

# 組込み向けファイルシステム G R - F I L E 旧構成への移行手順

第 1.00 版

2008 年 7 月

株式会社グレースシステム

**[注意事項]**

- すべての著作権は、株式会社グレープシステムにあります。
- 本ドキュメントの内容の一部または全部を無断で転載、複写、複製する事を禁じます。
- 本製品の仕様は予告なく変更される事があります。
- 本ドキュメントに記載されている会社名、製品名は各社の商標または登録商標です。

Copyright (C) 2008 Grape Systems, Inc. All Rights Reserved

## はじめに

本書は、組込みシステムでファイルアクセスを実現するためのミドルウェア「**GR-FILE**」のディレクトリ構成を、**GR-FILE** Version1.20 以前の構成に戻す手順を記述します。

## 改訂履歴

Rev.	日付	改訂内容
1.00	2008 年 7 月	初版

## 目次

1. 概要 .....	1
2. Version1.20 での変更点 .....	1
2.1 Version1.20 のディレクトリ構成について .....	1
2.2 移動された定義について .....	3
2.3 新規に作成されたファイルについて .....	4
2.3.1 grp_fat_param.h .....	4
2.3.2 grp_fs_sysdef.h .....	4
2.3.3 grp_fs_dev_sw_tbl.c .....	4
2.4 名称の変更されたファイルについて .....	4
3. 旧構成への移行手順 .....	5
3.1 ディレクトリの作成 .....	5
3.2 共通ファイルのコピー .....	5
3.2.1 baseファイルのコピー .....	6
3.2.2 includeファイルのコピー .....	6
3.2.3 文字コード処理関数ライブラリファイルのコピー .....	7
3.2.4 可変長メモリ管理関数ライブラリファイルのコピー .....	7
3.2.5 C言語標準I/Oインターフェースライブラリファイルのコピー .....	7
3.2.6 時刻情報変換関数ライブラリファイルのコピー .....	8
3.3 環境依存ファイルのコピー .....	8
3.3.1 μITRON依存ファイルのコピー .....	8
3.3.2 VOS依存ファイルのコピー .....	10
3.4 サンプルファイルのコピー .....	10
3.5 設定ファイルの確認 .....	11
3.6 ポーティングファイルの確認 .....	11
3.6.1 μITRON使用時 (mdep_itron) .....	12
3.6.2 VOS使用時 (mdep_vos) .....	12
3.6.3 文字列出力関数について (grp_fs_printf関数) .....	13
3.7 コンパイルオプションの削除 .....	13
3.8 カスタムソースファイルについて .....	13
3.9 パスの設定 .....	13
3.10 メイク環境へのファイルの追加 .....	13

## 1. 概要

組込み向けファイルシステム **GR-FILE** は、Version1.20 より使い易さの向上、ポーティング作業の明確化の為に、ディレクトリ構成が変更されています。

