## Hypertext Preprocessor





## インストール







### ↓ 前提事項

Linuxのディストリビューション(カーネルに各種ソフトウェアパッケージを組み合わせ た配布物または商品)においては、RedHat、Kondara、Vine、TurboLinux などのいわゆる RedHat系といわれるカテゴリが存在します。これらのディストリビューションでは、各種ア プリケーションのインストールにおいて、RPM(Redhat Package Manager)という管理シ ステムを使っています。RPM は簡単にパッケージのインストール / アンインストールができ、 make などの細かい作業をしなくてもよいという優れた特徴があります。その反面、最新バ ージョンに追随しにくい(最新バージョンが出ても、誰かがそのバージョン用のRPM パッケ ージを作ってくれるのを待つか、自分で作らなければならない)というデメリットも存在し ます。

特にPHP4に関しては、登場してまだあまり間がないことや、国際化の作業が継続中である ということもあり、安定して動作するRPM パッケージがまだ存在しないようです。そこで本 書では、ソース(tar.gz形式のアーカイブ。いわゆるtar ball)から各ソフトウェアをインスト ールする方法について説明します。この方法であれば、本書の出版後に各ソフトウェアのバ ージョンが上がっても、バージョン番号を読み替える程度で常に最新版をダウンロードして インストールすることができるようになるでしょう。また、Linux 以外のOSを使っている人 にとっても、以下の手順は参考になるでしょう。

オープンソース系であればどんなソフトウェアでも同じですが、インストール途中で何か トラブルが起こったら(できればトラブルが起こる前に)、まずはソース配布物に含まれる READMEやINSTALLといったテキストファイルを読んでください。ヒントになることが 書かれています。

以下で紹介するコマンドの実行例では、シェルのプロンプトが#で終わっている場合は root 権限で実行することを、\$で終わっている場合は特定の一般ユーザ権限で実行すること を表しています。

## **1.2** 必要な開発ツールの確認

各ソフトウェアをインストールするにあたっては、以下に示すいくつかのツールがインス トールされていることが前提となります。これらがインストールされていない場合は、ソー スを展開して生成されるINSTALLファイルに目を通してみてください。

以下に示すコマンドのバージョン表示やインストールの例は、原稿執筆時にプラットフォ ームとして使用した Vine Linux 2.0の実行結果です。ほかのディストリビューションやプラ ットフォームでは異なっていることもあります。Vine Linux 2.0を使用している場合には、 必要なツール群はすべてインストールされています。Linuxを使う限りにおいては、最近のデ ィストリビューションであればほかのものでもほぼ大丈夫でしょう。なお、RPM などの各コ マンドの使い方については、man rpm などを参照してください。

• GNU make

hotta@star:~\$ make -v GNU Make version 3.77, by Richard Stallman and Roland McGrath. Copyright (C) 1988, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. Report bugs to <bug-make@gnu.org>. hotta@star:~\$ rpm -qf /usr/bin/make

make-3.77-6

バージョン3.75以降のGNU makeが必要です。プラットフォームによっては、システム標準のmakeコマンドのほかに別途、gmakeという名前でGNU makeがインストールされているかもしれません。この場合はgmake -v で確認できます。gmakeを使う場合は、以下適宜makeをgmakeと読み替えてください。

●gcc(Cコンパイラ)

これは必須ではありませんが、各種商用UNIXに付属しているCコンパイラの中には、コンパイルできないものもあるかもしれません。

```
hotta@star:~$ gcc -v
Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/specs
gcc version egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)
```

hotta@star:~\$ rpm -qf /usr/bin/gcc
egcs-1.1.2-24vl1

- readline(コマンドライン編集用ライブラリ)
- history(コマンドヒストリ保持用ライブラリ)
   これらが組み込まれていると、psqlコマンドでコマンドライン編集やコマンドヒストリ機
   能が使えるようになり、使い勝手が格段に向上します。

```
hotta@star:~$ ls /usr/lib/lib{readline,history}*
/usr/lib/libhistory.a /usr/lib/libreadline.a
/usr/lib/libhistory.so.3@ /usr/lib/libreadline.so.3@
/usr/lib/libhistory.so.3.0 /usr/lib/libreadline.so.3.0
```

```
hotta@star:~$ ls /usr/include/readline/*
/usr/include/readline/chardefs.h /usr/include/readline/readline.h
/usr/include/readline/history.h /usr/include/readline/tilde.h
/usr/include/readline/keymaps.h
```

```
hotta@star:~$ rpm -qa|grep readline
readline-2.2.1_jp-5
readline-devel-2.2.1_jp-5
```



