

# KNOPPIX/Math 作成方法

濱田龍義

TATSUYOSHI HAMADA

福岡大学

FUKUOKA UNIVERSITY \*

本稿では, KNOPPIX/Math[1] の作成方法について解説する. 以下の作業を実施するにあたっては, 前提知識として UNIX 系システムをコマンドラインで操作できることが望ましい.

## 1 KNOPPIX について

KNOPPIX[2] とは, ドイツの Klaus Knopper 氏によって開発された CD 起動型 Linux の一種である. 現在, その開発は Klaus Knopper 氏を含む LinuxTAG というグループによって進められている. ハードウェアの自動認識機能が素晴らしく, 手軽に Linux デスクトップを体験できるため, 世界中で活発に利用されている. 日本では, 独立行政法人産業技術総合研究所 (AIST) の須崎有康氏を中心とする研究グループが日本語化された KNOPPIX を 2002 年 9 月に公開した [3]. KNOPPIX は, オープンソースソフトウェアによって構成されており, その簡便性と配布性の良さから, KNOPPIX を原型に再構築された様々な派生版の存在が知られている. 特に教育利用に特化した KNOPPIX Edu[4], USB メディアから起動可能な USB-KNOPPIX[5], 生物情報学に特化された KNOB[6] などが国内では有名である.

## 2 KNOPPIX/Math について

KNOPPIX/Math は, KNOPPIX 日本語版を原型に再構築された派生版の一種であり, 数学のために特化されたコンピュータ環境である. KNOPPIX/Math の CD もしくは DVD を PC に入れて再起動するだけで, すぐに様々な数学ソフトウェアを使うことが出来る. 収録している数学ソフトウェアは多岐に渡っており, 文書組版システム  $\text{\TeX}$ , 汎用数式処理システム Maxima や Risa/Asir, MATLAB 類似の数値計算ソフト Octave, S 言語類似の統計計算ソフトウェア R, 対話式幾何学ソフトウェア KSEG など, 高校, 大学等の数学, 理工教育に適したソフトウェアから, 最前線の数学研究に関連するツールまで幅広く収録している. また, プログラミング言語などの開発環境を複数備えているため, 国内の複数の大学でプログラミング教育にも活用されている. また, 基本的なツールとして Web ブラウザ Iceweasel (Mozilla Firefox の別名) やオフィス環境 OpenOffice.org を収録しているため, 数学ソフトウェアに限らず, 通常のコンピュータ環境としても利便性が高い. また, 早い段階から国際的なプロジェクトとしての有効性を意識して, 2004 年に英語版, 2005 年に朝鮮語版の公開を行った. 最新版の DVD では, 日本語, 中国語, 台湾語, 朝鮮語などに対応しており, DVD1 枚で多言語入力も可能である.

福岡大学の濱田, AIST の須崎有康氏, 飯島賢吾氏の 3 人によって 2003 年 2 月に始められたプロジェクトは, 2005 年 8 月に国内の数学者有志から構成される KNOPPIX/Math Project に発展し, 現在に至る. 現在

---

\*hamada@holst.sm.fukuoka-u.ac.jp

の主な活動として、KNOPPIX/Math CD/DVD の編集と公開、収録ソフトウェアに関する講演、展示による紹介、収録ドキュメントの執筆等を行っている。

### 3 Debian Package について

KNOPPIX は Debian GNU/Linux[7] と呼ばれる Linux ディストリビューションを原型に作成されている。KNOPPIX にインストールされている主要なソフトウェアは基本的に Debian GNU/Linux の標準パッケージである dpkg というシステムによって管理されている。ソフトウェアパッケージによる管理はインストール、アンインストールが大変容易になるが、パッケージ間の依存関係による問題が生じやすい。そこでパッケージ間の依存関係を自動的に管理し、アップグレード時に設定ファイルの存在まで考慮してくれるシステムが必要となった。それが、現在 Debian GNU/Linux や Vine Linux で標準的に使われている APT(Advanced Packaging Tool)[8] と呼ばれるパッケージ管理システムである。APT は利用可能なパッケージのソースとなる /etc/apt/sources.list や /etc/apt/sources.list.d/\* というファイルを使用しており、バイナリパッケージ (\*.deb) のアーカイブに関する情報が記述されている。場合によっては、インストールするパッケージに応じて sources.list を書き換えることで Debian Project から公式に配布されているソフトウェアパッケージ以外のソフトウェアを APT で管理することも可能である。

APT の基礎的な命令をまとめておく。ここで、<package\_name> はパッケージ名、<keyword>は検索したいキーワードに置き換えて欲しい。

パッケージの情報の更新	apt-get update
パッケージのインストール	apt-get install <package_name>
パッケージのアンインストール	apt-get remove <package_name>
パッケージの解説	apt-cache show <package_name>
キーワードによる検索	apt-cache search <keyword>