本書では、Version1.20 より採用されたディレクトリ構成を、Version1.20 以前の構成に戻す為の手順について説明します。

## 2. Version1.20 での変更点

**GR-FILE** Version1.20 では、**GR-FILE** のベースとなる部分と、環境に依存する部分に分けられています。その為、ディレクトリが細分化されています。

また、Version1.20 以前では複数のファイルに存在していた、**GR-FILE** の動作を決定する設定値も可能な限りまとめられています。

### 2.1 Version1.20 のディレクトリ構成について

Version1.20 で採用されているディレクトリ構成を、表 2.1-1 GR-FILE Version1.20 のディレクトリ構成に示します。

表 2.1-1 GR-FILE Version1.20 のディレクトリ構成

ディレクトリ/ファイル	内容
doc / lib /	ターゲット非依存 <b>GR-FILE</b> ベースソースファイル <b>GR-FILE</b> の生成したライブラリを格納するディレクトリ 各ディレクトリ下の Makefile を使ってライブラリを作成しますと、以下の 2 つのライブラリファイルが生成されます。 grp_file.a : grp_file 及び grp_stdio ファイルの関数ライブラリ grp_lib.a : その他のファイルの関数ライブラリ
src / grp_fs / base / fat.c fat.h fat_format_def.h grp_fat_format.c grp_fs.c grp_fs.h grp_fs_cfg.c grp_fs_cfg.h grp_fs_conv_lib.c grp_fs_dev_io_if.c grp_fs_error.c grp_fs_get_cwd_lib.c grp_fs_get_disk_part.c grp_fs_io_disk_part.c grp_fs_mdep_if.h grp_fs_readdir.c grp_fs_set_disk_part.c grp_fs_trace.c grp_fs_trace.h grp_queue.h Makefile	<b>GR-FILE</b> ソースファイル ターゲット非依存 <b>GR-FILE</b> ソースファイル <b>GR-FILE</b> ベースファイル FAT ファイルシステム依存処理 FAT ファイルシステム依存処理定義ヘッダファイル FAT フォーマット関数の内部ヘッダファイル FAT フォーマット処理関数 ファイルシステム非依存処理 ファイルシステム非依存処理定義ヘッダファイル ファイルシステムタイプテーブル OS 抽象化/デバイスドライバインタフェース定義ヘッダファイル 変換処理関数 デバイス直接制御インタフェース関数 <b>GR-FILE</b> エラー番号→メッセージ変換ライブラリ カレントディレクトリの取得関数 パーティション情報の取得関数 パーティション情報の読書き関数 OS 抽象化インタフェース定義ヘッダファイル POSIX インタフェース関数の opendir / closedir / readdir 関数 パーティション情報の設定関数 I/O トレースオプションコード I/O トレースオプションコードヘッダファイル リスト処理マクロ定義ヘッダファイル <b>GR-FILE</b> 関数ライブラリ (lib / grp_file.a) 生成用 Makefile
include / grp_char_conv.h grp_fat_format.h grp_fat_param.h grp_fs_conv.h grp_fs_dev_io_if.h grp_fs_disk_part.h grp_fs_if.h	アプリケーション向けターゲット非依存 <b>GR-FILE</b> ヘッダファイル 文字コード変換処理インタフェース定義 FAT フォーマットインタフェースの定義 <b>GR-FILE</b> のファイルシステム configuration パラメータ定義 POSIX 互換インタフェース定義 デバイス直接制御インタフェースの定義 パーティション設定/変更インタフェースの定義 <b>GR-FILE</b> 固有アプリケーションインタフェース定義

