Файловая система FX-FS

Описание прикладного интерфейса

Версия 0.9

Содержание

Введение

Об этом руководстве

Настоящий документ описывает интерфейс прикладного программирования (API) для реализации файловой системы FAT для встроенных систем.

Формат описания функций API

Описание функции API включает в себя 6 основных разделов:

Прототип

Данный раздел содержит прототип описываемой функции на языке C в том виде, в котором она описана в заголовочных файлах ФС.

Описание

После прототипа приводится краткое описание функционала данной функции.

Аргументы

Содержит список аргументов функции (в том виде, как он описан в разделе «прототип») и их краткие описания и/или возможные значения.

Возвращаемое значение

Для функций, возвращающих значение, оно описывается в данном разделе. Если возвращаемое значение имеет определенные символьные имена (например, коды ошибок и их символьные значения) – приводится описание для каждого из них.

Ремарки

Данный раздел содержит описание особых сценариев использования функции, а также дополнительную информацию для разработчиков.

Пример

Содержит пример кода на C, в котором используется описываемая функция.

Внешние зависимости

FX-FS может использоваться как совместно с ОС, так и в системах без ОС. В последнем случае синхронизация доступа к носителю (недопущение параллельного выполнения функций FX-FS) возлагается на пользователя.

При использовании в виде исходных текстов требуются следующие стандартные заголовочные файлы (и реализация соответствующих функций):

- `stdint.h`
- `stdbool.h`
- `stddef.h`
- `errno.h`
- `memory.h`
- `string.h`
- `ctype.h`

Используемые коды ошибок: `ENOSPC; EINVAL; EIO; ENOENT; EAGAIN; ENOTDIR; EISDIR; EPERM; ENOMEM`.

Драйверы

Для упрощения программирования FX-FS не определяет интерфейс драйвера устройства, при инициализации носителя информации указываются функции чтения и записи, которые используются в дальнейшем внутри ФС. Если ФС сконфигурирована для работы с поддержкой ОС, гарантируется, что функции для работы с носителем не будут выполняться параллельно, однако обработка прерываний и прочие действия выполняются на стороне приложения. При инициализации носителя указывается также аргумент, который передается в функции чтения и записи, и может использоваться в качестве контекста драйвера.

Функция чтения имеет следующий прототип:

```
int fs_device_read(
void* context, void* buffer, size_t* sz, uint32_t blk_offset);
```

Функция возвращает 0 в случае успешного завершения. Смещение указывается в блочном виде, например, смещение 1 в устройстве с секторами в 512 байт читает сектор 1 (512-ый байт), а не байт 1. Размер указывается в байтах, функция должна также возвращать в этой же переменной актуальное количество прочитанных данных.

Функция записи имеет аналогичный прототип:

```
int fs_device_write(
void* context, void* buffer, size_t* sz, uint32_t blk_offset);
```

Назначение аргументов аналогично функции чтения, только буфер используется в качестве источника, а не приемника данных.

Указатель на функцию с таким прототипом определен как fs_device_func_t.

Интерфейсы FX-FS

Управление носителем

fs_media_init

```
int fs_media_init(
   fs_media_t* media,
   fs_device_func_t read,
   fs_device_func_t write,
   void* info,
   void* ptr,
   size_t sz);
```

Описание

Инициализация объекта "носитель данных". Пользователь должен обеспечить существование объекта носителя в течении всего периода работы с ним, а также предотвратить любое использование носителя после вызова функции fs_media_deinit.

Аргументы

Аргумент	Описание
media	Указатель на выделенный пользователем объект "носитель".
read	Функция чтения драйвера устройства.
write	Функция записи драйвера устройства.
info	Контекст драйвера (передается в функции драйвера в качестве аргумента context).
ptr	Указатель на область временных данных, используемых файловой системой для внутренних нужд.
sz	Размер области временных данных, для файловых систем FAT без поддержки кэширования минимальный размер определяется как размер кластера + 3 сектора (2 для чтения и записи + 1 для текущего элемента каталога).

