

Debian大特集(その2)



付録メディアに本記事で扱った
コンテンツを収録しております

Wikiサイト作りで学ぶ

Debianの 基本

Debian/GNU Linuxはソフトウェア・
パッケージの管理が容易で、安定性も高いため、
サーバー構築に適しています。
特集2では、Debianの一般的な設定を紹介したのち、
Wikiサイトを例に、サーバーを構築する方法を
実践的に解説します。 (ライター 増井 雄一郎)

Keywords Debian GNU/Linux 入門 Wiki

Debian GNU/Linuxは、Ian Murdock氏がGNUプロジェクトの方針に沿って、フリーであることに重点を置いて1993年に開発したLinuxディストリビューションです。

多くのLinuxディストリビューション、特にサーバー用途のディストリビューションは米Red Hat社のRed Hat Enterprise Linuxや米Novell社のNovell SUSE LINUX Enterprise Serverなどのように企業が主体となっていて作成しています。無料版もありますが、有償がものが中心です。

一方、Debianは企業に依存せずボランティアのみで開発されるディストリビューションです。そのため、開発やリリース期間などが、企業戦略に左右されません。

最近では、Linuxディストリビューションの中でもフリーであることが着目されて、政府や自治体といった公的機関の公式ディストリビューションとして採用されるケースが増えてきました。日本では日本医師会のORCAプロジェクトというオープンソースの診療報酬明細計算システムのOSとして使われています。

Debianを選ぶ理由は、大きく3つあります。

●フリーである

Debianでは、「フリー」とは無料という意味だけではなく、「自由」も同

時に表しています。Debianは「Debian 社会契約」という文章をリリースしており、「Debianフリー・ソフトウェアガイドライン」という項目では、Debianに収録するパッケージのライセンスに対するガイドラインが定義されています。オープンソース・ライセンスの定義の元になったのは実はこの文章です。

●厳密なパッケージ管理

Debianでは、「APT/dpkg」という洗練されたパッケージ管理システムが採用されています。APT^{*1}とdpkg^{*2}が連携して動作するため、アプリケーションのインストールやアップデートは、Debian上で非常に簡単に実行できます。

●サーバーに向くバージョンアップ戦略

多くのLinuxディストリビューションは、1年、もしくはそれ以下のサイクルでメジャー・バージョンアップを実施します。新しいバージョンを使おうとすると、再インストールしなければならないことも珍しくはありません。これはサーバー用途には不向きな性質です。

さらに新しいバージョンが実施されると、一つ前、二つ前の古いバージョンはパッケージのメンテナンスがおろそかになる傾向があります。こうなると、ソースからインストールし

てパッケージ管理システムの管理下から外れるか、自分でパッケージを作成しなければなりません。

一方、Debianのメジャー・バージョンアップ間隔は広く取られています。実際、多くのLinuxディストリビューションがバージョンにして10を超える中、Debianは2005年に3.0がリリースされたほどです。1996年に1.1がリリースされて以来、メジャー・バージョンアップは2回しか行われていません。

APT/dpkgが導入された2.1以降では、ほとんどの場合「aptitude upgrade □」と打ち込むだけで、インストール済みのパッケージをアップデートできます。Debianはパッケージを不安定版(unstable)、試験版(testing)、安定版(stable)の3つに分けています^{*3}。本誌今月号の付録

^{*1} APTは、apt-getというコマンドを用いたパッケージ管理の手法を指す。apt-getはFedora Coreのyumに相当するコマンド。パッケージを取得する相手先となるリポジトリをAPTリポジトリと呼ぶ。

^{*2} dpkgコマンドはFedora Coreでのrpmコマンドに相当する。あるパッケージのインストールによって追加されたファイル一覧(dpkg -L パッケージ名)やパッケージ・ファイル(debファイル)の内容を閲覧(dpkg -c パッケージ名.deb)できる。

^{*3} Debianに加わった新しいパッケージはまず不安定版としてリリースされる。決められた手順を経て試験版に昇格し、リリース担当者の責任において安定版に追加される。不安定版は時々刻々と更新されるが、安定版はセキュリティ修正のみ施される。

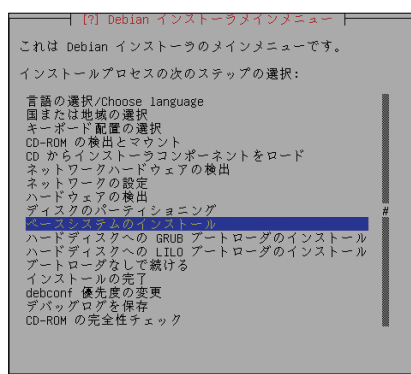


写真1 Debianのインストール画面

メディアに収録したDebianのバージョン3.1 (開発名Sarge)は、安定版とです。安定版は特に互換性を重視しているため、多くの場合、設定の変更など必要ありません。パッケージではなくディストリビューション自体のバージョンが上がった場合でも、「aptitude upgrade-dist □」と実行するだけでアップグレードできます。

このように、Debianは一度インストールをしたら、OSごと再インストールする必要はほぼありません。

Debianの弱点

しかしながら、サーバー用途にDebianを使おうとした場合、弱点が2つあります。

●商用ソフトウェアのサポートがない

多くの商用ソフトウェアは、Red HatやNovellなどの商用ディストリビューションを中心にサポートしています。実際にはDebianであっても問題なく動くことが大半ですが、企業内で導入する場合は、周囲を説得しにくいことがあるでしょう。

Debianは「Debian社会契約」で宣

言しているように、フリーソフト以外を収録しません。商用ソフトウェアが公式パッケージに入ることはないのです。これは致命的に見えますが、商用ソフトウェアのインストール支援ソフトという形で、米Sun Micro systems社のJDKや米Macromedia社のFlash Playerのインストーラを公式パッケージとして提供しています。公式のサポートはありませんが、多くの場合問題なくDebian上で動作します。

●パッケージのバージョンが古い

Debianでは非常に多くのパッケージが提供されているため、パッケージ管理者が慢性的に不足しています。このため、マイナーなパッケージでは、更新が遅れる場合があります。

特にstableと呼ばれる安定版のパッケージ群は、リリース後、重要なアップデートやセキュリティ修正のみを提供します。機能を追加したというだけのバージョンアップは行われません。スクリプト言語PHPなど開発が活発なソフトウェアでは、実際の利用に支障が出るほど古いバージョンが提供されていることもあります。

これを回避するために、試験版や不安定版と呼ばれる十分にテストされていないパッケージ群を使うことが多いのです。何らかの問題が起る場合があるので、管理者が十分に

注意する必要があります。

以上のような長所や短所を理解した上で、Debianを使うかどうかを判断しましょう。Debianは長期にわたって、安定したメンテナンスを提供することに重点を置いて開発されています。サーバー向けとして魅力的なLinuxディストリビューションといえるでしょう。

