

ディスク割り当て評価ツールv1.2.1

使用説明書

日立製作所

目次

1. はじめに	- 3 -
1.1. 目的	- 3 -
1.2. プロジェクト概要とスコープ	- 3 -
2. DAVL概要	- 4 -
2.1. DAVLとは	- 4 -
2.2. 前提条件	- 4 -
3. DAVL動作環境構築手順	- 4 -
3.1. ディレクトリ構成	- 4 -
3.2. インストール手順	- 5 -
3.2.1. 前準備	- 5 -
3.2.2. インストール	- 5 -
4. DAVL使用手順	- 7 -
4.1. 提供コマンド概要	- 7 -
4.2. cdavl使用方法	- 9 -
4.3. gdavl使用方法	- 14 -
4.4. davl_liveinfoモジュールについて	- 26 -

1. はじめに

1.1. 目的

サーバ分野において、Linux を適用したシステムが普及・拡大している。数年前までは、Web サーバやメールサーバ、ネームサーバといったネットワークのフロントエンドサーバへの適用が中心であったが、最近では、アプリケーションサーバ、DB サーバから構成されるエンタープライズシステムへの適用ニーズも出てきている。サーバ分野において、Linux を適用したシステムが普及・拡大している。数年前までは Web サーバやメールサーバ、ネームサーバといったネットワークのフロントエンドサーバへの適用が中心であったが、最近ではアプリケーションサーバ、DB サーバから構成されるエンタープライズシステムへの適用ニーズも出てきている。

エンタープライズシステムでは迅速な障害対応が求められるが、Linux にはダンプやトレースといった障害解析のための標準的なツールが無く、障害発生時は、各社固有のノウハウで対応しているのが現状である。障害の中でも特に Linux ファイルシステムのディスク割り当てに関しては、フラグメンテーションが起こりにくく障害の原因にはなりにくいといわれており、評価するためのツールもほとんど提供されていない。

しかし最近では、データのアクセス性能が落ちる障害が多々見受けられるようになってきている。この原因のひとつとしてディスクのフラグメンテーションが考えられるが、評価ツールがないために問題を特定することができない。また、ディスク全体の割り当て状況の評価するツールを開発した場合でも、特定ファイルのアクセス性能が落ちる障害の場合には問題を特定できない。

ディスク割り当て評価ツール（以降 DAVL : Disk Allocation Viewer for Linux と称する）ではこれらの問題を解決するため、ディスク割り当て状況を容易に取得 / 可視化でき、更にファイルを対象としたディスク割り当て状況の可視化も可能とすることを目的として開発を行った。

1.2. プロジェクト概要とスコープ

DAVL の適用範囲を表 1-1 に示す。

表 1-1 DAVL 適用範囲

項番	項目	マクロ名	備考
1	ハードウェア条件	PC: IA-32 をベースとした PC CPU: Pentium メモリ: 64MB 以上	
2	Linux カーネル	Upstream Kernel v2.6.9 Miracle Linux 3.0 (Asianux 1.0): v2.4.21	
3	ライブラリ	GTK+1.2、GTK+2	
4	ファイルシステム	ext2、ext3	

2. DAVL 概要

2.1. DAVL とは

DAVL は Linux の ext2 / ext3 ファイルシステムのフラグメンテーション状況を視覚化して表示するツールであり、下記の機能を持つ。

- ファイルシステムのマウント状態によらず、フラグメンテーション状況を取得できる。
- フラグメンテーション取得対象として、パーティション全体 / 任意のディレクトリ下の全ファイル / 単一ファイルのいずれかを指定できる。

DAVLはフラグメンテーション情報を取得するcdavl(CUI Disk Allocation Viewer for Linux)と、その情報をGUI表示するラッパーのgdavl(GUI Disk Allocation Viewer for Linux)の2つのプログラムおよび、マウント状態のパーティションのフラグメンテーション情報を取得するdavl_liveinfoモジュールによって構成される。フラグメンテーション状況を取得するcdavlは単体でも動作し、フラグメンテーション状況をテキスト形式で出力させることができる。

DAVLはカーネルをUpstreamカーネル2.6.9に変更したFedora Core 2とFedora Core 3および、Miracle Linux V3.0上で動作を確認している。

2.2. 前提条件

DAVLのコンパイルおよびインストールに必要な条件を下記に示す。

- カーネルバージョンが2.4以上であること。
- ext2fs/ext2_fs.hが標準インクルードファイルとして存在していること。
- GTK+1.2またはGTK+2がインストールされていること。
- カーネルソースが展開されていること(davl_liveinfoを利用する場合のみ必要)。

3. DAVL 動作環境構築手順

3.1. ディレクトリ構成

アーカイブを下記の手順で展開すると、davlのディレクトリツリーが展開される。(以降、davlディレクトリ)

```
$ tar jxvf davl-XXX.tar.bz2
```

XXX は、davl バージョン

davl ディレクトリ下の主な内容を下記に示す。

davl-XXX/	
+---- README	概要、インストール方法
+---- ChangeLog	変更点履歴ファイル
+---- COPYING	GPL ライセンス
+---- Makefile	メイクファイル
+---- path_list	パス設定ファイル
+---- doc/	ドキュメント
+---- rpm/	rpm パッケージ作成支援
+---- src/	ソースファイル
+---- common/	共通ソースファイル
+---- cdavl/	cdavl ソースファイル
+---- gdavl/	gdavl ソースファイル
+---- drv/	dav_liveinfo モジュールソースファイル

3.2. インストール手順

3.2.1. 前準備

DAVL では、カーネルソースおよびインストール先のディレクトリ、使用する GTK ライブラリのバージョンは、dav1 ディレクトリ直下の path_list 中で決まっており、変更する場合は直接 path_list を書き換える必要がある。各種ディレクトリの定義マクロ名と、デフォルトのディレクトリ等を表 3-1 に示す。

表 3-1 各指定マクロ一覧

項番	項目	マクロ名	デフォルトディレクトリ
1	カーネルソース	KERN_DIR	/lib/modules/XXXX/build
2	実行ファイルインストール先	BIN_DIR	/usr/local/bin
3	davl_liveinfo モジュールインストール先	DRV_DIR	/lib/modules/XXXX/kernel/drivers/davl
4	マニュアルインストール先	MAN_DIR	/usr/local/man
5	GTK ライブラリのバージョン指定	GTK_VER	GTK2

XXXX : uname -r の出力文字列

3.2.2. インストール

インストールは、下記の手順で行う。手順(1)～(7)までは、Fedora Core 2 の場合のカーネルソースを準備する手順であり、dav1_liveinfo モジュールを利用しない場合は、不要である。

- (1). スーパーユーザに移行

```
$ su
```

(スーパーユーザのパスワードを入力)

- (2). カーネルソースパッケージをダウンロードし、インストール

```
# rpm -i kernel-2.6.5-1.358.src.rpm
```

- (3). インストールしたカーネルソースパッケージを展開
rpmbuild -bp --target i686 /usr/src/redhat/SPECS/kernel-2.6.spec
- (4). カーネルソースディレクトリに移動
cd /usr/src/redhat/BUILD/kernel-2.6.5/linux-2.6.5
- (5). Makefile の EXTRAVERSION 行を編集し、uname -r コマンドを実行した結果が、2.6.5-1.358 の場合、「EXTRAVERSION = -1.358」となるようにする
vi Makefile
- (6). カーネルソースをモジュールコンパイル用に準備
make prepare-all
- (7). 一般ユーザに戻る
exit
- (8). davl ディレクトリに移動
\$ cd davl-XXX
- (9). ビルド (davl_liveinfo モジュールを生成する場合は、WITH_DRV=1 を指定する)
\$ make (WITH_DRV=1)
- (10). インストール (スーパーユーザで実行)
make (WITH_DRV=1) install

4. DAVL 使用手順

4.1. 提供コマンド概要

DAVL が提供するコマンドの一覧を下記に示す。

- cdavl フラグメンテーション状況取得とその出力（テキスト形式）を行う
- gdavl 内部で cdavl を起動し、その出力を GUI 表示する

cdavl / gdavl を実行する上での留意点・制限事項を下記に示す。

- スーパーユーザ権限で実行する必要がある。
- コマンド実行中に、フラグメンテーション状況取得対象パーティションのマウント状態を変更しないこと。
- アンマウントパーティションのフラグメンテーション状況を取得する場合、クリーンアンマウント状態であること。

DAVL では、ファイルブロックをファイル構成順に見て行き、下記の状態の場合に、フラグメントブロックと判断する。

- (1). ファイルブロックが、システムブロックを挟んでいる。
- (2). ファイルブロックが、連続しておらず、間に他のブロックが挟まっている。
- (3). ファイルブロックの順番が逆転している。

