1CD Linux + Sapid

小川 順平

平成 17 年 6 月 1 日

1 はじめに

自分の環境に Sapid をインストールしようとすると，Sapid をコンパイルするために予め環境を整えなければならない．例えば，Windows では Cygwin が必須であり，一般のユーザがこれをインストールしているとは限らない．また Linux でも，OS やユーザによりインストール済みのパッケージが異なり，Sapid のインストールが容易にできるとは限らない．

以上のような，問題を解決する方法として 1CD Linux がある．1CD Linux には以下の特色がある．

- ハードディスクやマシン一式を交換しても，CD さえあれば同じシステムが起動できる．
- 設定が変更されても，再起動するだけで元の設定で確実に動作する．

この特色を利用して，環境に依存するという問題を打開することができる．

そこで今回，環境に依存することなく Sapid を使用できる，Sapid インストール済みの 1CD Linux を作成する．これを利用することで，いつでも Sapid の環境の整った Linux システムを CD-R で起動させることができる．また勉強用としてや，不特定多数の人が触る端末においても安定した状態を保つことができる．

このドキュメントは 1CD Sapid を作成する際にスムーズになるようにマニュアルを提供する．具体的には，KNOPPIX を元に再構成してオリジナル Linux を作成する，リマスタリングという手順の途中に Sapid のインストールする．

マニュアル作成するにあたって，文献 [1] を参考にした．

2 目次

1. 必要環境と準備

   リマスタリングするに当たって，ハードディスク容量 4GB(ext2) と 1GB(SWAP) を必要とする．この準備について述べている．既に，条件が整っている方は読み飛ばしてもよい．また，ネットワーク設定についても述べている．

2. 作業環境の展開と準備

   KNOPPIX から必要なデータをハードディスクに展開する．

3. パッケージの追加と削除

   Sapid に最低限必要な環境を整える．ここで，他に必要なパッケージをインストールする．iso ファイルの容量削減のため不必要なパッケージを削除する．

4. Sapid のインストール

   SPIE を動作することを目的に Sapid をインストール．詳しくは，Sapid インストールマニュアルを参考に．

5. その他の調整

   無駄なファイルの削除．環境設定の保存など．

6. CD-ROM イメージの作成

   iso ファイルの作成．

7. 動作テスト

   SPIE の動作確認．

この文章は，1CD Linux KNOPPIX を用いて，Sapid をインストール済みの KNOPPIX ベースのカスタマイズ 1CD Linux を作成する手順を示す．

KNOPPIX の入手は，

`http://unit.aist.go.jp/itri/knoppix/index.html`

にてダウンロードする．

今回使用したものは

`knoppix_v3.8.1_20050408-20050415.iso`

である．

また，動作確認した計算機は，IBM ThinkPad T40, Sony vaio VGN-S73PB, Panasonic Let's note Y2 などが挙げられる．KNOPPIX が動作する環境であれば，問題なく動作すると思われる．動作しなかった計算機は，日立 FLORA 270W NA1 ．

`http://unit.aist.go.jp/itri/knoppix/hardware/index.html`

ここに KNOPPIX の動作確認を行った公式リストがあるので，ここに載っている計算機は動作すると思われる．

3 前準備

3.1 ハードウェア環境条件

CPU と RAM CPU は KNOPPIX が動作すれば問題ない．RAM と Linux SWAP の合計が 1GB 以上必要である．

ハードディスク リマスタリングの作業には，Windows の FAT や NTFS 領域を用いることができないため，データ格納や作業用の Linux(ext2)4GB 以上, 上記の SWAP 領域 1GB 程度の 2 つを用意する．実際のパーティショニングについては節 3.2 で述べる．

3.2 パーティショング

Windows の FDISK.EXE では，FAT や NTFS しか理解できないので，商用のパーティショニングソフトを使用してもよいが，KNOPPIX 側からでも `cfdisk` で切ることができる．

まず，KNOPPIX を通常通り起動させる．パーティショニングには root 権限がないと実行できないため，次に画面左下から 2 番目の「ぺんぎん」メニュー→「Root Shell」を立ち上げるか，Konsole から sudo -s として，root になってパーティショニングを行う．以降の説明で，`[home]$`は KNOPPIX の root シェルプロンプトを表す．

```
[home]$ cfdisk
```