インストールと環境設定

Debianはあまりくせの無いディストリビューションです。他のディストリビューションを使ったことのある方であれば、ほとんど苦も無く移行ができるでしょう。

本誌付録メディアに収録したのは約100Mバイトの最小CD-ROMイメージ(debian-31r0a-i386-netinst.iso)です。これは必要最小限の構成となっており、インストール時に必要なソフトウェア・パッケージをネットワークからダウンロードするものです。DHCP環境でインターネットに接続していれば、ほぼ何の設定もせずに画面に従ってインストールできます(写真1) *4

しかし、インストール直後の状態では、使いにくい部分があります。調整を施すには、Debian固有のコマンドを利用しなければなりません。そこでFedora Core 4などの他のLinux

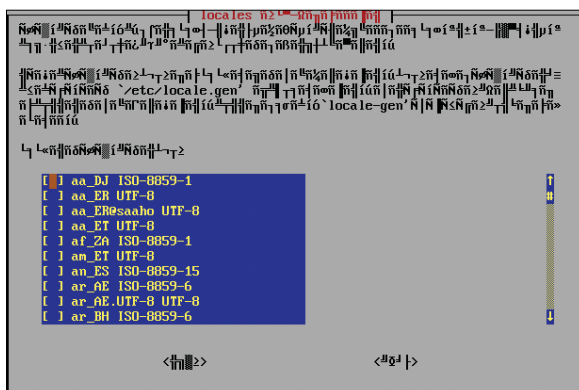


写真2 標準のロケール設定で、コンソール画面が文字化けしたところ

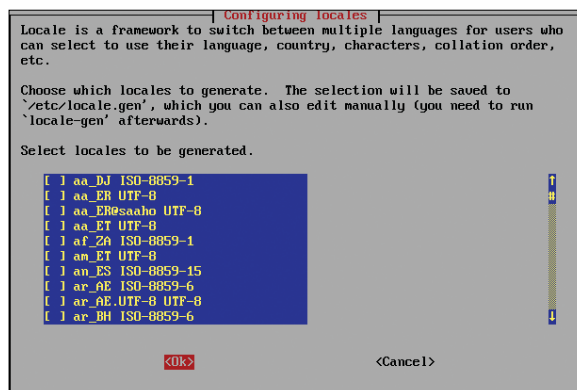


写真3 ロケールを設定しているところ

ディストリビューションから移行した場合につまずきやすいと思われる部分を重点的に解説しましょう。

日本語環境を設定する

Debianは国際化されたLinuxディストリビューションであり、日本語版などの言語別、地域別にはパッケージやインストーラを配布していません。デフォルトでは、インストール時に選択した言語が第一言語として扱われ、ロケールが設定されます。

インストーラで「Japanese - 日本語」を選んだ場合、インストール直後はロケールが「ja_JP.EUC-JP」になっています。

しかし、コンソールはこのままでは使えません。標準では日本語を表示できないため、コマンドによってはメッセージが化けてしまいます(写真2)。また、このロケールでは動作しないアプリケーションも存在します。

このロケール情報を変更しましょう。

```
# su
# LANG=
```

dpkg-reconfigure locals

このマークで改行

画面表示後、最初にシステムで利用できるロケールを選択します。現時点では「ja_JP.EUC-JP」のみにチェックが付いています。UTF-8や他の言語のロケールを使うには、ここで該当するロケールにスペース・キーでチェックを付けてください。ロケールを選択したら、タブキーを押し、OKボタンを選択して次の画面に移ります(写真3)。

続いて、デフォルト・ロケールを設定しましょう。現在は「ja_JP.EUC-JP」が選択されていますが、上キーを押して「None」を選択したのち、OKを選んで設定を終えてください。

これでデフォルト・ロケールが未設定状態になりました。変更を有効にするために、一度ログアウトして、再度ログインしましょう。これでメッセージの文字化けは直りました。

ここまでに変更した設定は、「/etc/environment」にテキストとして記されています。何らかの理由でコマンドを使わない場合は、直接こ

のファイルを編集してください。

キーボードを変更する

言語環境で左右されるものがもう一つあります。キーボードです。標準では、インストール時の設定がそのまま使われます。変更したい場合には、先ほどと同じdpkg-reconfigureコマンドを起動してください。

```
# dpkg-reconfigure
console-data
```

説明画面を読んだ後、「Select keymap from full list」を選択しましょう(写真4)。キーボードの種類が表示されています。適当なものを選んでください(写真5)。

*4 詳しくは日経LinuxのWebサイトから、2006年1月号の補足情報を参照(<http://itpro.nikkeibp.co.jp/linux/backnum/200601/correct.shtml>)。インストール方法をまとめたPDFファイルがダウンロードできる。Debianのインストール方法としては、特集3「Mac miniはマジで速い」も参考になる。

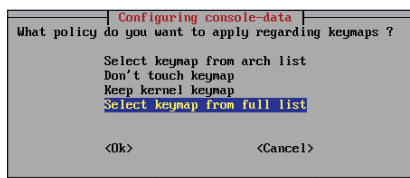


写真4 キーマップを変更する画面

ほとんどの場合は日本語106キーボードを使っているでしょう。この場合「pc / qwerty / Japanese / Standard / Standard」を選択します。

コンソールで日本語を表示する

Debianをサーバー用途で利用する場合、インストール後の作業はsshなどを経由し、ネットワークを使うことが多いでしょう。しかしながら、ネットワークが使えない状況もあり得ます。直接コンソールを操作できるよう準備が必要です。

まず、コンソールで日本語が表示できるように、日本語コンソールのアプリケーションをインストールしておきます。日本語対応のものとしては、kon2とjfbtermが広く使われていますが、多くのビデオ・ボードに対応していることから、ここではjfbtermで構築します。

jfbtermはパッケージとして提供されていますので、ここでは簡易GUIを備えたaptitudeコマンドでインストールしてみましょう。同時にフォント集であるxfonts-baseもインストールします。

```
# aptitude install jfbterm
xfonts-base
```

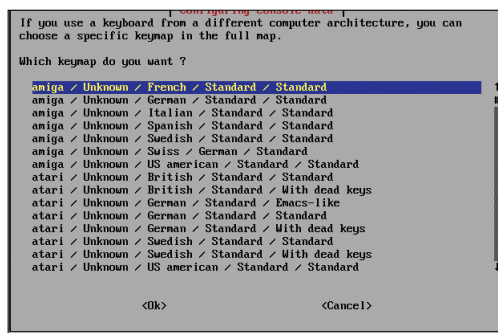


写真5 キーボードを選択しているところ

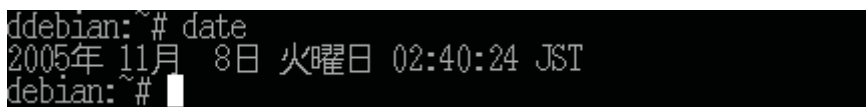


写真6 jfbtermで日本語を表示したところ

続いてjfbterm自体を起動します。日本語を使うには同時にロケールを設定しておく必要があるからです。

```
# LANG=ja_JP.EUC-JP
jfbterm
```