### 4 KNOPPIX 再構築

KNOPPIX の再構築は、次のような流れになる。

1. 原型となる KNOPPIX からソースディレクトリを作成する。
2. ソースディレクトリを圧縮して cloop ファイルを作成する。
3. cloop ファイルを同梱した ISO イメージを作成する。

KNOPPIX が日本で紹介されてから 4 年が経ったが、基本的な再構築の手順はほとんど変わっていない。[10] によって公開された再構築方法を元にここでは作業を進める。最近では、大学の講義で KNOPPIX の再構築を行う試み [11] もあり、もっと大勢の人に KNOPPIX の再構築を楽しんでもらえると嬉しい。さて、再構築において一番時間を要する作業がソースディレクトリを作成する作業である。原型からシステム全体を複製し、不要なパッケージを削除し、必要なパッケージのインストールを行う。CD 版の場合には、収録できるファイル容量が限られているため、最も神経を使い、人間の手による作業が必要な部分でもある。

一方、計算機パワーを必要とするのが 2 番目の作業である。最近の計算機であれば、CD 版であれば十数分で作業が終了するが、容量の大きい DVD を作成する場合には、圧縮作業を終えるのに 1 時間近くを費さなければいけない場合もある。

## 4.1 作成機材について

現在、実際に KNOPPIX/Math を作成している PC の性能は以下の通りである。

CPU	Pentium4 3GHz
実メモリ	2GB
Swap メモリ	2GB
ハードディスク	IDE 120GB

KNOPPIX では、cloop という圧縮ファイルシステムを採用している。標準的な CD の収録容量は 700MB、DVD の収録容量は 4.7GB であるので、ハードディスクインストール状態で、上記数字の約 3 倍弱の容量を、圧縮することでメディアに収録することができる。

この圧縮作業は、上記環境の場合、CD で 15 分前後、DVD で 1 時間前後かかる作業であるため、CPU の速度は速ければ速いほど良い。また、メモリの量は、圧縮作業において大変重要であり、KNOPPIX HACKS[9] によると CD 版で実メモリ+スワップで 1GB と言われている。現在のメモリ価格を考えると、実メモリを最低でも 1GB は確保したい。もちろん、DVD 版を作成する場合には、さらにメモリを用意しておいた方が良いでしょう。しかし、DVD に用いている iso9660 ファイルシステムの制限から、1 個のファイルは 2GB に制限されているため、今のところ実メモリ+スワップで 4GB の環境であれば、十分に DVD の再構築が可能なのである。また、このマシンは DVD+ReWritable ドライブを装備しており、再構築時のオペレーティングシステムは、産総研版の KNOPPIX を利用する。また、ハードディスクは IDE 環境で接続されているものとする。

## 4.2 KNOPPIX 原型の複製

この節では KNOPPIX 再構築を行うための環境の作成について述べる。原型となる KNOPPIX は産総研から配布されている日本語版 `knoppix_v5.1.1CD_20070104-20070122+IPAFont_AC20070123.iso` とする。これを複製して、再構築を行う。これ以降、主な作業は `su` によって `root` で行う。`root` のプロンプトを `#` で表すことにする。また、`chroot` 環境での作業のプロンプトは `##` で表すことにする。

1. まずは作業領域のハードディスクを `ext2` 形式でフォーマットする。

```
# mke2fs /dev/hdb1
```

2. `ext2` ハードディスクをマウントする。仮に IDE ハードディスクを `/dev/hdb1` としよう。

```
# mount -t ext2 /dev/hdb1 /media/hdb1
```

3. 作業ディレクトリ `source`, `source/KNOPPIX` を作成する。

```
# cd /media/hdb1
# mkdir -p source/KNOPPIX
```

4. 原型の KNOPPIX を複製する。

```
# cp -Rp /KNOPPIX/* source/KNOPPIX
```

5. ドメイン名を解決するため /etc/resolv.conf を複製する.

```
# cp /etc/resolv.conf source/KNOPPIX/etc/
```

6. /media/hdb1/source/KNOPPIX をルートファイルシステムにする.

```
# chroot source/KNOPPIX
```

7. chroot 環境で proc ファイルシステムをマウントする.

```
## mount -t proc /proc proc
```

### 4.3 不要パッケージの削除

まずは、パッケージの一覧を取得し、不要なパッケージを削除する。パッケージの一覧を取得するには幾つか方法がある。一般的なのは、次の方法である。

```
## dpkg -l | less
```