これでハードディスクに対するパーティショングを開始する．

まず，節 3.1 で述べた作業領域のために 5GB の空き領域を確保する．次に開き領域をカーソルでハイライト表示させた状態で「新規作成」を選択すると，領域の種別を着替えるので，基本領域を選択する．次に，作成するパーティションのサイズを聞かれる．`cfdisk` は MB 単位で入力するようになっていて，ここでは 4GB を ext2 にするので，4000 と入力しリターンをする．そのパーティションの置き場所は「最初から」を選択する．ここまで進むと，hda1 として 4GB のパーティションが作成される (名前が違う場合は読み替えるように)．この方法を繰り返し，Linux SWAP パーティション用の領域 (hda2) も確保する．

次に，パーティションのファイルシステム (FS) タイプを変更する．hda1 は既に「Linux」となっているので変更の必要はない．hda2 については Linux SWAP を割り当てたいので，hda2 にカーソルを合わせた状態で「FS タイプ」を選択する．すると，FS タイプの一覧が表示されるので Linux SWAP の数値 (82) を入力して FS タイプを変更する．

実際にハードディスクに対して行うために「書き込み」を選択して実際の書き込みを行う．作業途中でミスがあれば,「終了」を選択して最初からやり直す．

3.3 パーティションの初期化

パーティションの初期化を行う．

ルートシェルから

```
[home]$ mke2fs /dev/hda1
[home]$ mkswap /dev/hda2
```

と入力する．mke2fs コマンドで Linux パーティションの初期化，mkswap コマンドで Linux SWAP パーティションの初期化を行っている．

ここまでの作業が終わったら，一旦 KNOPPIX を再起動する．

3.4 ネットワーク環境の整備

リマスタリングを行う際のパッケージの追加や削除はインターネットを経由しなければならない．そこで，起動した KNOPPIX において，ネットワーク接続が必要である．

DHCP が接続の場合には，本節の作業は必要ないが，固定 IP アドレスを利用している場合は，以下のような接続設定を行う．ここでは LAN または ADSL の説明だけを行う．ISDN や PPPoE などの接続設定は別の方法で行う．

接続設定は，画面左下から 2 つ目の「ぺんぎん」メニュー→「Network/Internet」→「ネットワークカードの設定」で行う．設定は対話形式で，DHCP の有無，IP アドレス, ネットワークマ

スク, ブロードキャストアドレス, デフォルトゲートウェイ, ネームサーバの順で入力する．なお，ここで入力された情報は，

/etc/dhcpc/resolv.conf

/etc/network/interfaces

の 2 つのファイルに格納されている．

4　作業環境の展開と準備

以下，作業領域 (Linux パーティション) を/dev/hda1，SWAP 領域を/dev/hda2 と仮定して説明する．

4.1　データの展開とコピー

リマスタリングに関するこの後の全ての作業は root アカウントで実施するので，デスクトップが立ち上がったらルートシェルを起動する．

まず，Linux 領域をマウントする．

```
[home]$ mount /dev/hda1 /mnt/hda1
```

KNOPPIX システム全体をハードディスクに展開するために，作業領域に source と master の 2 つのディレクトリを作成する．

```
[home]$ mkdir -p /mnt/hda1/source/KNOPPIX
[home]$ mkdir -p /mnt/hda1/master/KNOPPIX/KNOPPIX
```

次に，起動している KNOPPIX からリマスタリングに必要なファイルをコピーする．まず cloop ファイルの中身を source ディレクトリにコピーする．数十分程度かかる．

```
[home]$ cp -Rpv /KNOPPIX/* /mnt/hda1/source/KNOPPIX
```

起動用のイメージファイルやドキュメント類を master ディレクトリにコピーする．

```
[home]$ cp -p /cdrom/KNOPPIX/*.* /mnt/hda1/master/KNOPPIX/KNOPPIX
[home]$ cp /cdrom/*.* /mnt/hda1/master/KNOPPIX
[home]$ cp -Rpv /cdrom/boot /mnt/hda1/master/KNOPPIX
```