以上で、jfbtermを使って日本語が表示できるようになりました(写真6)。

ネットワークを設定する

Debianはインストール時にネットワーク上でDHCPが稼働していると、自動的にDHCPを使う設定になってしまいます。しかし、DebianをサーバーOSとして運用する場合は、一般的に固定IPアドレスを使うでしょう。

DebianのTCP/IPの設定は、ファイル「/etc/network/interfaces」に記されています。DHCPが有効になった状態では、interfacesファイルの内容は、図1のようになっています。

通常はloopbackのためのloと、Ethernetのeth0が設定されていま

す。loに触れる必要はありません。「auto[インタフェース名]」で指定されているインタフェースが起動時に設定されます。autoの部分がないとシステム起動時にネットワークは起動しません。

IPアドレスなどを設定するには「iface」の行を書き替えます。この行を図2のように変えることで、固定IPアドレスを設定できます。アドレスは適宜正しいものを記述してください。

interfacesファイルを変更後、設定を有効にするためには、以下のコマンドを実行して、ネットワークを再起動してください。

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

これで新しいネットワークの設定が有効になりました。

現在のIPアドレスを表示する、一時的にIPアドレスの変更する、などの操作には他のLinuxディストリビューションと同様、ifconfigやrouteコマ

ドを使ってください。

DNSを設定する

DNSサーバーの設定は、Debian独自の部分はほとんどありません。設定ファイルとしては、`/etc/resolv.conf`を書き換えます。しかし、Debianでは`resolvconf`パッケージを導入することで、DNSを先ほどの`interfaces`ファイルで同時に設定できるようになります。

`resolvconf`パッケージは標準では導入されていないので、`aptitude`コマンドを使ってインストールします。

```
# aptitude install
resolvconf
```

ただし、前提としてAPTサーバーが設定された状態でなければなりません。先にp.52にまとめた「APTを設定する」を参考にして、APTを使える状態にしてください。

`interfaces`ファイルでDNSサーバーを指定するために、図3のように末尾に2行追加します。追加したのは「`dns-search`」で始まる行と「`dns-nameservers`」という行です。

書き換え後に、「`# /etc/init.d/networking restart`」を実行してください。設定が有効になり、定義に従って「`/etc/resolv.conf`」が変更されます。

`resolvconf`パッケージをインストー

```
/interfaces --
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

図1 DHCP利用時の`interfaces`ファイルの内容
`/etc/network/interfaces`を示した。

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1
```

図2 固定IPアドレス利用時の`interfaces`ファイルの例

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1
dns-search example.com
dns-nameservers 192.168.1.1
```

図3 DNSサーバーの指定内容
`/etc/network/interfaces`を示した。

表1 DebianとFedora Coreのラン・レベルの意味

ラン・レベル	Debian	Fedora Core
0	停止	停止
1	シングル・ユーザー・モード	シングル・ユーザー・モード
2	マルチユーザー・モード	ネットワーク無効
3	2に同じ	テキスト・ログイン
4	2に同じ	未使用
5	2に同じ	グラフィカル・ログイン
6	再起動	再起動

ルした後、DNSの設定を変更する場合は、直接`/etc/resolv.conf`ファイルは編集しないでください。代わりに`resolvconf`コマンドを使います。

ラン・レベルと起動項目の設定

Debianのラン・レベルは、Fedora Coreなど他のLinuxディストリビューションと少し違います(表1)。

Fedora Coreなど多くのディストリビューションでは、コンソールを使う場合はラン・レベル3、X Window System(以下、X)を使う場合にはラン・レベル5で起動します。

しかし、Debianではラン・レベル2しか利用しません。「テキスト」と「グラフィカル」という区別がないのです。

これは、Debianでは利用するアプリケーションのみをインストールする

表2 DebianとFedora Coreのラン・レベル関連のコマンド

コマンド	Debian	Fedora Core
ラン・レベルでのサービスの起動・停止	update-rc.d	chkconfig
サービスの起動停止	/etc/init.d/??	service
ラン・レベルの変更	telinit	telinit
ラン・レベル・エディタ	パッケージで提供	ntsysv

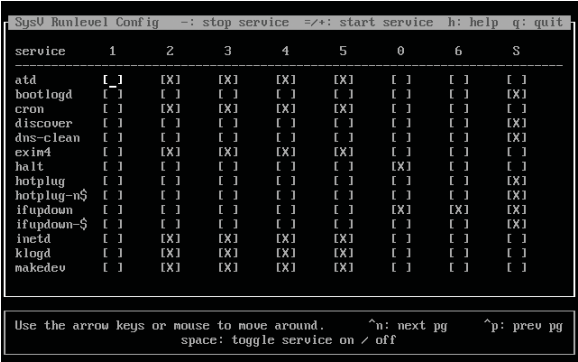


写真7 ラン・レベル・エディタの画面
sysv-rc-confを実行したところ。

というポリシーがあるためです。XDMやGDMなどのディスプレイ・マネージャをインストールすると、自動的にディスプレイ・マネージャが起動し、グラフィカル・ログインになります。

サービスごとの起動・終了スクリプトは、/etc/init.dで管理されており、ラン・レベルごとに/etc/rc?.d/にあるスクリプトが動きます。この部分は多くのLinuxディストリビューションと同じです。Fedora Coreでは、ラン・レベルと起動アプリケーションの管理のために、chkconfigコマンドを使用します。Debianで対応するコマンドはupdate-rc.dコマンドです。

```
# update-rc.d サービス名
start|stop 優先度 ラン・レベル
. 
```

例えば、GDMが起動して常にグラフィカル・ログインになっているとします。これをテキスト・ログインに戻し

たい場合は、以下のように実行します。

```
# update-rc.d gdm stop 99
1 2 3 4 5 6 . 
```

ここでは、優先度99のgdmサービスをラン・レベル1から6で起動するように設定しています。

続いて、ラン・レベル5の場合のみ、gdmサービスが起動するように設定します。

```
# update-rc.d gdm start 99
5 . 
```

これで、他のLinuxディストリビューションと同様にラン・レベル5の場合のみ、グラフィカル・ログインを使うように変更できました。

Debianではラン・レベルのカスタマイズが、管理者の手にゆだねられているのです(表2)。

ラン・レベルを大きく変更する場合は、sysv-rc-confなどのラン・レベル・エディタをインストールして使う方が楽でしょう(写真7)。

```
# aptitude install sysv-rc-conf 
```

現在のラン・レベル自体を変えるには、telinitコマンドを使います。ラン・レベル5へ移行するには次のように入力します。

```
# telinit 5 
```

起動時のラン・レベルを変えるには、/etc/inittabを書き換えてください。初期状態では、ラン・レベル2で起動しますから、次のように書かれているはずです。