フラグメントブロックの例（ファイルブロック数=6）を、図 4-1 に示す。

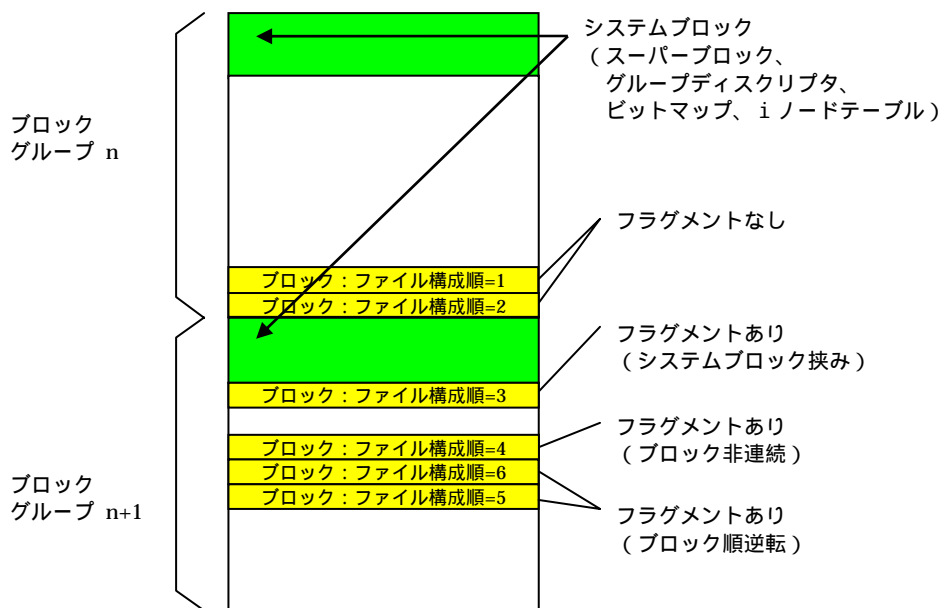


図 4-1 フラグメントブロック例

(1)のシステムブロックによるフラグメントは不可避であるため、他のフラグメントとは区別し、別に計数している。

DAVLでのフラグメンテーション状況のテキスト形式出力/ GUI表示では、以下の2つのブロック表示方法がある。

- (1). 物理的なブロック番号順でシステムブロックを含めて表示する表示。(ブロック番号順表示)
- (2). 確認対象のファイルを構成するブロックだけを、その構成順に表示。(ファイル構成順表示)

図 4-1 の例を2つの表示方法でそれぞれ表示した例を図 4-2 に示す。

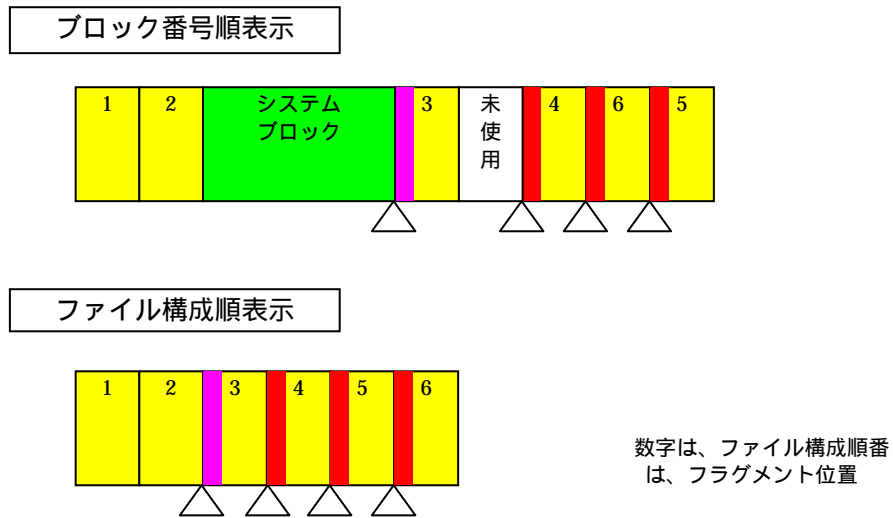


図 4-2 ブロック番号順/ファイル構成順での表示例

どの場合にどちらの方法で表示するかの概要は、表 4-1 を参照。また、どちらの表示の場合でも、フラグメントの判定方法は変わらない。

表 4-1 確認対象とブロック表示方法

項番	確認対象	付加情報	ブロック表示	備考
1	パーティション全体	-	ブロック番号順	
2	ディレクトリ	-	ブロック番号順	
3	ファイル	-	ブロック番号順 / ファイル構成順	選択可能
4	上記すべて	フラグメントファイル情報	ファイル構成順	

4.2. cdavl 使用方法

cdavl は、パーティション全体 / 任意のディレクトリ下の全ファイル / 単一ファイルのいずれかのフラグメンテーション状態を取得し、標準出力にテキスト形式で出力する。また、パーティションの一覧をマウント状態と合わせて表示することができる。

cdavl コマンドの説明を表 4-2 に示す。

表 4-2 cdavl コマンド説明

コマンド	cdavl	
書式	cdavl [-cChnstTv[v[v]]] [devname [path]]	
引き数	devname	デバイス名。デバイス名が指定されない場合は、システム内のパーティション一覧を /proc/partitions から読み出し、マウント状態と合わせて表示する。この時 /proc/partitions に含まれないデバイスに関しては表示しない
	path	パーティション devname 内のディレクトリまたはファイル名。先頭に / (スラッシュ) が必要。path が指定されない場合は、パーティション全体のフラグメント状況を表示する
オプション	c	出力を、エスケープシーケンスを用いたカラー表示にする
	C	ファイルシステム中のブロック / inode ビットマップと cdavl 解析結果を比較チェックする。(デバッグ用)
	h	コマンドのヘルプを表示する
	n	ディレクトリをチェックする場合、そのディレクトリだけのフラグメント状況を表示し、下位のディレクトリおよびファイルのフラグメント状況は含めない。このオプションを指定しない場合は、下位のディレクトリおよびファイルのフラグメント状況を含めて表示する
	s	マウントパーティション中のフラグメント状況を確認する際に、davl_liveinfo モジュールを使用しない。davl_liveinfo については「4.4 davl_liveinfo モジュールについて」参照
	t	ブロック情報出力フォーマットを、ブロック種別 + 同一種別連続数のカンマ区切り文字列にする。カラム数 80 で行を折り返す。ブロック種別については後述の cdavl 出力例を参照
	T	ブロック情報出力を折り返さずに 1 行で全て表示する以外は、t オプションと同じ
	v	詳細情報表示レベル。v、vv、vvv と 3 つまで指定でき、v の数が多いほど詳細な情報を表示する。vv、vvv はデバッグ用

cdavl をオプション -Tv[Cns] で実行した出力をファイルに保存しておけば、いつでもそれを gdavl に読み込ませて GUI 表示することが可能である。

cdavl 実行手順を下記に示す。

- (1). スーパーユーザでログイン
- (2). davl_liveinfo モジュールをロード (一度ロードすれば、次回の cdavl / gdavl 実行時は不要)

```
# /sbin/modprobe davl_liveinfo
```

- (3). cdavl 実行

```
# cdavl (必要に応じてオプションおよび引数を指定)
```

cdavl の出力例とその見方を図 4-3 ~ 図 4-6 に示す。

----- partitions information -----		
name	mount	#blocks
/dev/hda	unmount	58593840
/dev/hda1	mount	12289693
/dev/hda2	unmount	1020127
/dev/hda3	unmount	1
/dev/hda5	unmount	11727418
/dev/hda6	unmount	3911796
/dev/hda7	unmount	2353491
/dev/hda8	unmount	240943
/dev/hda9	unmount	56196
/dev/hda10	unmount	2353491
/dev/hda11	unmount	240943
/dev/hda12	unmount	56196
/dev/hda13	unmount	8001