もしリマスタリングした KNOPPIX で QEMU や coLinux などを実行したい場合には，それらのディレクトリもコピーする．

```
[home]$ cp -Rpv /cdrom/coLinux /mnt/hda1/master/KNOPPIX
[home]$ cp -Rpv /cdrom/qemu-0.6.0-windows /mnt/hda1/master/KNOPPIX
```

4.2　chroot とネットワーク設定

/mnt/hda1/source/KNOPPIX に展開されたファイル群は，Debian のパッケージ管理システムで管理された状態になっている．その状態を壊さずに作業をしなければならないので，上記ディレクトリを起点 (ルートディレクトリ) として chroot する．更に，proc ファイルシステムをマウントする．以下，chroot したシェルプロンプトを，区別のため [root]#と表す．

```
[home]$ chroot /mnt/hda1/source/KNOPPIX
[root]# mount -t proc /proc proc
```

chroot した環境は，下のシステムの環境を一部引き継ぐものの，ルートディレクトリの位置が変わってしまったため，設定ファイルの内容が異なる．そのため，元々の KNOPPIX で設定したネットワーク関連の設定が利用できない．そこで，そのファイルをコピーするのだが，chroot したルートシェルからは，元々の KNOPPIX 側のファイルを見ることができない．そこで，もう一つ別のルートシェルを開き，ネットワーク関連の設定ファイルをコピーする．

まず，chroot した側で，ファイルのバックアップを取る．

```
[root]# cd /etc/dhcpc
[root]# mv resolv.conf resolv.conf.org
[root]# cd ../network
[root]# mv interfaces interfaces.org
```

次に，新しく開いたルートシェルから，設定ファイルをコピーする．

```
[home]$ cd /etc/dhcpc
[home]$ cp resolv.conf /mnt/hda1/source/KNOPPIX/etc/dhcpc/
[home]$ cd ../network
[home]$ cp interfaces /mnt/hda1/source/KNOPPIX/etc/network/
```

その後，chroot した側でインターネットへの接続ができるかどうか，

```
[root]# host www.google.co.jp
```

などで確認する．もし接続できなければ，インターフェイスを再起動する．

```
[root]# ifdown eth0
[root]# ifup eth0
```

5　パッケージの追加と削除

5.1　パッケージの追加

5.1.1　管理情報のアップデート

Debian で用いるパッケージ管理情報のアップデートを行う．apt-get update コマンドで更新を行う．ただし，http/ftp 接続にプロキシサーバを経由する場合には，以下のように環境変数を設定しておく．

```
[root]# export http_proxy=http://サーバ名:ポート名/
[root]# export ftp_proxy=http://サーバ名:ポート名/
```

実際の更新は次のように行う．

```
[root]# apt-get update
```

読み込み時のタイムアウトなどで,「E:いくつかのインデックスファイルのダウンロードに失敗しました．無視されたか，あるいは古いものが使用されました.」と表示されて終了した場合は，繰り返し apt-get update を正常終了するまで行う．

5.1.2　パッケージの追加

Debian パッケージ化されているアプリケーションの追加は,「apt-get install パッケージ名」でインターネットからダウンロード→インストールされる．今回は，Sapid のインストールを目的としているので，特に必要なパッケージはない．必要なパッケージがある時は，この段階でインストールする．

5.2 パッケージ削除作業

Sapid を利用するには不必要だと考えられるパッケージ，または自分が不必要だと思うものを削除して，/mnt/hda1/source/KNOPPIX/の容量を 1.8GB 程度に減らすことが目的である．今回は，wine, OpenOffice, mozilla, kdevelop, scribus を削除すれば，1.8GB まで減らせる．また，flex パッケージも削除する．Sapid のビルドで必要とするが，最新版を利用したいので，ここで古いものを削除しておく．

パッケージの削除の例として wine を削除する様子を示す．

最初に wine 関連パッケージがどの程度あるのかを調べる．

```
[root]# dpkg -l | grep wine
```

この出力を見ると，ベースとなるライブラリが libwine のようなので，これを消すとまとめてどれだけ消されるかシュミレートする．

```
[root]# apt-get -s remove --purge libwine
```

winesetuptk パッケージが消されないので，これを付けて再シュミレートする．

```
[root]# apt-get -s remove --purge libwine winesetuptk
```