```
id:2:initdefault:
```

この部分を次のように変えることで、ラン・レベル5で起動します。

```
id:5:initddefault:
```

APTを設定する

Debianの最大の特徴は、APT (aptitude) /dpkgを使ったパッケージ管理にあります。

表3 aptitudeコマンドの利用方法

機能	コマンド
CUIインタフェース起動	aptitude
お勧めパッケージを同時にインストール	aptitude -r -g install パッケージ名
特定のバージョンをインストール	aptitude install パッケージ名=バージョン番号
パッケージを検索	aptitude search キーワード
パッケージの削除	aptitude remove パッケージ名
パッケージと設定ファイルの削除	aptitude purge パッケージ名
パッケージの情報を表示	aptitude show パッケージ名
パッケージのアップグレード	aptitude upgrade [パッケージ名]
パッケージのアップグレードを行わない	aptitude hold [パッケージ名]

aptitudeコマンド単体で多くのアプリケーションをインストールできるため、他のLinuxディストリビューションに移ると落差に驚くこともあるでしょう。

Debianは、2005年6月にリリースされた3.1 (Sarge) 以前には、apt-getコマンドやapt-cacheコマンドを使って、パッケージをインストールしていました。しかし、Debian 3.1以降は新しいパッケージ管理コマンドaptitudeを使います。そこで、表3に代表的なコマンド例を紹介します。「[]」の部分は省略が可能です。

APTを使うには、まず利用するAPTサーバーを/etc/apt/sources.listに指定する必要があります。ファイルの内容を見てみましょう(図4)。

debで始まる行でパッケージを提供しているサーバーを指定しています。deb-srcはソース・パッケージ向けの指定です。標準では、公式パッケージ群が設定されています。

このdebやdeb-srcで始まる指定を「apt-line」と呼びます。公式パッケージのみを使う場合は、多くの場合、図4にある3行を指定します。非公式パッケージや、開発版のパッケージを利用する場合には、このファイルに該当するサーバーのapt-lineを追加しなければなりません。通常はインストール時に使用するAPTサーバーを選択しているはずなので、図4のま

```
#dev file:///cdrom/ sarge main

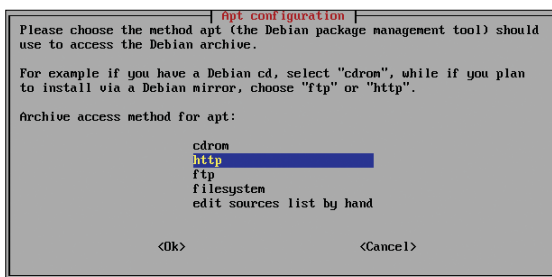
deb http://www.ring.gr.jp/archives/linux/debian/debian/ stable
main

deb-src http://www.ring.gr.jp/archives/linux/debian/debian/
stable main

deb http://security.debian.org/ stable/updates main
```

図4 APTの設定ファイルの内容

/etc/apt/sources.listの内容を示した。2行目で公式パッケージを、3行目で公式ソース・パッケージを、4行目でセキュリティ・アップデートを指定している。

写真8 APTサーバーを指定しているところ
apt-setupコマンドを用いた。

ま利用することにしましょう。

インストーラでAPTをCD-ROMからインストールすると指定した場合などは変更が必要です。apt-setupコマンドを実行するとAPTサーバーの選択画面になりますので、httpかftpを選択し、適当なミラー・サーバーを選択してください(写真8)。

apt-setupの実行後はもちろん、/etc/apt/sources.listを変更した場合には、aptitudeコマンドを使ってパッケージ・リストを更新する必要があります。

```
# aptitude update
```

コマンドを起動すると、APTサーバーからパッケージ情報を取得します。パッケージをインストールする前などは、念のためこのコマンドを実行しておく方とよいでしょう。

アプリケーションを導入するには

aptを使ってアプリケーションをインストールするのは、驚くほど簡単です。例えば、Ruby (<http://www.ruby-lang.org/>) をインストールした

い場合には、次のように実行するだけです。

```
# aptitude install ruby
```

aptitudeコマンドのメッセージは日本語化されているため、必要に応じてロケールを設定して実行します。

rubyパッケージをインストールすると、自動的に最新版のruby1.8とlibruby1.8パッケージが選ばれます。rubyをインストールするには、さらにzlib1gなど複数のパッケージが必要ですが、必要なパッケージは、aptがすべて自動で探し出してインストールしてくれます。

APTの特徴は登録パッケージの多さ

APTを使ったパッケージ管理はDebianの最大の特徴です。確かに現在ではFedora Coreなど他のLinuxディストリビューションでも、APTと同じように使えるパッケージ管理システムが提供しています。

しかし、現在でもAPTは他のパッケージ管理システムよりも充実しています。Debianでは提供されているパッケージ数が多いからです。現在、1万5000個以上が公式パッケージとして配布されています。Debianでオープンソースのアプリケーションを使いたいなら、ほとんどの場合、aptitudeコマンドだけでインストール

できるのです。

パッケージ管理システムを使う上で一番重要になるのは、パッケージの依存関係でしょう。例えばRPMファイルで提供されていないライブラリなどをソースからインストールし、競合を起こして困った経験はありませんか。さらに、パッケージの依存関係が循環してしまい、アンインストールできないような状態に陥ったことがあるかもしれません。Debianでは、ほとんどの場合、ソースからインストールする必要があるため、このような問題は起こりにくいのです。

Debianでサーバーを作る準備

ひとたびインストールしてしまえば、OSの再インストールはほとんど必要ない、メンテナンス性の高さがDebianの魅力です。以下では、OSにDebian (Linux)、WebサーバーにApache HTTP Server、データベース管理ソフトにMySQL、スクリプト言語としてPHP/Perlを用いたサーバー、いわゆるLAMP (Linux, Apache, MySQL, PHP/Perlの頭文字)サーバーを作ってみましょう。

例題として取り上げるのは、データベースを用いたWikiと、Javaを用いたTomcatサーバーです。いずれもWebアプリケーション・サーバーとして利用できます。

まずはSSHをインストールする

通常、DebianなどのOSをインストールした後は、SSHを使って外部のマシンから作業を続けます。Debianは、インストーラで追加パッケージを選択していない限り、SSHをインストールしません。

まずは、以後の作業のために、SSHでDebianサーバーにログインできるようにSSHサーバーをインストールしておきましょう。

もちろん、APTを使います。SSHクライアントとサーバーが一つになったsshパッケージが用意されていますので以下のようにインストールします。

```
# aptitude install ssh
```

インストール時にいくつかの質問に答える必要があります。

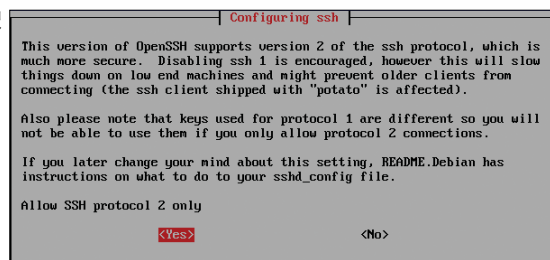
まずは、SSH v1プロトコルを使用しない設定にするかどうかを決めます。特に必要がなければ「Yes」を選択してください(写真9)。

続いてSSH2のホスト認証に使うファイルをSUIDしてよいかどうか決めます。ここでも「Yes」を選択します(写真10)。

そして最後に、SSHサーバーを起動してよいかという質問があります。SSHによるログインを許さないのであれば、ここで「No」を選択してください。

写真9 SSH v1プロトコルを選択する画面

通常は、Yesを選択してインストールしない。



い。ほとんどの場合、SSHサーバーを起動するでしょうから、ここでも「Yes」を選択します(写真11)。

sshパッケージの詳細を調べる

以上で、sshのパッケージがインストールできました。パッケージについて学ぶよい機会なので、実際にどのようなファイルがインストールされたのか、表示してみましょう。