grp_fs_param.h grp_fs_readdir.h grp_fs_sysdef.h grp_mem.h grp_sem.h grp_stdio.h grp_time.h grp_time_lib.h grp_types.h	<b>GR-FILE</b> の configuration パラメータ定義 POSIX インタフェース opendir / closedir / readdir の定義 <b>GR-FILE</b> コンパイル定義 可変長メモリ管理インタフェース grp_mem_alloc / free の定義 セマフォ処理インタフェース定義 C 言語標準 I/O インタフェース定義 時刻設定/読出し関数インタフェース定義 時刻情報変換（トータル秒⇔日付時刻）インタフェース定義 基本タイプ定義
lib / grp_char / grp_char_sjis_conv.c grp_char_sjis_tbl.h Makefile	<b>GR-FILE</b> ターゲット非依存ライブラリ 文字コード処理関数ライブラリソース シフト JIS ⇔ UNICODE 変換ライブラリ関数 シフト JIS ⇔ UNICODE 変換テーブル 文字コード処理関数生成、及び、組込み向け関数ライブラリ（lib / grp_lib.a）への追加用 Makefile
grp_mem / grp_mem_vl_pool.c grp_mem_vl_pool.h Makefile	可変長メモリ管理関数ライブラリソース 可変長メモリ管理関数 可変長メモリ管理関数定義ヘッダファイル 可変長メモリ管理関数生成、及び、組込み向け関数ライブラリ（lib / grp_lib.a）への追加用 Makefile
grp_stdio / grp_stdio_default_io.c grp_stdio_fclose.c grp_stdio_fflush.c grp_stdio_fill.c grp_stdio_fopen.c grp_stdio_fprintf.c grp_stdio_fread.c grp_stdio_fseek.c grp_stdio_ftell.c grp_stdio_fwrite.c grp_stdio_getc.c grp_stdio_gets.c grp_stdio_put.c grp_stdio_vprintf.c Makefile	C 言語標準 I/O インタフェース関数ライブラリソース（オプション） 標準入出力(stdin / stdout / stderr)変数定義 fclose 処理 fflush 処理 I/O バッファ fill 処理 fopen 処理 fprintf 処理 fread 処理 fseek 処理 ftell 処理 fwrite 処理 getc 処理 gets 処理 putc 処理 vprintf 処理 C 言語標準 I/O インタフェース関数生成、及び、 <b>GR-FILE</b> 関数ライブラリ（libs / grp_file.a）への追加用 Makefile
grp_time_lib / grp_time_lib.c Makefile	時刻情報変換（トータル秒⇔日付時刻）関数ライブラリソース 時刻情報変換（トータル秒⇔日付時刻）関数 時刻情報変換関数生成、及び、組込み向け関数ライブラリ（lib / grp_lib.a）への追加用 Makefile
mdep_itron / base / grp_fs_dev_io.c grp_fs_dev_sw_tbl.c grp_fs_mdep_if.c include / grp_fs_mdep_types.h grp_mdep_sem.h lib / grp_itron_id.h grp_mem / grp_mem.c grp_sem / grp_sem.c grp_time / grp_time_get.c grp_time_set.c	$\mu$ ITRON 向けソースファイル  I/O 関数の空スタブコード デバイスドライバスイッチテーブル $\mu$ ITRON 向け OS 抽象化インタフェース  $\mu$ ITRON 向け <b>GR-FILE</b> タイプ定義 $\mu$ ITRON 向けセマフォタイプ定義  $\mu$ ITRON 向けセマフォ、メモリプール ID 定義  $\mu$ ITRON 向け可変長メモリ管理関数（grp_mem_alloc / free 関数）  $\mu$ ITRON 向けセマフォ関数  現在時刻取得関数の空スタブソース 時刻設定関数の空スタブソース
mdep_vos / base / grp_fs_dev_io.c grp_fs_dev_sw_tbl.c grp_fs_mdep_if.c include / grp_fs_mdep_types.h grp_mdep_sem.h grp_vos.h lib / grp_mem /	VOS 向けソースファイル  I/O 関数の空スタブコード デバイスドライバスイッチテーブル VOS 向け OS 抽象化インタフェース  VOS 向け <b>GR-FILE</b> タイプ定義 VOS 向けセマフォタイプ定義 VOS 向け基本タイプ定義

grp_mem.c grp_sem / grp_sem.c grp_time / grp_time_get.c grp_time_set.c	VOS 向け可変長メモリ管理関数 (grp_mem_alloc / free 関数)  VOS 向けセマフォ関数  現在時刻取得関数の空スタブソース 時刻設定関数の空スタブソース
sample / app / base / grp_fat_format_sd.c grp_fat_format_sd.h grp_fs_dev_io_ram.c grp_fs_dev_io_ram.h grp_fs_proc_event.c grp_fs_proc_event.h	サンプルコード サンプルアプリケーション サンプル関数 SD カードのフォーマット処理関数ライブラリ*1 SD カードのフォーマット処理関数ライブラリヘッダファイル*1 RAM ディスク I/O 関数 RAM ディスク I/O 関数ヘッダファイル プラットフォーム依存の挿抜処理関数 プラットフォーム依存の挿抜処理関数

※grp\_fat\_param.h、grp\_fs\_sysdef.h、grp\_fs\_confirmed\_porting.h は、新たに作成されたヘッダファイルです。

## 2.2 移動された定義について

移動された定義を、表 2.2-1 grp\_fs\_cfg.hからgrp\_fs\_param.hへ移動された定義と表 2.2-2 fat.hからgrp\_fat\_param.hへ移動された定義に示します。

