

Práctica 2 Opcional : utilidad simpleAR

Introducción

Objetivos

- Gestión de opciones línea de comandos
- Revisión librería de C manejo de ficheros
- Conocer detalles de como se implementan las utilidades para archivado de ficheros
- Revisión shell scripting

simpleAR, un archivo simplificado

En esta practica se debe crear una utilidad *simplear* para manipular archivos *ar*. Un archivo *ar* es un fichero que contiene otros ficheros (los miembros del archivo) con un formato que permite la recuperación (extracción) de los miembros preservando atributos originales como propietario, grupo, fechas o permisos. Para ello, además del contenido de los fichero miembros, el archivo incluye esta información para que se pueda restaura al extraerlos.

Las distribuciones de Linux suelen incluir la utilidad *ar* para la manipulación de estos ficheros. Aunque su uso no esta limitado a ficheros de tipo binario, forma parte de las *binutils* de GNU ya que en la actualidad su uso se restringe a la creación de librerías estáticas ¹.

En este ejercicio implementaremos una versión reducida de *ar* con algunos cambios en el formato y en las opciones para simplificar. Las sintaxis de esta utilidad es la siguiente:

```
simplear [opciones] archivoar [argumentos...]
```

Las tres opciones básicas son:

- creación de un archivo (*-c*)
- inserción de un fichero en el archivo (*-r*), que por defecto reemplaza si ya existiera en el archivo una versión anterior de este fichero.
- extracción de un fichero (*-x*). Si no se especifica ningún miembro, extraerá todos los ficheros incluidos en el archivo.

La opción *-c* se puede combinar con la opción *-r* para insertar miembros al archivo. Por ejemplo, la orden:

```
./simplear -cr ejemplo.ar simplear.c simplear.h sroutines.c makefile
```

deberá crear un archivo *ejemplo.ar* que incluya el contenido de los ficheros *simplear.c*, *simplear.h*, *sroutines.c* y *makefile*.

Un vez que se ha creado el archivo la orden:

```
./simplear -x practica2.ar
```

¹Los ficheros *.deb* usados para la distribución de paquetes en las distribuciones basadas en Debian también son archivos con formato *ar*.

extraerá el contenido del archivo *practica2.ar*, es decir, los ficheros *simplear.c*, *simplear.h*, *s routines.c* y *makefile* en el directorio actual.

Cabecera Global simplear

Cada archivo *simplear* comienza con un encabezado de longitud fija que contiene el número mágico, que identifica el archivo *simplear*, y un contador del número de miembros incluidos con el siguiente formato:

```
#define ARMGICO "<arspl>\n"      /* Numero Magico simplear */

#define MAGSIZE 8                /* Longitud Numero Magico */

struct ar_ghdr                   /* Cabecera global */
{
    char ar_magic[MAGSIZE]; /* Numero Magico fichero simplear */
    int nfiles;             /* Numero de miembros */
};
```

Aclaración: Cabecera Global en los ficheros ar

En el formato ar, la cabecera contiene desplazamientos a miembros especiales del archivo y para ordenar los miembros del archivo se utiliza una lista doblemente enlazada, de modo que los miembros del archivo podrían no estar ordenados secuencialmente dentro del archivo. Los desplazamientos contenidos en la cabecera permiten ubicar al primero y al último de los miembros. El orden de los miembros está determinado por el orden en esta lista enlazada. Un desplazamiento cero indica que el correspondiente miembro no está presente en el archivo. Por ejemplo:

```
struct ar_ghdr {                /* Cabecera global */
    char ar_magic[MAGSIZE]; /* Numero Magico fichero simplear */
    char ar_fstmoft[12];    /* Desplazamiento al primer miembro */
    char ar_lstmoft[12];    /* Desplazamiento al ultimo miembro */
    char ar_freeoff[12];    /* Desplazamiento a la lista de miembros libres */
    char fl_memoff[12];     /* Desplazamiento a la tabla de miembros */
    char fl_gstoff[12];     /* Desplazamiento a la tabla de símbolos (librerías) */
};
```

Este tipo de cabeceras facilita la implementación de la eliminación de miembros. Para ello se utiliza una lista de miembros libres, que enlaza los miembros del archivo que han sido eliminados. Al añadir nuevos miembros se utilizaría si es posible el espacio de la lista libre antes de expandir el tamaño del archivo.