```
# dpkg --getfiles ssh
```

コマンドの出力結果はsshパッケージで提供されているファイルを示しています。出力から、sshの設定ファイルが/etc/ssh/にインストールされているのだろう、と推測できます。

初めてインストールしたパッケージについては、どこに設定ファイルがあるのかが分かりづらいこともあるでしょう。このようにdpkgコマンドを使うことで、関連するファイルの位置が一目瞭然になります。

逆にインストールされたファイルから、パッケージを検索することもできます。何に使うのか分からない孤立したファイルに対して使うと便利でしょう*5。

```
# dpkg --getfiles ssh
/usr/sbin/sshd
ssh: /usr/sbin/sshd
```

写真10 SSH2のホスト認証に関して選択する画面

Yesを選択して、認証ファイルをSUIDする。

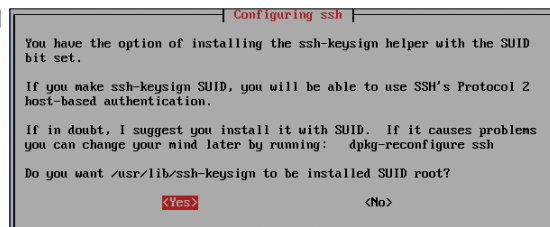
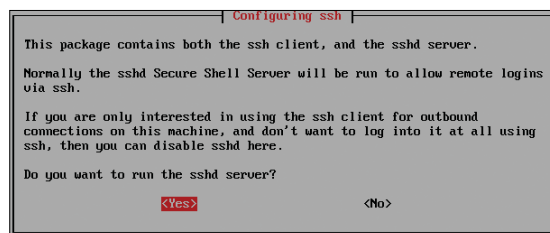


写真11 SSHサーバーを起動するかどうか選択する画面

Yesを選択して起動する。



インストール後にパッケージを再設定することもできます。

```
# dpkg-reconfigure ssh
```

インストール時に設定を選択するよう促したスクリプトをもう一度実行できます。このように実行するとほとんどの場合、設定ファイルなどが上書きされて元に戻ってしまいます。設定ファイルを書き換えた後は、設定ファイルのバックアップを取ってから実行してください。

目的のパッケージを探すには

Debianでは、ソフトウェアをインストールする場合、パッケージ名を知っておく必要があります。しかし、パ

ッケージ名を暗記しているはずもないでしょう。そこでDebianでは、目的のパッケージを探す方法が提供されています。

まず、表3にまとめたようにaptitudeコマンドのsearch機能があります。この場合、パッケージ名や説明文からパッケージを検索します。もう一つは、Debianの公式Webサイトです。「Debian - パッケージ」(<http://www.debian.org/distrib/packages>)のWebページでは、ディストリビューションやセクションを超えた検索も可

*5 ただし、dpkgを使った場合、sshパッケージのインストーラがスクリプトによって自動生成した/etc/ssh/ssh_host_dsa_keyなどファイルは表示・検索できない。

ここには、Debianで最新のパッケージがあります。パッケージが安定性と品質の基準を満たせば、テスト版に含まれるようになります。不安定版もセキュリティアプローチにはサポートされていません。

不安定版のパッケージは、ほとんどテストされてなく、そしてシステムの安定性に影響するなどの問題があるかもしれません。このディストリビューションは、経験豊かなユーザーだけがよく考えてから使うべきです。

詳細は、不安定版 (unstable) ディストリビューションのページを見てください。

パッケージディレクトリを検索

キーワード:

検索対象: ☒ パッケージ名のみ ☐ パッケージ説明文 ☐ ソースパッケージ名

単語の部分検索: ☒

大文字小文字の区別: ☐

ディストリビューション: セクション:

いくつかの検索に対して、簡単な方法があります。

- <http://packages.debian.org/name> は、バイナリパッケージ名で検索します。
- <http://packages.debian.org/src/name> は、ソースパッケージ名で検索します。

パッケージの内容を検索

この検索エンジンでは、Debian ディストリビューションの中からパッケージの一部である、いかなるファイル(あるいはファイル名の一部)でも検索することができます。また、あるパッケージに含まれるファイルの一覧を見ることができます。

キーワード:

結果表示方法:

- ☒ キーワードに似た名前のファイルを含むパッケージ
- ☐ キーワードに似た名前のファイルやディレクトリを含むパッケージ
- ☐ キーワード文字列を含む名前のファイルやディレクトリを含むパッケージ
- ☐ キーワードで示されるパッケージに含まれるファイルの一覧

大文字小文字の区別: ☐ ディストリビューション: アーキテクチャ:

[Debian プロジェクトホームページに戻る。](#)

このページは以下の言語でもご覧になれます

[Български \(Bulgarian\)](#) [català](#) [čeština](#) [dansk](#) [Deutsch](#) [Ελληνικά \(Ellinika\)](#) [English](#) [español](#) [français](#) [한국어 \(Hangul\)](#) [Italiano](#) [magyar](#) [Nederlands](#) [polski](#) [Português](#) [Русский \(Russian\)](#) [slovensky](#) [suomi](#) [svenska](#) [українська \(ukrajinska\)](#)

[デフォルトの言語を設定するには](#)

写真12 Debianの公式Webサイトの情報
目的のパッケージを検索できる。

能です。コマンド名やファイル名からも検索できます(写真12)。

Wikiサーバーを作る

準備が整ったので、いよいよLAMPサーバーを構築してみましょう。単純なWebサーバーでは面白くありませんから、だれもがWebブラウザを使ってWebページを編集できる「Wikiサーバー」などはどうでしょうか。

手軽に使えるWikiソフトとしてはPukiwiki (<http://pukiwiki.sourceforge.jp/>) などが有名ですが、データベースを使いません。そこで、フリーの百科事典サイトWikipedia (<http://wikipedia.org/>) などで行われているWikiソフトであるMediaWiki (<http://www.mediawiki.org/>) を組み込んでみましょう。

MediaWikiは、MySQLをバックエ

ンドに持ったWikiクローンで、アクセス数が世界で50位以内に入るWikipediaで使われていることから分かるように、大規模なサイトでの運用や、ユーザー認証が必要な社内Wikiでの利用に向いています*6。

まずは、公式サイトにあるインストール・ドキュメント (<http://meta.wikimedia.org/wiki/Help:Installation>) を読み、必要なソフトを確認します*7。MediaWikiを稼働させるには、Apache、PHP、MySQLが必要となるので、インストールします(表4)。