しかし、より詳しい情報が欲しい場合には dpkg-query が便利である。

```
## dpkg-query -W --showformat='${Package}\t${Version}\n'
```

dpkg-query で指定できる形式は、Installed-Size, Section, ... 等、多数あり、詳しくは man dpkg-query を参照されたい。

さて、実際にパッケージファイルを削除する際には、設定ファイルも同時に削除するケースが多い。

```
## apt-get remove --purge <package_name>
```

--purge を忘れて、設定ファイルを残したが、後で設定ファイルだけを削除したい場合には、以下の命令を用いる。

```
## dpkg --purge <package_name>
```

### 4.4 新規パッケージのインストール

Debian の公式パッケージに収録されている汎用数式処理ソフト Maxima を例にインストールの手順を記す。例えば APT の apt-get を用いる場合には、

```
## apt-get install maxima
```

だけで良い。/etc/apt/sources.list を参照して自動的に Maxima に必要なパッケージを選択してダウンロード、インストールを行う。

さらに、Maxima の GUI 環境をインストールしたい場合には、xmaxima, wxmaxima 等のパッケージ名を指定してインストールすれば良い。

```
## apt-get install xmaxima
## apt-get install wxmaxima
```

一方、dpkg を利用してインストールする際は、具体的なファイル名を指定する必要がある。

```
## dpkg -i maxima_5.10.0-6_i386.deb
```

それでは、公式パッケージに収録されていないソフトウェアをインストールする場合にはどうしたら良いか。まずは、開発元で Debian に対応していないかどうかを調べる。例えば、計算機代数ソフトウェア Singular[12] は開発元で Debian の APT にも対応しており、Singular-release.deb というパッケージをインストールすることで/etc/apt/に必要なソースを書き加える。以下は、Singular の Download ページに書かれているインストール方法の例である。

1. まずは、wget を用いて Singular-release.deb というファイルを取得する。

```
## wget http://www.mathematik.uni-kl.de/%7Emschulze/repo/DEBS/Singular-release.deb
```

2. 次に Singular-release.deb を dpkg という命令でインストールする。この作業によって Singular のパッケージをどこからダウンロードするかなどの情報を/etc/apt/sources.list.d/Singular.list として追加する。

```
## dpkg -i Singular-release.deb; apt-get update
```

3. APT の命令 apt-get を用いて singular-icons, singular-help をインストールする。

```
## apt-get install singular-icons singular-help
```

これだけの作業で、依存関係を利用して singular-core がインストールされ、Singular が実際に利用できるようになる。

#### 4.5 chroot 環境の終了

さて、ソフトウェアパッケージのインストールが完了したらダウンロードしたパッケージファイルは不要なので、消去する。次の命令によって、Download された /var/cache/apt/archives/\*.deb が消される。

```
## apt-get clean
```

/etc/resolv.conf には作成者のドメインにおける DNS 情報が記載されているので、気になるようであれば、削除しておく。

```
## rm /etc/resolv.conf
```

KNOPPIX 上で locate コマンドを実行できるようにするためには、データベースの更新が必要なので、以下の命令で更新しておく。

```
## updatedb
```

ただし、更新されたデータベースは、ISO イメージに焼き付けられた途端に更新することができなくなるため、後日、KNOPPIX を CD/DVD から起動して、実際に locate を実行した際には「locate: 警告: データベース '/var/cache/locate/locatedb' は 8 日以上古い」というエラーメッセージが出ることを覚悟しないとイケない。

忘れずに行わなければいけないのが、proc ファイルシステムのマウント解除である。この操作を忘れると、予想よりも大きな cloop ファイルができてしまう。数時間も待つて cloop が完成しないときなどは、この操作を行ったかどうか疑った方が良い。

```
## umount /proc
```

chroot 環境の終了する際は

```
## exit
```

で終了する。これでソースファイルの準備ができた。

#### 4.6 マスターファイルの作成

さて、chroot での作業を終え、カレントワーキングディレクトリは /media/hdb1 とする。次に ISO イメージを作成するためのマスター作業用のディレクトリを作成する。

```
# mkdir master
```

まずは、CD/DVD に含まれるファイルの複製を行う。

```
# rsync -a --exclude=KNOPPIX/KNOPPIX /cdrom/ master
```

最後に cloop を作成する。

```
# mkisofs -R -l -V "KNOPPIX iso9660 filesystem" -hide-rr-moved -v \  
/media/hdb1/source/KNOPPIX | create_compressed_fs - \  
65536 > /media/hdb1/master/KNOPPIX/KNOPPIX
```