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на структуру объекта или функции драйвера.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t disk;

device_driver_context_t context;
int disk_read(void* context, void* buf, size_t* size, uint32_t offset);
int disk_write(void* context, void* buf, size_t* size, uint32_t offset);

static int scratch[(SECTORS_PER_CLUSTER_MAX + 3) * SECTOR_SIZE];

uintptr_t my_thread(void* arg)
{
  Driver initialization...
  ...
  fs_media_init(&disk,
   disk_read, disk_write, &context,
   scratch, sizeof(scratch));

  return 0;
}
```

fs_media_deinit

```
int fs_media_deinit(fs_media_t* media);
```

Описание

Деинициализация объекта "носитель данных". Пользователь должен гарантировать что устройство не используется в момент вызова данной функции, а также что оно не будет использоваться после, до повторной инициализации.

Аргументы

Аргумент	Описание
media	Указатель на объект "носитель", который предварительно был успешно проинициализирован с помощью функции fs_media_init.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на структуру объекта.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t disk;

device_driver_context_t context;
int disk_read(void* context, void* buf, size_t* size, uint32_t offset);
int disk_write(void* context, void* buf, size_t* size, uint32_t offset);

static int scratch[(SECTORS_PER_CLUSTER_MAX + 3) * SECTOR_SIZE];

uintptr_t my_thread(void* arg)
{
  int error;
  Driver initialization...
  ...
  error = fs_media_init(&disk,
   disk_read, disk_write, &context,
   scratch, sizeof(scratch));

  if(!error) fs_media_deinit(&media);

  return 0;
}
```

fs_media_lock

```
void fs_media_lock(fs_media_t* media);
```

Описание

Блокировка носителя. При необходимости обратиться к устройству напрямую с помощью функций драйвера, требуется предварительно заблокировать обращения к носителю со стороны ФС.

Аргументы

Аргумент	Описание
media	Указатель на объект "носитель", который предварительно был успешно проинициализирован с помощью функции fs_media_init.

Возвращаемое значение

Нет.

Ремарки

Нет.

Пример

```
fx_media_t media;

uintptr_t my_thread(void* arg)
{
  ... Media & driver initialization...

  fs_media_lock(&media);
  ... media is locked
  fs_media_unlock(&media);

  return 0;
}
```

fs_media_unlock

```
void fs_media_unlock(fs_media_t* media);
```

Описание

Разблокировка носителя. При необходимости обратиться к устройству напрямую с помощью функций драйвера, требуется предварительно заблокировать обращения к носителю со стороны ФС. Носитель блокируется с помощью функции fs_media_lock.

Аргументы

Аргумент	Описание
media	Указатель на объект "носитель", который предварительно был успешно проинициализирован с помощью функции fs_media_init и заблокирован функцией fs_media_lock.

Возвращаемое значение

Нет.

Ремарки

Нет.

Пример

```
fx_media_t media;

uintptr_t my_thread(void* arg)
{
  ... Media & driver initialization...

  fs_media_lock(&media);
  ... media is locked
  fs_media_unlock(&media);

  return 0;
}
```

Логические разделы

fs_volume_open

```
int fs_volume_open(fs_media_t* media, fs_vol_t* volume, uintptr_t part);
```

Описание

Функция инициализирует объект "раздел".

Аргументы

Аргумент	Описание
media	Указатель на объект "носитель", который предварительно был успешно проинициализирован с помощью функции fs_media_init.
volume	Объект "раздел", который требуется проинициализировать.
part	Идентификатор раздела. Для носителей со стандартной структурой MBR - номер одного из основных разделов (0-3).

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на структуру объекта. Несвпадение типа файловой системы (раздел не является разделом, поддерживаемым FX-FS).
EIO	Ошибка ввода-вывода при работе с устройством.
ENOMEM	Недостаточно места для размещения внутренних буферов файловой системы. Попробуйте увеличить размер буфера, указанного при инициализации носителя в функции fs_media_init.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume1;

uintptr_t my_thread(void* arg)
{
  int error;

   ... Driver initialization...
   fs_media_init(&media, ...

   error = fs_volume_open(&media, &volume1, 1);
  return 0;
}
```

fs_volume_close

```
int fs_volume_close(fs_vol_t* volume);
```

Описание

Завершение работы с разделом ФС. Пользователь должен предотвратить использование раздела после вызова этой функции.

Аргументы

Аргумент	Описание
volume	Объект "раздел", который требуется закрыть.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на структуру объекта.
0	Успешное завершение.

Ремарки

Функция должна вызываться только в контексте обработчиков (программных или аппаратных) прерываний, при вызове в другом контексте результат ее выполнения не определен.

Пример