この節の話題はRedHat系のLinuxを使っている人だけに関係する話です。それ以外の人は読み飛ばしてもらってかまいません。

RedHat系を使っている場合は、必要なソフトウェアが(ある意味、中途半端に)インストー ルされていることがあります。このまま本書の推奨手順で同じソフトウェアをソースから構 築してしまうと、新旧バージョン間のコンフリクト(衝突)が起こり、トラブルの元になりま す。お勧めは、それらプリインストールされたものをばっさり削除してしまうことです。もし これらプリインストールされたソフトウェアを使ってすでに何らかのデータを登録済みであ れば、それらのバックアップを取っておいてください。

まず、インストールしようとしているソフトウェアがプリインストールされているかどう かを調べます。こちらの環境では以下のようになっているとします<sup>\*1</sup>。



この例は擬似的に作成したも ので、かならずしもVine Linux 2.0のデフォルトの状態を表す ものではありません。

root@star:~# rpm -qa grep postgres
postgresq1-7.0.2-2v12
postgresql-devel-7.0.2-2vl2
postgresql-jdbc-7.0.2-2vl2
postgresql-odbc-7.0.2-2vl2
postgresql-perl-7.0.2-2vl2
postgresql-python-7.0.2-2vl2
postgresql-server-7.0.2-2v12
postgresql-tcl-7.0.2-2vl2
postgresql-test-7.0.2-2vl2
postgresql-tk-7.0.2-2vl2
root@star:~# rpm -qa grep apache
apache-1.3.12-2v13
apache-devel-1.3.12-2vl3
apache-manual-1.3.12-2vl3
root@star:~# rpm -qa grep php
php-3.0.15-2
php-imap-3.0.15-2
php-ldap-3.0.15-2
php-manual-3.0.15-2
php-pgsql-3.0.15-2

インストールされているこれらのパッケージをすべて削除します。この作業はroot権限で 行います。

```
root@star:~# rpm -qa|grep postgres|xargs rpm -e
root@star:~# rpm -qa|grep apache|xargs rpm -e
root@star:~# rpm -qa|grep php|xargs rpm -e
```

インストールの前に

以下のようなエラーが出ることがあります。

/var/lib/pgsql/data を削除できません - ディレクトリが空でありません

これは、ユーザが作成したデータや設定ファイルなどは削除しないようにガードがかかっ ているためです。必要なものはバックアップを取ってから、手作業で削除してください。 きれいに削除されたか、もう一度確認します。なにも出力されなくなれば大丈夫です。

root@star:~# rpm -qa|grep postgres root@star:~# rpm -qa|grep apache root@star:~# rpm -qa|grep php





この章では、PostgreSQL+Apache(DSO)+PHP3国際化パージョン(Apacheモジュ ール)という組み合わせのインストールを行います。



#### **2.1.1** 専用アカウントの作成

PostgreSQLはセキュリティ保護のために、root権限では起動しないようになっています。 また、インストールやデータベースの作成といったPostgreSQLの運用管理をする場合にも 専用のアカウントで行うことが推奨されています。ここでは慣例にしたがって、postgresと いう一般ユーザをadduserコマンドで作成し、このアカウントでインストール作業のほとん どを行っています。

root@star:~# adduser postgres root@star:~# passwd postgres Changing password for user postgres New UNIX password: (パスワード) Retype new UNIX password: (パスワード) passwd: all authentication tokens updated successfully

#### 2.1.2 作業用ディレクトリの作成

ソースの展開用およびインストール用としてディレクトリを作成します。

```
root@star:~# mkdir -p /usr/local/src/postgresql-7.0.2
root@star:~# chown postgres /usr/local/src/postgresql-7.0.2
root@star:~# mkdir -p /usr/local/pgsql
root@star:~# chown postgres /usr/local/pgsql
```

基本構成のインストール



本書付属のCD-ROMをマウントします。Vine Linuxでは以下の方法でマウントできますが、ほかのプラットフォームでは異なる場合もあります。

root@star:~# mount /mnt/cdrom

これ以降、CD-ROMの基点は/mnt/cdromであることを前提に作業します。必要に応じて 適宜読み替えてください。

#### **7.1.4** ソースの展開とパッチの適用

これ以降はユーザpostgresで作業を行います。

```
root@star:~# su - postgres
postgres@star:~$ cd /usr/local/src/
postgres@star:/usr/local/src$ tar xvzf /mnt/cdrom/arc/postgresql-7.0.2-patched.tar.gz
postgres@star:/usr/local/src$ zcat /mnt/cdrom/arc/psqlj-7.0.2.diff.gz | patch -p0
```