私の作業環境の場合、CD で 15 分、DVD で 1 時間程度かかる。最近のコンピュータであれば、もう少し速くなる可能性がある。

DVD を作成する際の注意点としては、ISO イメージは一つのファイルが 2GB を越えることができないことである。cloop ファイルが 2GB に収まらない場合には、master/KNOPPIX/KNOPPIX, master/KNOPPIX/KNOPPIX2 などのように cloop を分割する。KNOPPIX/Math/2007 では、/usr/local 以下を KNOPPIX2 に収録している。KNOPPIX の起動時に KNOPPIX, KNOPPIX2 が自動的にマウントされるように設定されているので、どのディレクトリを分割するかだけを作成者が判断して作業を行えば、それほど面倒な設定をせずに DVD 化できる。

## 4.7 CD/DVD イメージの作成

cloop ファイルが作成できたら、ISO イメージを作成する。以下の例では、作成する ISO イメージのファイル名を `knoppix_v5.1.1-math-dvd-20070307-ja.iso`、メディア名を `KNOPPIX/Math` としているので、適宜、自分の好みの名前に変更する。

```
# cd /media/hdb1/master
# mkisofs -l -r -J -V "KNOPPIX/Math" -hide-rr-moved \
  -b boot/isolinux/isolinux.bin \
  -o /media/hdb1/knoppix_v5.1.1-math-dvd-20070307-ja.iso \
  -no-emul-boot -boot-load-size 4 -boot-info-table /media/hdb1/master
```

## 5 KNOPPIX 高速起動化

KNOPPIX は、CD/DVD というメディアの特性上、起動に時間がかかるという問題があった。この問題に対して、(株) アルファシステムズでは LCAT (Live CD Acceleration Tool kit)[13] という高速化ツールを開発し、オープンソースソフトウェアとして公開している。ここでは細かいチューニングについては述べないが、高速起動化を行うことにより、KNOPPIX を 1 分前後で起動できるようになる。詳しいことが知りたい場合には (株) アルファシステムズによるドキュメント [13] や、山城潤氏による Web 上での解説 [14] を参考にしてほしい。

### 5.1 高速起動化ツール

まずは、作成した ISO イメージから起動時にどの部分にアクセスしているかの情報、起動プロファイルを取得する。ISO イメージを元に何らかの方法で CD/DVD に焼き付け、一時的な起動メディアを作成し、起動させる。この際、VMware Player や VirtualBox などの仮想環境を利用して ISO イメージから起動すると平均的なハードウェア環境となり便利である。起動プロファイルを取得するための手順を以下に示す。

1. 起動時に “profile” という cheatcode <sup>1)</sup> を入力する。

```
boot: profile
```

<sup>1)</sup> 起動時に入力する設定変更のための文字列を cheatcode と呼ぶ。その他の cheatcode については、志子田有光氏による日本語訳が参考になる [16]。

2. 起動プロファイルは `/proc/cloop/read_blocks` に記録されるので、これを何らかの手段で作業領域 `/media/hdb1` に複製する。<sup>2)</sup>

```
# cat /proc/cloop/read_blocks > boot.blk
# cp boot.blk /media/hdb1
```

3. さらに、OpenOffice.org などの起動に時間がかかるアプリケーションのための起動プロファイルを取得するため、一旦、記録されているデータをリセットする。

```
# echo 1 > /proc/cloop/reset_read_blocks
```

4. OpenOffice.org 等を起動後に、再度アプリケーション用の起動プロファイルを取得する。

```
# cat /proc/cloop/read_blocks > appli.blk
# cp appli.blk /media/hdb1
```

カレントワーキングディレクトリは `/media/hdb1` とする。Sourceforge から LCAT のバイナリを取得する。 <http://sourceforge.jp/projects/lcat/> にアクセスして `lcat_1.0.1-2.05_with-binary.tar.gz` をダウンロードする。ファイルを `/media/hdb1` 以下に展開し、さしあたり必要なファイルは、`cloopoptimizer` と `rblk2bl` なので、二つのファイルに対してシンボリックリンクをはり、使いやすいようにする。

```
# tar xvzf lcat_1.0.1-2.05_with-binary.tar.gz
# ln -s lcat_1.0.1-2.05/util/cloopoptimizer
# ln -s lcat_1.0.1-2.05/util/rblk2bl
```

## 5.2 cloop の最適化

cloop の原型 `master/KNOPPIX/KNOPPIX` を `/media/hdb1` に移動する。

```
# mv master/KNOPPIX/KNOPPIX KNOPPIX.normal
```

`KNOPPIX.normal` を元に、`boot.blk`、`appli.blk` 等の起動プロファイルを参考にして最適化を行う。

```
# ./cloopoptimizer KNOPPIX.normal boot.blk appli.blk > master/KNOPPIX/KNOPPIX
```

さらに、`rblk2bl` を用いて先読みリストの作成を行う。

```
# ./rblk2bl ./boot.blk > master/KNOPPIX/KNOPPIX.boot.lst
```

