Utiliza una técnica básica en donde los bits de cada byte son rotados hacia la derecha <desp> bits.

Por ejemplo, considere los siguientes bits: 00100000, rotarlos 2 bits a la derecha produce la secuencia: 00001000.

7. -y <desp>: descifra el contenido del archivo. Se usa sólo con las opciones -x y -t. Utiliza una técnica básica en donde los bits de cada byte son rotados hacia la izquierda <desp> bits.

Por ejemplo, considere los siguientes bits: 00001000, rotarlos 2 bits a la izquierda produce la secuencia: 00100000.

8. -x: extrae todos los archivos en <nombreTar>. Si los archivos están cifrados, debe extraerse descifrado. Si la opción -y no fue especificada, debe dar un error. Si un archivo destino existe, será sobre escrito. Y debe ser idéntico al original correspondiente, es decir, los comandos diff o cmp no podrán encontrar diferencias.
9. -s nombreArchivo: extrae el archivo nombreArchivo en caso de existir. Si el archivo está cifrado, debe extraerlo descifrado. Si la opción -y no fue especificada, debe dar un error. Si un archivo destino existe, será sobre escrito. Y debe ser idéntico al original, es decir, los comandos diff o cmp no podrán encontrar diferencias.

La realización de esta opción tendrá una evaluación adicional de **hasta 2** pts en base a la

implementación realizada.

10. -v [archivo]: escribe en archivo una descripción de lo que va haciendo. En caso de no ser especificado el nombre, esta información aparecerá en el terminal.

Note que [x] indica que x puede estar presente o no en la línea de comando.

Ejemplo de algunas invocaciones válidas del comando:

Las siguientes es para dar una idea de algunas cosas que se pueden solicitar al comando. Note que hay otras invocaciones que cumplen con las definiciones dadas.

- **mytar -cf archivo.mytar .**

Crea el paquete *archivo.mytar* con el contenido del directorio actual

- **mytar -vcf archivo.mytar .**
- **mytar -cvf archivo.mytar .**

Crea el paquete *archivo.mytar* con el contenido del directorio actual. En el terminal se imprime lo que el comando va haciendo.

- **mytar -cvf archivo.mytar Dir1**

Crea el paquete *archivo.mytar* con el contenido del directorio *Dir1*. En el terminal se imprime lo que el comando va haciendo.

- **mytar -cvf archivo.mytar Dir1 -v mensajes**

Crea el paquete *archivo.mytar* con el contenido del directorio *Dir1*. En el terminal se imprime lo que el comando va haciendo en archivo *mensajes*.

- **mytar -z 3 -cf archivo.mytar .**

Crea el paquete *archivo.mytar* con el contenido del directorio actual. El archivo *archivo.mytar* es cifrado desplazando 3 bits a la derecha cada byte.

- **mytar -tf archivo.mytar**

Muestra los archivos, sus nombres, permisos, dueño, etc. Usando formato 'ls -l'.

- **mytar -xf archivo.mytar**

Extrae el contenido del *archivo.mytar* hacia el directorio actual.

- **mytar -f archivo.mytar -x pepe**

Extrae el archivo *pepe* del *archivo.mytar* hacia el directorio actual.

- **mytar -xf archivo.mytar -o salida**

Extrae el contenido del *archivo.mytar* hacia el directorio *salida*. Si no existe lo crea. Y si existe, el contenido inicial es borrado.

Recomendaciones

Para la realización del proyecto se les sugiere que siga las siguientes recomendaciones:

- Leer el manual de tar y de gzip
- Entender bien el problema antes de diseñar su propuesta.
- Diseñe toda su solución antes de comenzar a programar.
- Lea la información dada por las páginas de manual del sistema sobre las funciones que debe usar.
- Vaya documentado su código a medida que lo vaya generando.
- Trabaje en forma ordenada!
- Si tiene alguna duda, con tiempo aclárela directamente con su profesor.

Entrega de la tarea:

Código impreso: Jueves de la semana 7 a las 12 mediodía en el salón de clases.

El código debe estar engrapado e identificado con sus nombres, carnets y *el número del grupo*

Digital: Hasta las 11:59pm del Miércoles semana 7.

En la sección de documentos del grupo, crear el directorio Proy1. Ahí cada grupo debe colocar los fuentes de su proyecto (incluido el Makefile) en un archivo que con nombre GXX.tar.gz. Donde XX es el número del grupo. Recuerde ejecutar make clean y borrar los archivos finalizados en ~ antes de crear el archivo.

Note que debe estar suscrito a algún grupo en aula Virtual para poder optar a esta opción, ***no espere al día de la entrega para notificar que tiene problemas o que no se ha registrado.***

NOTAS IMPORTANTES:

1. Proyecto que no sea entregado en el lugar, fecha y hora prevista, NO será recibido.
2. Proyecto que no cumpla con algunas de las especificaciones establecidas en este enunciado corre el riesgo de no ser corregido.
3. No se corregirán proyectos que no sean colocados de esta forma.

Recuerde que el proyecto debe poder compilarse utilizando el Makefile, y será corrido en las máquinas del LDC. Por consiguiente, es su responsabilidad garantizar que esto pueda hacerse.