```
# aptitude install apache2
libapache2-mod-php4 php4-
mysql mysql-server
```

aptitudeコマンド1つだけで、MediaWikiに必要なソフトが一気にインストールできました。

次に、外部のPC上でWebブラウザ

を起動してアクセスします(写真13)。インストールが正常であれば、いつものApacheの起動画面が見えるはずです。

Apacheの設定を確認する

Apacheの起動が確認できたら、次にその設定を確認しましょう。

Debianでは、パッケージに関するドキュメントを、`/etc/share/doc`ディレクトリ以下にまとめています。パッケージを設定する場合は、まずはこちらにあるファイルを参照してください。apache2ディレクトリのファイル一覧を見てみましょう。

```
$ ls
/usr/share/doc/apache2/
README.etc
changelog.Debian.gz
changelog.gz  copyright
examples
```

Apacheの場合は、README.etcにDebian固有の説明が書かれています。Apacheでは違いましたが、他の多くのパッケージでは、README.Debianという名前で提供されます。

README.etcを読むと、`/etc/apache2/apache2.conf`がメインの設定ファイルであること、ポート番号やIPアドレスに関する設定は、`/etc/apache2/ports.conf`に記述するとあ

表4 インストールするパッケージの一覧

パッケージ名	内容
apache2	Apache httpd 2.x
libapache2-mod-php4	Apache2.xで稼働するPHP4モジュール
php4-mysql	PHP4用MySQL接続モジュール
mysql-server	MySQLサーバー

ります。

Webサイトごとの設定は、`/etc/apache2/site-available/`の中に適当なファイルを作って記述すること、このうち有効にしたい設定ファイルは、`/etc/apache2/site-enabled/`にシンボリック・リンクを張るようにと書かれています。デフォルト・ホストの設定は、`/etc/apache2/sites-available/default`になるので、これを確認してみましょう。

今回は、「DocumentRoot `/var/www/`」と設定されていました。Apacheから参照されたディレクトリは、`/var/www`になります。

PHPの設定を確認する

次にPHPの確認に入ります。次のようにして`phpinfo.php`を作り、Webブラウザから確認します。

```
# echo "<?php phpinfo(); ?>"
> /var/www/phpinfo.php
```

`/var/www/`にPHPのファイルが作成されたら、Webブラウザから「`http://サーバーのIPアドレス/phpinfo.php`」にアクセスします。PHP4が正常にインストールされていれば、写真14のような画面が表示されます。

続いて、設定内容を変更してみましょう。現在の設定では、PHPのエ

ラー・メッセージが画面に表示されるようになっています。これをログに出力するように変えましょう。

通常、PHPの設定は、`php.ini`ファイルに施します。最初にパッケージから、`php.ini`を検索してください。

```
$ dpkg --search php.ini
```

図5のようにひな形となる、`php.ini-dist`は見つかりましたが、`php.ini`自体はインストール後のスクリプトで生成されているようです。ここでは検索できませんでした。

しかし、図5から`php.ini`を管理しているのは、`php4-common`パッケージだと分かったので、ドキュメントを参照してみましょう。`/usr/share/doc/php4-common/`に、圧縮された`README.Debian.gz`が見つかりましたので、次のようにして`zcat`コマンドで確認します。

```
$ zcat
/usr/share/doc/php4-common
/README.Debian.gz |more
```

このドキュメントには、`/etc/php4/apache/php.ini`がPHPの設定ファイルであると書かれていました。

```
$ cat
```

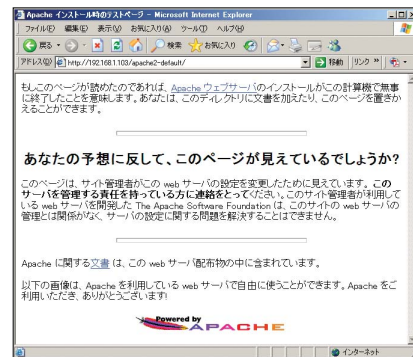


写真13 Apacheの起動画面を確認したところ

```
/etc/php4/apache/php.ini
cat:
/etc/php4/apache/php.ini:
No such file or directory
```

ディレクトリを1つ戻ってみると、`/etc/php4/apache2`ディレクトリがあり、この中に`php.ini`がありました。

この`php4-common`パッケージはApache 1.3とApache 2.xの共有パッケージのため、ドキュメントがApache 1.3x用になっていたようです。

`php.ini`のログ出力に関する部分を、図6に従って変更してください。変更後、設定を反映させるために、Apache2を再起動します。

*6 英Netcraft社が提供する情報による (http://toolbar.netcraft.com/site_report?url=http://en.wikipedia.org)。

*7 MediaWikiの公式サイトは一部日本語化されている。

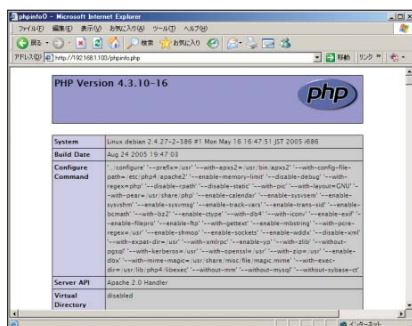


写真14 PHPの出力画面を確認したところ

```
# /etc/init.d/apache2
restart
Forcing reload of web
server: Apache2.
```

先ほどWebブラウザで開いていた `phpinfo.php` をリロードし、「display_errors」の項目が「off」になっていることを確認してください。

MySQLを設定する

MySQLはインストール直後、rootのパスワードが指定されていません。このままでは非常に危険なので、まずはrootのパスワードを設定します。

```
$ mysqladmin -u root
password "新しいパスワード"
```

新しいパスワードには、適切だと思われるパスワードを入力します。

さらに初期状態では、testというデータベースが登録されていますが、必要ありませんので、削除します。

```
$ mysqladmin -u root -p
```

```
php4-common: /usr/share/php4/php.ini-dist
php4-common: /usr/share/doc/php4-common/examples/php.ini-
recommended
php4-common: /usr/share/doc/php4-common/examples/php.ini-dist
php4-common: /usr/share/doc/php4-common/examples/php.ini-
paranoid
```

図5 php.iniの検索結果

drop test

コマンドの出力に対しては図7のように答えてください。

Debianでは、MySQLにdebian-sys-maintと呼ばれるreloadとshutdownの権限を持った管理用のユーザーが登録されています。このユーザーはmysqlサービスの起動・停止時、さらにcronからも呼び出されます。消さないようにしてください。特に他のPCで稼働していたデータベースをDebianに移行する際に、誤ってこのユーザーを消さないように注意してください。

MediaWikiをインストールする

MediaWikiが必要とするソフトのインストールが完了したので、MediaWiki自体をインストールしましょう。

MediaWikiは、SourceForgeで配布されています。ダウンロード・ページ (<http://www.mediawiki.org/wiki/Download>) をたどって、tarballのURLを探してください。本誌の付録メディアにも収録しました。

MediaWikiのtarballをwgetでダウンロードし、`/var/www`に展開します。rootで作業するのは、セキュリティ上

好ましくないのでオーナーをexampleユーザーに変更しておきます。

