

I386 SBC用のCFへ Linuxをインストールする方法

Copyright © 2003

Twise Labo Inc.

ボードにLinuxを インストールする方法

- CD, FDを接続して、Debianのインストールを行う方法
 - 128MB以上のCFが必要です。
 - 通常のインストーラーでインストールします。
- 別のPCにCFまたは、DOMなどを接続してインストールする方法
 - 128MB以上のCFが必要です。
 - 通常のインストーラーでインストールします。
- Linuxマシンに、CFリードライトを接続して行う方法
 - インストールするファイルを整理することで、16MB程度まで削減できます。(今回は、128MB版を使用)
 - 沢山インストールする場合に、適しています。
- WindowsのCFリーダーで書き込む方法
 - TwRwとイメージファイルをダウンロードしてください。
 - TwRwをインストールして、イメージをCFに書き込んでください。

開発機にUSB CFリーダを接続

- LANボードの確認
 - lspci,dmesgにより、使っているLANカードを確認します。
 - カーネルの入れ替え後に、LANカードが認識されなくなる場合があります。
- USB対応のカーネルへの入れ替え
 - USBの使えるカーネルに入れ替えます。
 - 2.4.18-386か2.2.20にします。
 - 2.4.18-386の方が、手順は難しいのですが、動作するCFリーダが多くなります。
- ドライバーの再設定
 - modprobe でLANカード、USBを認識させます。
 - /etc/modules にドライバーを登録します。

カーネルへの入れ替え(1)

```
WSV1:~# apt-get install kernel-image-2.4.18-386
The following extra packages will be installed:
  ash cramfsprogs initrd-tools
The following NEW packages will be installed:
  ash cramfsprogs initrd-tools kernel-image-2.4.18-386
0 packages upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 8897kB of archives. After unpacking 24.5MB will be used.
Do you want to continue? [Y/n]
Get:1 ftp://ftp.jp.debian.org stable/main ash 0.3.8-37 [71.6kB]
|
Unpacking kernel-image-2.4.18-386 (from ../kernel-image-2.4.18-386_2.4.18-5_i386.deb) ...
```

You are attempting to install an initrd kernel image (version 2.4.18-386)
This will not work unless you have configured your boot loader to use
initrd. (An initrd image is a kernel image that expects to use an INItial
Ram Disk to mount a minimal root file system into RAM and use that for
booting).
As a reminder, in order to configure lilo, you need to
add an 'initrd=/initrd.img' to the image=/vmlinuz
stanza of your /etc/lilo.conf
I repeat, You need to configure your boot loader. If you have already done
so, and you wish to get rid of this message, please put
'do_initrd = Yes'
in /etc/kernel-img.conf. Note that this is optional, but if you do not,
you'll continue to see this message whenever you install a kernel
image using initrd.
Do you want to stop now? [Y/n]n

これは、注意事項です。
/etc/lilo.confの
image=/vmlinuzの下の
Initrd=/initrd.imgを書かないと
BOOTできなくなると書いてあ
ります。
止めますかとの質問には
NOと答えます。

カーネルへの入れ替え(2)

```
/boot/initrd.img does not exist. Installing from scratch, eh?  
Or maybe you don't want a symbolic link here. Hmm? Lets See.  
I notice that you do not have initrd.img symbolic  
link. I can create one for you, and it shall be  
updated by newer kernel image packages. This is  
useful if you use a boot loader like lilo.  
Do you want me to create a link from /boot/initrd.img-2.4.18-386 to initrd.img?[Yn]Y  
A new kernel image has been installed, and usually that means  
that some action has to be taken to make sure that the new  
kernel image is used next time the machine boots. Usually,  
this entails running a ``bootloader" like SILO, loadlin, LILO,  
ELILO, QUIK, VMELILO, ZIPL, or booting from a floppy. (Some  
boot loader, like grub, for example, do not need to be run on  
each new image install, so please ignore this if you are usingsuch a boot loader).  
A new kernel image has been installed. at /boot/vmlinuz-2.4.18-386 (Size: 643kB)  
Initial rootdisk image: /boot//initrd.img-2.4.18-386 (Size: )  
Symbolic links, unless otherwise specified, can be found in /  
LILO sets up your system to boot Linux directly from your hard  
disk, without the need for booting from a boot floppy.  
WARNING  
If you are keeping another operating system or another version  
of Linux on a separate disk partition, you should not have LILO  
install a boot block now. Wait until you read the LILO documentation.  
That is because installing a boot block now might make the other  
system un-bootable. If you only want to run this version of Linux,  
go ahead and install the boot block here. If it does not work, you  
can still boot this system from a boot floppy.  
  
You already have a LILO configuration in /etc/lilo.conf  
Install a boot block using the existing /etc/lilo.conf? [Yes]  
Testing lilo.conf ...  
Testing successful.  
Installing the partition boot sector...  
Installation successful.
```

