

ご利用になる前に必ずお読みください

このPDFファイルの内容についてのご質問・お問い合わせは株式会社アスキー・メディアワークスでは一切お受けできません。ご自身の責任においてご利用ください。



この作品は、クリエイティブ・コモンズの表示-非営利-継承 2.1 日本ライセンスの下でライセンスされています。この使用許諾条件を見るには、<http://creativecommons.org/licenses/by-nc-sa/2.1/jp/>をチェックするか、クリエイティブ・コモンズに郵便にてお問い合わせください。住所は：171 Second Street, Suite 300, San Francisco, California 94105, USA です。

このファイルをクリエイティブ・コモンズの表示-非営利-継承 2.1 日本ライセンスに基づいて利用する際には、下記クレジットを必ず作品や配布物に表示する必要があります。

クレジット：

- 文/vine_user(「独学 Linux」 http://blog.livedoor.jp/vine_user/)
- 写真/岡田清孝
- デザイン/シオズミタロウ
- 初出/株式会社アスキー・メディアワークス「Ubuntu Magazine Japan vol.02」(<http://ubuntu.asciimw.jp/>) 2009 年 11 月 30 日発行

PCと連携させれば、さらに活躍度アップ!!

ネットウォーカー

ネットウォーカーでの作業の続きはPCで！
NetWalkerならではのPCとの連携テクニックをバ
ッチリと解説するぞ!! ●文 vine_user (ブログ「独学Linux」)

自宅PC

ラクラク連携術

NetWalker PC-Z1



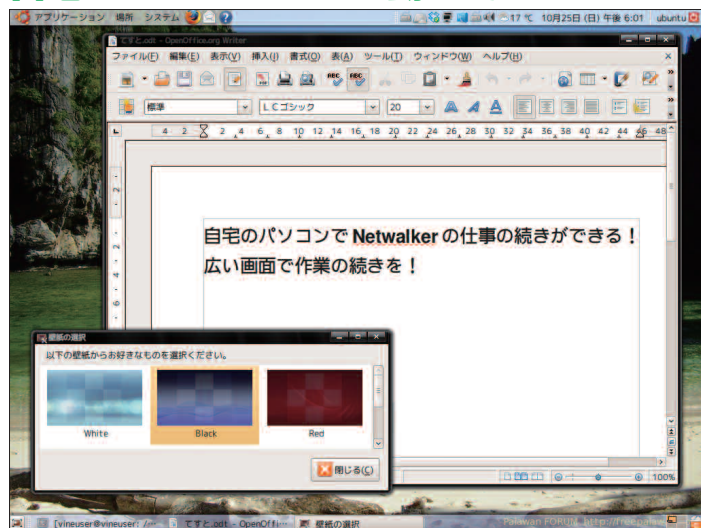
●シャープ
●<http://www.sharp.co.jp/netwalker/>

実売価格
4万円前後

シャープカスタマイズによる
ARM版Ubuntu9.04を搭載した
モバイルインターネット端末。タ
ッチパネル付きの5インチ液晶に、
両手で持ちながら打てるフルキ
ーボード、電源オンから3秒起動、
約10時間のバッテリー駆動時間
など、外出先でもフル機能のUb
untuを使えるのが魅力の1台だ。

**自宅PCの広々画面で
作業を続けられる!!**
ネットブックと携帯電話の中間
ガジェットとして登場したシャ
ープのネットウォーカー。Ubuntu
カスタマイズ版が搭載された、約
400グラムの超軽量PCといえ
るだろう。この携帯性は外出先
に持ち歩くには非常に便利だが、自
宅に持ち帰ってから狭い画面で作
業の続きをやるのはあまりにも不
便だ。そこで、Linuxの特性を
生かして、自宅PCの大きな画面
でも作業できるようにしてみよう。
初期設定はやや面倒だが、一度設
定してしまえば、快適な環境で使
いこなすことができる。

ネットウォーカーのアプリを 自宅のUbuntu PCで動かそう!