execute "cdavl -h" for help

} パーティション一覧ヘッダ

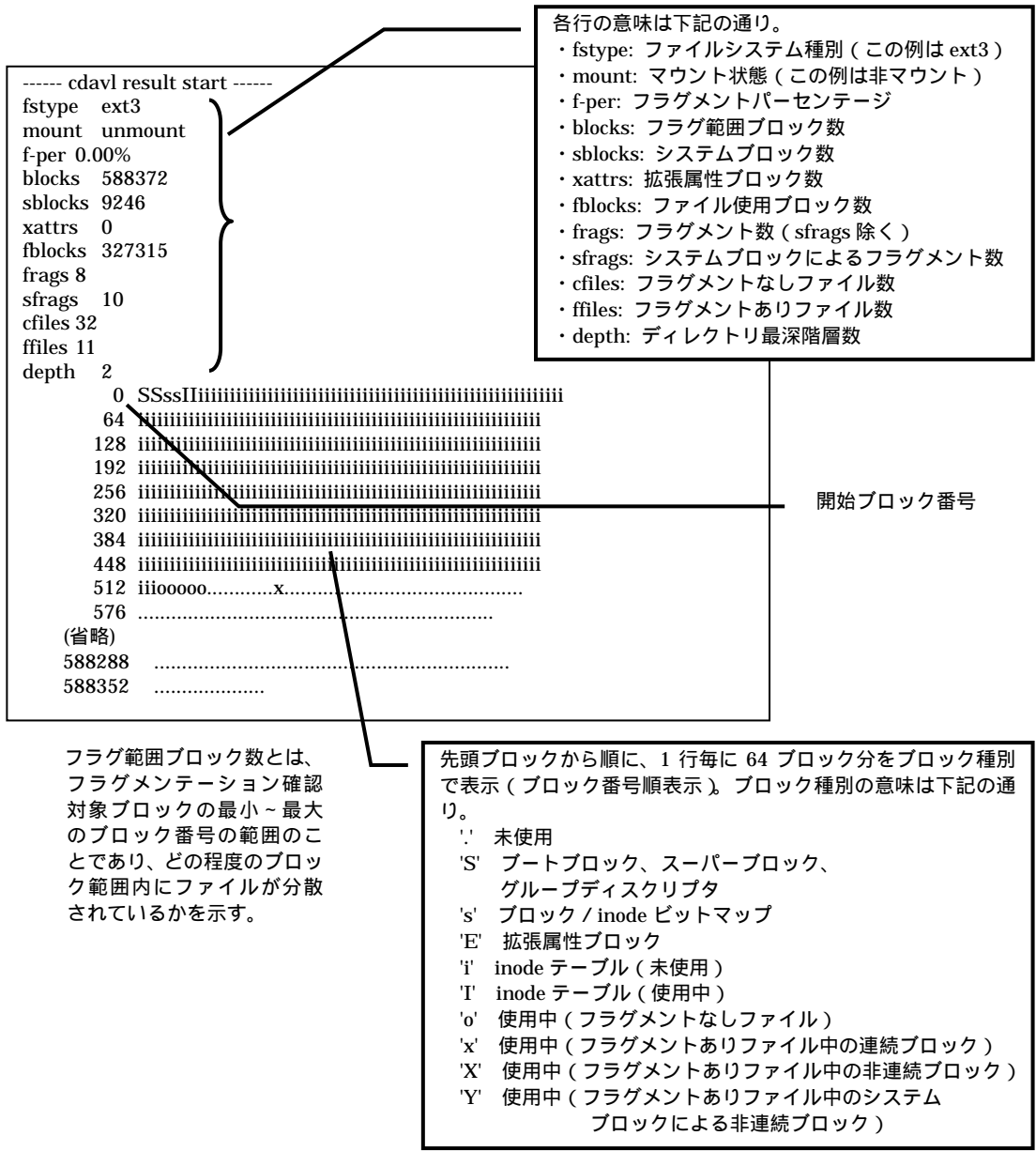
デバイス名

マウント状態

ブロック数

cdavl (オプションなし) の出力例

図 4-3 cdavl 出力例 (1 / 4) - パーティション一覧の出力



cdavl /dev/hda10 の出力例

図 4-4 cdavl 出力例 (2 / 4) - 対象がパーティション全体の場合 - その 1

gdavl 読み込み可能形式タグ

```

data for gdavl (cdavl ver: 1.2.1)
----- check start -----
"/6"(ino: 12) is fragmented (0.00%, total: 11909/12422, frag: 0, sfrag: 1), xattr: 0
block sequence: x4952,Y1,x6956
"/9"(ino: 15) is fragmented (0.01%, total: 8652/9165, frag: 0, sfrag: 1), xattr: 0
block sequence: x8030,Y1,x621
"/14"(ino: 18) is fragmented (0.01%, total: 7781/8296, frag: 0, sfrag: 1), xattr: 0
block sequence: x1730,Y1,x6050
"/17"(ino: 21) is fragmented (0.00%, total: 12437/12950, frag: 0, sfrag: 1), xattr: 0
block sequence: x10530,Y1,x1906
"/18"(ino: 22) is fragmented (0.01%, total: 9314/9829, frag: 0, sfrag: 1), xattr: 0
block sequence: x5205,Y1,x4108
"/20"(ino: 23) is fragmented (0.00%, total: 13359/13874, frag: 0, sfrag: 1), xattr: 0
block sequence: x13059,Y1,x299
"/22"(ino: 25) is fragmented (0.00%, total: 10400/10915, frag: 0, sfrag: 1), xattr: 0
block sequence: x6483,Y1,x3916
"/23"(ino: 26) is fragmented (0.11%, total: 6899/15080, frag: 8, sfrag: 0), xattr: 0
block sequence: x1,X1,x1,X7,x6889
"/29"(ino: 32) is fragmented (0.01%, total: 8701/9216, frag: 0, sfrag: 1), xattr: 0
block sequence: x7734,Y1,x966
"/19"(ino: 36) is fragmented (0.01%, total: 8977/9490, frag: 0, sfrag: 1), xattr: 0
block sequence: x4537,Y1,x4439
"/2"(ino: 37) is fragmented (0.01%, total: 8161/8674, frag: 0, sfrag: 1), xattr: 0
block sequence: x4704,Y1,x3456

----- cdavl result start -----
fstype  ext3
mount    unmount
f-per    0.00%
blocks   588372
sblocks  9246
xattrs   0
fblocks  327315
frags    8
sfrags   10
cfiles   32
ffiles   11
depth    2
0  S2,s2,I2,i509,o5,.12,x1,.1024,X1,x1,.1024,X1,.1024,X1,.1024,X1,.
1024,X1,.1024,X1,.1024,X1,.1013,X1,x6889,o11951,x5205,S2,s2,i511,Y1,x4108,o
23440,x4704,s2,i511,Y1,x3456,o22315,x6483,S2,s2,i511,Y1,x3916,o20306,x8030
,s2,i511,Y1,x621,o29903,x1730,S2,s2,i511,Y1,x6050,o21250,x4952,s2,i511,Y1,x6
956,o12239,x13059,S2,s2,i511,Y1,x299,o21423,x10530,s2,i511,Y1,x1906,o22614,
x7734,S2,s2,i511,Y1,x966,o26749,x4537,s2,i511,Y1,x4439,o8530,.19285,s2,i511,.
32255,s2,i511,.32255,s2,i511,.32255,s2,i511,.32255,s2,i511,.32255,s2,i511,.3225
5,s2,i511,.30803

```

フラグメントファイル情報 (2行で1ファイル分。vレベルが1以上で出力)

パス名

inode 番号

フラグメントパーセンテージ

ファイル使用ブロック数

フラグ範囲ブロック数

フラグメント数 (次の sfrags は除く)

システムブロックによるフラグメント数

拡張属性ブロック数

図 4-4 と同じ

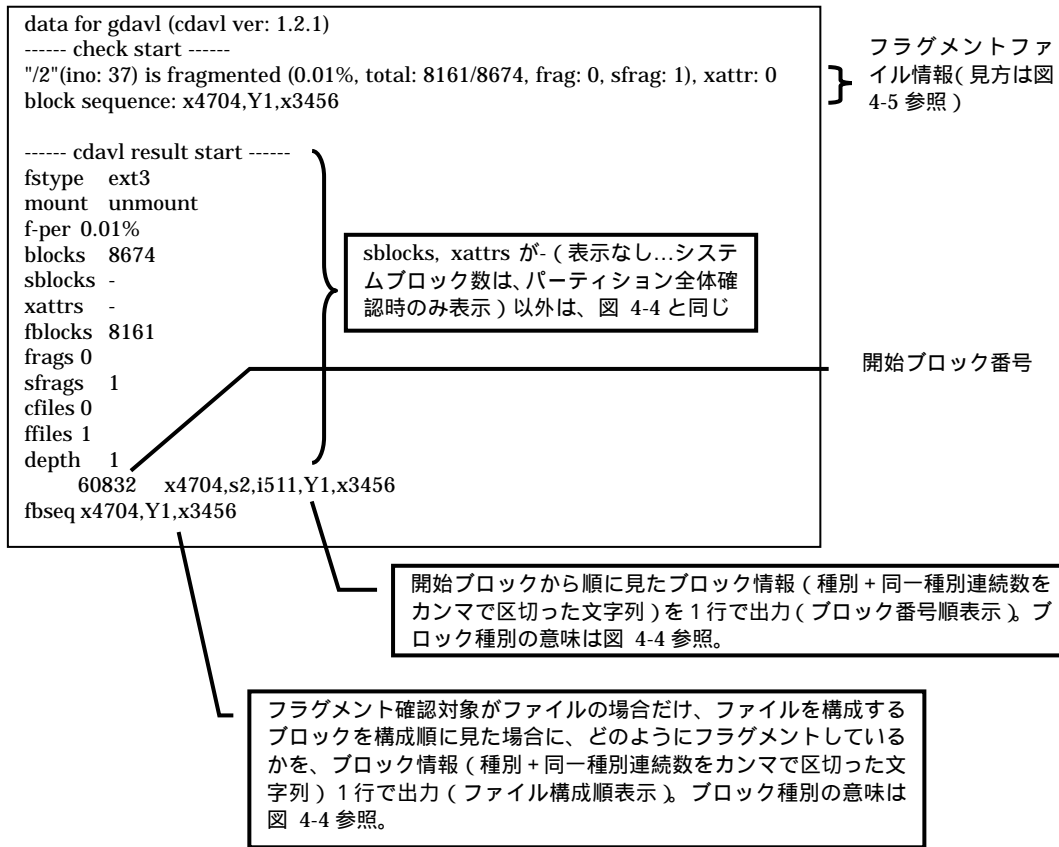
開始ブロック番号

ファイルを構成するブロックだけを構成順に見た場合に、どのようにフラグメントしているかを、ブロック情報 (種別 + 同一種別連続数をカンマで区切った文字列) 1行で出力 (ファイル構成順表示)、ブロック種別の意味は図 4-4 参照。