これで全て消去できるようなので，実際の作業を行う．消去途中で消してもいいか聞かれる際は，デフォルトが Yes なのでそのままリターンをする．そのときに，開放するディスク容量が表示されるので，パッケージ削除の参考にできる．

```
[root]# apt-get remove --purge libwine winesetuptk
```

このように，他のパッケージについても，最初に関連パッケージについて調べ，依存関係をシュミレートした上で削除するとよい．

以下に著者が削除したパッケージの一覧を記述する．

- wine

  libwine, libwine-print, libwine-twain, wine, wine-utils, winesetuptk

- mozilla

  mozilla-firefox, mozilla-firefox-locale-de-de, mozilla-firefox-locale-ja, mozilla-thunderbird, mozilla-thunderbird-locale-de, mozilla-thunderbird-offline

- kdevelop

  kdevelop3, kdevelop3-plugins, kdevelop3-data

- scribus

  scribus

- openoffice

  openoffice.org, openoffice.org-bin, openoffice.org-debian-files, openoffice.org-help-ja, openoffice.org-l10n-en, openoffice.org-l10n-ja, openoffice.org-l10n-ja-knoppix, openoffice.org-mimelnk, ttf-openoffice

- flex

  flex

5.3 パッケージ関連のクリーンアップ

パッケージ削除作業を通じて，他から参照されなくなったライブラリパッケージが出る可能性がある．そのようなパッケージの有無をチェックする．

```
[root]# deborphan
```

ここで出力があった場合，これらを削除する．

```
[root]# deborphan | xargs apt-get -y remove --purge
```

次に，インストールしたパッケージそのものを消す．パッケージの中身は既にインストールされたが，インターネットからダウンロードしたパッケージは/var/lib/apt/archives/ディレクトリに保存されている．そこで，apt-get clean コマンドで削除する．

```
[root]# apt-get clean
```

5.4 開発環境のインストール

Sapid をインストールするための環境を整えるための作業を行う．このマニュアルは Sapid の CASE ツール SPIE を動作させるのを目的にしている．その他のツールを使用したい場合は，巻末の付録を参考にパッケージをインストールして設定するように．

5.4.1 Java のインストール

http://java.sun.com/j2se/から最新版の Java 2 SDK Standard Edition (J2SE) をダウンロードする．今回は，jdk-1_5_0_03-linux-i586.bin をダウンロードしたとする．このファイルを chroot した側から見えるところに置く．

ブラウザからダウンロードした際の UID は root のものではないでので，とりあえず/home ディレクトリなどにおいて，その後 root 権限で目的の箇所にファイルを移動させるとよい．

bin ファイルは，実行したカレントディレクトリにインストールするので，インストールしたい場所に置く．例えば，/mnt/hda1/source/KNOPPIX/usr/local/java/に置いたとする．このディレクトリは予め作っておいた．インストールは以下のように行う．

```
[root]# cd /usr/local/java
[root]# chmod u+x jdk-1_5_0_03-linux-i586.bin
[root]# ./jdk-1_5_0_03-linux-i586.bin
```

すると，ライセンス承諾について聞かれるので yes と入力すると，/usr/local/java/jdk1.5.0_03 に Java がインストールされる．

環境変数 JAVA_HOME を設定する．

```
[root]# export JAVA_HOME=/usr/local/java/jdk1.5.0_03
```

また，先ほどインストールした Java のコマンドに PATH を通す．java, javac コマンドは KNOPPIX が最初からサポートしている gcj で扱っているが，Java のコマンドを優先したいので先に PATH を通す．

```
[root]# export PATH=/usr/local/java/jdk.1.5.0_03/bin:$PATH
```

5.4.2 flex のインストール

ftp://ftp.gnu.org/non-gnu/flex/から最新版の flex ソースをダウンロードする．今回は，flex-2.5.4a.tar.gz をダウンロードした．これも同様に chroot した側から見えるところに置く．例えば，/mnt/hda1/source/KNOPPIX/home/に置いたとする．インストールは以下のように行う．

```
[root]# cd /home
[root]# tar xvzf flex-2.5.4a.tar.gz
[root]# cd flex-2.5.4
[root]# ./configure --prefix=/usr/local
[root]# make
[root]# make install
```