表 2.2-1 grp\_fs\_cfg.h から grp\_fs\_param.h へ移動された定義

定義	内容
GRP_FS_MAX_FSTYPE	ファイルシステムテーブルの最大エントリ数 ※廃止
GRP_FS_MAX_MOUNT	同時に mount するファイルシステムの最大数
GRP_FS_MAX_FILE	同時にオープン可能なファイルの最大数
GRP_FS_MAX_FHDL	同時にオープン中のファイルハンドルの最大数
GRP_FS_FBLK_SHIFT	ファイル管理ブロックキャッシュのブロックサイズのシフト値
GRP_FS_DBLK_SHIFT	ファイルデータキャッシュのブロックサイズのシフト値
GRP_FS_FBLK_CNT	ファイル管理ブロックキャッシュのブロック数
GRP_FS_DBLK_CNT	ファイルデータキャッシュのブロック数
GRP_FS_BLK_NHASH	キャッシュブロックのハッシングバケットの数
GRP_FS_MAX_TASK	同時にファイルシステムをアクセスするタスクの最大数
GRP_FS_FILE_NHASH	オープン中のファイルのハッシングバケット数
GRP_FS_FNAME_CACHE_CNT	ファイル名称キャッシュの最大数
GRP_FS_FNAME_NHASH	ファイル名称キャッシュのハッシングバケットの数

※廃止

GRP\_FS\_MAX\_FSTYPE 定義は使用されていない為、Version1.20 より廃止されました。

表 2.2-2 fat.h から grp\_fat\_param.h へ移動された定義

定義	内容
FAT_BLK_SHIFT	FAT セクタサイズのシフト値
FAT_MAP_CNT	オープン中の各 FAT ファイルの領域情報のキャッシュ数
FAT_FREE_TBL	フリーブロックキャッシュの数
FAT_COMP_SZ	FAT ファイル名の各コンポーネントの最大長 (NULL を含む)
FAT_COMP_CHCNT	FAT ファイル名の各コンポーネントの最文字数 (NULL を含む)
FAT_CNT_BUF_SZ	フリークラスタ数を算出する際の FAT 用バッファサイズ

## 2.3 新規に作成されたファイルについて

Version1.20 で新たに作成されたファイルの説明をします。

### 2.3.1 grp\_fat\_param.h

表 2.2-2 fat.h から grp\_fat\_param.h へ移動された定義で示される、FAT ファイルシステムの各種設定を行います。

Version1.20 以前は、fat.h に定義されていました。

### 2.3.2 grp\_fs\_sysdef.h

**GR-FILE** のコンパイルスイッチを定義します。

本ヘッダファイルは、全てのソースファイルよりインクルードされます。

統合開発環境で開発されている場合は、統合開発環境側の定数定義は不要になります。

ご注意ください。

また、Version1.20 より、環境に依存する処理を分離しており、従来よりありました「GRP\_VOS」や「ITRON」と言った定義は不要となりました。

代わりに「mdep\_vos」、「mdep\_itron」を参考に、ポーティングして頂く必要があります。

### 2.3.3 grp\_fs\_dev\_sw\_tbl.c

デバイスドライバ関数テーブルの、登録テーブルが実装されます。

旧バージョンでは、「grp\_fs\_cfg.c」に実装されていたテーブルを、環境に依存する為、抜き出しています。

デバイスドライバを登録する場合は、このテーブルへ追加します。

## 2.4 名称の変更されたファイルについて

環境に依存するファイルは、ディレクトリ/ファイル名を同じ構成にして、ルートのディレクトリ名のみ異なる名称に変更されています。

この構成により、異なる環境へのポーティングが行い易くなっています。



### 3. 旧構成への移行手順

ここでは、**GR-FILE** Version1.20 を、Version1.20 以前の構成へ変更する手順を説明します。

ただし、Version1.20 では、ファイルの新規作成、ファイル名変更、定義の移動が行われています。

ここでは、ファイル単位の移動のみ説明します。

また、本作業を行う前に、ファイルのバックアップを行うことをお勧めします。

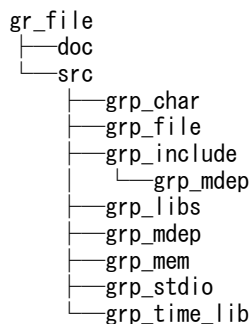
本作業を行い、Version1.20 以前の構成にした場合は、各ディレクトリの **Makefile** は使用できなくなりますので、環境に合わせ作成を行ってください。