先頭ブロックから順に見たブロック情報 (種別 + 同一種別連続数をカンマで区切った文字列) を 1行で出力 (ブロック番号順表示)、ブロック種別の意味は図 4-4 参照。

cdavl -Tv /dev/hda10 の出力例

図 4-5 cdavl 出力例 (3 / 4) - 対象がパーティション全体の場合 - その 2



cdavl -Tv /dev/hda10 /2 の出力例

図 4-6 cdavl 出力例(4 / 4) - 対象がファイルの場合

4.3. gdavl 使用方法

gdavl は、パーティション全体 / 任意のディレクトリ下の全ファイル / 単一ファイルのいずれかのフラグメンテーション状態を取得し、GUI 表示する。gdavl 読み込み可能形式で cdavl 出力を行い、それをファイルに保存（以降、フラグメンテーション状態の「スナップショット」と呼ぶ）しておけば、いつでも cdavl 実行時のフラグメンテーション状況を確認することができる。また、dav1 バージョン 1.2.1 からは、複数のスナップショットを比較した結果を差分情報として GUI 表示することができる。

gdavl コマンドの説明を表 4-3 に示す。

表 4-3 gdavl コマンド説明

コマンド	gdavl	
書式	gdavl [-bhs] [-S n] [devname [path]] ----- 通常表示用 gdavl [-b] [-S n] [-d file-before file-after] ----- 差分表示用 gdavl [-S n] [-a interval file-1 file-2 [file-3 ...]] ----- アニメーション表示用	
引き数	devname	デバイス名または、cdavl 出力ファイル名。cdavl 出力ファイルの場合は、gdavl 読み込み可能形式で出力されている必要がある。
	path	パーティション devname 内のディレクトリまたはファイル名。先頭に / (スラッシュ) が必要。path が指定されない場合は、パーティション全体のフラグメント状況を表示する。cdavl 出力ファイルを読み込ませる場合は、path 指定は無視される。
オプション	b	バッチ指定。コマンド実行すると自動的に実行ボタンを押した状態になる
	h	コマンドのヘルプを表示する
	s	マウントパーティション中のフラグメント状況を確認する際に、dav1_liveinfo モジュールを使用しない。dav1_liveinfo については「4.4 dav1_liveinfo モジュールについて」参照
	S	サマライズ指定。2^n ブロック分のブロックを 1 ブロックとして集約表示する
	d	差分表示指定。続けて 2 つのスナップショットファイルをスペースで区切って指定する (file-before と file-after)
	a	アニメーション表示指定。続けて表示更新周期 (秒数) スナップショットファイル群をスペースで区切って指定する

gdavl 実行手順を下記に示す。

- (1). スーパーユーザでログイン
- (2). dav1_liveinfo モジュールをロード（一度ロードすれば、次回の cdavl / gdavl 実行時は不要）


```
# /sbin/modprobe dav1_liveinfo
```
- (3). gdavl 実行


```
# gdavl (必要に応じてオプションおよび引数を指定)
```