最後のステップはオプションです。このパッチにより、会話型SQLクライアントである psqlの日本語版psqljが作られます。

#### **2.1.5** configure の実行

ほかの多くのオープンソース・ソフトウェアと同様に、PostgreSQLでもconfigureスクリ プトが用意されています。これは各種プラットフォーム間の差異を吸収し、コンパイル (make)の際に必要となる適切なMakefile(GNUmakefile)を自動生成してくれるものです。 メジャーなプラットフォームの場合は以下のオプションだけでGNUmakefileが作られます。

postgres@star:/usr/local/src\$ cd postgresql-7.0.2/src/ postgres@star:/usr/local/src/postgresql-7.0.2/src\$ ./configure --enable-multibyte=EUC\_JP

> --enable-multibyte=EUC\_JPオプションは、多バイト拡張機能を有効にします。 PostgreSQLのバージョン6.4からは、データベースの作成時に明示的に文字コードを指定す ることができるようになりました。このオプションでは、コード指定がない場合のデフォル ト文字コードとしてEUCを指定しています。残念ながらconfigureでエラーとなってしまっ

た場合、--with-templateオプションを使って明示的にプラットフォームを指定してみるとう まくいくかもしれません。詳細は./configure --helpで表示されるヘルプを参照してください。

#### **2.1.6** コンパイルおよびインストール

コンパイルを行います。マシンスペックにより数分から数十分かかります。

postgres@star:/usr/local/src/postgresql-7.0.2/src\$ make (中略)

All of PostgreSQL is successfully made. Ready to install.

/usr/local/pgsql配下にバイナリをインストールします。

postgres@star:/usr/local/src/postgresql-7.0.2/src\$ make install

/usr/local/pgsql/doc 配下にhtml形式のマニュアルをインストールします。

```
postgres@star:/usr/local/src/postgresql-7.0.2/src$ cd ../doc/
postgres@star:/usr/local/src/postgresql-7.0.2/doc$ make install
```

以上でインストールは完了です。なお、添付CD-ROMの/docs/pgsql配下には日本語マニュアルも入っています。

#### 2.1.7 環境設定

管理者ユーザであるpostgres、およびPostgreSQLを利用するすべてのユーザについて、 コマンドサーチパスと環境変数の設定を行います。シェルとしてbashを使っている場合は /.bashrcに、csh / tcshを使っている場合は /.cshrcに以下の設定を追加します。

#### ● /.bashrc への追加設定

PATH="\$PATH":/usr/local/pgsql/bin export POSTGRES\_HOME=/usr/local/pgsql export PGLIB=\$POSTGRES\_HOME/lib export PGDATA=\$POSTGRES\_HOME/data export MANPATH="\$MANPATH":\$POSTGRES\_HOME/man export LD\_LIBRARY\_PATH="\$LD\_LIBRARY\_PATH":"\$PGLIB" ● /.cshrc への追加設定

```
set path = ($path /usr/local/pgsql/bin)
setenv POSTGRES_HOME /usr/local/pgsql
setenv PGLIB $POSTGRES_HOME/lib
setenv PGDATA $POSTGRES_HOME/data
if ($?MANPATH) then
    setenv MANPATH "$MANPATH":$POSTGRES_HOME/man
else
    setenv MANPATH $POSTGRES_HOME/man
endif
if ($?LD_LIBRARY_PATH) then
    setenv LD_LIBRARY_PATH "$LD_LIBRARY_PATH":"$PGLIB"
else
    setenv LD_LIBRARY_PATH $PGLIB
endif
```

source /.bashrc(またはsource /.cshrc)を実行して、設定を反映させます。 このあと実際にいくつかのプログラムを起動します。各プログラムには多くの引数が用意 されていますが、ここでは動作確認に必要な最小限の使用にとどめます。