これで，/usr/local/bin 以下に flex がインストールされた．

5.4.3 SWIG のインストール

http://jaist.dl.sourceforge.net/sourceforge/swig/から最新版の SWIG ソースをダウンロードする．今回は，swig-1.3.24.tar.gz をダウンロードした．これも同様に chroot した側から見えるところに置く．例えば，/mnt/hda1/source/KNOPPIX/home/に置いたとする．インストールは以下のように行う．

```
[root]# cd /home
[root]# tar xvzf swig-1.3.24.tar.gz
[root]# cd SWIG-1.3.24
[root]# ./configure --prefix=/usr/local
[root]# make
[root]# make install
```

これで，/usr/local/bin 以下に SWIG がインストールされた．

5.4.4 Ruby のインストール

KNOPPIX は Ruby1.8 をサポートしているが，Sapid をビルドする際に不具合が生じるので，改めてインストールすることにする．ftp://ftp.ruby-lang.org/pub/ruby/から最新版の Ruby ソースをダウンロードする．今回は，ruby-1.8.2.tar.gz をダウンロードした．これも同様に chroot した側から見えるところに置く．例えば，/mnt/hda1/source/KNOPPIX/home/に置いたとする．インストールは以下のように行う．

```
[root]# cd /home
[root]# tar xvzf ruby-1.8.2.tar.gz
[root]# cd ruby-1.8.2
[root]# ./configure --prefix=/usr/local
[root]# make
[root]# make install
```

これで，/usr/local/bin 以下に Ruby がインストールされた．

5.4.5 Tcl の環境

KNOPPIX は Tcl8.4 をサポートしているが，Sapid をビルドするためのヘッダファイルやライブラリがない．そのため，これらを手に入れる必要がある．今回はソースから必要なものをコピーすることにする．http://jaist.dl.sourceforge.net/sourceforge/tcl/から最新版の Tcl ソースをダウンロードする．今回は，tcl8.4.9-src.tar.gz をダウンロードした．これも同様に chroot した側から見えるところに置く．例えば，/mnt/hda1/source/KNOPPIX/home/に置いたとする．インストールは以下のように行う．

```
[root]# cd /home
[root]# tar xvzf tcl8.4.9-src.tar.gz
[root]# cd tcl8.4.9/generic
```

ヘッダファイルを置くディレクトリを作成する．

```
[root]# mkdir -p /usr/local/include/tcl-private/generic/
```

ヘッダファイルのコピー．

```
[root]# cp *.h /usr/local/include/tcl-private/generic/
```

ライブラリファイルを作成し，コピーする．

```
[root]# cd ../unix
[root]# ./configure --prefix=/usr
[root]# make
[root]# cp libtcl* /usr/local/lib/
```

これで，Sapid のビルドに必要な Tcl 環境は整った．

5.4.6 Tk の環境

KNOPPIX は Tk8.4 をサポートしているが，Sapid をビルドするためのヘッダファイルやライブラリがない．そのため，これらを手に入れる必要がある．今回はソースから必要なものをコピーすることにする．http://jaist.dl.sourceforge.net/sourceforge/tcl/から最新版の Tk ソースをダウンロードする．今回は，tk8.4.9-src.tar.gz をダウンロードした．これも同様に chroot した側から見えるところに置く．例えば，/mnt/hda1/source/KNOPPIX/home/に置いたとする．インストールは以下のように行う．

```
[root]# cd /home
[root]# tar xvzf tk8.4.9-src.tar.gz
[root]# cd tk8.4.9/generic
```

ヘッダファイルを置くディレクトリを作成する．

```
[root]# mkdir -p /usr/local/include/tk-private/generic/
```

ヘッダファイルのコピー．

```
[root]# cp *.h /usr/local/include/tk-private/generic/
```

ライブラリファイルを作成し，コピーする．

```
[root]# cd ../unix
[root]# ./configure --prefix=/usr
[root]# make
[root]# cp libtk* /usr/local/lib/
```