▲X Forwardingを使って、ネットウォーカーの「OpenOffice.org Writer」を自宅PCのディスプレイに表示している。ネットウォーカーの背景を遠隔操作で変更するなど可能だ。

X Forwardingを使って 画面をPCに転送する

デスクトップLinuxのデスク
トップやアプリのウィンドウなどは
「X Window」という描画シス
テムを利用している。これは、X

Ubuntu PCで ネット ウォーカーを 操作する

X Forwarding編

サーバを稼働させてネットワー
ク経由でデスクトップやアプリなど
をクライアントとして起動させる
というしくみ。単体のPCでは、
特に設定しなくてもPC内部で処
理されている。もともとネットワ
ーク機能を使っているの、この
機能を利用すれば、ネットワー
クに接続された他のPCでクライ
アントのアプリを実行させることが
できるというわけだ。
「X Forwarding」は、このしくみ
を利用して、他のPCでGUIの
アプリケーションを起動させる機
能だ。外出先から操作することも
可能だが、ルータでポート開放を
行う必要がある、セキュリティ上
おすすめできないため、今回はL
AN内での接続方法を例として解
説しよう。

4 gdmの設定ファイルを変更する

```
$ sudo gedit /etc/gdm/gdm.conf
↑gdmの設定ファイルをテキスト・エディタで開く
# DisallowTCP=true ←コメントアウトする
DisallowTCP=false ←この行を追加して保存
```

■ログイン時（gdmの起動時）に6000番ポートが無効にならないようにgdmを設定する。設定ファイルを開き、上記のように266行目を変更、新たな1行を追加して保存する。

5 クライアントPCからのログインを許可する

```
$ xhost +192.168.x.x
↑調査したIPアドレス(192.168.x.x)を指定
```

■PC側でNetwork Managerのアイコンを右クリックし「接続情報」を開き、IPアドレスを確認しておく。LAN内なら「192.168.x.x」の形で表示されるはずだ。このIPアドレスで上のコマンドを実行する。

6 gdmとSSHを再起動する

gdm の再起動

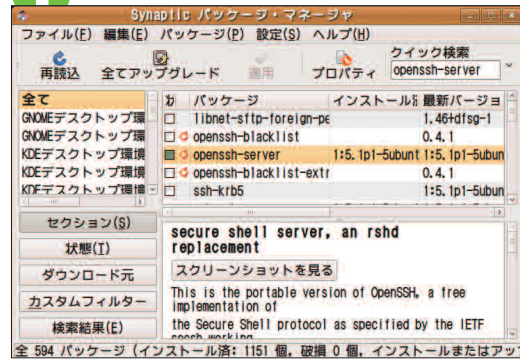
- 起動しているアプリケーションを終了
- [Ctrl] + [Alt] + [F2] キーでテキストモードに移行
- \$ sudo /etc/init.d/gdm restart を実行

SSH の再起動

- 端末を起動する。
- \$ sudo /etc/init.d/ssh restart を実行

■設定の変更を反映させるため、gdmとSSHを再起動しておく。ネットワーク側を再起動させる方が簡単だが、再起動せずに設定を反映するには上の手順を踏めばいい。なお、[Ctrl] + [Alt] + [F2] キーでテキストモードに移行したら、ユーザ名とパスワードを入力してログインする必要がある。

1 SSHをインストールする



■Synapticパッケージ・マネージャでクイック検索の入力欄に「openssh-server」と入力し、[Enter]キーを押す。検索結果から「openssh-server」をインストール。

2 SSHの設定を行う

```
$ less /etc/ssh/sshd_config | grep X11F
X11Forwarding yes ←ここが「yes」になっていればOK
$ sudo gedit /etc/ssh/sshd_config
PermitRootLogin no ←「yes」を「no」に変更して保存
```

■端末から上のコマンドを実行。SSHの設定は「/etc/ssh/sshd_config」というファイルで指定される。X Forwardingが許可されているかを確認し、root（管理者）でのアクセスを無効（no）に変更して保存する。

3 Xサーバの設定をする

```
$ sudo gedit /etc/X11/xinit/xserverrc
↑xserverrcファイルをテキストエディタで開く
# exec /usr/bin/X11/X -nolisten tcp
↑コメントアウト(行頭に#を加えて設定を無効にする)する
```

■Xサーバの6000番ポートを開けておく必要があるの、上記のコマンドで設定ファイルを開き、上記の1行をコメントアウトして保存する。

Ubuntu PC(クライアント側)からの接続方法

3 ネットワーカを操作できるように!!

