

本稿は [Linux Japan 誌](#) 2001 年 3 月号に掲載された記事に補筆修正したものです。

フロッピーディスクの活用

DIMM メモリは投機筋が蠕いているらしく値段の上がり下がりが激しいですね。今は底値で 128MB がノーブランドなら 5000 円 以下です。安いといえば、CD-R は普及と共に健全に価格が低下し今や 1 枚 50 円です。50 円で買えるものというところ、3.5" 2HD(High Density の略) のフロッピー。強引なイントロですが (^_^;、今回はこのフロッピーディスクの活用についてお話しします。話題の中心は、Fduutils というユティリティです。これは Linux でフロッピードライブを扱うためのを集めたものです [1] [W³](#)。

Fduutils の概要

Fduutils には以下のものが含まれており、各配布系で標準的にインストールされている筈です。

```
superformat, fdmount, xdfcopy,
MAKEFLOPPIES, getfdprm, setfdprm,
fdacmd, floppycontrol, floppymeter,
diskd, diskseekd
```

このうち、一般ユーザーが日常使って幸せになれるのは superformat による、ディスク容量の増加 (3.5" 2HD で 1992 kB まで) でしょう。

コンパイル

標準でインストールされている筈ですが、詳しいドキュメント Fduutils.dvi を得るためにソースを展開して configure, make(make install は不要ですね) しましょう。この記事の内容は、Fduutils.dvi に全て記載されていますので、筆者の怪しい説明よりも原文を読んだ方が正しく理解できることと思います。

パーミッションの確認

フロッピーデバイスにアクセスするには、`/dev/fd*` に対して書き込みの権限が必要です。“`ls -l /dev/fd0*`” としてみてください。

```
brw-rw---- 1 root floppy 2, 0 ... /dev/fd0
...
brw-rw---- 1 root floppy 2, 84 ... /dev/fd0u1040
brw-rw---- 1 root floppy 2, 88 ... /dev/fd0u1120
brw-rw---- 1 root floppy 2, 28 ... /dev/fd0u1440
...
```

のように (字詰めしてます) となっていることと思います。この設定では、フロッピーを使うには floppy のグループに所属しなければなりません。それには `/etc/group` の floppy グループにユーザー名を追加します (再度ログイン後有効になります)。登録すべきユーザーが多く煩わしい場合には、セキュリティが低下しますが、

```
chmod 666 /dev/fd0
```

により `/dev/fd0` のパーミッションを 666 にして other にも解放するしかないでしょう。

物理フォーマット

現在 Linux のフロッピーディスクは `/usr/src/linux/drivers/block/floppy.c` の 400 行目あたりに記述されているように、31 種類が定義されており (固定パラメタ)、デフォルトで使えるようになっています。これ以外にもパラメタを変更してフォーマットすることがもちろん可能です。例えば、Fduutils のサイトには `fdacmd` を用いて、CP/M のディスクを解析、アクセスする方法などが紹介されています。

superformat

superformat は、種々のパラメタを指定してフロッピーを物理フォーマットします。ついでに、その物理フォーマットに合わせて MS-DOS 論理フォーマットも実行します。表 1 に、3.5" 2HD に対してデフォルトで定義されているタイプを示します。例えば 1920 kB (ただし、この値は ext2 で論理フォーマットした場合の容量で、DOS のものとは違います) は

```
superformat /dev/fd0 hd tracksize=12kB mss
```

とします。ここで mss は Mixed Sector Size の略で、読んで字の如く大きさの異なるセクターを混ぜて使うことでディスク容量を拡大する方式です。ディスク容量の拡大方法は、基本的にはシリンドラ (cyl) やセクター (sect) を通常の値よりも大きくすること (1440kB は cyl=80, sect=18) や、上記の mss などの方式を導入することで計られています。なお、`-f` を付けるとベリ

ファイルを省略しますからフォーマットに要する時間が短くなります。表 1 には superformat でフォーマット可能な最大容量 1992kB タイプと 1826kB タイプも示しました。1992kB の場合は mss のおかげでアクセスがかなり遅くなります、1826kB はアクセスがあまり遅くならない範囲での最大容量です。

以降このフロッピーは DOS のフロッピーとして使うことができます。例えば、mdir すると

```
Volume in drive A has no label
Volume Serial Number is 71BF-CB26
Directory for A:/

File "*" not found

1 947 648 bytes free
```

と使用可能な残り容量が表示されることで確かめることができます。もちろん、以下のようにして ext2 で論理フォーマットも可能です。

```
/sbin/mke2fs /dev/fd0
```