Para acelerar el acceso individual a los miembros, los archivos ar suelen incluir una tabla de miembros al final del archivo que sirve de índice. Tampoco implementaremos esta funcionalidad.

En librerías estáticas es útil también incluir una tabla de símbolos que le facilita la búsqueda al linker.

Miembros y Cabeceras Locales

Cada miembro del archivo está precedido por una cabecera local, que almacena los atributos del fichero original. Para simplificar, *simplear* utiliza una cabecera local de tamaño fijo. El nombre de los miembros está limitado a un máximo de 15 caracteres (se utiliza un `\0` para indicar su final). El formato de una cabecera local es el siguiente:

```
struct ar_lhdr                   /* Cabecera de miembro de archivo */
{
    char ar_memname[16]; /* Nombre del miembro (maximo de 15 caracteres) */
    long ar_size;        /* Tamaño del contenido del miembro (en bytes) */
    char ar_date[12];    /* Fecha */
    int ar_uid;          /* ID de usuario */
    int ar_gid;          /* ID de grupo */
    char ar_mode[12];    /* Modo - octal */
};
```

```
};
```

El contenido del correspondiente fichero se almacena a continuación de su cabecera.

Script de comprobación para *simplear*

Para comprobar el funcionamiento de *simplear* se debe desarrollar un *script bash* para *testear* las opciones básicas. El script creará un archivo *ar* con el contenido de unos ficheros utilizando nuestro programa *simplear*, y luego extraerá el contenido y comprobará que los ficheros extraídos son idénticos a los originales. Si todo es correcto terminará devolviendo 0. Si hay errores terminará devolviendo 1. El script deberá seguir el siguiente esquema:

- Comprobará que el programa *simplear* está en el directorio actual y que es ejecutable. En caso contrario mostrará un mensaje informativo por la salida de error estándar y terminará.
- Comprobará si existe un directorio *tmp* dentro del directorio actual. Si existe lo borrará, incluyendo todo lo que haya dentro de él (mirar la opción *-r* del comando *rm*).
- Creará un nuevo directorio temporal *tmp* dentro del directorio actual y cambiará a este directorio.
- Creará tres ficheros (dentro del directorio):
 - *file1.txt*: con el contenido *Hello world!*, utilizando la orden *echo* y redirigiendo la salida al fichero.
 - *file2.txt*: con una copia de las 10 primeras líneas del fichero */etc/passwd*. Se puede hacer con el comando *head* y redirigiendo la salida al fichero.
 - *file3.dat*: con un contenido aleatorio de 1024 bytes, tomado del dispositivo */dev/urandom*. De nuevo conviene utilizar *head* con la opción *-c*.
- Invocará el programa *simplear* que hemos desarrollado, para crear un archivo *miarchivoar.ar* con el contenido de los tres ficheros anteriores.
- Creará un directorio *out* (dentro del directorio actual, que debe ser *tmp*) y copiará el fichero *archivoar.ar* al nuevo directorio.
- Cambiará al directorio *out* y ejecutará el programa *simplear* para extraer el contenido del archivo.
- Usará el programa *diff* para comparar los ficheros extraídos con los originales, que estarán en el directorio anterior (*..*).
- Si los tres ficheros extraídos son iguales que los originales, volverá al directorio original (*../.*), mostrará el mensaje “Correcto” por la salida estándar y devolverá 0. Si hay algún error, volverá al directorio original, mostrará un mensaje descriptivo por la salida de error estándar y devolverá 1.