■PC側とネットワークのシステムモニタ(コマンド名は「gnome-system-monitor」)を同時に表示させてみた。左下の画面はプロセッサが「ARMv7」となっている。こちらがネットワークのウィンドウだ。



! 初回起動時はRSAキーを登録

```
$ ssh -l ubuntu -X 192.168.0.2
The authenticity of host '192.168.0.2 (192.168.0.2)'
can't be established.
RSA key fingerprint is
f6:0c:xx:xx:xx:xx:1a:56:xx:xx:2a:24:3a:fc:xx:xx.
Are you sure you want to continue connecting (yes/no)?
yes ←入力
Warning: Permanently added '192.168.0.2' (RSA) to the
list of known
hosts.
ubuntu@192.168.0.2's password: ←パスワードを入力
```

■最初の接続時に、SSHのRSAキーにクライアントのIPアドレスを追加するか確認される。上のように「yes」を指定すると、以後はパスワードのみで接続できる。

1 SSHでネットワークに接続

```
$ ssh -l 「ユーザ名」 -X 「ネットワークのIPアドレス」
```

SSHで接続するには「openssh-client」というパッケージが必要だが、Ubuntuの場合、デフォルトでインストールされているはずだ。接続するには、ネットワークのIPアドレス・ユーザ名・ログイン時のパスワードが必要なので、事前に控えておく必要がある。

SSHでの接続は、端末で上記のコマンドを実行すればいい。接続が完了すると、端末のシェルが切り替わり、プロンプトの左側にあるユーザ名とコンピュータ名がネットワークのものに変化する。この時点で、端末での作業はネットワーク上で行っていることになる。

2 アプリケーションを起動する

```
$ 「コマンド名」 2> /dev/null &
```

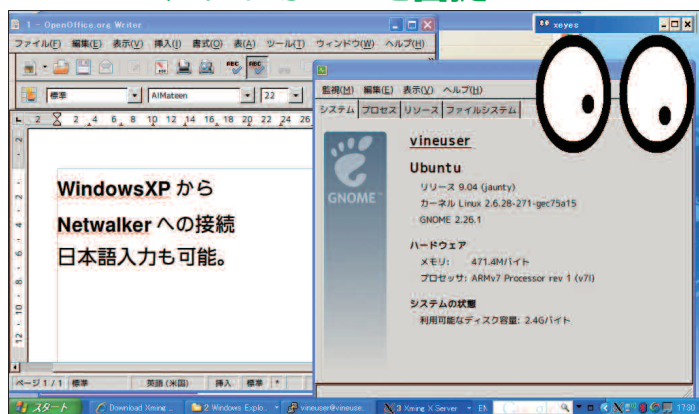
クライアントPC上でネットワーク側にあるアプリを起動するには、SSH接続した状態で、端末から上のようにコマンドを実行すればよい。

「コマンド名」の部分には起動したいネットワークのアプリのコマンドを入れよう。「2> /dev/null」は、標準エラー出

力を表示させないようにするための指定で、末尾の「&」はバックグラウンドで実行することを意味する。

ちなみに、バックグラウンドで動いているジョブを強制終了するには、psコマンドでプロセスIDを調べて、killコマンドを実行すればよい。

UbuntuアプリがXPを占拠!?



XPのデスクトップ上でネットウォーカーの「OpenOffice.org Writer」、「システムモニタ」、「xeyes」を実行してみた。

Xming

http://sourceforge.jp/projects/sfnet_xming/

日本語版PuTTY

<http://hp.vector.co.jp/authors/VA024651/>

XPでネットウォーカーアプリを起動しよう!
Windows XPでX Forwardを実行するには、XP側にXサーバとSSHのクライアントプログラムを導入しておく必要がある。

Windows XP上でネットウォーカーを操作しよう!