#### 7.1.8 データベースの初期化

データベースを使用する前に、データベースの入れ物を用意する作業を行います。具体的には\$PGDATAディレクトリに、管理用データベースとユーザデータベースのひな型をコピーします。

postgres@star:~\$ initdb

なお、initdbを起動したユーザ(通常はpostgres)がそのデータベース領域の所有者となり ます。このあとユーザが作成するデータベースやテーブルの実体は、特に指定しない限り \$PGDATA/base 配下に置かれます。

「**7.1.9**」postmasterの起動

PostgreSQLのデーモンプログラムであるpostmasterを起動します。postmaster コマンドで有効なオプションを以下に示します。

Usage: postma -B nbufs -D datadir -S -a system -b backend -d [1-5] -i -N nprocs -n -o option	ster [options] 共有バッファ数のセット データディレクトリのセット サイレントモード(ttyから切り離す) この認証システムを使用する 指定したバックエンドサーバを使用する デバッグレベルの指定 UNIX ソケットに加え TCP/IP ソケットからの接続を受けつける バックエンドの最大数(11024, デフォルト 32) 異常終了後に共有メモリを再初期化しない 各バックエンドサーバへ 'option' を渡す 接続を待つポート委号
-o option -p port	各バックエンドサーバへ 'option' を渡す 接続を待つポート番号
-5	バックエンドのうちーつが死んだらほかのバックエンドにもSIGSTOPを送る

実際には、postmaster プログラムの起動スクリプトpg\_ctlを使って起動すると便利です。 pg\_ctlの起動オプションを以下に示します。

pg\_ctl [-w][-D DIR][-p PATH] [-o "OPTS"] start
pg\_ctl [-w][-D DIR][-m s|f|i] stop
pg\_ctl [-w][-D DIR][-m s|f|i] [-o "OPTS"] restart
pg\_ctl [-D DIR] status

-h help -w -D DIR -p PATH -m s f i	ヘルプを表示する 既存のpostmaster プロセスが終了するのを待ってから起動する \$PGDATA ディレクトリを指定する postmaster が存在するパスを指定する シャットダウン時のモード指定
	s(mart) スマートなシャットダウン(SIGTERMを送る) f(ast) 早いシャットダウン(SIGINTを送る)
	i(mmediate) 即時シャットダウン(SIGQUITを送る)
-o "OPTS"	postmaster プログラムに渡す起動時オプションを指定する
start	postmasterを起動する
restart	postmasterを再起動する
stop	postmasterを終了する
status	postmasterの動作状況を得る

- o オプションが指定されない場合、pg\_ctlはpostmasterに渡すべきオプションを /usr/local/pgsql/data/postmaster.opts.defaultファイルから読み込みます。

chmod +w /usr/local/pgsql/data/postmaster.opts.default

としてpostmaster.opts.defaultを書き込み可能としてから、適当なエディタでこのファイル にデフォルトのオプションを書き込んでおきましょう。一例を挙げます。

-s WebサーバとDBサーバが同一マシンの場合。postmasterが動作しているホストからの接続のみを許可する -s -i 他のホストからの接続を許す場合

そしてpg\_ctlコマンドでpostmasterを起動します。

postgres@star:~\$ pg\_ctl -w start
Waiting for postmaster starting up...done.
postmaster successfully started up.

7.1.10 アクセス制御

postmasterの起動時オプションに-iをつければ別ホストからの接続を許可するようにな りますが、さらにアクセス制御ファイル/usr/local/pgsql/data/pg\_hba.confを適切に設 定する必要があります。別ホストからの接続を行わない場合は、デフォルトのままでかまい ません。

とりあえずどこからでもアクセスできるようにしたい場合は、140行目付近にある

	#host	all	0.0.0.0	0.0.0.0	trust	
		のコメントをl trustのとこ ます。	はずしてください。 ころをpasswdもしくに	tcryptに代えると、パ	スワード認証ができるように	なり
		●passwd 普通のファ pg_passwd(1) ●crypt PostgreSC alter_user(I)n	イルにユーザ名とパス man ページを参照して にのデータベースでパ nan ページを参照してく	、ワードを書いておく方 ください。 ペスワードを管理する方 ください。	法。詳しくはpg_hba.conf( 法。詳しくはpg_hba.conf(	(5)と
30	0)	Pa	rt-4	インス	ストール	

#### 2.1.11 regression test(回帰テスト)

次に、PostgreSQLがシステムに正しくインストールされたかどうかを確認するための regression testを行います。regression testは標準SQL機能に加え、PostgreSQLで用意さ れている拡張機能のテストも行います。

```
postgres@star:~$ pg_ctl -w start(postgresが動作していない場合のみ)
postgres@star:~$ cd /usr/local/src/postgresql-7.0.2/src/test/regress/
postgres@star:/usr/local/src/postgresql-7.0.2/src/test/regress$ make all runtest
```