```
fs_vol_t volume;

uintptr_t my_thread(void* arg)
{
   fs_volume_close(&volume);

  return 0;
}
```

Каталоги

fs_dir_create

```
int fs_dir_create(fs_vol_t* volume, char* directory_name);
```

Описание

Создание каталога на указанном разделе.

Аргументы

Аргумент	Описание
volume	Открытый ранее раздел.
directory_name	Полный путь к создаваемому каталогу. В качестве корневого каталога раздела используется имя /. Например, для создания каталога MY_DIR в корневом каталоге открытого раздела, следует указывать имя "/MY_DIR". Создается только последний каталог, если какой-то промежуточный компонент пути отсутствует - функция возвращает ошибку.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на имя создаваемого каталога.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOSPC	Недостаточно места для создания нового каталога.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_dir_create(&volume, "/MY_DIR");

  return 0;
}
```

fs_dir_delete

```
int fs_dir_delete(fs_vol_t* volume, char* directory_name);
```

Описание

Удаление каталога на указанном разделе.

Аргументы

Аргумент	Описание
volume	Открытый ранее раздел.
directory_name	Полный путь к удаляемому каталогу. В качестве корневого каталога раздела используется имя /. Например, для удаления каталога MY_DIR в корневом каталоге открытого раздела, следует указывать имя "/MY_DIR".

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на имя удаляемого каталога.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOENT	Указанный каталог не найден.
0	Успешное завершение.

Ремарки

Удалять разрешается только пустые каталоги (не содержащие файлов и других каталогов)!

Пример

```
fs_media_t media;
fs_vol_t volume;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_dir_delete(&volume, "/MY_DIR");

  return 0;
}
```

fs_dir_open

```
int fs_dir_open(fs_vol_t* volume, char* directory_name, fs_dir_t *dir);
```

Описание

Открытие каталога для перечисления его элементов.

Аргументы

Аргумент	Описание
volume	Открытый ранее раздел.
directory_name	Полный путь к каталогу. В качестве корневого каталога раздела используется имя /.
dir	Структура открытого каталога, которая инициализируется данной функцией.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела, открываемого каталога или нулевой указатель на строку, содержащую путь.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOENT	Указанный каталог не найден.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_dir_t directory;
static char buf[MAX_PATH];

uintptr_t my_thread(void* arg)
{
  int error;
   ... driver & media initialization, volume opening...

   error = fs_dir_open(&volume, "/MY_DIR", &directory);
   if(!error)
  {
        fs_dir_read_first_entry(&directory, buf);
        fs_dir_close(&dir);
  }

  return 0;
}
```

fs_dir_close

```
int fs_dir_close(fs_dir_t *dir);
```

Описание

Закрытие каталога после перечисления его элементов.

Аргументы

Аргумент	Описание
dir	Структура каталога открытого ранее с помощью функции fs_dir_open.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_dir_t directory;
static char buf[MAX_PATH];

uintptr_t my_thread(void* arg)
{
  int error;
   ... driver & media initialization, volume opening...

   error = fs_dir_open(&volume, "/MY_DIR", &directory);
   if(!error)
  {
        fs_dir_read_first_entry(&directory, buf);
        fs_dir_close(&dir);
  }

  return 0;
}
```

fs_dir_read_first_entry

```
int fs_dir_read_first_entry(fs_dir_t *dir, char* entry);
```

Описание

Получение имени первого элемента в каталоге. Функция также инициализирует внутренний итератор, находящийся внутри объекта каталога, для чтения второго и следующих элементов каталога следует пользоваться функцией fs_dir_read_next_entry.

Аргументы

Аргумент	Описание
dir	Структура каталога открытого ранее с помощью функции fs_dir_open.
entry	Буфер, где будет сохранено имя первого элемента в каталоге.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела каталога или нулевой указатель на буфер.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOENT	Указанный каталог не содержит ни одного элемента.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_dir_t directory;
static char buf[MAX_PATH];

uintptr_t my_thread(void* arg)
{
  int error;
   ... driver & media initialization, volume opening...