この際に表示されるメッセージは

```
mke2fs 1.14, 9-Jan-1999 for EXT2 FS 0.5b,
95/08/09
Linux ext2 filesystem format
Filesystem label=
480 inodes, 1920 blocks
96 blocks (5.00...
```

となって、ext2 では 1920 block すなわち 1920kB のディスクと認識されます。

ところで、初めて superformat を行った際には、次のようなメッセージが表示されるはずですが。

```
old capacity=12500
Measuring drive 0's raw capacity
(略)...,
add the following line to /etc/driveprm :
drive0: deviation=-2400
```

これは、フロッピードライブの物理的な容量が回転速度の影響を受けて変化するので、規定の値からの偏差を測定した結果なのです。指示通り /etc/driveprm に

```
drive0: deviation=-2400
```

を書き込みましょう。なお、この値は正確ではないらしいので、floppymeter で一度詳しく測定してみるとよいでしょう。

読み書き：mcopy, cp

DOS のフロッピーは Mtools を用いてシステムに mount せずに読み書きが可能です。mcopy でフロッピーにファイルをコピーするとき問題になるのが、名前の長さ制限です。DOS の 8 文字+拡張子 3 文字を超

える長さのファイルは変な名前(もちろん規則性はあります)に変換されてしまいます。例えば

```
mdir a:
...
TORI_P~1 GIF      58995 ... tori_pm3d.gif
TORI_P~1 GP        444 ... tori_pm3d.gp
TORI_P~1 JPE      42648 ... tori_pm3d.jpeg
TORI_P~2 JPE      10300 ... tori_pm3d_s.jpeg
```

となってしまうのです。これを元の名前で復元してコピーするのは面倒ですね。ところがこのフロッピーを mount すると

```
ls -l /floppy
...
-rwxr-xr-x ... 58995 ... tori_pm3d.gif
-rwxr-xr-x ... 444 ... tori_pm3d.gp
-rwxr-xr-x ... 42648 ... tori_pm3d.jpeg
-rwxr-xr-x ... 10300 ... tori_pm3d_s.jpeg
```

と元の長い名前で扱われますから、cp コマンドで TAB による名前補間まで有効になって、簡単にコピーできるようになります。mcopy ではオプション -t が UNIX ↔ DOS のテキストの改行コードを自動変換してくれるので便利です。

mount

以上、DOS のフロッピーは mount にかかわらず読み書き可能な方法があって、DOS らしい使い方ができます。一方 UNIX ではストレージデバイスは mount してシステムに組み込む必要がありますから、ext2 ディスクも当然 mount する必要があります。一般にデバイスの mount/umount は root に委ねられており一般ユーザーには解放しないのが安全です。ところが、リムーバブルなものはそうも言ってもらえませんから、ユーザー権限で可能にする道があります。御託を並べてしまいましたが、方法は /etc/fstab に

```
/dev/fd0 /floppy auto user,rw,noauto,exec 1 1
```

と登録するだけです。2 番目のフィールドはマウントポイントで、筆者は /floppy としていますが、読者ご自身のホストの環境に合わせください(/fd0 や /mnt/fd0 が多いようです)。こうすると、“cat /proc/filesystem” で表示されるファイルシステムは自動判別 (auto) されて

```
mount /floppy
```

で mount できるようになります。ext2 は問題ないでしょうが、もし DOS が判別されないときは、“modprobe msdos” を実行、それでもだめならカーネルの再構築となります。mount した場合には (DOS であっても) cp

表 1 3.5" 2HD の物理フォーマットにおけるパラメタ

番号	容量 (kB)	メディア定義	ext2	dd 容量
7	1440	hd		1,474,560
11	1680	hd sect=21	×	1,720,320
15	1722	hd sect=21 cyl=82	×	1,763,328
19	1743	hd sect=21 cyl=83	×	1,784,832
24	1760	hd sect=11 ssize=1kB		1,802,240
25	1920	hd tracksize=12kB mss		1,966,080
29	1840	hd tracksize=23b mss	×	1,884,160
31	1600	hd sect=20	×	1,638,400
-	1992	hd tracksize=12kB cyl=83 mss		2,039,808
-	1826	hd sect=11 cyl=83 ssize=1kB		1,869,824

コマンドしか使えません。

ここでちょっと問題がありました。ext2 フロッピーの一旦抜いてしまうと次には mount はされるのですが実際には読み書きができないものがあるのです。表 1 の ext2 欄の × 印のもので、これは、フロッピーを抜くと、ディスク情報がリセットされてしまうことによるものです。× 印のものを新規に入れ直した場合には getfdprm でディスクがどう認識されているか調べると

```
mount /floppy (マウント場所はホストの環境によります)
getfdprm
DS HD sect=18
```