このあと数行のメッセージの後テストが開始され、順次結果が表示されます。テスト完了 までに手元のマシンでは15分ほどかかりました。

上記のように各機能に対しての動作チェックが行われ、逐次okまたはfailedが表示されま す。この表示結果は、テスト終了後にregress.outというファイルに保存されます。これで全 項目がokとなれば申し分ありませんが、そうでないこともあるかもしれません。参考までに、 手元の環境(Vine Linux 2.0)ではすべてokになりました。

テストが失敗する原因は、テスト方法に起因するものがほとんどです。regression test で は、結果として期待される文字列をテキストファイル(expected/\*.out)として前もって用 意しておき、これらと実行結果をファイルに落としたものとの単純な差分を取ることでokか どうかを判定しています。差分が発生したらその項目をfailedとするとともに、差分をregre ssion.diffsというファイルに保存します。このファイルをみてfailedが妥当であるかどうか を判断します。failedになる典型的なパターンとしては以下のものが挙げられます。いずれも プラットフォーム間の差異に起因するものです。

- ・エラーメッセージの違い(perror(3)の出力)
- ・浮動小数点の表現方法の違い
- ・乱数を使ったテストにおける違い

詳細は同ディレクトリにあるREADMEファイルを参照してください。

# **2.2 Apache**のインストール

ここから先は再びroot権限で作業を行います。

```
root@star:~# cd /usr/local/src/
root@star:/usr/local/src# tar xvzf /mnt/cdrom/arc/apache_1.3.12.tar.gz
root@star:/usr/local/src# cd apache_1.3.12/
root@star:/usr/local/src/apache 1.3.12# OPTIM="-O2" ./configure --enable-module=so
root@star:/usr/local/src/apache 1.3.12# make
root@star:/usr/local/src/apache 1.3.12# make install
    _____
You now have successfully built and installed the
Apache 1.3 HTTP server. To verify that Apache actually
works correctly you now should first check the
(initially created or preserved) configuration files
   /usr/local/apache/conf/httpd.conf
 and then you should be able to immediately fire up
 Apache the first time by running:
   /usr/local/apache/bin/apachectl start
 Thanks for using Apache.
                               The Apache Group
                               http://www.apache.org/
```

このように表示されればApacheのインストールは成功です。ではApacheを起動してみましょう。

root@star:/usr/local/src/apache\_1.3.12# /usr/local/apache/bin/apachectl start /usr/local/apache/bin/apachectl start: httpd started

> 手元のブラウザから、http://localhost/にアクセスしてみましょう。図4-1の画面が表示 されたらインストール成功です。





PHP3の国際化バージョンをインストールします。次章ではPHP4のインストールを行いま すが、PHP3とPHP4は混在可能です。ここではPHPをApacheのDSO(Dynamic Shared Object)としてインストールします。

```
root@star:/usr/local/src/apache_1.3.12# cd ..
root@star:/usr/local/src# tar xvzf /mnt/cdrom/arc/php-3.0.15-i18n-ja.tar.gz
root@star:/usr/local/src# cd php-3.0.15-i18n-ja/
root@star:/usr/local/src/php-3.0.15-i18n-ja# ./configure ¥
    --with-pgsql --enable-track-vars --with-apxs=/usr/local/apache/bin/apxs ¥
    --enable-i18n --enable-mbregex --enable-versioning --without-gd
root@star:/usr/local/src/php-3.0.15-i18n-ja# make
root@star:/usr/local/src/php-3.0.15-i18n-ja# make
```

これで実行イメージがコピーされました。後は、アクセスされるコンテンツに関する拡張 子の関連づけを行います。/usr/local/apache/conf/httpd.confの中に

#AddType application/x-httpd-php3 .php3
#AddType application/x-httpd-php3-source .phps

という2行があるので、行頭の #( コメント行)をはずしてください。これによって、.php3 と いうファイルがアクセスされると、ApacheによりPHP3スクリプトであると判断され、 PHP3実行ルーチンに処理が移ります。また、PHP3スクリプトに.phpsという拡張子をつけ てプラウザからアクセスすると、PHP3ソースをカラーで見やすく表示してくれるようにな ります。