最後に、最適化された `cloop` ファイル `master/KNOPPIX/KNOPPIX` を含んだ高速化 CD/DVD イメージを作成する。

<sup>2)</sup>あらかじめ `/media/hdb1` をマウントしておく必要がある。



```
# cd /media/hdb1/master
# mkisofs -l -r -J -V "KNOPPIX/Math" -hide-rr-moved \
  -b boot/isolinux/isolinux.bin \
  -o /media/hdb1/knoppix_v5.1.1-math-dvd-20070307-ja-lcat.iso \
  -no-emul-boot -boot-load-size 4 -boot-info-table /media/hdb1/master
```

これで、完成である。

## 6 KNOPPIX の起動過程

KNOPPIX を再構築する際に、パッケージのインストール、アンインストールは比較的、簡単に行うことができる。しかし、デスクトップの背景画面を変更したり、メニューシステムを変更したり、デスクトップ環境を KDE 以外のものを選択したいといったときには、KNOPPIX がどのような手順を経て起動しているか、その様子を良く知ることが大事である。

KNOPPIX で最初に実行されるのは ISOLINUX というブートローダである。ISOLINUX は CD/DVD 内の boot/isolinux/isolinux.cfg の設定を読み込んだ後、ミニルートを実行し、cloop ファイルを展開する。cloop 内の各種設定ファイルが読み込まれながら、起動スクリプトが実行される。起動時に読み込む重要なファイルを列挙すると、

- CD/DVD 内の boot/isolinux/isolinux.cfg
- CD/DVD 内の boot/isolinux/minirt.gz
- minirt.gz 内の linuxrc
- /etc/init.d/knoppix-autoconfig (minirt.gz 内の accel/\* から起動時に複製)
- /sbin/hwsetup
- /usr/sbin/mkxorgconfig (minirt.gz 内の accel/\* から起動時に複製)
- /usr/sbin/rebuildfstab
- /etc/init.d/xsession (minirt.gz 内の accel/\* から起動時に複製)
- /etc/X11/xinit/xinitrc
- /etc/X11/Xsession.d/\*
- /etc/X11/Xsession.d/45xsession (minirt.gz 内の accel/\* から起動時に複製)

などが挙げられる。

例えば、デスクトップ環境や、言語環境を変更するだけあれば、isolinux.cfg を変更するだけで良い。しかし、もう少し込み入ったことをするときには、knoppix-autoconfig や 45xsession 等を編集する必要がある。このようなファイルは元々は cloop 内にあったため、設定を変更するたびに cloop 圧縮を行う手間が必要であった。最近の KNOPPIX では、UNIONFS や AUFS といったスタックファイルシステムを採用し、CD/DVD から起動しているにも関わらず仮想的にファイルを上書きすることも可能になっている。これを利用して、最近の高速起動版 KNOPPIX では重要な設定ファイルは minirt.gz 内に収録されており、起動時にスタッ

クファイルシステムを利用して元々の場所に上書きする形式を取っている。この方式を採用したことにより、設定を変更するたびに圧縮する手間が省け、容易に設定を煮詰めることができるようになった。

それでは、/cdrom/boot/isolinux/minirt.gz 内に含まれている linuxrc, knoppix-autoconfig を調べてみる。minirt.gz を適宜作業ディレクトリに複製し、gzip を用いて圧縮ファイルを展開する。

```
# cp /cdrom/boot/isolinux/minirt.gz .
# gzip -d minirt.gz
```

展開されたファイル minirt を /media/test に loop デバイスとしてマウントする。

```
# sudo mount -t ext2 -o loop minirt /media/test
```

マウントした minirt の中身を見てみる。特に大事なのが、linuxrc と accel/\* である。

```
# cd /media/test
# ls
KNOPPIX/ boot@ etc/ linuxrc.orig* mnt/ proc/ sys/ accel/ cdrom/ lib@ lost+found/
modules/ sbin@ tmp/ bin@ dev/ linuxrc* media/ opt@ static/ usr@
```

linuxrc の後半では、accel/\* を本来の場所に複製する作業が行われている。そこで、accel ディレクトリ内を観察する。

```
# ls accel
accel:
45xsession* accton* cloopreadahead* knoppix-autoconfig* xsession* 45xsession_bt*
bootchartd* inittab mkxorgconfig*
```

特に knoppix-autoconfig, 45xsession 等のシェルスクリプトは、一度は通読して欲しい。起動時に knoppix-autoconfig は/etc/init.d/knoppix-autoconfig に上書きされて起動スクリプトとして実行される。また、45xsession は /etc/X11/Xsession.d/45xsession に上書きされ、X Window System 起動後の初期スクリプトとして実行される。なにか環境設定を変更したいときは、ほとんどの場合、この2つのファイルを熟読し、適宜、変更することで実装できる。また、起動時の様子については KNOPPIX Temporary Directory [15] などで詳しく解説されており参考になる。