#### 3.1 ディレクトリの作成

Version1.20 以前のディレクトリを作成します。

以下のツリー構造を作成します。（「**gr\_file**」は任意のディレクトリ名）

以降の説明は、「**gr\_file / src**」をルートとして説明します。



#### 3.2 共通ファイルのコピー

**GR-FILE** Version1.20 の各ディレクトリより、3.1 ディレクトリの作成で作成したディレクトリへ、ファイルをコピーします。

表 3.2-1 旧構成のコピー先で示すとおり、3.1 ディレクトリの作成で作成したディレクトリへ、ファイルをコピーします。

### 3.2.1 baseファイルのコピー

**GR-FILE**のベースとなるソースファイルを、表 3.2-1 旧構成のコピー先（ベースファイル）で示すディレクトリへコピーします。

表 3.2-1 旧構成のコピー先（ベースファイル）

src / grp_fs / base /		
fat.c, fat.h		
fat_format_def.h, grp_fat_format.c		
grp_fs.c, grp_fs.h		
grp_fs_cfg.c, grp_fs_cfg.h		
grp_fs_conv_lib.c		
grp_fs_dev_io_if.c	全てのファイルをコピーする	
grp_fs_error.c		
grp_fs_get_cwd_lib.c	⇒	grp_file
grp_fs_get_disk_part.c		
grp_fs_io_disk_part.c		
grp_fs_mdep_if.h		
grp_fs_readdir.c		
grp_fs_set_disk_part.c		
grp_fs_trace.c, grp_fs_trace.h		
grp_queue.h		

### 3.2.2 includeファイルのコピー

**GR-FILE**のベースとなるインクルードファイルを、表 3.2-2 旧構成のコピー先（インクルードファイル）に示すディレクトリへコピーします。

表 3.2-2 旧構成のコピー先（インクルードファイル）

src / grp_fs / include /		
grp_char_conv.h		
grp_fat_format.h		
grp_fat_param.h		
grp_fs_conv.h		
grp_fs_dev_io_if.h		
grp_fs_disk_part.h		
grp_fs_if.h	全てのファイルをコピーする	
grp_fs_param.h	⇒	grp_include
grp_fs_readdir.h		
grp_fs_sysdef.h		
grp_mem.h		
grp_sem.h		
grp_stdio.h		
grp_time.h		
grp_time_lib.h		
grp_types.h		

### 3.2.3 文字コード処理関数ライブラリファイルのコピー

**GR-FILE**で使用する、文字コード変換処理関数ライブラリファイルを、表 3.2-3 旧構成のコピー先（文字コード変換処理ライブラリファイル）で示すディレクトリへコピーします。

表 3.2-3 旧構成のコピー先（文字コード変換処理ライブラリファイル）

src / grp_fs / lib / grp_char /	
grp_char_sjis_conv.c	全てのファイルをコピーする
grp_char_sjis_tbl.h	⇒ grp_char

### 3.2.4 可変長メモリ管理関数ライブラリファイルのコピー

**GR-FILE**で使用する、可変長メモリ管理関数ライブラリファイルを、表 3.2-4 旧構成のコピー先（可変長メモリ管理関数ライブラリファイル）で示すディレクトリへコピーします。

表 3.2-4 旧構成のコピー先（可変長メモリ管理関数ライブラリファイル）

src / grp_fs / lib / grp_mem /	
grp_mem_vl_pool.c	全てのファイルをコピーする
grp_mem_vl_pool.h	⇒ grp_mem