これで設定は終了です。Apacheを再起動してみましょう。

root@star:~# /usr/local/apache/bin/apachectl restart
/usr/local/apache/bin/apachectl restart: httpd restarted

環境によっては、

Syntax error on line 203 of /usr/local/apache/conf/httpd.conf: Cannot load /usr/local/apache/libexec/libphp3.so into server: Can't find shared library "libpq.so.2.0" /usr/local/apache/bin/apachectl start: httpd could not be started

> というエラーが表示されるかもしれません。これは、PHP3の実体であるlibphp3.soが、 PostgreSQLのAPIライブラリlibpq.so.2.0を読み込めなかったことを示しています。この 場合、以下のいずれかを試してみてください。

● LD\_LIBRARY\_PATHを指定する

bash の 場合

root@star:~# LD\_LIBRARY\_PATH=/usr/local/pgsql/lib ¥
 /usr/local/apache/bin/apachectl start

csh/tcsh の場合

root@star:~# env LD\_LIBRARY\_PATH=/usr/local/pgsql/lib ¥
 /usr/local/apache/bin/apachectl start

Part-4

● /etc/Id.so.conf に以下の1行を追加する

/usr/local/pgsql/lib

そのあとでIdconfig コマンドを実行します。

次にPHP3のサンプルを表示することにより動作を確認します。/usr/local/apache/ htdocs/phpinfo.php3という名前のスクリプトをエディタで作ってみてください。中身は 以下の1行のみです。

<? phpinfo(); ?>

できたら、ブラウザからこのスクリプトを呼び出してみます。

http://localhost/phpinfo.php3

図4-2の画面が表示されれば、インストールは成功です。

	¥I								PICE.
Drak	編集 教社 ジャ	-27 CB	nenario201						340
4	= 3	11 2	4	4	16	3	- H		N
6++	- 77 - 7 A 400	hereite	e der	eres.	11.00	. 86.755	11.2		
		Pot on		2000000	10.112	0.022	22 20 M 2	57	
			PHP V	ersi	on 3.	0.15-	i18n-ja	<u>9</u>	
by Rasma	a Levent And Do	iners 240	re Darashi, St.	te Banke	e. Then	e Cararas	Jan Wantes	1 and counters other	10 A
20111	Savo savo se os	2022240	e-county)	ditati		120232	201012122		52
	System Li	inkasi ether 30	antware.co.	022.14	11/6 #	17ue Ma	14 09:55:12.2	ST 2000 1586 unknow	ai (
				Bullet	Date Ja	121 2000			
				E	intens	ions			
	Externa	anu -		E	xtens Ad	ions Withmul	Information		1
	Extern PHP cure	iona i	CITLAGE- HORBOEX-	E	A/	tons Wittend	Information		
	Extens PHP care Basic Fares	kets	CPLAGE+ HRRBOEK+ No oddikes	E	A/	ions Vittered	nternation		1
	Extens PHP care Book Fanes PHP_DL	ikens Gieres	CPLAGD- HORBOEX- No oddiana Dynamic Lib	E Ibdanoa rary equ	Ad Ad Gots Fort ead	ions Mittimul Sied	Information		
	Extense PHP cure Book Parset PHP_DL PHP_dr	ketu Genes	CPLAGE= HSPBOEE= No addisse Dysenic Lib No addisse	E Isdanoa racy e4a isdanoa	Adens Ad tion.	ions Withmul Sled	Information		
	Externa PHP care Book Paras PHP DL PHP dar PHP filesta	kets Genes	CIFLAGE = HBRBOEL = No additana No additana No additana No additana	E Isdanov Isdanov Isdanov	Ad Ad fort fort fort fort	ions Mittend   sled	information		
	Extense PHP care Basic Parace PHP DL, PHP da PHP filesta PHP files	kens Genes	CPLAGE= HBROEE= No oddiana Dynamic Lib No oddiana No oddiana No oddiana	E Isdansa recy equ Isdansa Isdansa Isdansa	Ad Ad tota part cad tion. Son. tion.	ions Wittend	Information		
	Extern PHP care Basic Paract PHP DL PHP dar PHP dis PHP file PHP file	ikens Genes	CPLAGE= HRROEI= No addiana Dynamic Lib No addiana No addiana No addiana No addiana	E Islanov Islanov Islanov Islanov Islanov	Ad Ad toos toos toos toos toos toos	ions Withmul Sied	Information		
	Externs PHP cure Basic Paract PHP DL PHP da PHP dasta PHP filesta PHP filesta Sembasii	kets Geres	CPLAGE- HORBOEX- No oddiana Dynamic Lib No oddiana No oddiana No oddiana No oddiana	E Isdanos Isdanos Isdanos Isdanos Isdanos Isdanos Isdanos	Ad ad both point ead tran. both t	ions Mitimul Sied	information		
	Externs PHP cure Basic Parast PHP DL PHP dir PHP dir PHP dist PHP file Serrhauf Systeg	keu Gens	CPLAGE= HOROSES= No oddiana Dynamic Lib No oddiana No oddiana No oddiana No oddiana No oddiana	E Lidanos (ety 44) Lidanos Lidanos Lidanos Lidanos Lidanos Lidanos	Ad ton ton tion tion tion tion tion	ions Mitimul Sied Ln/zonz	information		