```
# cd /var/www
# wget -O -
http://jaist.dl.sourceforge
e.net/sourceforge/wikipedi
a/mediawiki-1.5.2.tar.gz |
tar xvz
# mv mediawiki-1.5.2 wiki
# chown -R example
# su example
$ cd wiki
```

MediaWikiの設定にはWebブラウザを使います。その前に、設定を保存するディレクトリをPHPから書き込めるように、パーミッションを変更しておきましょう。

```
$ chown a+w config
```

パーミッションの変更が終わったら、Webブラウザから「`http://サーバーIP/wiki/`」にアクセスしてください。写真15の画面が表示されますので、「set the wiki up」をクリックし、設定を始めましょう。

まずは現在の環境をチェックしま

```
display_errors = On
↓
display_errors = Off

log_errors = Off
↓
log_errors = On
```

図6 php.iniの変更部分

```
Enter password: ← 設定したパスワード
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'test' database [y/N] y ← データ
ベース名を確認してy
Database "test" dropped
```

図7 MySQLの不要なデータベースを削除しているところ

す。写真16のような画面が表示されるまで多少時間がかかります。

すると、2つ気になるメッセージが出力されました。

```
-PHP's memory_limit is 8M.
If this is too low,
installation may fail!
Attempting to raise limit
to 20M... ok.

-Couldn't find GD library
or ImageMagick; image
thumbnailing disabled.
```

最初のメッセージによると、PHPの使用できるメモリーが8Mバイトに設定されているものの、20Mバイト以上必要とあります。そこで余裕を持って32Mバイトに設定しましょう。この設定内容は、先ほど編集したphp.iniに記載されていますので、/etc/php4/apache2/php.iniの「memory_limit」の行を「32M」に設定してください(図8)。

設定ファイルを変更後、Apacheを再起動します。

```
# /etc/init.d/apache2
restart
```

2番目のメッセージによると、GDやImageMagickといった画像処理の機能拡張がインストールされていないため、画像のサムネイルが表示ができないようです。早速PHP4のGDモジュールとImageMagickモジュールをインストールします。Apacheも再起動しましょう。

```
# aptitude install php4-gd
php4-imagick
# /etc/init.d/apache2
restart
```

ここでWebブラウザをリロードしてください。

先ほどのメッセージは、以下のよう内容

```
-PHP's memory_limit is
32M. If this is too low,
installation may fail!

-Found GD graphics library
built-in, image
thumbnailing will be
enabled if you enable
uploads.
```

に変わりました。問題ないようです。

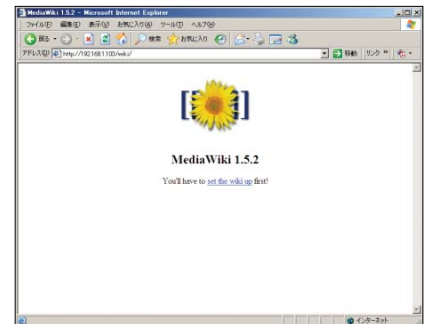


写真15 MediaWikiにアクセスしたところ



写真16 MediaWikiの設定画面
警告メッセージが2つあるようだ。

MediaWikiの動作を設定する

以下では、写真16の画面に従って、MediaWikiを設定していきます。

最初はWebサイト自体に関する設定です。「Site Name」には適当なWebサイト名を入力してください。「Contact Email」には、MediaWikiからの通知メールを受け取るメール・アドレスを記入します。このアドレスは公開されません。「Language」には、メッセージの言語を選びます。「Copyright」はコンテンツのライセンスを指定します。


```
memory_limit = 8M      ; Maximum amount of memory a script may
consume (8MB)
↓
memory_limit = 32M     ; Maximum amount of memory a script may
consume (32MB)
```

図8 php.iniの変更内容

特に指定がなければ「no license metadata」を選択します。

「Sysop account name」には、MediaWikiの管理者アカウント名とパスワードを指定します。「Shared Memory Caching」は、今回使いませんので「no caching」を選択します。電子メールに関する設定は、デフォルトのまま使用します。

次はデータベースに関する設定欄です。画面では、MySQLに接続して新しいデータベースやユーザーを作成しています。

「Database name」, 「DB username」, 「DB password」には、MediaWikiで使うデータベース名やアカウントを指定します。無ければ自動作成されます。パスワードは必ず入力してください。

「Database table prefix」には、テーブル名の頭に付ける文字列を指定します。1つのデータベースで複数のWikiを利用する場合には、テーブル名が重複しないように指定が必要です。今回はデータベースを共有しないので、何も入力しません。「Database charset」ではデータベースの文字コードを指定してください。「Backwords-compatible UTF-8」がよいでしょう。

最後に「DB root password」に、

MySQLのrootのパスワードを指定します。ここでは、先ほどmysqladminで設定したパスワードを指定してください。

以上で、MediaWikiの設定が完了しました。最下部の「install!」ボタンを押してください。設定ファイルが作成されます。筆者の環境では、ページの読み込みが終わりませんでした。ページをリロードすることでインストールが完了しました。

Webブラウザを閉じ、ターミナルから、このインストーラで作成された設定ファイルをMediaWikiのフォルダにコピーします。念のため、先ほど変更したパーミッションも戻しておきましょう。

```
$ mv
config/LocalSettings.php
./
$ chmod 755 config
```

「http://サーバーのIPアドレス/wiki/」をWebブラウザで開いてください(写真17)。Wikiサイトが動いています。

実際にWikiとして利用するには「編集」とあるタブをクリックします。編集画面が開きますので、表5の文法に従って、Webページを作成して

ください*8。

このようにDebianでは、複数の機能拡張を要求するようなアプリケーションを、非常に簡単にインストールできます。

Javaアプリ・サーバーを動かす

冒頭で、Debianの弱点として、フリーではない商用ソフトがパッケージに含まれないことを挙げました。

商用ソフトのうち、最も困るのがWebアプリケーション・サーバーで多用するJavaでしょう*9。幸い、Sun社が提供しているJ2SE (Java 2 Standard Edition)のバイナリ・パッケージは、そのままDebianにインストール可能です。とはいえ、J2SEをそのままインストールしてしまうと、Debianのパッケージ管理から外れてしまいます。Apacheなどとの連携時に設定が面倒になってしまいます。

そこでDebianでは、Sun社が配布しているバイナリ・パッケージをDebianにインストールするためのインストーラを提供することにしました。これを使えば、Sun社のJ2SEをDebianのパッケージ管理の中で扱うことができます。

J2SE SDKのインストール

まずは、J2SEのダウンロード・ページ(<http://java.sun.com/j2se/>)

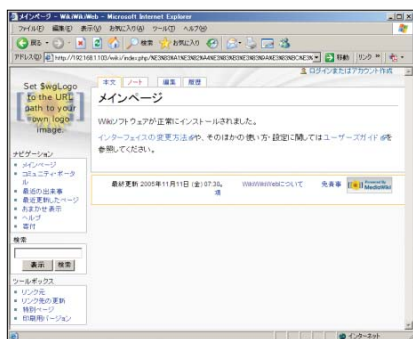


写真17 MediaWikiの動作画面
この状態でWikiクローンとして利用できる。

1.4.2/ja/download.html)に行き、「Linux Platform - Java (TM) 2 SDK, Standard Edition」の「self-extracting file」を選択し、ダウンロードしてください*10。これをscpコマンドなどを使って、Debianをインストールしたマシンにコピーします。

次に、J2SEをDebianパッケージに変換するjava-packageをインストールします。同時にJavaの共通パッケージもインストールします。

しかし、このパッケージはcontribというセクションで配布されており、現在のAPTの設定ではインストールできません*11。そこでまずはapt-lineを編集し、contribセクションもインストールに含めるようにします。

そのためには、/etc/apt/sources.listを編集しなければなりません。現在の設定はmainセクションのみになっているので、行末にcontribを追加してください(図9)。sources.listを編集したら、aptitudeでupdateするのを忘れないでください。