これで，Sapid のビルドに必要な Tk 環境は整った．

5.4.7 その他の開発環境

ここまでの作業で Sapid のビルドに必要な環境は整えることができた．他に，開発環境として必要なものは，各自この段階でインストールするように．

6 Sapid のインストール

6.1 ダウンロード

http://www.sapid.org/FTP から，Sapid の最新リリースをダウンロードし，展開する．その後で，<SAPID_TOP> に移動する．これ以降，インストールが終るまでの作業は全て <SAPID_TOP> で行う．

```
[root]# tar zxvf Sapid-<RELEASE_NO>.tar.gz
```

6.2 設定ファイルの変更

<SAPID_TOP>/Sapid/SapidSite.def の内容を変更する．

- Japid(l.36), SWIG(l.92), RUBY(l.95), Tcl/Tk(l.135) をコメントアウトする．
- ruby コマンドのパスを`/usr/local/bin/ruby` に変更する (l.129).
- Tcl/Tk の設定 (l.146〜151) をコメントアウトし以下のように変更する．

  SapidTclIncDir `/usr/local/include/tcl-private/generic`

  SapidTkIncDir `/usr/local/include/tk-private/generic`

  SapidTclLibDir `/usr/local/lib`

  SapidTkLibDir `/usr/local/lib`

  SapidTclVersion `8.4`

  SapidTkVersion `8.4`

6.3 インストール

設定ファイルの変更を行ったら，<SAPID_TOP> にて以下のコマンドを実行して Sapid のインストールを行う．

```
[root]# xmkmf -a
[root]# make
[root]# make install
```

これで，`/usr/local/Sapid` に Sapid がインストールされる．実際にコマンドが使用するために，以下のコマンドを実行する．

```
[root]# source /usr/local/Sapid/lib/SetUp.sh
```

すると，sdb4-cc などのコマンドを使用できるようになる．

7　その他の調整

7.1　chroot 終了前の作業

chroot を抜ける前に，最後の作業が必要である．

まず，ネットワーク設定の復元を行う．最初にバックアップを取ったファイルを元に戻す．

```
[root]# cd /etc/dhcpc
[root]# mv resolv.conf.org resolv.conf
[root]# cd ../network
[root]# mv interfaces.org interfaces
```

次にパッケージの追加/削除で用いたパッケージデータベースのうち，特に必要でないものを削除する．

```
[root]# cd /var/lib/dpkg
[root]# rm *-old
[root]# cd /var/lib/apt/lists
```

lock と partial 以外のファイルを削除する．

```
[root]# rm security*
[root]# rm ftp*
[root]# rm ndiswrapper*
[root]# ls
lock partial
```

次に，KNOPPIX を立ち上げてすぐに Sapid を使えるようにする設定を行う．/etc/skel には KNOPPIX デスクトップ起動時に用いられる各種初期設定ファイルの雛形がある．ここにある .bashrc ファイルが，ホームディレクトリにコピーされて使われているので，このファイルの最後に以下の内容を加える．

```
[root]# cd /etc/skel
[root]# emacs -nw .bashrc
```

以下の 7 行を付け加える↓

```
export JAVA_HOME=/usr/local/java/jdk1.5.0_03
export PATH=/usr/local/java/jdk1.5.0_03/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/lib
export DISPLAY=127.0.0.1:0.0
if [ -f "/usr/local/Sapid/lib/SetUp.sh" ]; then
.  "/usr/local/Sapid/lib/SetUp.sh"
fi
```

これが終われば，proc ファイルシステムをアンマウントし，chroot 環境を抜ける．

```
[root]# umount /proc
[root]# exit
[home]$ ← 抜けた
```

7.2　不要なファイルの削除

chroot を抜けた後で，これまでの作業履歴を消す．

```
[home]$ rm /mnt/hda1/source/KNOPPIX/home/knoppix/.bash_history
```

KNOPPIX のシステム自体に関する作業は終了したが，それ以外の部分で iso イメージ作成に影響のないものを必要に応じて削除する．

8 CD-ROM イメージの作成

リマスタリング作業は一発で希望通りになるとは限らず，何度かバージョンアップが必要になることがほとんどである．そこで CD-ROM の内容の区別が付かなくなることを防ぐため，バージョン情報の埋め込みを行う．

```
[home]$ touch /mnt/hda1/master/KNOPPIX/1CD-Sapid0521
```