   error = fs_dir_open(&volume, "/MY_DIR", &directory);
   if(!error)
  {
        fs_dir_read_first_entry(&directory, buf);
        fs_dir_close(&dir);
  }

  return 0;
}
```

fs_dir_read_next_entry

```
int fs_dir_read_next_entry(fs_dir_t *dir, char* entry);
```

Описание

Получение имени второго и следующих элементов каталога. После каждого успешного вызова производится инкремент встроенного в объект итератора элементов каталога. Вызову данной функции всегда должен предшествовать вызов функции fs_dir_read_first_entry.

Аргументы

Аргумент	Описание
dir	Структура каталога открытого ранее с помощью функции fs_dir_open.
entry	Буфер, где будет сохранено имя элемента в каталоге.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела каталога или нулевой указатель на буфер.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOENT	Указанный каталог не содержит следующего по счету элемента.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_dir_t directory;
static char buf[MAX_PATH];

uintptr_t my_thread(void* arg)
{
  int error;
   ... driver & media initialization, volume opening...

   error = fs_dir_open(&volume, "/MY_DIR", &directory);
   if(!error)
  {
        fs_dir_read_first_entry(&directory, buf);

        while(fs_dir_read_next_entry(&directory, buf) != ENOENT) ...

        fs_dir_close(&dir);
  }

  return 0;
}
```

fs_dir_rename

```
int fs_dir_rename(fs_vol_t* vol, char* old_name, char* new_name);
```

Описание

Переименование каталога.

Аргументы

Аргумент	Описание
vol	Открытый ранее раздел.
old_name	Полное путь к каталогу, который требуется переименовать.
new_name	Полное новое имя каталога (путь к нему не должен изменяться).

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на строки содержащие путь.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOENT	Указанный каталог не найден.
ENOTDIR	Хотя бы один промежуточный компонент пути не является каталогом.
0	Успешное завершение.

Ремарки

При вызове данной функции должен разичаться только последний компонент пути, при несоблюдении этого требования, поведение не определено.

Пример

```
fx_vol_t volume;

uintptr_t my_thread(void* arg)
{
  fs_dir_rename(&volume, "/MY_OLD_DIR", "/MY_NEW_DIR");
  return 0;
}
```

Файлы

fs_file_create

```
int fs_file_create(fs_vol_t* volume, char* file_name);
```

Описание

Создание нового файла на указанном разделе.

Аргументы

Аргумент	Описание
volume	Открытый ранее раздел.
file_name	Полный путь к создаваемому файлу. В качестве корневого каталога раздела используется имя /. Например, для создания файла MY_FILE в корневом каталоге открытого раздела, следует указывать имя "/MY_ FILE ".

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на имя создаваемого файла.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOSPC	Недостаточно места для создания нового файла.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_create(&volume, "/MY_FILE");

  return 0;
}
```

fs_file_delete

```
int fs_file_delete(fs_vol_t* volume, char* file_name);
```

Описание

Удаление файла на указанном разделе.

Аргументы

Аргумент	Описание
volume	Открытый ранее раздел.
file _name	Полный путь к удаляемому файлу. В качестве корневого каталога раздела используется имя /. Например, для удаления файла MY_FILE в корневом каталоге открытого раздела, следует указывать имя "/MY_ FILE ".

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на имя удаляемого файла.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOENT	Указанный файл не найден.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_delete(&volume, "/MY_FILE");

  return 0;
}
```

fs_file_open

```
int fs_file_open(
   fs_vol_t* vol, fs_file_t* file_ptr, char* file_name, int open_type);
```

Описание

Открытие файла для чтения или записи.

Аргументы

Аргумент	Описание
vol	Объект "раздел", на котором требуется открыть файл.
file_ptr	Указатель на структуру открытого файла (инициализируется данной функцией).
file_name	Полный путь к открываемому файлу. В качестве корневого каталога раздела используется имя /. Например, для открытия файла MY_FILE в корневом каталоге открытого раздела, следует указывать имя "/MY_ FILE ".
open_type	Зарезервировано, следует устанавливать этот аргумент в 0.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела, файла или нулевой указатель на путь к файлу.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOTDIR	Хотя бы один элемент пути не является каталогом.
ENOENT	Указанный файл не найден.
EISDIR	Последний компонент пути является каталогом.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
  return 0;
}
```

fs_file_close

```
int fs_file_close(fs_file_t *file_ptr);
```

Описание

Закрытие файла открытого ранее для чтения или записи.

Аргументы

Аргумент	Описание
file_ptr	Указатель на структуру файла, проинициализированный функцией fs_file_open.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект файла.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;

uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
   ...
   fs_file_close(&file);
  return 0;
}
```

fs_file_read

```
int fs_file_read(
   fs_file_t *file_ptr,
   void* buffer_ptr, uint32_t request_size, uint32_t* actual_size);
```

Описание

Чтение файла.