起動時に必要な、ルートディ
スクイメージのシンボリック
リンクを作成しますか？
に “YES ”と答えます。

liloを実行するか聞いてきます
ので、とりあえず YESと答えま
す。NOでもいいです。
でも、このままリブートしても
動作しません。

nano(又はvi)/etc/lilo.conf

```
#vi /etc/lilo.conf
Boot up Linux by default.
#
default=Linux

image=/vmlinuz
  label=Linux
  initrd=/initrd.img
  read-only
```

その後liloを実行、エラーがないか確認
debian:/# lilo
Added Linux*
Added Linux O 1d
リブートします。

この行を追加後
Ctrl-Xで、終了時に保存
(viの場合はESC、ZZで保存)

ブートできなかったら
BDS画面が終了した瞬間に、ALTキーを押します。
カーネル選択メニューが表示されます。
そこで、古いOSを選択すれば、再度編集できます。

ドライバーの再設定

- コンソールからroot権限でログインします。
- LANドライバーを確認します。
 - modprobe LANドライバー名
 - 正常にインストールできるドライバーを捜します。
- USBドライバーを確認します。
 - modporbe usb-uhci
 - modprobe usb-ohci
 - どちらか、うまくいく方を調べます。
- USB - STORAGEドライバーを確認します。
 - modporbe usb-storage
 - で、CFリーダのドライバーを登録します。
- これらの正常にインストールできたドライバーを/etc/modulesに記録します。

CFにLinuxを書き込む

- TWISE LABOのHPからダウンロードしたファイルを自分のHOMEディレクトリにコピーします。
 - icop6047.tar.gz :通常インストールしたもの
 - icop6047-ramdisk.tar.gz :RAMディスクで動作するようにしたもの
 - icop6047-64MB.tar.gz: サイズ64MBに入るようにしたもの
 - lilo.conf.cf.icop6047 :LILO設定ファイルです。
 - twlinux.tar.gz TW-Linuxです。
- CF上にパーティションを作成する。
- CFのフォーマット(ファイルシステムの作成)
- CF上にOSの書き込み
- CFのブートローダーを登録する。(LILO)
- CFをアンマウントし、CFを外します。

ファイルの準備

```
debian:/home/test# ls  
icop-6047.tar.gz    icop6047-ramdisk.tar.gz  lilo.conf.cf.icop6047  
icop6047-64MB.tar.gz  icop6047.tar.gz  twlinux.tar.gz  
debian:/home/test#
```

全部は、必要ありません。書き込みものとILO設定ファイルのみ必要です。

パーティションを作成

fdisk /dev/sdaを実行します。

```
Tera Term - 192.168.1.16 VT
File Edit Setup Control Window Help

      cfdisk 2.11n

      Disk Drive: /dev/sda
      Size: 128188416 bytes
      Heads: 4   Sectors per Track: 62   Cylinders: 1009

-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
                Pri/Log    Free Space                128.12

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

Print help screen
```

パーティションの作成

- 他のパーティションがある場合 (デジカメ用のものは、FATでフォーマットされています。)
 - パーティションを選択して、<Delete>コマンドで削除します。全てFreeスペースにします。
- Linuxパーティションを作成します。
 - Freeの領域を選択して、<New>を実行します。
 - サイズを、最大値にします。
 - あとは、デフォルトで、<ENTER>を押します。
 - 作成後、選択し< Bootable>を実行してFlagsにBootと表示されるようにします。
 - sda1というパーティションになります。
- <Write>コマンドを実行して、書き込みます。この時点まで、まだ、変更はされていません。
- <Quit>します。

フォーマットとマウント (ファイルシステムの作成)

```
始めて使うCFの場合は、おまじないで
debian:/home/test# mke2fs /dev/sda
mke2fs 1.27 (8-Mar-2002)
/dev/sda is entire device, not just one partition!
Proceed anyway? (y,n)y
を実行します。
debian:/home/test# mke2fs /dev/sda1
mke2fs 1.27 (8-Mar-2002)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
31360 inodes, 125085 blocks
6254 blocks (5.00%) reserved for the super user
First data block=1
16 block groups
8192 blocks per group, 8192 fragments per group
1960 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
debian:/home/test# mount /dev/sda1 /mnt
debian:/home/test# ls /mnt
lost+found
debian:/home/test#
```