### 3.2.5 C言語標準I/Oインターフェースライブラリファイルのコピー

**GR-FILE**で使用する、C言語標準I/Oインターフェースライブラリファイルを、表 3.2-5 旧構成のコピー先（C言語標準I/Oインターフェースライブラリファイル）で示すディレクトリへコピーします。

表 3.2-5 旧構成のコピー先（C 言語標準 I/O インターフェースライブラリファイル）

src / grp_fs / lib / grp_stdio /	
grp_stdio_default_io.c	
grp_stdio_fclose.c	
grp_stdio_fflush.c	
grp_stdio_fill.c	
grp_stdio_fopen.c	
grp_stdio_fprintf.c	
grp_stdio_fread.c	全てのファイルをコピーする
grp_stdio_fseek.c	⇒ grp_stdio
grp_stdio_ftell.c	
grp_stdio_fwrite.c	
grp_stdio_getc.c	
grp_stdio_gets.c	
grp_stdio_put.c	
grp_stdio_vprintf.c	

### 3.2.6 時刻情報変換関数ライブラリファイルのコピー

**GR-FILE**で使用する、時刻情報変換関数ライブラリファイルを、表 3.2-5 旧構成のコピー先（C言語標準I/Oインターフェースライブラリファイル）で示すディレクトリへコピーします。

表 3.2-6 旧構成のコピー先（時刻情報変換関数ライブラリファイル）

src / grp_fs / lib / grp_time_lib / grp_time_lib.c	全てのファイルをコピーする ⇒ grp_time_lib
---	------------------------------------

### 3.3 環境依存ファイルのコピー

Version1.20 より環境に依存する部分が分離されています。

旧構成へ移行するには、必要な依存ファイルをコピーします。

下記に、 $\mu$ ITRON と VOS の依存ファイルのコピー先を説明します。

「src / mdep\_itron」と「src / mdep\_vos」は、ご使用されている環境に合わせてファイルをコピーします。

$\mu$ ITRON、VOS 以外の環境をご使用の場合は、ポーティングされているファイルを、 $\mu$ ITRON、VOS の説明を参考にコピーを行ってください。

既にポーティング済みのファイルがある場合も、 $\mu$ ITRON、VOS の説明を参考にコピーを行ってください。

#### 3.3.1 $\mu$ ITRON依存ファイルのコピー

表 3.3-1 旧構成のコピー先（環境依存ファイル  $\mu$ ITRON）に  $\mu$ ITRONの環境依存ファイルのコピー先を示します。

表 3.3-1 旧構成のコピー先（環境依存ファイル  $\mu$ ITRON）

src / mdep_itron / base / grp_fs_dev_io.c grp_fs_dev_sw_tbl.c grp_fs_mdep_if.c(※1)	⇒   grp_file
src / mdep_itron / include / grp_fs_mdep_types.h grp_mdep_sem.h	⇒  grp_include / grp_mdep
src / mdep_itron / lib / grp_itron_id.h	⇒ grp_mdep
src / mdep_itron / lib / grp_mem / grp_mem.c	⇒ grp_mdep
src / mdep_itron / lib / grp_sem / grp_sem.c	⇒ grp_mdep
src / mdep_itron / lib / grp_time / grp_time_get.c grp_time_set.c	⇒ grp_mdep

※1 grp\_fs\_mdep\_if.c は、旧バージョンよりファイル名が変わっています。旧バージョンでは「grp\_fs\_mdep\_if\_itron.c」の名称でした。



### 3.3.2 VOS依存ファイルのコピー

表 3.3-2 旧構成のコピー先（環境依存ファイル VOS）にVOSの環境依存ファイルのコピー先を示します。

表 3.3-2 旧構成のコピー先（環境依存ファイル VOS）

src / mdep_vos / base / grp_fs_dev_io.c grp_fs_dev_sw_tbl.c grp_fs_mdep_if.c(※1)	⇒	grp_file
src / mdep_vos / include / grp_fs_mdep_types.h grp_mdep_sem.h grp_vos.h	⇒	grp_include / grp_mdep grp_include
src / mdep_vos / lib / grp_mem / grp_mem.c	⇒	grp_mdep
src / mdep_vos / lib / grp_sem / grp_sem.c	⇒	grp_mdep
src / mdep_vos / lib / grp_time / grp_time_get.c grp_time_set.c	⇒	grp_mdep