## 7 容量問題

DVD は最大 4.7GB まで収録できるが、CD は最大でも 700MB の容量しかない。インストールを繰り返して KNOPPIX の cloop ファイルを作成し、ISO イメージを構築してみると 700MB を越えていることに気づくことがある。ついつい、食べすぎた後に体重計に乗ってみると思っていた以上に体重が増えていたようなもので、このような場合には CD のサイズに収めるためのダイエットが必要である。<sup>3)</sup> 目標体重(容量)は 690MB。700MB 以内であれば、CD の ISO イメージを作成することはできる。しかし、経験則として、どの PC でも起動できるようにということを考えると、理想的な体重は 690MB 以内に収めた方が良いでしょう。

<sup>3)</sup> 実際のダイエットと同じで、食べるものを選択するという強い意志が必要である。

まずは、不必要なパッケージを削除することを考える。不必要なパッケージと言っても、どのパッケージが、どのような役割を果たしているかということを知ることは意外に難しいものである。このような作業には、普段から Linux デスクトップでの経験を積んでおくことが必要である。最終的には、パッケージの内容を閲覧する命令、

```
## apt-cache show <package_name>
```

を行って、一つ一つの解説を読み、判断しなければならない。しかし、多数あるパッケージを一つ一つ調べて、削除していくのは、あまりに効率が悪い。そこで、自分が再構築したいシステムの編集方針にもよるが、一般に次の Section: devel, games, graphics, net, sound, ... などからパッケージを削除していくのが良いだろう。この際に、有益なのが、dpkg-query である。

```
## dpkg-query -W --showformat=?{Section} ${Package}\n? | sort -n
```

また、大きなパッケージを削除することも重要である。パッケージのインストールサイズ順に並べるためには、次のような命令を用いる。

```
## dpkg-query -W --showformat=?{Installed-Size} ${Package}\n? | sort -n
```

パッケージを削除する際に、注意しなければいけないのが、パッケージの依存性である。あるパッケージを削除すると、それに引きずられて他のアプリケーションまで削除されることがある。特にライブラリを消す場合には、あらかじめ逆依存性 (reverse depends) を調べると良い。

```
## apt-cache rdepends libwine | sort | uniq
```

また、KNOPPIX には、他のパッケージから依存されていないパッケージを探す命令 deborphan が収録されている。

```
## deborphan -z | sort -n
```

さて、不必要なパッケージを全て削除しても、まだ希望体重をクリアできていない場合にはどうするか。まずは、APT のデータベース /var/lib/apt/lists/\* を削除しよう。

```
## cd /etc/apt
## mv sources.list sources.list.math
## touch sources.list
## apt-get update
```

APT はネットワーク上からパッケージの情報を取得し、その情報を /var/lib/apt/lists/\* に蓄えている。この情報は意外に容量があり、これを削除するだけでも、かなりのダイエットを行うことができる。

さて、ここまでのダイエット法は、必要であれば比較的容易に元の状態に戻すことができる方法であった。これ以降に解説する方法は、APT で管理されているパッケージ情報から逸脱しているので、適用する場合には注意が必要である。

まず、第一の方法は、不必要な言語情報ファイル (LOCALE ファイル) を削除することである。これには、localepurge というパッケージをインストールすれば良い。必要な言語情報ファイルを設定することで、インストールやアップグレードを行う際に、自動的に不必要な言語情報ファイルの削除を行ってくれる。

また、不必要なドキュメントを削除することも一つの方法である。デスクトップ環境 KDE のドキュメント等を個別に削除することで、ダイエットを行うことができる。しかし、このような方法は APT によるパッケージ管理を逸脱しており、様々な問題を生じる恐れがあるので、お勧めしない。

## 8 Debian Package の簡易作成

Debian GNU/Linux は世界的なオープンソースソフトウェア活動であり、膨大な数の Debian Package が提供されている。数学ソフトウェアも豊富に提供されているが、世界中で開発されている数学ソフトウェア全てが Debian パッケージになっているわけではない。

Debian パッケージになっていないソフトウェアをインストールする際に、`/usr/local` 等に直接バイナリを展開するという選択肢もある。しかし、追加インストールするソフトウェアが少数であれば良いが、数が増えるにつれて管理が難しくなる。Debian パッケージを作ることは技術的に難しそうで躊躇するかもしれない。そこで、Debian の公式パッケージとしての体裁は満たさないが、簡易な方法で作成する方法を紹介する。