Xming/PuTTY編

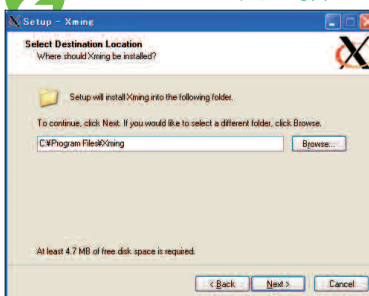
ここでは、Windows上で動作するXサーバとして「Xming」を使い、SSHクライアントとして「PuTTY」を利用してみよう。まず、左記のサイトから「Xming-6-9-0-31-setup.exe」をダウンロードして、この順番にインストールしよう。
Xming-fontsのインストール手順は省略するが、インストール先にXmingと同じ場所を指定しておくところだけ注意すればOKだ。なお、ネットウォーカー側については53ページの「ネットウォーカー側での設定」と同じ手順で設定しておこう。ネットウォーカーの性能に依存するので、アプリの起動には少し時間がかかるが、しばらく待てば表示されるはずだ。

1 Xmingのインストール



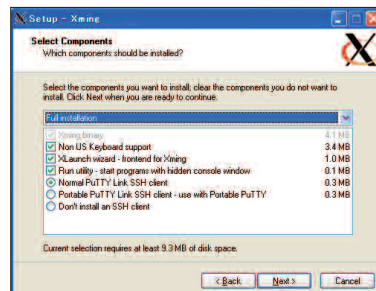
■「Xming-6-9-0-31.exe」をダブルクリックして実行する。「Next」ボタンで次へ進む。

2 インストール先を指定



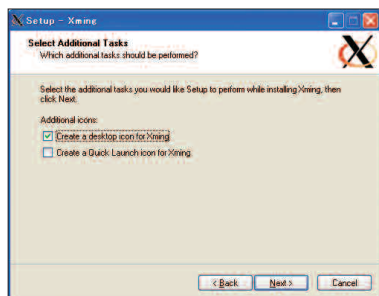
■インストール先を指定する。デフォルトのままOK。「Next」ボタンで次へ進む。

3 フルインストールを選ぶ



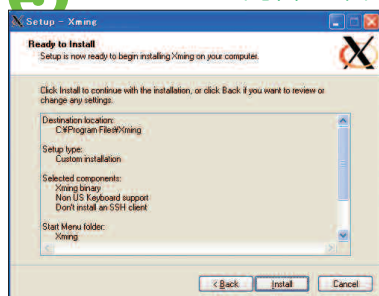
■インストール方法を指定。「Full installation」を選択し、「Next」ボタンで次へ進む。

4 デスクトップにショートカット



■「Create a desktop icon for Xming」にチェックを入れ、「Next」ボタンで次へ進む。

5 インストール内容の確認



■指定した内容を確認して「Install」ボタンでインストールを開始する。

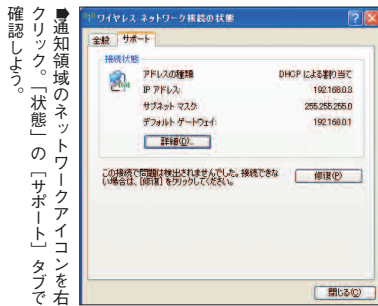
6 インストール完了



■この画面が出ればインストールは終了。「Finish」ボタンをクリックして一度終了しよう。

ネットウォーカーのラクラク連携術

IPアドレスの確認



Puttyのダウンロード

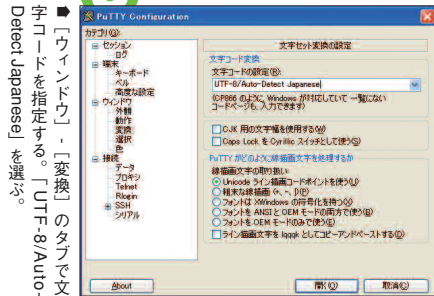


■「putty-0.60-jp20070603.zip」というファイルをダウンロード。

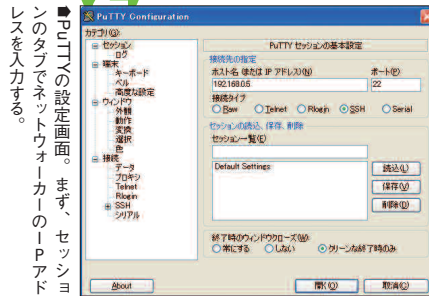
日本語化されたPuTTYが右ページ記載のサイトで配布されているので、「putty-0.60-jp20070603.zip」というファイルをダウンロードして解凍しよう。また、後の手順に必要となるので、Windows PCのIPアドレスを確認しておこう。

