

Resumen proyecto (changelog): “Tar tools”

Utilidad para gestionar archivos tar de formato gnu

1. Actividad A4.6 (Crear tar)

create_mytar v0.1.0 - Create_mytar (actividad A4.6)

- Utilidad que aparentemente crea bien el archivo tar dado un fichero regular pasado como argumento.
 - El resultado era correcto al verificar con “tar tvf archivo.tar”

2. Actividad A5.3 (Insertar y extraer ficheros regulares)

tar_tools v0.2.0 - Primer intento de insertar_fichero

- Tras muchos errores de E_TARFORM, nos dimos cuenta de que **no estábamos escribiendo bien los “ceros” del padding** del tar para fuera múltiplo de 10KiB.
 - "tar tvf " funciona con que haya 2 bloques de 0s al final del tar.
 - (el tamaño solo era 10240 al insertar un archivo creando el tar) desde cero)

tar_tools v0.2.1 - Segundo intento de insertar_fichero

- Identificamos que el **error estaba en que no actualizábamos bien la variable que contaba el tamaño del fichero tar al recorrerlo para llegar al final de la última entrada del mismo.** (en la función seek_pos_last_header)

tar_tools v0.3.1 - Versión final de insertar_fichero_regular y extraer_fichero_regular

- En extraer_fichero, al recorrer los header ya no comprobábamos si header.name fuese "" (string vacío), sino que comprobábamos los primeros 5 bytes del campo magic, es decir, la palabra “ustar”.
- La utilidad funcionaba en la mayoría de los casos al insertar/extraer ficheros regulares.
 - Pero si el tar estaba corrupto, pero con tamaño múltiplo de 10 KiB, el programa desbordaba la pila, había un error “segmentation fault” en strcpy() al comprobar el nombre para buscar un archivo para extraer. (solucionado versión final bug*)

3. Versión gestionar directorios, enlaces y permisos

tar_tools v0.3.0

- Se solventaron cantidad de bugs:
 - **Bug “calculo erróneo de data offset en el header”:**
 - Calculábamos de manera errónea el offset en el caso de que el tamaño del archivo fuese múltiplo de TAMAÑO DE BLOQUE (512).
 - **Bug “calculo erróneo de padding” (igual que antes):**
 - Calculábamos de manera errónea el padding en el caso de que el tamaño del ya fuera múltiplo de 10 KiB.
 - **Bug “calculo erróneo de módulo” en WriteDataBlock:**
 - Al igual que los 2 bugs anteriores, hacíamos mal el cálculo. (añadíamos un bloque de 0s innecesario)
- Bug* “desbordamiento” al realizar strcpy o strcmp al campo header.name:
 - Al leer el header no comprobamos si **name** tenía el último carácter nulo. (al pasar un tar completamente corrupto)
 - -> **read_header_get_size()** (soluciona esto)
- La otra función que recorre el tar, seek_eof_last_header(), ahora comprueba 5Bytes del campo magic.

tar_tools v1.0.0 (final)

- En esta versión se terminó la implementación de insertar/extraer directorios; y se hizo la parte de permisos.
- Se **eliminaron cantidad de strcpy()** innecesarios (tanto en la comparación de campos del header como en el paso de parámetros de la función main a los subprogramas).
 - Pasamos a comprobar todo el campo magic 6 Bytes (no sólo los primeros 5 bytes como antes). Ahora se usa la función **strncmp()** para ello.
- Para evitar repeticiones de código se decide modularizar al máximo el proyecto:
 - Se crea carpeta lib, con diversos subprogramas:
 - **seek_headers.h**
 - -> Se sitúan aquí las funciones que desplazan la cabeza lectora del descriptor fichero del tar (encuentran fichero a extraer o la posición para insertar un fichero)
 - **Auxiliary functions.h** (muy usadas, simplifican el código)

- -> Funciones para asignar permisos a ficheros, para leer un header y realizar comprobaciones y obtener campo size, para abrir el archivo tar comprobar formato y realizar stat (en una sola función), etc.
 - De la misma manera, en la función insertar_fichero o extraer_fichero, ahora se realizan operaciones comunes como gestionar la apertura de fichero tar, en vez de hacerlo en los subprogramas correspondientes.
- FINALMENTE, en esta versión los **permisos se gestionan correctamente**. (no en la apertura de los archivos; sino modificándolos una vez creados los archivos.)
- Se han intercambiado los valores de un par de errores en insertar (error abrir_tar por abrir_fichero_dat) y añadido alguno código de error más.
- Bug solventado:
 - En la inserción se comprueba que los argumentos f_dat y f_mytar no sean el mismo fichero (mismo nombre) para evitar un bucle infinito de escritura/lectura.