まずは、作業ディレクトリ `work` 内で `hoge` というパッケージを作成することを考える。かなり極端な例であるが、`work` 内のディレクトリ構成は次のような木構造で構成されているとしよう。

```
etc/hoge
usr/bin/hoge
usr/share/doc/hoge/*
usr/lib/hoge/*
```

カレントワーキングディレクトリを `tar` でアーカイブ作成し、`alien` でパッケージを作成する。

```
# cd work
# tar cvzf hoge-0.0.1.tar.gz .
# alien -d hoge-0.0.1.tar.gz
```

この作業によって Debian パッケージ `hoge_0.0.1-2_all.deb` が作成される。このままでもインストールは可能であるが、`copyright` 等を編集したい場合には、次のような命令を実行する。

```
# alien -d -g hoge_0.0.1-2_all.deb
# cd hoge-0.0.1/debian
```

`hoge-0.0.1/debian` 内の `changelog`, `control`, `copyright` などのパッケージ情報を編集し、

```
# cd ..
# make -f debian/rules binary
```

を実行すれば、パッケージ情報が更新された Debian パッケージが構成される。Debian パッケージを作成することで、容易にインストール、アンインストールを行うことができる。

ここで紹介した作成方法は、あくまで簡易的な方法であるので、正式な Debian パッケージを作成する場合には、Debian ユーザ文書にある開発者向けマニュアルを参照して欲しい。 <http://www.debian.org/doc/> から辿れる「Debian 新メンテナガイド」[17]、「Debian 開発者用リファレンス」[18]などが参考になる。また、最近ではパッケージに特化した書籍も複数出版されているので、そちらも参考にしてほしい。[19], [20].

## 9 Debian Menu System

KNOPPIX は KDE というデスクトップ環境を採用している。KDE により、ユーザはメニュー形式の GUI により、容易に目当てのアプリケーションを探すことができる。しかし、Debian 公式パッケージであっても、KDE のメニューが登録されているかどうかは、各パッケージに依存している。また、自分で簡易パッケージを作成した場合には、自分でメニューに登録する必要がある。そこで、この章では、主に KDE を対象として“Debian Menu System”[21]に基づいて、メニューシステムの作成方法について解説する。

### 9.1 メニューの作成

KDE ではデスクトップ最下段に Kicker と呼ばれるメニューシステムが表示される。まずは、Kicker のボタンアイコンに登録する方法だが、これは、`/etc/skel/.kde/share/config/kickerrc` を編集する。

また、KDE のメニューシステムだが、現在のところ、KNOPPIX/Math では `/usr/share/menu/*` に各パッケージごとにメニュー情報を置く方法を採用している。これは、`menu` という Debian パッケージに依存しており、`updatemenus` という命令で `/var/lib/menu-xdg/applications/menu-xdg/*.desktop` を作成し、KDE のメニューとして登録を行う。この方法の長所としては、他の WindowManager のメニューと併せて一括して管理することが可能であることだ。一方、短所としては、既存のパッケージで作成されているメニューに対して上書きを行っている場合、`apt-get update` を行うことで、公式 Debian パッケージによって、さらにメニューシステムが上書きされる可能性がある。また、公式 Debian パッケージの中には、`/usr/share/menu/*` を採用せずに `/usr/lib/menu/*` や `/usr/share/applications/*.desktop` によってメニューに登録しているアプリケーションもあり、二重にメニューに登録されたり、繁雑になる可能性がある。

そこで、KNOPPIX 日本語版の派生版として有名な KNOPPIX Edu 等は別の方法を採用している。`/etc/skel/.local/share/applications/*.desktop` を編集することで、ホームディレクトリに書き込まれる設定ファイルのカスタマイズを行っている。この方法の長所としては、公式 Debian パッケージと衝突しにくいこと、`apt-get update` 等のアップデートに強いこと。短所としては、Window Manager ごとに設定が必要となる可能性があることである。しかし、実際問題として KDE 以外のデスクトップ環境を起動時に選択して使うケースが少ないことを考えると、今後、KNOPPIX/Math についても同様の方法を検討して良いのではないかと考えている。

ところで、KDE を使う場合、要求される PC のスペックが高いことが問題となる。今後、Xfce 等の軽量デスクトップを用いた構成も期待されるが、USB メモリーディスクの処理や、プリンタの登録等、解決しなければいけない問題も多い。

