

Documento para la verificación final

Programa o utilidad a verificar: extrae_fichero (v1.0.0) ([hasta actividad 9.7](#))

Desarrollado por: Telmo Sendino Sáinz, Marcos Chouciño Grijalbo

Autores de los casos de prueba: Telmo Sendino Sáinz, Marcos Chouciño Grijalbo

Verificadores: Telmo Sendino Sáinz, Marcos Chouciño Grijalbo

(grupo ISO-1-10)

Notas:

- El binario **extraer fich** ha sido compilado con el comando:
 - `gcc ./lib/*.c ./tar_tools_extraer/*.c -o extraer_fich`
- El archivo tar ha sido creado con la utilidad inserta_fichero.

(VÉASE *script de verificación para más detalles*)

Caso de prueba 1: <i>Ejecutar el programa con menos de 2 parámetros</i>
Objetivo: Ejecutar el programa con menos del mínimo de parámetros necesarios. Entrada: <code>./extraer_fich</code>
Comportamiento previsto: Devuelve mensaje de error (indicando el correcto uso del programa).
Resultado de la verificación: <i>El esperado.</i>

Caso 2: <i>Ejecutar el programa con un archivo tar inexistente</i>

Objetivo: Ejecutar el programa con un archivo tar inexistente.

Entrada: `./extraer_fich fichero1.txt inexistente.tar`

Comportamiento previsto: Devuelve mensaje de error (no se puede abrir el fichero tar)

Resultado de la verificación: *El esperado*

Caso 3: *Ejecutar el programa con un archivo que no sea tar*

Objetivo: Ejecutar el programa con un archivo que no tenga el formato de un tar.

Entrada: `./extraer_fich fichero1.txt fichero2.txt`

Comportamiento previsto: Devuelve mensaje de error (no se puede abrir f_dat)

Resultado de la verificación: *El esperado*

Caso 4: *Extraccion (desde un archivo tar válido que lo contiene) de un fichero regular de tamaño mayor que el bloque de datos*

Objetivo: Extrae el fichero regular con el contenido e información de control que tuviera (y su tamaño es menor que 512 Bytes)

Entrada: `./extraer_fich fichero1.txt mytar.tar`

Comportamiento previsto: Recupera el archivo insertado con el contenido que tuviera.

Resultado de la verificación: Se extrae correctamente.

Caso 5: *Fichero regular que no se encuentre en el tar*

Objetivo: Intentar extraer un fichero que no está en el archivo tar.

Entrada: ./extraer_fich ficheroinexistente.txt mytar.tar

Comportamiento previsto: No se extrae el fichero, no debe crear un fichero vacío tampoco.

Resultado de la verificación: Devuelve un error indicando que el fichero a extraer no está en el archivo tar.

Caso 6: *Fichero a extraer con tamaño menor que tamaño de bloque de datos y archivo tar correcto*

Objetivo: Extrae el fichero regular con el contenido e información de control que tuviera (y su tamaño es menor que 512 Bytes)

Entrada: ./extraer_fich fichero1.txt mytar.tar

Comportamiento previsto: Recupera el archivo insertado con el contenido que tuviera.

Resultado de la verificación: El esperado

Caso 7: *Ejecutar el programa con un enlace que este en el tar*

Objetivo: Ejecutar el programa un enlace que se encuentre en el tar

Entrada: ./extraer_fich enlace mytar.tar

Comportamiento previsto:

Resultado de la verificación: Funciona como se esperaba.

Caso 8: *Ejecutar el programa con un directorio vacío*

Objetivo: Ejecutar el programa un directorio *vacío*

Entrada: `./extraer_fich DirectorioVacio mytar.tar`

Comportamiento previsto: Si el directorio no existe, lo crea en directorio correctamente.

Resultado de la verificación: Funciona como se esperaba.

Caso 9: *Ejecutar el programa con un directorio con más de un elemento dentro*

Objetivo: Ejecutar el programa a extraer un directorio

Entrada: `./extraer_fich directorioLleno mytar.tar`

Comportamiento previsto:

Resultado de la verificación: funciona

Caso 10: *Ejecutar el programa con un directorio con que tiene el mismo nombre que otro para ver si se sobrescribe el contenido del mismo*

Objetivo: Comprobar si un fichero dentro de un directorio se sobrescribe (como indica la especificación) o no.

Entrada:

`echo "Esto debe sobrescribirse"> directorioLleno/ficheroDentro.txt`

`./extraer_fich directorioLleno mytar.tar`

`cat ./directorioLleno/ficheroDentro.txt`

Comportamiento previsto: Se sobrescriben y se ve en la pantalla la palabra “funciona” después de hacer un cat al nuevo archivo

Resultado de la verificación: El esperado, funciona.

Caso 11: Comprobación de que la información de control (permisos y propietario, etc) se almacena en el tar

Objetivo: *Para ver como se mantiene la información de control se realiza la sobrescritura de un fichero regular existente, y la extracción de un directorio (previamente borrado)*

Entrada:

ls -la fichero1.txt (estos permisos cambiaran)

**** borramos directorio y fichero ****

./extraer_fich directorioLleno mytar2.tar

./extraer_fich fichero1.txt mytar2.tar

***** observamos los permisos que tiene directorio (rwxrwxrwx) y el fichero (rwxr-xr-x)*****

ls -la

Comportamiento previsto: *El archivo fichero1.txt debería tener permisos de ejecución (en propietario-grupo-otros) y el directorio directorioLleno debería tener todos los permisos rwx activados (también para prop.-grupo. -otros).*

Resultado de la verificación: Vemos que funciona perfectamente; los permisos se mantienen.