となり sect=18 すなわち 1440kB と認識されてしまうことが判ります。したがって正しいディスク情報を与え直す必要があります。root ならば明示的にドライブ (ディスク容量) を指定して

```
mount /dev/fd0u1743 /floppy
```

のように正しく設定することが可能です。一般ユーザーでは、この形式では mount コマンドを実行できません。そこで登場するのが setfdprm です。ディスクパラメタは /etc/fdprm に記述されており、1743 kB は 1743/1440 と登録されていますから (登録されていない場合は floppy.c を参照して追記してください)

```
setfdprm /dev/fd0 1743/1440
```

と実行して正しくセットすれば、以降アクセスが可能になります。getfdprm で確かめてください。

floppycontrol

実は、フロッピーを入れ直して自動認識するかどうか表 1 について試すと、1760 kB が mdir すらできないことに気付くと思います。その訳は、floppycontrol で調べると判ります。

```
floppycontrol -p
...
autodetect seq.= 7,4,25,22,31,21,29,11
...
```

autodetect のリストに 24 番 (1760kB) が載っていません、従ってスキャンすらされないのです。では、15 番 (1722kB) が mdir されるのは何故でしょう? スキャンの指定はセクター (sect) に関するものでトラック (cyl) は違っていても構わないと説明があります (最初のトラックをスキャンするだけです。また、ディスクの情報はその最初のトラックに書き込まれています)。従って、11 番があれば 15 番、19 番はスキャン可能となります。これより 3.5" 2HD に関して、mdir で自動認識させるために必要な登録番号は 7,11,24,25,29,31 の 6 個になります。リストには最大 8 個の登録が可能ですから、4 番 (720kB) と 22 番 (1120kB: 意味が良くわかりませんが default の設定に敬意を表して) を加えて、root となって

```
floppycontrol --autodetect 7,11,24,25,29,31,4,22
```

と自動認識のタイプの設定をやり直しましょう。ところで、これはあくまで物理フォーマットの自動認識です。その上の ext2 ファイルシステムが認識されることを保証するものではありません。改めて、表 1 の ext2 欄は上記のようにして物理フォーマットが自動認識されるように設定した場合、× 印のタイプは ext2 も認識されるようになったということです (何故なのかは勉強

不足で説明できません)。

fdformat : 時代遅れ

筆者はフロッピーの物理フォーマットをずっと fdformat で行っていました。3.5" 2HD では標準的には 1440kB ですから、

```
fdformat /dev/fd0u1440
```

です。これより大きい容量で物理フォーマットするつもりで、

```
fdformat /dev/fd0u1920
```

とすると、エラーが出ます。実際 fdformat のマニュアルには、3.5HD では一見 360,720,1440 kB しか物理フォーマットできないような記述になっています。が、実は 1760 kB までは、例えば

```
fdformat /dev/fd0u1760
```

で、物理フォーマットが可能です。superformat がある以上、このコマンドは確かにもう過去の遺物 (obsolete) かもしれません。

dd, tar と共に

tar も dd も論理フォーマットに関係な物理フォーマットが認識できるデバイスに直接データを読み書きします。従って、フロッピーが物理的に認識されれば

```
dd if=filename of=/dev/fd0
dd of=/dev/fd0 if=filename
tar zcvf /dev/fd0 files
tar zxvf /dev/fd0
```

のようなコマンドを用いて、/dev/fd0 に対して直接読み書きが可能となり、例えば 1992kB(ext2 上) のタイプは 2,039,808 bytes(1024 × 24 × 83) の容量を持ちます。

まとめ

今回は自分で読み返してもどうも見通しが悪いです。フロッピーを楽に使うためのマメ知識と Fdutils の設定をここでまとめます。

- フロッピーを自由に使うには /dev/fd0 にアクセスできる権限を持たないといけない。
- root で floppycontrol を用いて自動認識するタイプの設定を行う
- superformat でフォーマットすると DOS フロッピーが作成でき、新規挿入時にも自動認識される。

- DOS フロッピーは mount しなくとも Mtools で読み書きできる。
- DOS フロッピーは mount して使うと長い名前のまま扱える。
- ext2 でフォーマットしたもので自動認識されるのは 7,24,25 番系列 (cyl 数が違っていても良いという意味) のものである。

うーん、まだすっきりしないですね。一般ユーザーにはこの複雑さが見えないようにしてフロッピーを使えるようにすることも管理者の腕前なのでしょう。

参考文献

- [1] Fdutils の公式サイト
<http://fdutils.linux.lu>