Аргументы

Аргумент	Описание
file_ptr	Указатель на объект "файл", проинициализированный ранее с помощью вызова fs_file_open.
buffer_ptr	Буфер-приемник для данных, читаемых из файла.
request_size	Запрашиваемый объем данных.
actual_size	Указатель на переменную, где будет сохранено актуальное количество прочитанных данных.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на обязательный аргумент.
EIO	Ошибка ввода-вывода при работе с носителем.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;
char buf[READ_BUF_SIZE];
uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
   if(!error)
   {
         uint32_t read_sz;
         fs_file_read(&file, buf, READ_BUF_SIZE, &read_sz);
         fs_file_close(&file);
   }
  return 0;
}
```

fs_file_seek

```
int fs_file_seek(fs_file_t *file_ptr, uint32_t byte_offset, int method);
```

Описание

Установка текущего указателя чтения и записи для файла.

Аргументы

Аргумент	Описание
file_ptr	Указатель на объект "файл", проинициализированный ранее с помощью вызова fs_file_open.
byte_offset	Смещение в файле относительно точки определяемой параметром method.
method	0 - смещение относительно начала файла 1 - смещение относительно текущей позиции 2 - смещение относительно конца файла

Возвращаемое значение

0 в случае успешного завершения, EINVAL в случае некорректных аргументов.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;
char buf[READ_BUF_SIZE];
uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
   if(!error)
   {
         fs_file_seek(&file, 0x500, 0);
         fs_file_close(&file);
   }
  return 0;
}
```

fs_file_trunc

```
int fs_file_trunc(fs_file_t *file_ptr, uint32_t size);
```

Описание

Установка размера файла. Если файл имеет больший размер он будет уменьшен до указанного, если файл имеет меньший размер, на диске будет зарезервировано место под указанный объем данных.

Аргументы

Аргумент	Описание
file_ptr	Указатель на объект "файл", проинициализированный ранее с помощью вызова fs_file_open.
size	Новый размер файла.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на обязательный аргумент.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOSPC	Недостаточно места на диске для увеличения файла.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;
char buf[READ_BUF_SIZE];
uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...

   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
   if(!error)
   {
         fs_file_trunc(&file, 0x10000); //Reserve 64K for the file.
         fs_file_close(&file);
   }
  return 0;
}
```

fs_file_write

```
int fs_file_write(fs_file_t *file_ptr,
   void* buffer_ptr, uint32_t request_size, size_t* size);
```

Описание

Запись в файл.

Аргументы

Аргумент	Описание
file_ptr	Указатель на объект "файл", проинициализированный ранее с помощью вызова fs_file_open.
buffer_ptr	Буфер-источник для данных, записываемых в файл.
request_size	Записываемый объем данных.
actual_size	Указатель на переменную, где будет сохранено актуальное количество записанных данных.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на обязательный аргумент.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOSPC	Недостаточно места на носителе для завершения операции.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fs_media_t media;
fs_vol_t volume;
fs_file_t file;
char buf[WRITE_BUF_SIZE];
uintptr_t my_thread(void* arg)
{
  int error;

   ... driver & media initialization, volume opening...
   ... set buffer with data to be written...
   error = fs_file_open(&volume, &file, "/FOLDER/TEST.TXT", 0);
   if(!error)
   {
         uint32_t wr_sz;
         fs_file_write(&file, buf, WRITE _BUF_SIZE, &wr_sz);
         fs_file_close(&file);
   }
  return 0;
}
```

fs_file_rename

```
int fs_file_rename(fs_vol_t* volume, char* old_name, char* new_name);
```

Описание

Переименование файла.

Аргументы

Аргумент	Описание
vol	Открытый ранее раздел.
old_name	Полное путь к файлу, который требуется переименовать.
new_name	Полное новое имя файла.

Возвращаемое значение

Код ошибки	Описание
EINVAL	Некорректный (нулевой) указатель на объект раздела или нулевой указатель на строки содержащие путь.
EIO	Ошибка ввода-вывода при работе с носителем.
ENOENT	Указанный файл не найден.
ENOTDIR	Хотя бы один промежуточный компонент пути не является каталогом.
0	Успешное завершение.

Ремарки

Нет.

Пример

```
fx_vol_t volume;

uintptr_t my_thread(void* arg)
{
  fs_file_rename(&volume, "/MY_OLD_FILE", "/MY_NEW_FILE");
  return 0;
}
```