ヘルプオプション (-h) 以外であれば、ここから GUI 画面が表示される。

gdavl の GUI 画面の各パーツの概要を図 4-7 に示す。

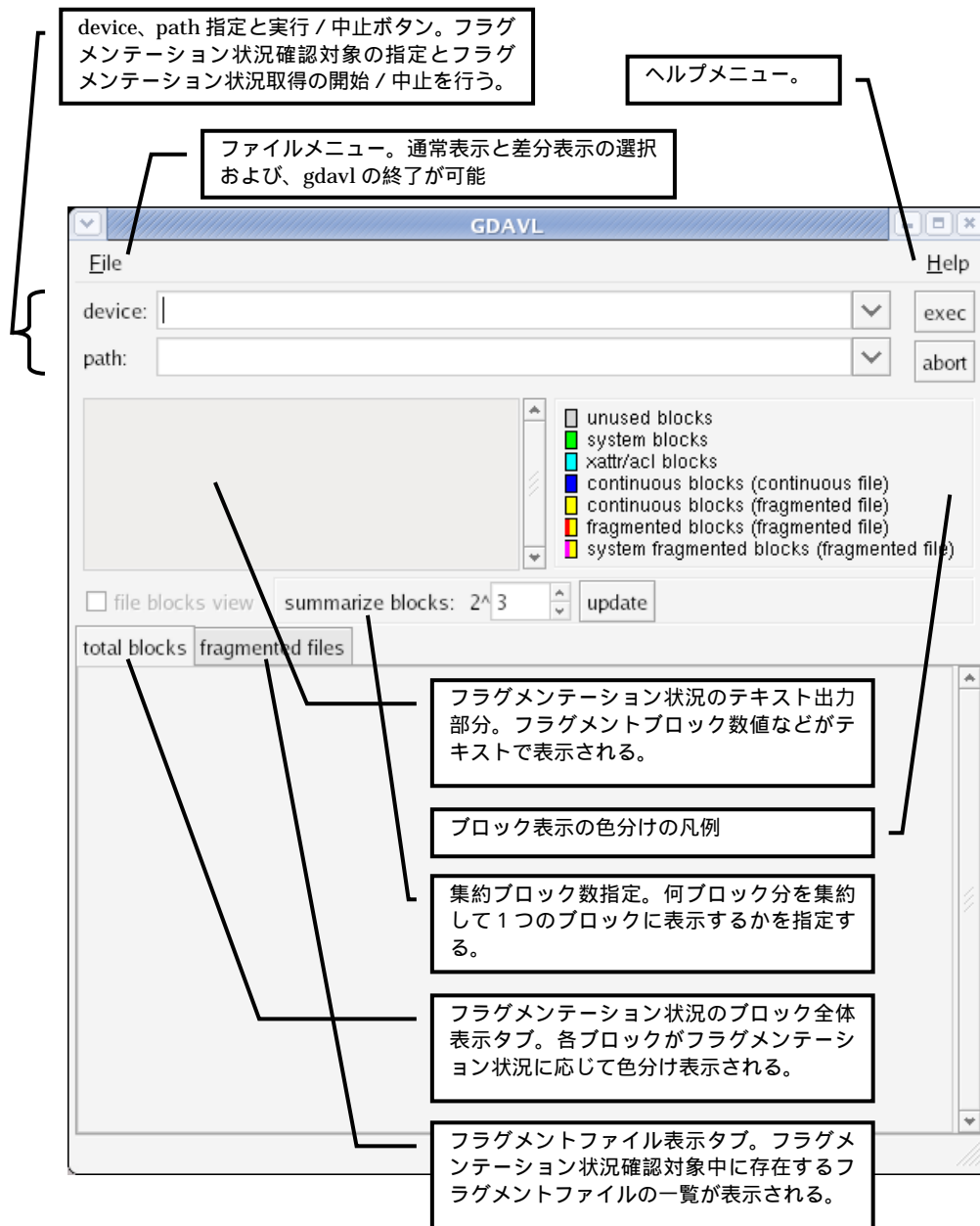


図 4-7 gdavl GUI 各パーツの概要

gdavl の GUI 画面での操作手順を図 4-8 ~ 図 4-14 に示す。

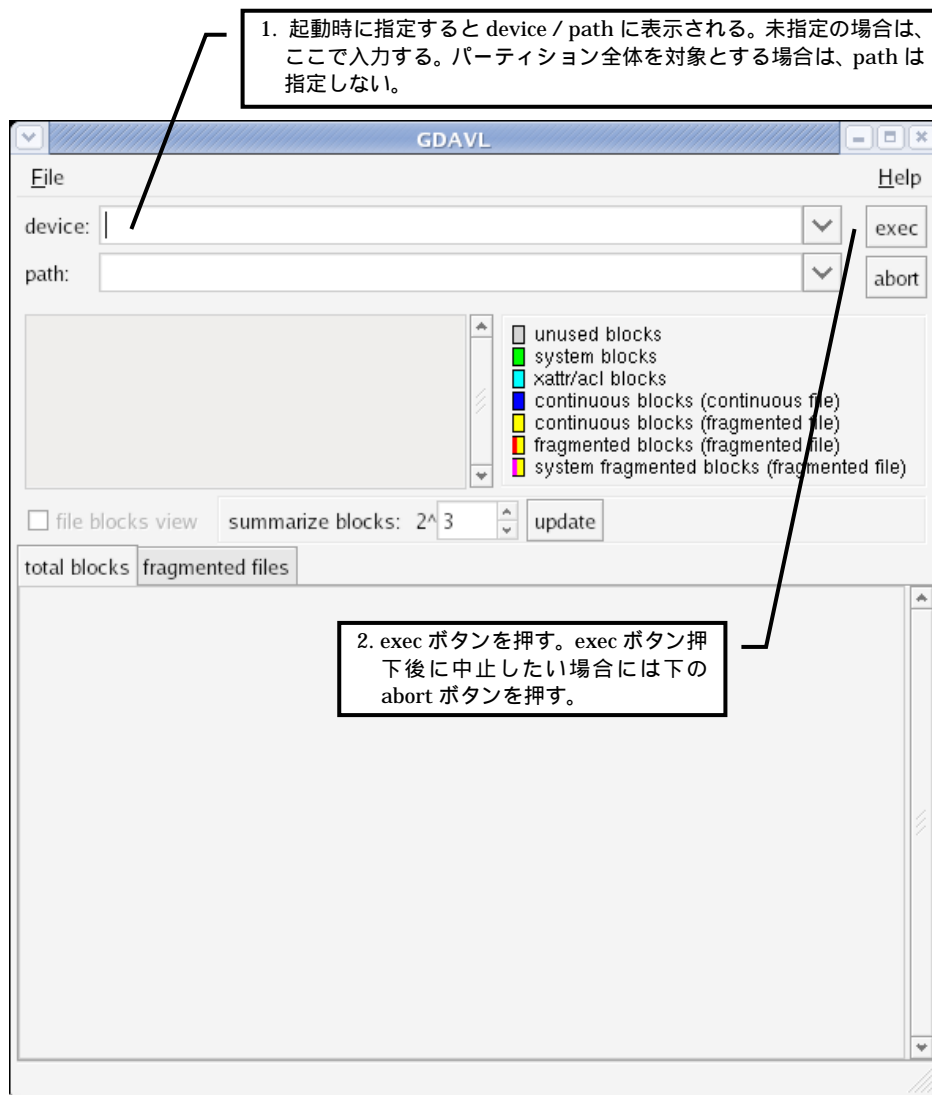
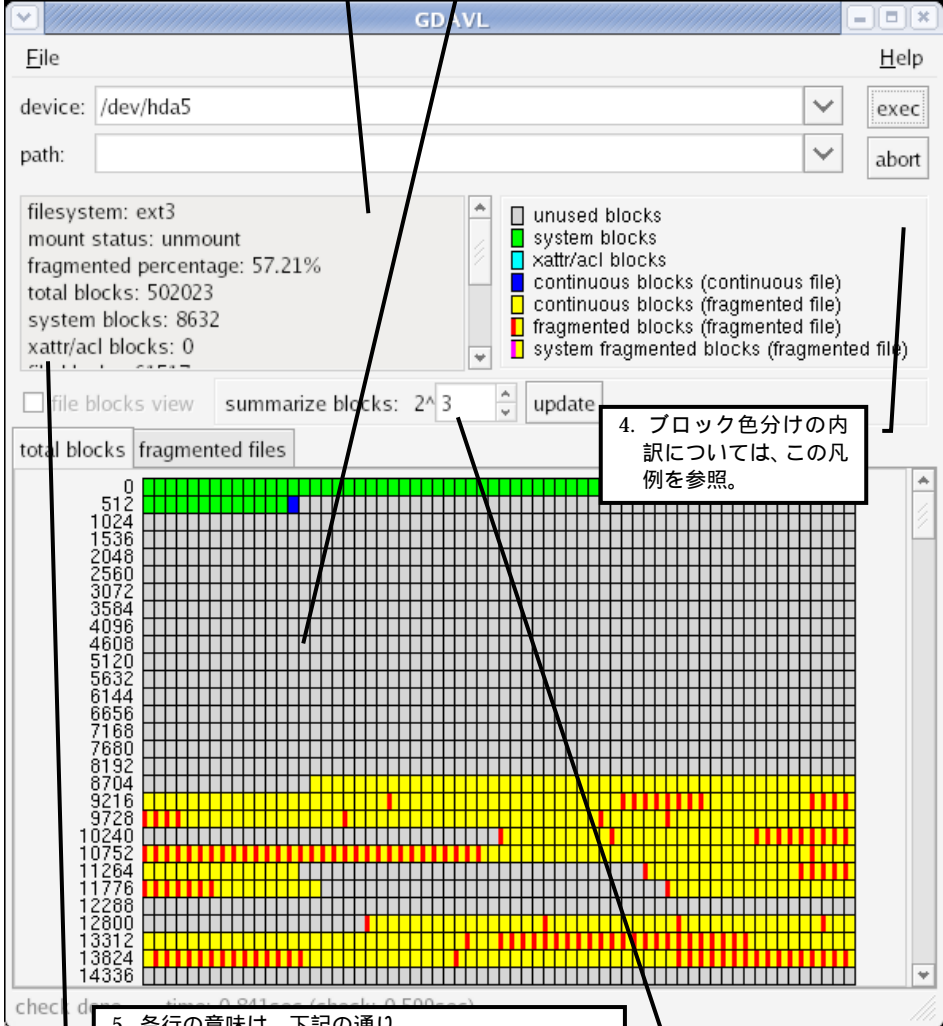


図 4-8 gdavl 操作手順 (1 / 1 0)

3. フラグメント状況取得が終了すると、テキストおよび... (3'の説明に続く)

3'. 物理的なブロック番号順に、色分けしたブロックでフラグメント状況が表示される(ブロック番号順表示)。左側の数字は、ブロック番号を示す。



4. ブロック色分けの内訳については、この凡例を参照。

5. 各行の意味は、下記の通り。
- filesystem: ファイルシステム種別
 - mount status: マウント状態
 - fragmented percentage: フラグメントパーセンテージ
 - total blocks: フラグ範囲ブロック数
 - system blocks: システムブロック数
 - xattr/acl blocks: 拡張属性ブロック数
 - file blocks: ファイル使用ブロック数
 - fragmented blocks: フラグメント数 (sys-fragmented blocks 除く)
 - sys-fragmented blocks: システムブロックによるフラグメント数
 - continuous files: フラグメントなしファイル数
 - fragmented files: フラグメントありファイル数
 - max dir depth: ディレクトリ最深階層数

6. 初期状態では $2^3(=8)$ ブロック分が1ブロックに集約されて表示される。集約ブロック数を変えるには、ここに直接数値を入力するか、すぐ右隣の / ボタンを押して値を変更してから、右側の update ボタンを押す。ブロック集約の優先度は、凡例の上から下になるほど高い。(例えば、青色と黄色を集約すると黄色になる)

図 4-9 gdavl 操作手順 (2 / 1 0)

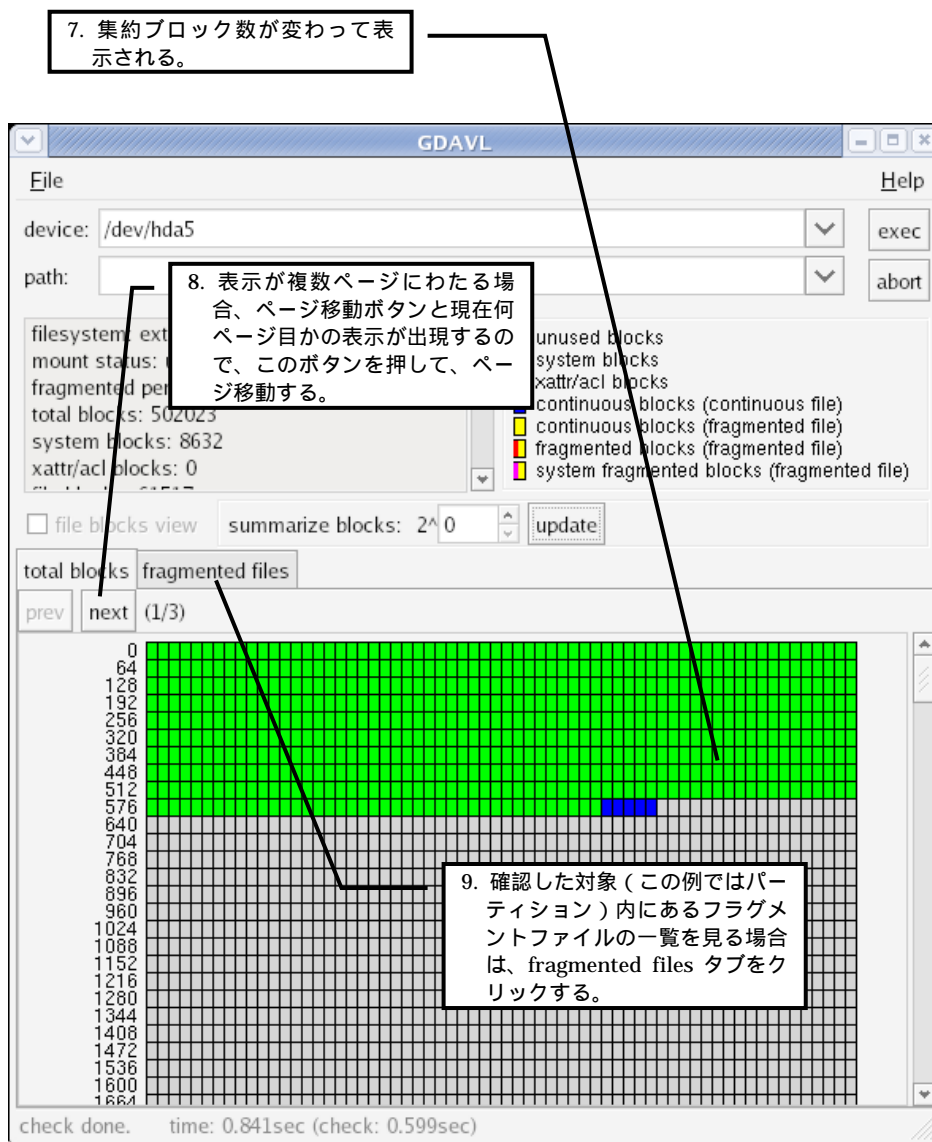


図 4-10 gdavl 操作手順 (3 / 10)