上記のおまじないをしないと特定のCFでは、`lost+found`のように文字化けします。
この状態だと、
正しく書き込みできません。

CF上にOSの書き込み

```
debian:/home/test# cd /mnt
```

書き込むファイルを変えることができます。

```
debian:/mnt# tar xvzf 書き込むファイル
```

```
|
```

```
debian:/mnt# ls
```

```
bin cdrom etc home lib mnt proc sbin usr vmlinuz
```

```
boot dev floppy initrd lost+found opt root tmp var
```

```
debian:/mnt#
```

ボード用の設定変更 (TW-Linuxの場合は不要)

- ホスト名
 - /mnt/etc/hostnameを編集します。
- IPアドレス、ルータなどの設定変更
 - /mnt/etc/network/interfacesを編集します。
 - man interfaceに設定方法が書いてあります。

```
/etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card - this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
    address 192.168.1.20
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.250
```

固定 IPアドレスを付
ける場合

ブートローダーの登録 (LILO)

```
debian:/mnt# cd /home/test
debian:/home/test# ls
icop-6047.tar.gz  icop6047-ramdisk.tar.gz  lilo.conf.cf.icop6047
icop6047-64MB.tar.gz  icop6047.tar.gz
debian:/home/test# lilo -C /mnt/etc/lilo.sda_hda.conf
```

Added Linux *

アンマウントします。

```
debian:/home/test# umount /mnt
```

```
debian:/home/test#
```

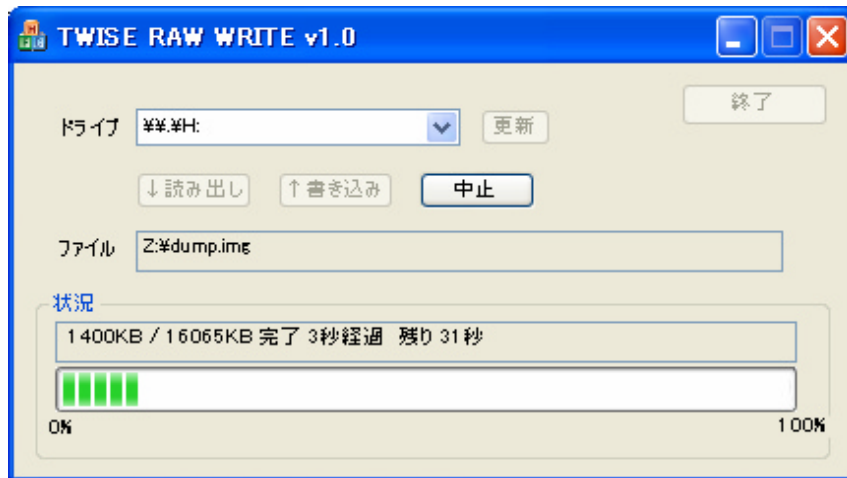
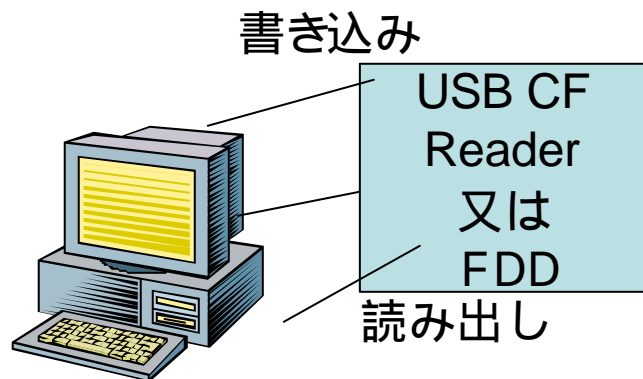
ここで、CFを取り外せます。

CFリーダのアクセスランプを確認してください。

できたものを、ボードに付けて、電源ONで起動するはずです。

このファイルにブートローダーの書き込み設定が入っています。これは、CFを/mntにマウントし、書き込みデバイスが/dev/sda、ブートデバイスが、/dev/hadの場合です。

CFの作成 (Windows環境で)



TWISE RAW WRITE for Windows

(<http://www.twise.co.jp>からダウンロードできます。)

ダウンロードしたイメージを、このソフトで、CFに書き込んでください。