PuTTYは、特にインストール作業を必要としないので、先にXP側でデスクトップ上に作成されたアイコンをダブルクリックしてXmingを起動しておこう。タスクバーの通知領域にアイコンが表示されていればOKだ。

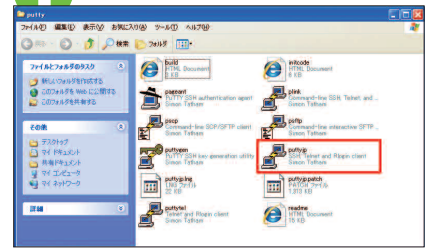
③ 文字コードの指定



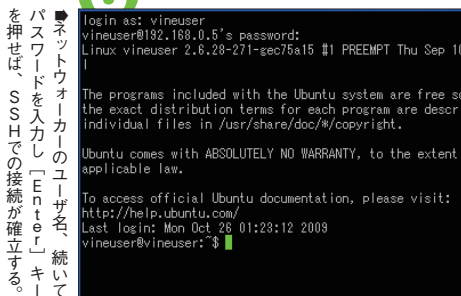
② NetWalkerとの接続



① PuTTYを起動する



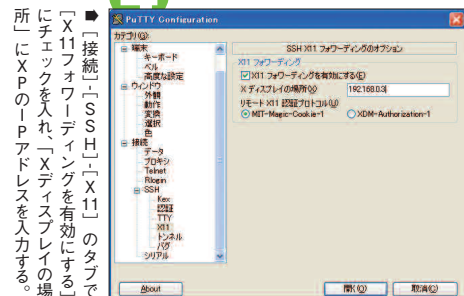
⑥ ユーザ名とパスワード



⑤ [開く]ボタンを押す



④ XPのIPアドレスを入力



Windows PCとファイルを共有する

これまでネットウォーカーのアプリをXP上で直接実行する方法を述べてきたが、作成したファイルはすべてネットウォーカー内に保存され、自宅PCには保存できない。そこで、自宅PCと直接ファイルをやりとりするために、Sambaというファイル共有ソフトを利用してアクセスの制御など詳細な設定を行うこともできるが、最も簡単に導入するには、以下の手順で設定する。

STEP1

ネットウォーカーの端末で、以下のコマンドを実行する。smbpasswdコマンドの実行後、パスワードを訊かれるので同じものを2回入力する。

```
$ sudo apt-get install samba
$ sudo smbpasswd ユーザ名
```

STEP2

ファイルブラウザでネットウォーカー上の共有したいフォルダを選び、右クリックして「共有のオプション」を選ぶ。「このフォルダを共有する」と「他人がこのフォルダへ書き込むのを許可する」にチェックを入れ、「Create Share」をクリックする。

STEP3

次のコマンドでSambaを再起動する。

```
$ sudo /etc/init.d/samba restart
```

これで、自宅PCのネットワーク上にネットウォーカーのフォルダが見えてははずだ。ユーザ名とパスワードを入力すれば接続できる。

アプリの実行方法

```
$ コマンド名 2> /dev/null &
```

■コマンド名のところに、起動したいアプリのコマンドを入れて入力する。

ネットウォーカーのアプリを起動してみる！

以上で、ネットウォーカーのアプリをXP上で実行する手順は終了。実際に動かすには、それぞれのアプリのコマンドを左の書式で入力する。下記の主なアプリのコマンドを入力して試してみよう。

ファイルをPCとネットウォーカーで共有しよう

アプリの種類	コマンド名
エディタ	gedit
ワープロ	oowriter
表計算	oocalc
プレゼンテーション	oaimpress
WEBブラウザ	firefox
メール	thunderbird
システムモニタ	gnome-sysytem-monitor

こんな感じで動く!