※1 grp\_fs\_mdep\_if.c は、旧バージョンよりファイル名が変わっています。旧バージョンでは「grp\_fs\_mdep\_if\_vos.c」の名称でした。

### 3.4 サンプルファイルのコピー

**GR-FILE** のご提供しているサンプルファイルを使用する場合も、ファイルのコピーが必要です。

また、RAM ディスク機能を使用される場合は、ファイルのコピーと「grp\_fs\_sysdef.h」のコンパイルスイッチの設定が必要ですのでご注意ください。

Version1.20 以前のファイルを使用される場合は、各ソースファイルで「grp\_fs\_sysdef.h」をインクルードする必要があります。

表 3.4-1 サンプルファイルのコピー先

sample / base / grp_fat_format_sd.c(※1) grp_fat_format_sd.h(※1) grp_fs_dev_io_ram.c grp_fs_dev_io_ram.h grp_fs_proc_event.c grp_fs_proc_event.h	⇒	grp_file
---	---	----------

※1 grp\_fat\_format\_sd.c、grp\_fat\_format\_sd.h はご要望があった場合のみご提供されます。

### 3.5 設定ファイルの確認

ファイルのコピーが出来ましたら、**GR-FILE** の設定値を確認します。

表 3.5-1 設定ファイルに、**GR-FILE**の設定に関するファイルを記します。

表 3.5-1 設定ファイル

ファイル名	内容
grp_fat_param.h	FAT ファイルシステムのパラメータ
grp_fat_format.h	FORMAT 機能を使用する場合のパラメータ
grp_fs_disk_part.h	パーティションに関するパラメータ
grp_fs_param.h	ファイルシステム全般のパラメータ
grp_fs_sysdef.h	コンパイルスイッチの設定
grp_stdio.h	C 言語 I/F を使用する際のパラメータ

### 3.6 ポーティングファイルの確認

最低限必要なポーティングファイルの確認を行います。

**GR-FILE** Version1.20 のご提供時は、ポーティング対象である「mdep\_itron」、「mdep\_vos」は両方ともスタブファイルを提供しています。

ポーティングは、各処理をスタブファイルに実装することで行います。

以下に、最低限必要なポーティングファイルを示します。

※ポーティングの必要なファイルには、その旨コメントが記述されています。

コメントは C 言語のコメント形式にはなっていないので、そのままではコンパイルエラーが発生します。

ポーティング部分を確認しましたら、コメントを削除するか、コメントアウトしてください。

### 3.6.1 μITRON使用時 (mdep\_itron)

μITRON依存ファイルを使用した場合の、ポータリング確認ファイルを、表 3.6-1 μITRON使用時のポータリング確認ファイルに示します。

表 3.6-1 μITRON 使用時のポータリング確認ファイル

Version 1.20 での ディレクトリ / ファイル	旧構成での ディレクトリ	内容	ポータリング の有無
src / mdep_itron / base / grp_fs_dev_io.c	src / grp_file	デバイスドライバ I/O インターフェース関数	○
grp_fs_dev_sw_tbl.c		デバイススイッチテーブル	△
grp_fs_mdep_if.c		各種依存処理関数	○
src / mdep_itron / lib grp_itron_id.h	src / grp_mdep	μITRON のセマフォ、メモリプール ID	○
src / mdep_itron / lib / grp_time / grp_time_get.c	src / grp_mdep	現在時刻取得処理関数	○
grp_time_set.c		現在時刻設定処理関数	○

