

Seminario de lenguajes - Opción C

2016

Trabajo final - Primera entrega

Debés implementar utilizando el lenguaje C una versión simplificada de una herramienta de empaquetamiento de archivos, una especie de [GNU Tar](#). El programa debe permitir poner dentro de un solo archivo varios archivos y luego permitir recuperarlos y/o listarlos. El empaquetador que desarrolles deberá utilizar el formato descrito en el apartado “Formato” de este documento, y a partir de esa información tendrás que implementar las siguientes operaciones:

1. Crear un archivo empaquetado.
2. Desempaquetar uno o más archivos de los que contiene un paquete.
3. Listar los archivos ordenados por diferentes criterios (nombre, tamaño y fecha).
4. Agregar un archivo al paquete.
5. Ver los detalles de un archivo en particular dentro del paquete.

¿Para qué sirve tar? En sistemas basados en Unix es típico tener pequeñas herramientas que realizan una sola tarea simple, de manera que combinando estas herramientas se logre una gran flexibilidad. Es así que existen muchas herramientas que permiten comprimir un único archivo, como gzip, bzip2 o xz. Estas herramientas sólo resuelven el problema de comprimir un archivo, de manera que si queremos comprimir una carpeta o una serie de archivos (como haríamos con herramientas como WinZip o WinRar) primero es necesario combinarlos de alguna manera en un único archivo. Combinar una serie de archivos en uno solo es el objetivo del formato y la herramienta tar. Es por esto que típicamente se empaqueta un carpeta con tar y luego se comprime con xz, el archivo resultante por convención suele tener un nombre de la forma: paquete.tar.xz.

¿Entonces hay que implementar el formato tar? No. El formato tar es demasiado complejo para los objetivos de este trabajo, por lo que el formato a implementar será uno inventado para este trabajo: EMP. Éste formato se encuentra descrito en el siguiente apartado.

Funcionalidad a implementar

El programa deberá aceptar los siguientes argumentos de línea de comandos. En cada caso, primero se presenta la versión larga (*long*) y luego la corta (*short*), haciendo referencia ambas versiones a la misma operación.

- `--pack 0 -p`
- `--unpack 0 -u`
- `--repack 0 -r`
- `--unpack-all 0 -a`
- `--detail 0 -d`
- `--list 0 -l`
- `--input 0 -i`

- `--output` o `-o`

Ejemplos de uso:

Empaquetar archivos con `--pack` (es obligatorio indicar el nombre del paquete con `--output`), crea un paquete nuevo:

```
emp --pack hola.txt chau.txt tres.bin --output file.emp
```

Agregar un archivo a un paquete existente con `--repack` (también es obligatorio `--output`):

```
emp --repack zuatro --output file.emp
```

Desempaquetar archivos con `--unpack` (si no se indica `--outputdir` los archivos se extraen en el directorio actual):

```
emp --unpack hola.txt tres.bin --input file.emp [--outputdir /tmp]
```

Desempaquetar todos los archivos con `--unpack-all` (`--input` es obligatorio y `--outputdir` opcional):

```
emp --unpack-all --input file.emp [--outputdir /tmp]
```

Listar archivos empaquetados con `--list` (`--input` es obligatorio). `--list` recibe un parámetro “name”, “date” o “size” indicando el criterio de ordenación de los archivos listados (notar que el valor indicado, para el caso de `date` o `size`, debe mostrarse al lado del nombre de cada archivo listado):

```
emp --list [name|date|size] --input file.emp
```

Ejemplo:

```
$ emp --list name --input file.emp
chau.txt
hola.txt
tres.bin
zuatro
$ emp --list size --input file.emp
zuatro 24
tres.bin 1622
hola.txt 2260
chau.txt 20050
```

Listar con detalles con `--detail` (`--input` es obligatorio)

```
emp --detail hola.txt --input file.emp
hola.txt 2016-05-10 13:06:46.431724357 11B
```

Nota: la primer entrega debe permitir empaquetar archivos, no se consideran directorios ni subdirectorios

Formato

El paquete cuenta con un registro principal en la posición 0 con los siguientes datos:

<----- 64 bits ----->	
Magic + Version	Entradas
Reservado	
Reservado	

- *Magic* es un código de 3 bytes en ASCII identificando el tipo de archivo, nuestro formato tendrá como código magic: "EMP".
- *Versión* es un byte con la versión, en la primer entrega el código de versión será un 0 binario.
- *Entradas* es un entero sin signo con la cantidad de archivos almacenados en este paquete.
- Los 2 campos *reservados* de 64 bytes deben cargarse en cero.

Seguido del encabezado se deben agregar los archivos, donde cada uno de esos archivos tendrá también su propio encabezado con sus atributos, siguiendo la siguiente estructura:

<----- 64 bits ----->	
Tamaño	
Fecha de modificación (epoch)	
Reservado	
Reservado	
Path del archivo (longitud variable, puede tener más o menos de 64 bits)	

- *Tamaño* es el tamaño del archivo en bytes (entero sin signo).
- *Fecha de modificación* es un entero sin signo, con la fecha de modificación del archivo (vea "man 2 stat").
- Los 2 campos reservados deberán cargarse con cero en esta etapa.
- *Path del archivo* es un string terminado en '\0' con el nombre del archivo empaquetado. Este campo es de longitud variable, por lo que puede ocupar 2 o más bytes, no hay un límite para la longitud del nombre.

- Seguido de este encabezado debe volcarse el contenido del archivo correspondiente.

Ejemplo:

- La siguiente es una visualización hexadecimal de un paquete con este formato que contiene adentro 2 archivos de texto: “hola.txt” y “chau.txt”.

```

00000000  45 4d 50 00 02 00 00 00 00 00 00 00 00 00 00 00 |EMP.....|
00000010  00 00 00 00 00 00 00 00 0b 00 00 00 00 00 00 00 |.....|
00000020  f3 e4 28 57 00 00 00 00 68 6f 6c 61 2e 74 78 74 |..(W....hola.txt|
00000030  00 48 6f 6c 61 20 4d 75 6e 64 6f 0a 13 00 00 00 |.Hola Mundo.....|
00000040  00 00 00 00 ff e4 28 57 00 00 00 00 63 68 61 75 |.....(W....chau|
00000050  2e 74 78 74 00 41 64 69 c3 b3 73 20 6d 75 6e 64 |.txt.Adi..s mund|
00000060  6f 20 63 72 75 65 6c 0a                                |o cruel.|
00000068

```

Para ver los archivos en hexadecimal, se puede usar el comando de Linux **hexdump** que se puede abreviar como **hd**. Esta visualización es útil para verificar si la herramienta está generando los archivos con el formato esperado.

El archivo “hola.txt” tiene como contenido “Hola Mundo\n” y el archivo “chau.txt” tiene como contenido “Adiós mundo cruel\n”.

Fechas importantes

- Entrega 1 (3/7/2016 con coloquios individuales en la misma semana)
 - Formato EMP: 0
 - Esta entrega consiste de la implementación de todas las funcionalidades pero empaquetando solamente archivos. Es decir, el empaquetador no se usará para comprimir directorios.
- Entrega 2 (31/7/2016 con coloquios individuales en la misma semana)
 - Formato EMP: 1
 - Modificaciones a definir.