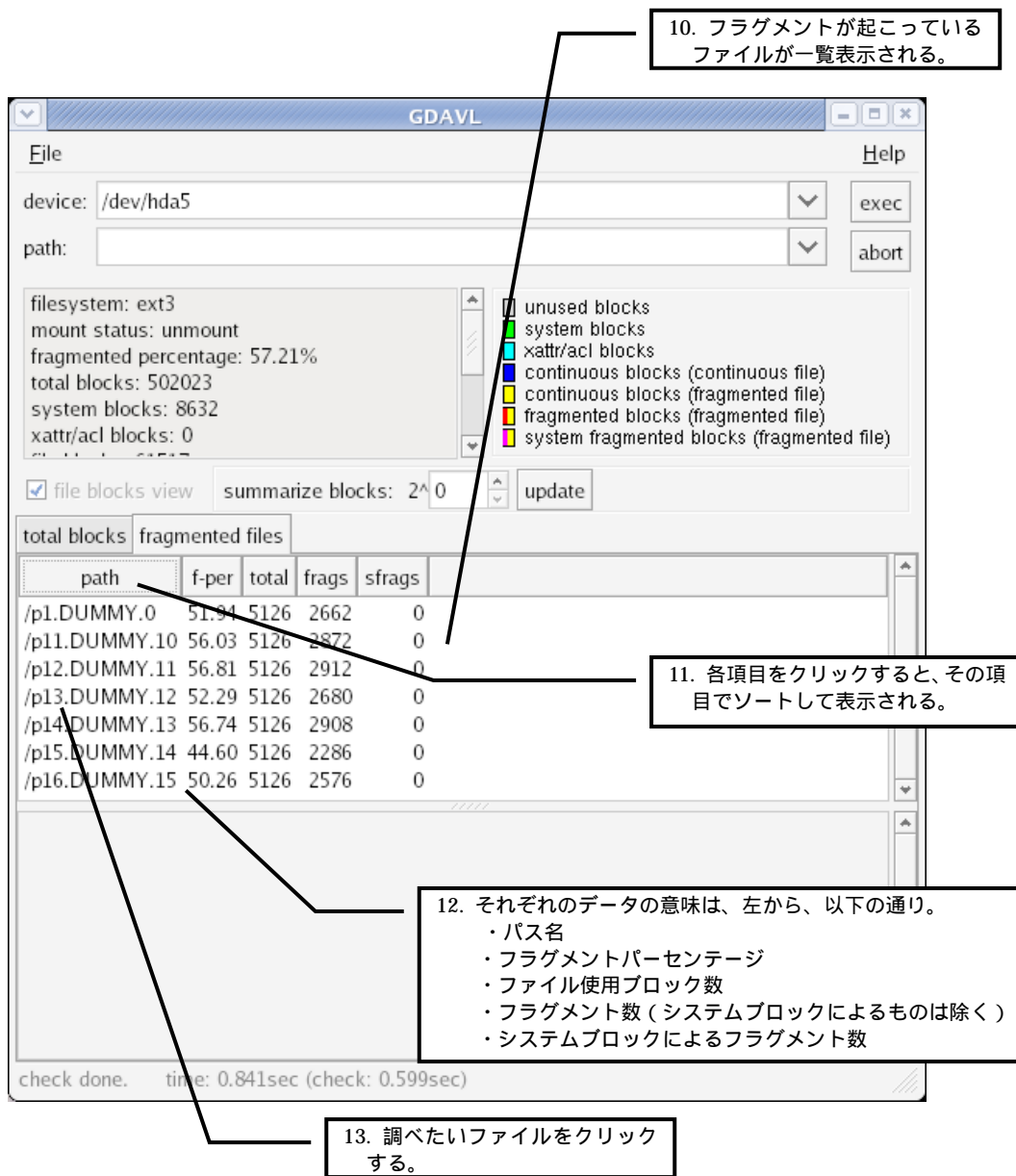
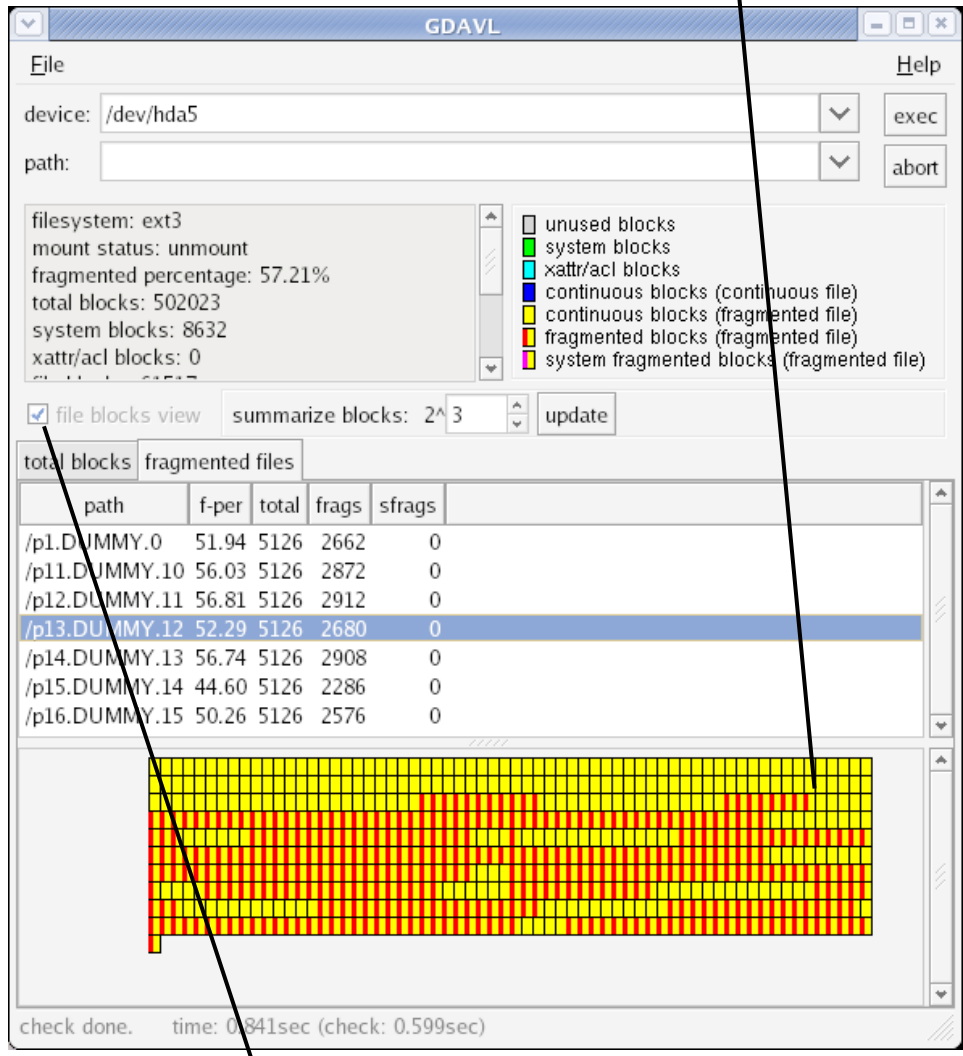


図 4-11 gdavl 操作手順 (4 / 1 0)

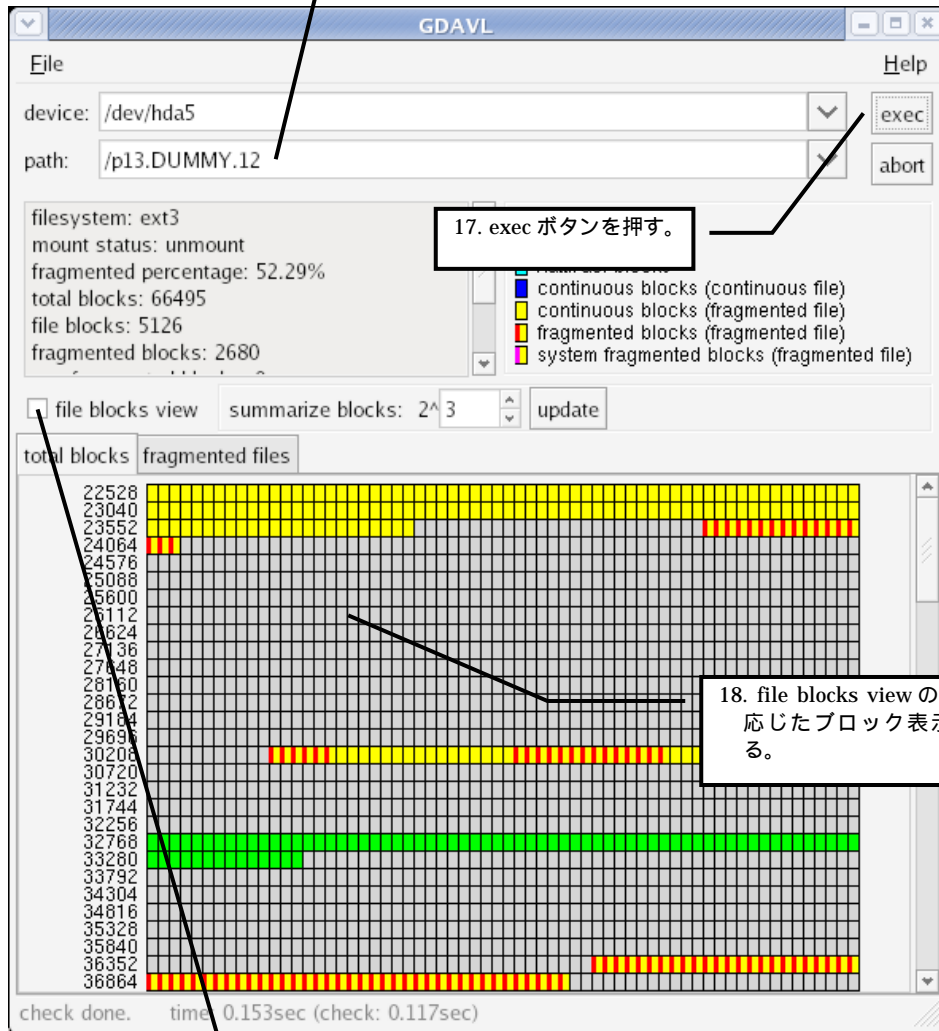
14. 選択されたファイルについて、ファイルを構成するブロックだけを構成順に見た場合に、どのようにフラグメントしているかを表示する（ファイル構成順表示）。ファイル構成順表示の場合は、ブロック番号順ではないため、左側にブロック番号が表示されない。