1CD-Sapid0521 の部分は任意である．

次に，圧縮ルートイメージの作成を行う．KNOPPIX 本体である cloop ファイルは以下のコマンドで作成する．

```
[home]$ mkisofs -R -l -V ”1CD Sapid iso9660 filesystem” \
-hide-rr-moved -v /mnt/hda1/source/KNOPPIX | \
create_compressed_fs - 65536 > /mnt/hda1/master/KNOPPIX/KNOPPIX/KNOPPIX
```

この作成には相当な時間を要する．筆者の環境でも 30 分程度かかった．

次に，iso イメージファイルを作成する．

```
[home]$ cd /mnt/hda1/master/KNOPPIX
[home]$ mkisofs -l -r -J -V ”1CD-Sapid” -v \
-b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat \
-o ../1CD-Sapid.iso \
-no-emul-boot -boot-load-size 4 -boot-info-table /mnt/hda1/master/KNOPPIX
```

この作業は筆者の環境で 15 分程度で行うことができた．上記のコマンドを実行したら，最後の行に iso イメージのサイズが表示されている．完成した iso イメージのサイズが 700MB を超えていなければよいのでそのファイルを CD-R に焼く．もし超えていれば，パッケージの削除の作業までもどって更にサイズを減らすようにする．

なお，700MB の制限は CD-R の容量だけであるので DVD-R に焼くことを想定していれば，容量の問題はなくなる．あとは同様に iso ファイルを DVD-R に焼くだけである．

以上で 1CD Sapid の作成作業は終了である．

9 動作テスト

最後に Sapid の動作確認をする．ここで紹介するのは最低限のテストなので，詳しくは Sapid のテストセットを参照するように．

作成した 1CD Sapid を使って計算機の CD ドライブで CD ブートさせる．まずコンソールを立ち上げる．/etc/skel での設定がされていれば，SetUp.sh が読み込まれるので Sapid のコマンドを使用できる．

SPIE の動作テストをするために簡単な C ファイルを作成する．

```
[home]$ emacs sample.c
```

sdb4-cc コマンドでこの C ファイルを解析し SDB を作成する．

```
[home]$ sdb4-cc sample.c
```

SDB の内容を解析・閲覧する spie3 コマンドを実行

```
[home]$ spie3
```

そして，/SPEC/0/index.html をブラウザで開くことによって，SPIE での解析内容を閲覧することができる．

ここまで実行できれば，最低限の Sapid の動作確認ができたといえる．

10 追記

KNOPPIX を利用するにあたって，既存のハードディスクのデータにアクセスしたいときがある．そのためには，KNOPPIX のデスクトップにある HD の形をしたアイコンをまずクリックする必要がある．すると，エクスプローラが立ち上がり，ハードディスクの内容を閲覧することができる．パスをみると，/mnt/hda*/のようになっている．このパスをコンソールで入力すれば，コンソールでも通常のデータのようにアクセスすることが可能である．ただし，他 OS の領域は UID や GID の関係で読み込みしかできないので，KNOPPIX で作業したファイルを保存したい場合は，開いている Linux パーティションに書き込むか，ssh などを利用する．また，USB メモリも HD と同様に/mnt/sda*/でアクセスできる．USB を認識させるためには，`knoppix usb2` と `usb2` オプションをつける必要がある．

参考文献

[1] KNOPPIX を用いた LaTeX 環境の構築とその活用, 船越裕介
`http://www.ntt.co.jp/qos/tools/KNOPPIX-Customization.pdf`

付録

Sapid で利用できるものとそのインストールに必要な環境を表にした．

	Sapid-base	SPIE	SAT	Japid	JSPIE	SWIG/AR5	VCRGL
bison	○		○	○	○	○	
flex	○	○	○	○	○	○	○
GL							○
glut							○
Java				○	○	○	
Ruby						○	
SWIG						○	
Tcl			○				○
Tk			○				○
Tomcat					○		
xmkmf	○	○	○	○	○	○	○

表 1: 必要環境