### 9.2 アイコンの作成

アイコンについても、Debian では仕様が決められている。しかし、全てのアプリケーションで対応しているわけではないようである。また、数学ソフトウェアに関して特に言えることだが、コマンドラインベースのインターフェースしか用意されていないため、アイコンが作成されていない場合がある。メニューシステム上、アイコンがなくても、アプリケーション名だけで特に支障はないが、やはりアイコンがあるのと、ないのとではデザイン面での統一感が大きく異なってくる。そこで KNOPPIX/Math では、アイコンが存在しない数学ソフトウェアについては独自にアイコンを作っている。アイコンを作るためのアプリケーションとしては、KIconEdit が存在し、比較的簡単にアイコンを作成することができる。“Debian Menu System”で定められている指針によると、xpm 形式でサイズは  $32 \times 32$  を越えないこと、可能であれば背景は透明化することなどが定められている。

## 10 今後の課題

現在、収録するソフトウェアを厳選した KNOPPIX/Math/2007 CD の作成を計画中である。DVD 化によって、ファイル容量による制限を考慮する必要がなくなったが、一方でネットワーク上から気軽に取得することが難しくなった。また、一世代前の PC での実行を考えた場合、CD からの起動手段を用意しておくことは十分な価値があると思われる。しかし、ここ数年で KDE を始めとするデスクトップ環境の肥大化は深刻であり、今後、ますますダイエットに頭を悩ませることが予想される。そろそろ、他のデスクトップ環境を検討する時期にきているのかもしれない。また、これまで様々な数学ソフトウェアを KNOPPIX/Math に収録してきたが、Debian GNU/Linux の公式な基準に対応できるような dpkg を構築しているとはいえない。今後の更新を考えると、もう一度、作成手順を見直し、公式パッケージを維持できるような体制を目指す必要がある。KNOPPIX/Math に収録するソフトウェアが増大するにしたがって、そのテスト環境が問題となっている。数学ソフトウェアに特化したデスクトップ環境をどのようにテストし、リリースしていくかの方針も議論を進めていかなければならない。

現在、KNOPPIX/Math Project では、開発者向け、ユーザ向けのメーリングリストを開設している。我々のプロジェクトに興味を持った方は積極的にメーリングリストに参加して欲しい。メーリングリストへの参加方法は、<http://www.knoppix-math.org/> から参照できる。

- [1] KNOPPIX/Math, <http://www.knoppix-math.org/>
- [2] KNOPPIX, <http://www.knopper.net/knoppix/>
- [3] KNOPPIX 日本語版, <http://unit.aist.go.jp/itri/knoppix/>
- [4] KNOPPIX Edu, <http://www.knoppix-edu.org/>
- [5] 小菅貴彦, USB-KNOPPIX, <http://kosuge.or.tp/kserv/>
- [6] 二階堂愛, KNOB, <http://knob.sourceforge.jp/>
- [7] Debian GNU/Linux, <http://www.debian.org/>
- [8] APT HOWTO, <http://www.debian.org/doc/manuals/apt-howto/>
- [9] Kyle Rankin(著), クイープ(訳), 須崎有康(監修) KNOPPIX HACKS - カスタマイズとシステム管理のテクニック, オライリージャパン, 2005, ISBN 978-4873112282
- [10] Kinneko, Build Own Knoppix, <http://sourceforge.jp/projects/ya-knoppix-jp/document/BOK/ja/1/BOK.txt>
- [11] 長坂耕作, 神戸大学 数理・情報総合演習授業資料, <http://wwwmain.h.kobe-u.ac.jp/nagasaka/lecture/2005/remastering-open.pdf>
- [12] Singular, <http://www.singular.uni-kl.de/>
- [13] (株) アルファシステムズ, KNOPPIX 起動高速化適用マニュアル, <http://sourceforge.jp/projects/lcat/>
- [14] 山城潤, SYON テクニカル: LCAT を利用した KNOPPIX の高速化, <http://www.syon.co.jp/syontech/>
- [15] 柘植昭秀, KNOPPIX Temporary Directory, <http://buxus.s31.xrea.com/pukiwiki/>
- [16] 志子田有光, CheatCode (チートコード) の設定, <http://www.eng.tohokugakuin.ac.jp/knoppix/cheatcode.html>

- [17] Josip Rodin, Debian 新メンテナガイド, <http://www.debian.org/doc/devel-manuals#devref>
- [18] Ian Jackson, Christian Schwarz, Adam Di Carlo, Raphaël Hertzog, Josip Rodin, Debian 開発者用リファレンス, <http://www.debian.org/doc/devel-manuals#maint-guide>
- [19] やまだあきら (監修), 鶴飼文敏 (著), 入門 Debian パッケージ, 2006, 技術評論社, ISBN 978-4774127682
- [20] 西村めぐみ, らぶらぶ Linux 〈3〉「もっと知りたい」人のためのパッケージ管理&構築入門, ソシム, ISBN 978-4883371914
- [21] Joost Witteveen, Joey Hess, Christian Schwarz, Debian Menu System, <http://www.debian.org/doc/devel-manuals#menu>