

man(ISO)

Especificación de la función `extrae_fichero`

NOMBRE

`extrae_fichero` - extrae un fichero de otro fichero con formato `mysha256_repo`

SINOPSIS

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

DESCRIPCIÓN

La función `extrae_fichero` extrae del fichero `f_mysha256-Repo` un fichero cuyo

nombre se indica en `f_dat`.

El fichero a extraer (`nombre, datos, ...`) se corresponde con la información

indicada en un "file system object" `f_mysha256-Repo` cuyo nombre coincide con

el valor de `f_dat`.

El "file system object" puede ser cualquiera de los ficheros almacenados en

`f_mysha256-Repo`, es decir, los ficheros guardados dentro de `f_mysha256-Repo`

no se encuentran de forma ordenada (ver `s_my_sha256header.h`).

Al crear el archivo a extraer del fichero `f_mysha256-Repo`, se creará con los permisos que se guardaron al insertarlo.

VALOR DE RETORNO

Si todo funciona correctamente, `extrae_fichero` devolverá cero. En caso

contrario no creará el fichero a extraer y retornará los errores indicados

en el apartado de ERRORES.

ERRORES

`E_OPEN` (-1)

No se puede abrir `f_mysha256-Repo`.

`E_NOEXIST` (-2)

`f_mysha256-Repo` no contiene el fichero de nombre `f_dat`.

`E_REPOFORM` (-3)

`f_mysha256-Repo` no tiene el formato `mysha256_repo`.

`E_CREATDEST` (-4)

No se puede crear/extraer el fichero `f_dat`.

`E_DESCO` (-99)

Otro tipo de errores

NOTAS

Nota 1: En caso de errores, el programa que utilice esta función , deberá informar por

la salida estándar de errores 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

extrae_fichero se encuentra en un fichero diferente. Actualiza el fichero Makefile

proporcionado en la fase previa, añadiendo el nuevo fichero extrae_fichero.o

En el ejemplo de uso, la función extrae el fichero de nombre `"/fichero3.dat"`

del fichero `"/Repo-ejemplo.rep"`. Verifica que el fichero extraído es correcto

`"Repo-ejemplo.rep"` debe ser un fichero con formato `mysha256_repo`.

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

n = extrae_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>

#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

extrae_fichero() debería funcionar en cualquier sistema UNIX.

VEASE TAMBIEN

mysha256_repo(ISO), inserta_fichero(ISO).

AUTOR

Ander Serrano, Mikel Leon y Koldo Intxausti

1.0

14 Feb 2024

man(ISO)