man(ISO) Especificación de la función inserta_fichero

NOMBRE
inserta_fichero - inserta un fichero en otro fichero con formato de mysha256_repo

SINOPSIS
#include "s_my_sha256header.h"
int inserta_fichero(char * f_mysha256-Repo, char * f_dat);

DESCRIPCIÓN
La función inserta_fichero, inserta el fichero f_dat en el fichero f_mysha256-Repo
con formato mysha256_repo.
Si f_mysha256-Repo no existe o está vacío, se debe crear el fichero f_mysha256-Repo
y se añadirá f_dat tal y como lo realiza la utilidad mysha256_repo inicial
(header fle record + data file).
Si f_mysha256-Repo contiene ficheros con un formato mysha256_repo correcto, se
añadirá el fichero fdat con formato mysha256_repo (header fle record + data file)
al final del fichero f_mysha256-Repo.
El formato de f_mysha256-Repo se describe en la estructura s_my_sha256header.h .
(ver s_my_sha256header.h).

Si se pasa como parámetro un directorio, insertará en el fichero f_mysha256-Repo todos los ficheros contenidos en el directorio, con formato mysha256_repo.

Además de esto, si se pasa como parámetro un fichero regular, se guardarán sus permisos en la variable permissions del struct c_sha256header.

VALOR DE RETORNO
Si todo funciona correctamente, inserta_fichero devolverá el número correspondiente
de archivos que hay dentro del fichero f_mysha256-Repo (número de ficheros
contenidos en f_mysha256-Repo).
En caso contrario no actualizará/creará el fichero f_mysha256-Repo y la función
retornará uno de códigos de error indicados en el apartado de ERRORES.

ERRORES
E_OPEN1 (-1)
No se puede abrir f_dat.
E_OPEN2 (-2)
No se puede abrir o crear f_mysha256-Repo.

```
       E_REPOFORM (-3)
           f_mysha256-Repo no tiene el formato de mysha256_repo.
       E_DESCO  (-99)
           Otro tipo de errores

NOTAS
       Nota 1: El programa que utilice esta función , deberá informar por
la salida
              de error estándar un mensaje indicando el tipo de error.

      Ejemplo de utilización:
      Se supone que se utiliza compilación separada y el código (en
lenguaje C) de la función
      inserta_fichero se encuentra en un fichero diferente. Actualiza el
fichero Makefile
      proporcionado en la  fase previa, añadiendo el nuevo fichero
inserta_fichero.o
      En el ejemplo de uso, la función inserta el fichero "./fichero3.dat"
      del fichero "./Repo-ejemplo.rep".
      "Repo-ejemplo.rep" debe ser un fichero con formato mysha256_repo.

      #include "s_my_sha256header.h"
      extern int inserta_fichero(char * f_mysha256-Repo, char * f_dat);
      ...
      unsigned int ret;

      n = inserta_fichero("./Repo-ejemplo.rep", "./fichero3.dat");
      if (n < 0) // Error
      {
          ....
      }
      ...

/**
* @file s_my_sha256header.h
* @author   Gonzalo Alvarez - Dpto. ATC/KAT - UPV-EHU
* @date     05/02/2024
* @brief    Include file with struct c_header_sha256
* @details  A header file(.h) with the data structure definition
*           (c_header_sha256). This file will be used to create a
*           "special file" that will store sha256 hash codes of a
*           set of files.
*
*               "SPECIAL  FILE"
*           ++++++++++++++++++++++++++++
*           + Header Files Record   0   +
*           +--------------------------+
*           +      Data File 0         +
*           +    0... N-1 bytes        +
*           +   (data of File 0)       +
```

```
*           ++++++++++++++++++++++++++++
*           + Header Files Record   1   +
*           +--------------------------+
*           +      Data File 1         +
*           +    0... N-1 bytes        +
*           +   (data of File 1)       +
*           ++++++++++++++++++++++++++++
*           + Header Files Record   2   +
*           +--------------------------+
*           +      Data File 1         +
*           +    0... N-1 bytes        +
*           +   (data of File 2)       +
*           ++++++++++++++++++++++++++++
*           +           ...            +
*           ++++++++++++++++++++++++++++
*           + Header Files Record   K-1 +
*           +--------------------------+
*           +      Data File K-1       +
*           +    0... N-1 bytes        +
*           +   (data of File K-1)     +
*           ++++++++++++++++++++++++++++
*
*/
#include <stdlib.h>
#include <stdio.h>
#include <openssl/sha.h>
#include <stddef.h>
#include <errno.h>
#include <string.h>

#include <fcntl.h>

#include <math.h>

#include <sys/stat.h>

#include <dirent.h>

#define OK (0)
#define ERROR_WRONG_NUMBER_ARGUMENTS (1)
#define ERROR_OPEN_DAT_FILE          (2)
#define ERROR_READ_DAT_FILE          (3)
#define ERROR_OPEN_SHA_REPO_FILE     (4)
#define ERROR_OTHER_1                (5)
#define ERROR_OTHER_2                (6)

#define FILE_HEADER_SIZE     512
#define READ_BLOCK_SIZE     (16 * 1024)                    // 16 KBytes

// Return error Codes
#define HEADER_OK (1)
```

```
#define HEADER_ERR (2)

#define HEX_SHA256_HASH_SIZE  (SHA256_DIGEST_LENGTH*2 +1 )  // 65 Bytes

// Source OpenSSL Cryptography and SSL/TLS Toolkit
//  https://www.openssl.org/docs/manmaster/man3/SHA256_Init.html
//  SHA256_CTX are Deprecated functions. ( new EVP_Digest)

#define DATA_VALID_SIZE ( 256 + sizeof(off_t) + HEX_SHA256_HASH_SIZE )
#define UNUSED_DATA_SIZE ( FILE_HEADER_SIZE - DATA_VALID_SIZE)

struct c_sha256header {
        char fname[256];                  // file name
        off_t size;                       // similar to a 32-bit integer
        char hash[HEX_SHA256_HASH_SIZE];  // hash code of file fname
(hexadecimal string)
        mode_t permissions; //permisos del archivo
        // to complete in subsequent versions of the project
        char unused[UNUSED_DATA_SIZE];
};

/**
* end @file s_my_sha256header.h
**/

COMPATIBILIDAD
       inserta_fichero() debería funcionar en cualquier sistema UNIX.

VEASE TAMBIEN
       mysha256_repo(ISO), extrae_fichero(ISO).

AUTOR
       Ander Serrano, Mikel Leon y Koldo Intxausti
```