■Debian GNU/Linux (lenny)を入れてみた。いくつか不具合もあるが、日本語化も可能だ。

インストール環境の準備

1 再インストール用のmicroSDを準備

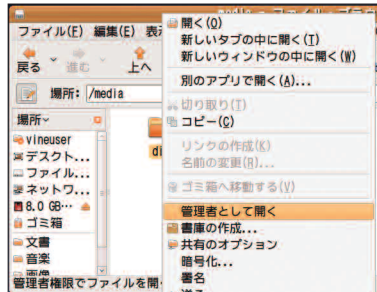
■まず下記のサイトを参考に、用意した8GBのmicroSDカードを使って再インストール用microSDを作成しよう。なお、リカバリー用途で用いる再インストール用microSDは別途作成して保存しておこう (こちらは2GBで十分)。

<http://www.sharp.co.jp/support/mit/doc/recovery.html>



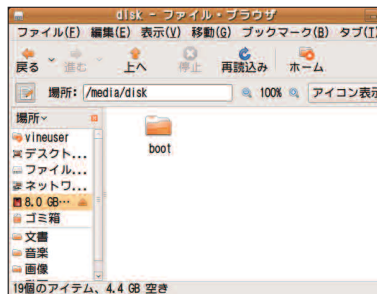
2 管理者権限で削除するために

■Synapticパッケージマネージャなどで「nautilus-gksu」をインストールしておけば、ファイルブラウザを管理者権限で開けるようになる。コマンドが苦手なら、導入しておくとう便利だろう。



3 /boot以外をすべて消去する

■作成したmicroSDを入れたままにしてネットワークを再起動しよう。ただし、左右ボタンの同時押しは必要ない。再起動後、microSDがマウントされるので、その中にある「boot」フォルダだけを残して、管理者権限で他のフォルダをすべて削除する。



4 「debootstrap」を導入する

```
$ sudo apt-get update
$ sudo apt-get install debootstrap
```

続いて、ネットワークのデフォルトのUbuntu上で、Debianインストーラが使用するツールである「debootstrap」というパッケージを導入しよう。この方法は、Debian基本シス

テムをインストールする公式の方法と認められているものだ。debootstrapは上のコマンドでインストールできる。もちろん、Synaptic パッケージマネージャを使ってもらってもいい。

再インストール用microSDを応用

ネットワークカードを購入した人なら、すでに再インストール用microSDを作成しているに違いない。ところで、この再インストール用microSDはなぜリカバリーに使えるのだろうか? 実は、再インストール用microSDには最小限のシステムがインストールされており、ネットワークカードはmicroSDからの起動に対応しているのだ。これをカスタマイズすれば、他のARM版デバイスに

実験編!

microSDからのブートでDebianを起動してみる!

ユーザのネットワークカードを起動することもできるはずだ。ちなみにネットワークカードの内蔵フラッシュにシステムをインストールするには、「ubifs」というファイルシステムでシステム全体のイメージファイルを作成しなければならず、さらに失敗すると本体が起動しなくなる危険もあるので、内蔵フラッシュへのインストールは難しいと判断した。ネットワークカードはmicroSDからの起動に対応しており、「debootstrap」という基本システムをインストールするためのプログラムもあるので、これを利用してmicroSDにインストールしたシステムを起動させてみよう。最新版のUbuntu 9.10の導入は難しく、本体と同じバージョンのU

buntuを入れてもあまり面白くないので、今回はUbuntuの派生元である「Debian GNU/Linux (lenny)」を入れてみた。ただし、いくつか不具合もあり、この記事はあくまで実験的なものである。現時点では常用には向かないと考えた方がいいだろう。この記事の内容を実行する場合は自己責任で行ってほしい。インストールに最小限必要なのは、8GB程度の容量をもつmicroSDカードとネットワークカード本体 (インターネット接続環境が必要) のみ。事前に、シャープのサイトから再インストール用microSDを作成するためのスクリプトファイルをダウンロードしておく必要がある。リカバリー用microSDは別途保管しておこう。

今回の実験のメリット/デメリット

● 本体内蔵フラッシュよりも大きな容量でネットワークカードを利用できる

● 使用感が比較的軽快。起動は若干遅いが、起動してしまえばUbuntuよりやや高速に動作

● ログイン時にキーボードが反応しない([Enter]キーを1、2回押すとgdmが再起動して入力可能になる)

● ダブルクリックが効かない

● クリックのたびにオプティカルポインタをほんの少し動かす必要がある

● タッチパネルの調整ができない

Debian JP Project



<http://www.debian.or.jp/>

Ubuntuの派生元ディストリビューション。Ubuntuは多くをDebianに依存しているので、パッケージ管理システムAPTを始め共通点が多い。さまざまなアーキテクチャに対応しており、当然ARMにも対応している。