```
# vi /etc/apt/sources.list
# aptitude update
```

表5 MediaWikiの書き込みに使える主なタグ

編集内容	記述方法
「文字列」を表示	文字列
強調文字を表示	'''強調文字'''
改行	2行改行する
通し番号を表示	#
個条書きを表示	*
区切り線を表示	----
Webサイトへのリンクを埋め込む	[URL]
WikiMedia内のページへのリンクを埋め込む	[[ページ名]]

これでjava-packageをインストールできるようにになりました。

```
# aptitude install java-  
package java-common
```

java-packageをインストールすると、先ほどダウンロードしたJ2SE SDKをDebianのパッケージに変換するmake-jpkgというコマンドが使えるようになります。

```
# chmod a+x j2sdk-  
1_4_2_10-linux-i586.bin  
# su nobody -c fakeroot  
make-jpkg j2sdk-1_4_2_10-  
linux-i586.bin
```

rootでは、パッケージを変換できないので、nobodyユーザーとして実行します。はじめにメンテナの名前とメール・アドレスに関する質問が現れます。パッケージ管理のための情報ですので、適切な内容を入力してください。

続いてJ2SE SDKのライセンスに対しては、確認後「yes」を入力してください。すぐにパッケージの変換が始まります。途中、ライブラリが不足し

ているなどといったメッセージが出ることがありますが、ほとんどはAWT関連のパッケージですので問題はありません。

変換後、作業中のディレクトリにsun-j2sdk1.4_1.4.2+10_i386.debという、Debianのパッケージができています。これをインストールします。

```
# dpkg -i sun-
```

*8 各ページには本文とノートが存在しており、それぞれに履歴が存在する。履歴を閲覧するには履歴タブを押せばよい。

*9 DebianではJavaの開発・実行環境として、Kaffe (http://www.kaffe.org/) が提供されている。これはGPLで公開されたJava仮想マシンの実装だが、米Sun Microsystems社の認定を受けていないため、公式のJava環境とはいえない。

*10 2005年10月時点の最新版はj2sdk-1_4_2_10-linux-i586.bin。

*11 Debianのパッケージは、DSFG (Debian Free Software Guidelines) というガイドラインを満たすかどうかによって、main, contrib, non-freeの3つに分かれている。DSFGには、ソース・コードを入手できるか、再配布・販売を認めるか、個人や団体を差別しないか、などの10個の条件が定義されている。ApacheなどにDSFGに沿ったソフトウェアはmainに入る。contribにはDSFGに沿っていないソフトウェアを利用するDSFGを守ったソフトウェアが分類されている。non-freeにはDSFGが適用できないソフトウェアがまとめられている。

```
deb http://www.ring.gr.jp/archives/linux/debian/debian/ stable
main contrib ← contribを追加した
deb-src http://www.ring.gr.jp/archives/linux/debian/debian/
stable main contrib ← contribを追加した

deb http://security.debian.org/ stable/updates main
```

図9 sources.listの変更内容
/etc/apt/sources.listを2カ所変更する。

```
# To enable mod_jk2, customize workers2.properties* from
# /usr/share/doc/libapache2-mod-jk2/examples and copy it to
# /etc/apache2/workers2.properties. Then uncomment the
following line:
JkSet config.file /etc/apache2/workers2.properties ← 先頭の#を取る
```

図10 mod_jk2の変更内容
/etc/apache2/mods-enabled/jk2.confを編集する。

```
j2sdk1.4_1.4.2+10_i386.deb
```

Javaのインストールは完了しました。念のためjavaコマンドで確認してみましょう。

```
# java -version
java version "1.4.2_10"
Java(TM) 2 Runtime
Environment, Standard
Edition (build 1.4.2_10-b03)
Java HotSpot(TM) Client VM
(build 1.4.2_10-b03, mixed
mode)
```

今回は、Sun J2SE 1.4 SDKをインストールしましたが、java-packageはJava2 1.5 SDKや米IBM社のSDKにも対応しています。

Tomcat環境を構築する

次はJavaやApacheと組み合わせて簡単なWebアプリケーション・サ

ーバーを作るために使えるTomcat (<http://tomcat.apache.org/>)をインストールしましょう。

Tomcatはパッケージとして提供されているため、aptitudeでインストールできます。ApacheからTomcatに接続するためのmod-jk2も提供されています。先ほどJavaをDebianパッケージとしてインストールしたことで、これらのJava関係パッケージをaptitudeでインストールできるようになりました。

```
# aptitude install tomcat4
libapache2-mod-jk2
tomcat4-webapps
```

これだけで、Tomcatとmod-jkのインストールが完了し、既に起動状態になっています。次にmod_jk2の設定に入ります。Apache経由でTomcatを稼働させるためです。

mod_jk2の設定ファイルは、/etc/apache2/mods-enabled/jk2.

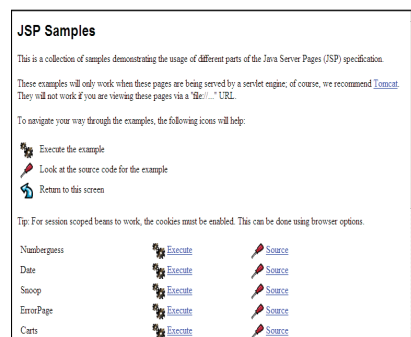


写真18 Tomcatの動作画面
デフォルトの表示画面が現れた。

confです。このファイルにはサンプル設定が書かれているので、末尾の行の先頭の#を取って設定を有効にします(図10)。

次にTomcatのexamplesから、Tomcatの設定ファイルをコピーし、以下のようにして、Apacheを再起動してください。

```
# cp
/usr/share/doc/libapache2-
mod-jk2/examples/workers2.
properties.minimal
/etc/apache2/workers2.prop
erties
# /etc/init.d/apache2
restart
```

これで、Tomcatの設定は終わりました。Webブラウザから、「http://サーバーのIPアドレス/jsp-examples/」を開くと、Tomcatのサンプル表示が見られます(写真18)。Tomcatを用いたServlet, JSPの作り方については触れませんが、動的にWebページを生成するための準備はこれで整っています。