15. このチェックは、ブロック番号順 / ファイル構成順の2つの表示を切り替えるものであり、ファイル構成順表示の場合に、チェックマークが付く。fragment files タブ側では、ファイル構成順表示のみ可能である。

図 4-12 gdavl 操作手順 (5 / 10)

16. ディレクトリまたはファイルを確認対象とする場合は、path を指定する。この例の場合だと、/dev/hda5 の直下にある p13.DUMMY.12 というファイル名のファイルを確認している。



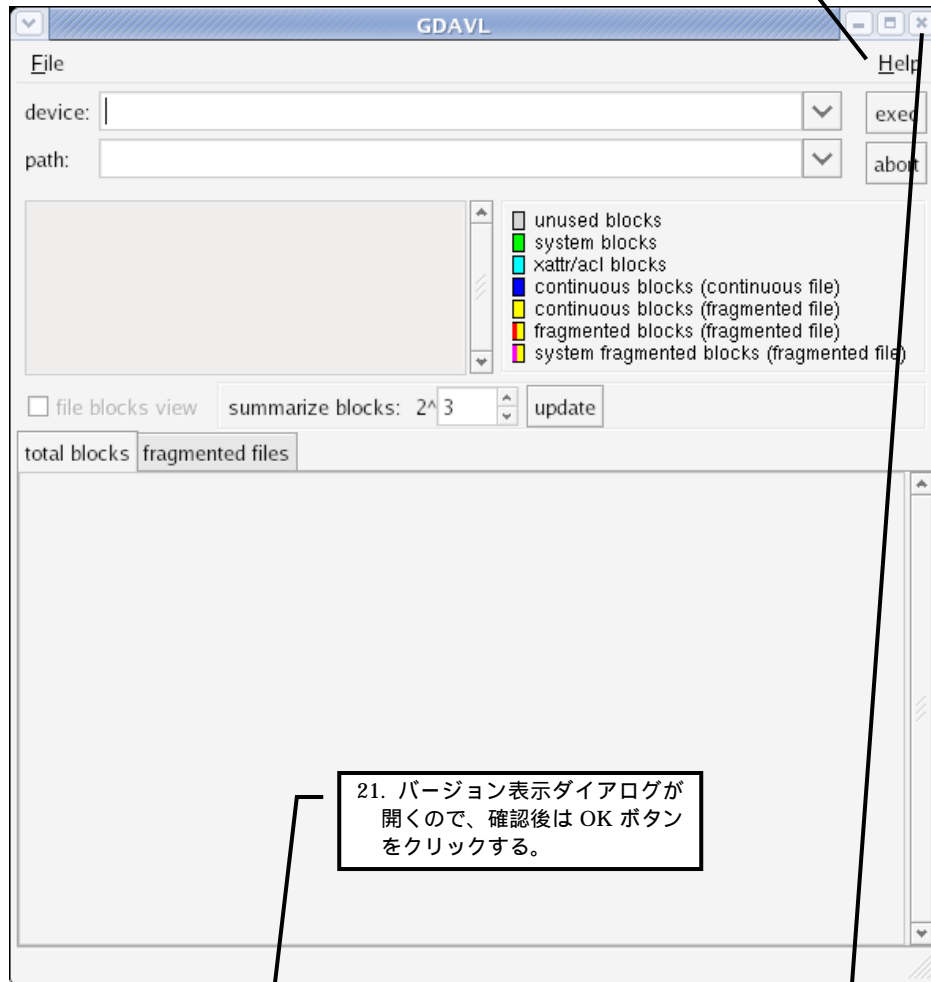
17. exec ボタンを押す。

18. file blocks view のチェックに応じたブロック表示が行われる。

19. ファイルを対象とした場合だけ、ブロックをファイル構成順に見るか (チェック ON)、ブロック番号順に見るか (チェック OFF) を選択できる。

図 4-13 gdavl 操作手順 (6 / 10)

20. バージョンを確認するには、メニューの help version とクリックする。



21. バージョン表示ダイアログが開くので、確認後は OK ボタンをクリックする。



22. 終了する場合は、x ボタンをクリックするかまたは、メニューから File Quit とクリックする

図 4-14 gdavl 操作手順 (7 / 10)

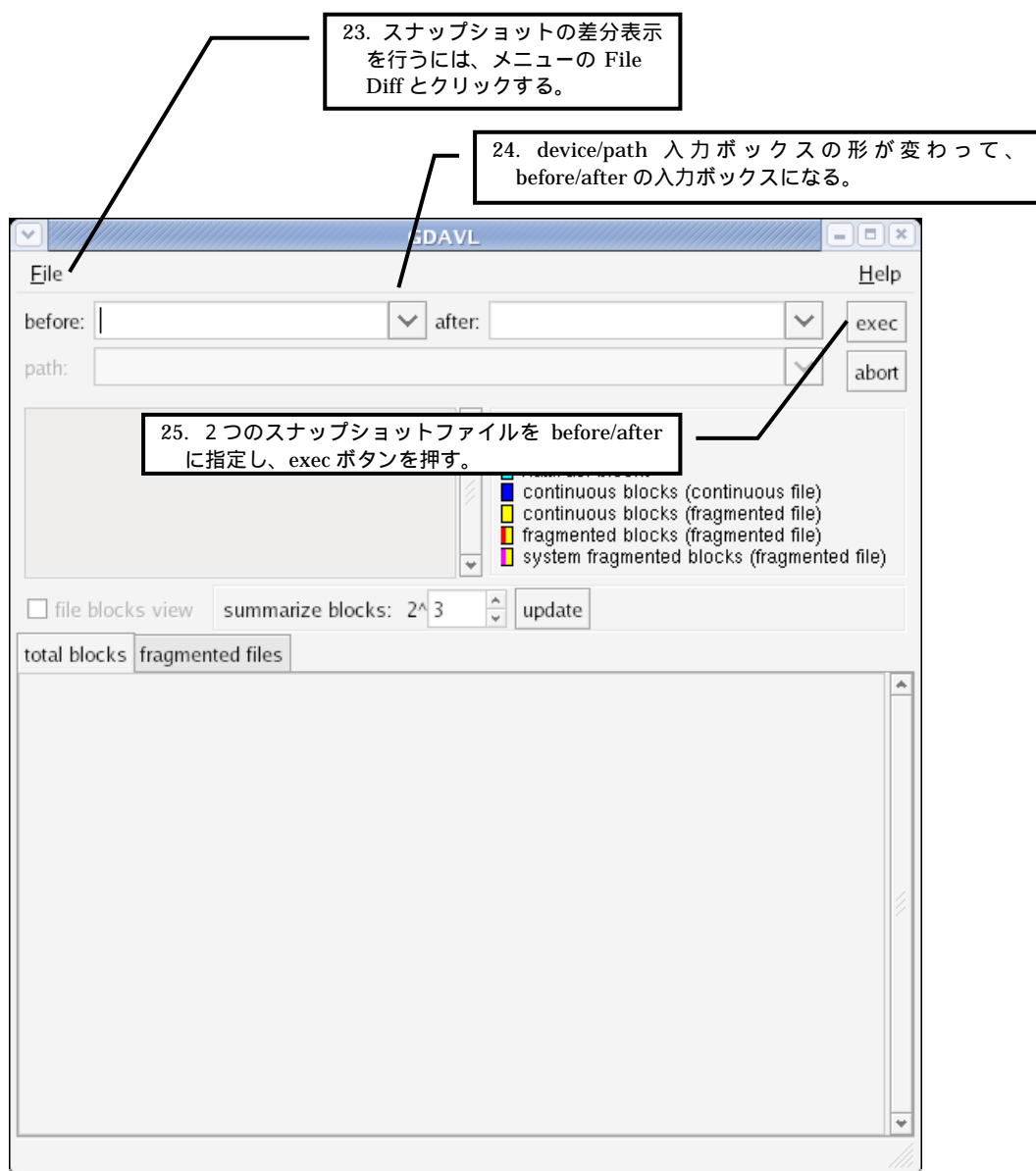
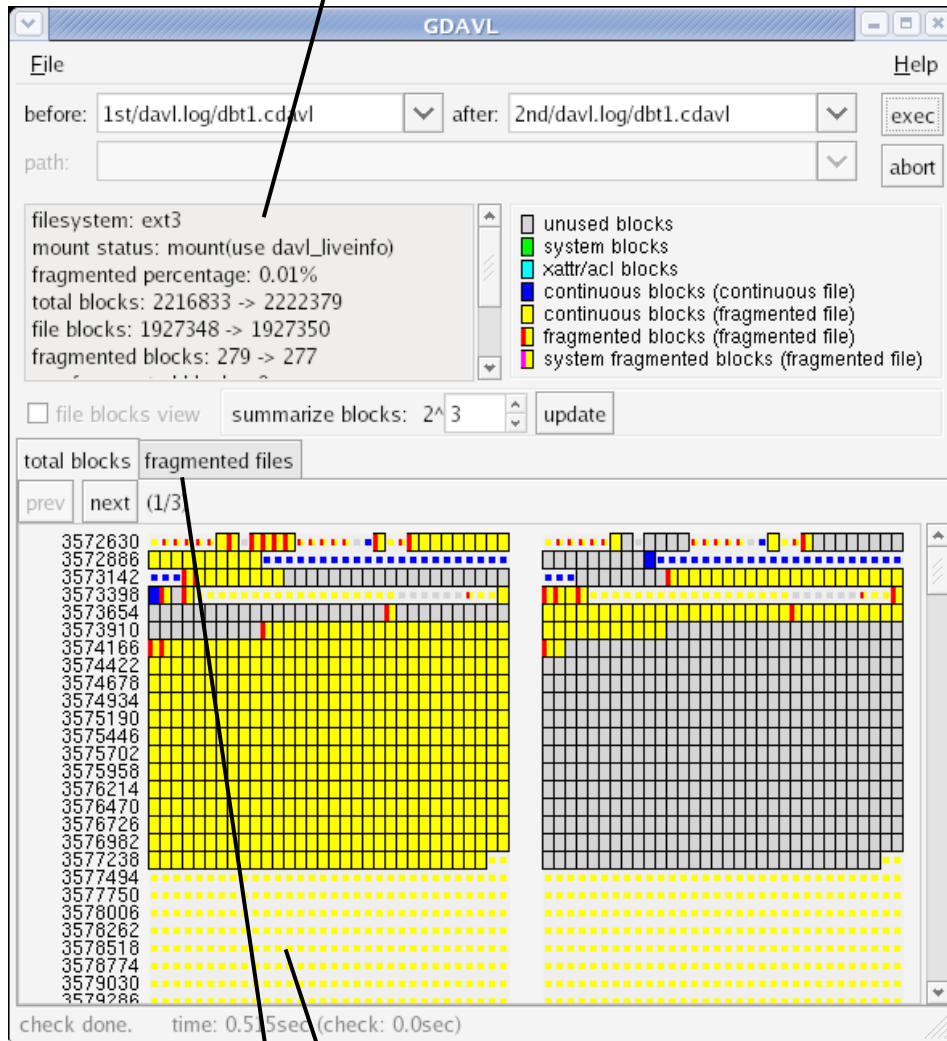