表中の「ポータリングの有無」の欄で、「○」となっているファイルは、空のスタブ関数が定義されています。

関数 I/F に合わせてポータリングを行って下さい。

「grp\_fs\_dev\_sw\_tbl.c」には、デバイスドライバのスイッチテーブルが定義されています。

必要に応じて登録を追加/削除します。

「grp\_fs\_mdep\_if.c」には、スタブ関数が既に実装されており、**GR-FILE** より呼び出されます。

必要に応じてポータリングを行って下さい。

### 3.6.2 VOS使用時 (mdep\_vos)

VOS依存ファイルを使用した場合の、ポータリング確認ファイルを、表 3.6-2 VOS使用時のポータリング確認ファイルに示します。

表 3.6-2 VOS 使用時のポータリング確認ファイル

Version 1.20 での ディレクトリ / ファイル	旧構成での ディレクトリ	内容	ポータリング の有無
src / mdep_itron / base / grp_fs_dev_io.c	src / grp_file	デバイスドライバ I/O インターフェース関数	○
grp_fs_dev_sw_tbl.c		デバイススイッチテーブル	△
grp_fs_mdep_if.c		各種依存処理関数	○
src / mdep_itron / lib / grp_time / grp_time_get.c	src / grp_mdep	現在時刻取得処理関数	○
grp_time_set.c		現在時刻設定処理関数	○

表中の「ポータリングの有無」の欄で、「○」となっているファイルは、空のスタブ関数が定義されています。

関数 I/F に合わせてポータリングを行って下さい。

「grp\_fs\_dev\_sw\_tbl.c」には、デバイスドライバのスイッチテーブルが定義されています。

必要に応じて登録を追加/削除します。

「grp\_fs\_mdep\_if.c」には、スタブ関数が既に実装されており、**GR-FILE** より呼び出されます。

必要に応じてポータリングを行って下さい。



### 3.6.3 文字列出力関数について(grp\_fs\_printf関数)

**GR-FILE** には、誤動作や致命的なエラーの際に、エラー情報を出力したり、文字列を出力する機能があります。

この出力時に、`grp_fs_printf` 関数を使用しておりますが、出力部分はターゲットに依存する為、出力を行う場合は、この部分のポーティングが必要となります。

`grp_fs_printf` 関数は、`grp_fs_mdep_if.c` に実装されており、出力関数として `cons_putchar` 関数を呼ぶように実装されています。

ですので、`cons_putchar` 関数はアプリケーションでご用意ください。

また、文字列出力を行わない場合は、`grp_fs_printf` 関数の `cons_putchar` 関数をコメントアウトして下さい。

### 3.7 コンパイルオプションの削除

Version1.20 では、コンパイルオプションをヘッダファイル「`grp_fs_sysdef.h`」に定義しており、このヘッダファイルを読み込むことで、同様の効果を得ています。

この事により、`Makefile` や統合開発環境で設定していた、**GR-FILE** に関する定義は不要となりますので削除して下さい。

### 3.8 カスタムソースファイルについて

旧バージョンの **GR-FILE** のソースファイルに、独自の変更（カスタム）を行っている場合で、旧バージョンのカスタムソースファイルをそのまま使用する場合、コンパイルスイッチの変更に対応できません。

カスタムソースファイルの先頭で、「`grp_fs_sysdef.h`」をインクルードするように追加を行ってください。

### 3.9 パスの設定

Version1.20 より、相対パスによるインクルードの記述が削除されました。

これに伴い、旧構成へファイルをコピーした場合は、インクルードパスの設定が追加となります。

`itron`、`vos` では追加するインクルードパスが異なりますのでご注意ください。

環境	追加インクルードパス名
itron	<code>grp_include / grp_mdep /</code>
vos	<code>grp_include / grp_mdep / grp_mem /</code>

### 3.10 メイク環境へのファイルの追加

Version1.20 より、下記ファイルが追加/名称変更されています。

ご使用のメイク環境への追加、変更を行ってください。

ファイル名	内容
<code>grp_fs_dev_sw_tbl.c</code>	新規ファイル
<code>grp_fs_mdep_if.c</code>	ファイル名変更

## 組込み向けファイルシステム GR-FILE 旧構成への移行手順

発行年月：2008 年 7 月 第 1.00 版

発行：株式会社グレープシステム

E-Mail : [gr@support.grape.co.jp](mailto:gr@support.grape.co.jp)

URL : <http://www.grape.co.jp>

Copyright (C) 2008 Grape Systems, Inc.

All rights reserved.