Debian GNU/Linux

9 microSDから起動する

■ここで、一度ネットワークカーをシャットダウンして、microSDから起動させてみる。キーボードの左上にある左右ボタンを同時押ししたまま、電源を入れよう。「SHARP」のロゴが出て左右ボタンは少し長めに押しておく。



10 デスクトップ環境の導入

右のような形で、コンソールが表示されるので、rootでログインしよう。パスワードはまだ設定していないので、直接ログインできるはずだ。次のコマンドでネットワークを有効にしよう（無線LANは使えないので、USB-LANアダプタを利用する）。



```
# /etc/init.d/networking restart
```

11 X Windowを導入する

```
Section "Device"
Identifier "Configured Video Device"
Driver "fbdev" ←この行を追加
Option "UseFBDev" "true"
EndSection
```

■# aptitude install xserver-xorg-video-fbdev xinit
でX Windowを導入し、上記のように「/etc/X11/xorg.conf」を変更する。デスクトップ環境はLXDEやGNOMEなど好きなものを入れよう。
aptitude install lxde ←LXDE の場合
aptitude install gnome ←GNOME の場合

12 ユーザの設定と日本語環境の導入

```
# adduser ユーザ名
(ここで新規パスワードを入力)
# aptitude update
# aptitude install locales
# dpkg-reconfigure locales
# aptitude install ttf-vlgothic scim-anthy
# export LANG=ja_JP.UTF-8
```

■adduserのコマンドで、ユーザを追加し、ロケールの設定、日本語フォントもインストールする。なお、vlgothicなどをインストールする前に「/etc/apt/sources.list」の「ftp.en.debian.org」を「ftp.jp.debian.org」に変更しておこう。

13 日本語デスクトップ環境の起動

■「halt」コマンドで一度シャットダウンさせて、再び左右ボタンの同時押しで、起動させよう。左のようなGDMのログイン画面が表示されるはずだ。右ページのデメリットにあるように操作にはコツがあるので注意しよう。



Debian基本システムをインストールしよう

それではさっそく「debootstrap」を使ってDebianのインストール作業を始めよう。以下では、「/mnt/debinst」というディレクトリを作成し、microSDをそこにマウントして、Debianの基本システム(X Windowやウィンドウマネージャなどは入っていない状態)をネット経由でインストールした後、好みのデスクトップ環境を導入して環境を整えるという手順を進める。途中、有線LANに接続してインターネットを利用するので、きちんと接続できることを確認してから始めよう。

Debian GNU/Linux インストール手順

5 マウントポイントの変更

```
$ sudo su
# mkdir /mnt/debinst
# umount /media/disk
# mount /dev/mmcblk0p1 /mnt/debinst
```

■管理者権限での作業が多くなるので、「sudo su」でrootのシェルに移行して上のコマンドを実行しよう。「/dev/mmcblk0p1」はmicroSDのデバイス名だ。

6 インストールを実行する

```
# debootstrap --arch armel lenny /mnt/debinst \
http://ftp.jp.debian.org/debian

# cp -r /lib/modules/* /mnt/debinst/lib/modules
```

■上のコマンドは、アーキテクチャーを「armel」に指定して「debootstrap」コマンドを実行する、という意味。Debianのバージョンは最新安定版のlennyとし、リポジトリは国内のjp.debian.orgを指定している。ネットワークカー本体のモジュールをコピーして利用するため下のコマンドも実行しよう。

7 設定ファイルを編集する

```
# gedit /mnt/debinst/etc/hosts
127.0.0.1 localhost ユーザ名
# gedit /mnt/debinst/etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet dhcp
```

■起動する前に、ネットワークの設定をすませる。microSD内の設定ファイルをgeditで開いて、上記のように変更する。ユーザ名は、Debianで使いたいユーザの名前を設定しておこう。

8 chrootでDebian環境に入る

```
# LANG=C chroot /mnt/debinst /bin/bash
# mount -t proc proc /proc
# cd /dev
# MAKEDEV generic
# dpkg-reconfigure tzdata
```

■上記は「chroot」というコマンドを使って、Debianの環境に入り、デバイスファイルの作成やタイムゾーンの設定を行っている。