図 4-15 gdavl 操作手順 (8 / 10)

26. 差分がある項目は、xxx -> yyy の形式で表示される。

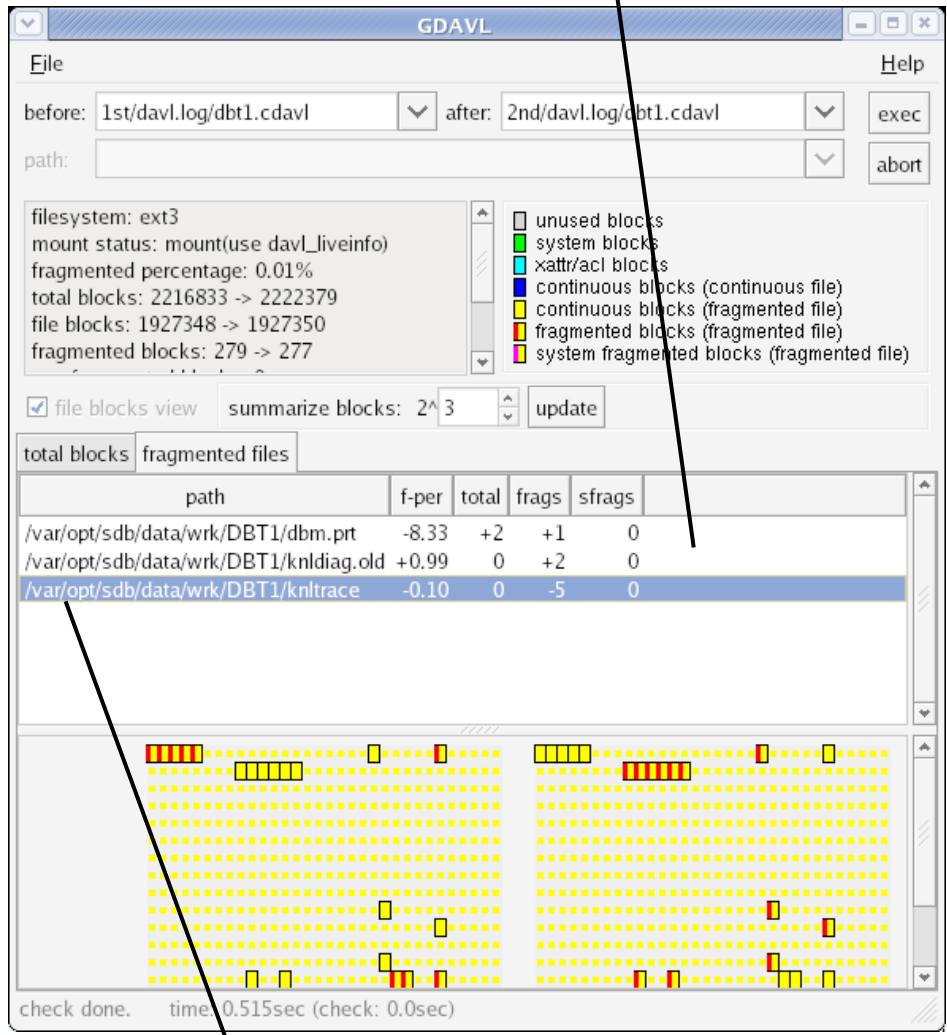


27. 左側に before、右側に after で指定したスナップショットの情報が表示される。差分があるブロックは枠付きで、そうでないブロックは枠なしで表示される。

28. 差分があるフラグメントファイルの一覧を見る場合は、fragmented files タブをクリックする。

図 4-16 gdavl 操作手順 (9 / 10)

29. 差分があるフラグメントファイルだけが一覧表示される。どちらか一方にだけ存在するフラグメントファイルについては、存在しない方はフラグメントなし、ブロック総数は存在している方と同じファイルと仮定して、差分を算出した結果が表示される。



30. 確認したいファイルの行をクリックすると、ファイル構成順での差分が表示される。

図 4-17 gdavl 操作手順 (1 0 / 1 0)

4.4. davl_liveinfo モジュールについて

DAVL は確認対象がアンマウント状態のパーティションの場合、ブロックデバイスを直接読み込んでファイルブロック情報を取得する。マウント状態のパーティションの場合も同じ処理を行うと、マウントパーティション上のファイルが変更された場合、ディスク I/O が実行されるまでの間は、その変更を読み出せないため、タイミングによってはファイル変更前の情報を表示してしまう問題がある。この問題を回避するため、カーネルからファイルブロック情報を直接取得するモジュールが `davl_liveinfo` モジュールである。

DAVL は、`davl_liveinfo` モジュールがロードされている状態で、かつ対象がマウント状態であれば、自動的にこのモジュールを使用する。システム負荷の軽減などの目的で、このモジュールを使いたくない場合は、`gdavl / cdavl` のコマンドオプションで `-s` を指定する必要がある。

ブロックデバイスへのファイル変更 I/O の遅延以外でも、`davl_liveinfo` モジュールを使用するかどうかで DA VL に見える inode が異なる場合があり、出力結果が異なる可能性がある。モジュール使用による出力結果の差分を表 4-4 に示す。

表 4-4 モジュール使用による出力結果差分 (ファイル変更 I/O 遅延以外)

項番	項目	モジュール使用時	モジュール未使用時	備考
1	シンボリックリンク	使用ブロックなしと認識	使用ブロックを正しく認識	リンク先パス名が 60 文字以上の場合のみ
2	ハードリンク	一番先にチェックしたパス名で表示	一番先にチェックしたパス名で表示	チェック順がモジュールを使用するかどうかで変わる可能性があるため、表示されるパス名が異なる場合がある
3	他ファイルシステムをマウントしているディレクトリ	使用ブロックなしと認識	使用ブロックを正しく認識	
4	inode 番号 1,3~10	使用ブロックなしと認識	使用ブロックなしと認識	ext3 ジャーナルの使用ブロックも含む

【補足～商標について】

- Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標である。
- MIRACLE LINUX は、日本オラクル株式会社が登録し、ミラクル・リナックス株式会社が使用権許諾を受けている登録商標である。
- その他、記載されている会社名、製品名は、各社 の登録商標または商標である。