Chapter - 2

基本構成のインストール

なお、PHP3にはphp3.iniという設定ファイルが存在します(存在しない場合はデフォルトの動作をします)。PHP3のデフォルトの動作を変更したい場合は、以下のように設定ファイルを作成しておいてください。

root@star:~# cp /usr/local/src/php-3.0.15-i18n-ja/php3.ini-dist ¥
/usr/local/lib/php3.ini

php3.iniについての内容は、第3部「4.2 PHPオプションおよび情報」を参照してください。



マシンをブートする際に自動的にPostgreSQL+Apache+PHPを起動するには、以下の設 定を行います。ここはrootで作業してください。 /etc/rc.d/rc.localに、以下の設定を追加します。

```
if [ -x /usr/local/pgsql/bin/pg_ctl ]; then
    su postgres -c "/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start"
fi
if [ -x /usr/local/apache/bin/apachectl ]; then
    /usr/local/apache/bin/apachectl start
fi
```

このあとリブートを行い、再起動後正常にpostmasterとhttpdが起動しているのを確認しておきましょう。



この章では、第2章でインストールした内容に加え、PHP4のApacheモジュール、PHP3 のコマンドラインバージョン、およびPHP4コマンドラインバージョンのインストールについ て説明します。なお、第2章までの作業はすべて完了しているものとします。

本章における作業は、特に必須ではありません。たとえばPHP4を組み込まなければ、本書 で紹介した機能のうち、セッション管理などPHP4で取り入れられた新機能が使えないに過ぎ ません。

# 3.1 рнр4

PHP4に関しては、国際化バージョンの開発が始まったばかりということもあり、まだ日本 語関係のモジュールがそろっていません。ここではオリジナルのPHP4をインストールする 手順を紹介します。PHP4用の国際化関連アドインモジュールについては、第3部「Chapter-7 国際化(日本語)関連」を参照してください。

インストールの手順は以下の通りです。

```
root@star:/usr/local/src# tar xvzf /mnt/cdrom/arc/php-4.0.1pl2.tar.gz
root@star:/usr/local/src# cd php-4.0.1pl2
root@star:/usr/local/src/php-4.0.1pl2# ./configure ¥
    --with-pgsql --enable-track-vars --with-apxs=/usr/local/apache/bin/apxs ¥
    --enable-versioning --without-gd --enable-trans-sid
root@star:/usr/local/src/php-4.0.1pl2# make
root@star:/usr/local/src/php-4.0.1pl2# make install
```

次に、/usr/local/apache/conf/httpd.confにある

```
#AddType application/x-httpd-php .php
```

という行のコメントをはずして有効にします。PHP4におけるスクリプトの拡張子は.phpとなります。最後にApacheを再起動します。

### root@star:~# /usr/local/apache/bin/apachectl restart /usr/local/apache/bin/apachectl restart: httpd restarted

PHPのサンプルを表示することによりPHPの動作を確認します。前項で作成した/usr/ local/apache/htdocs/phpinfo.php3をphpinfo.phpにコピーしてアクセスするだけ です。

プラウザからこのスクリプトを呼び出してみます。         http://localhost/phpinfo.php         図4-3 の画面が表示されれば、インストールは成功です。         図4-3 phpinfo()         プライク・クーン・         プライク・         プライク・     <	 			
http://localhost/phpinfo.php B4-3 maintification and a standard a	ブラウサ	「からこのスクリプ	トを呼び出してみます。	
B4-3 memory Sequences         B4-3 memory         B4-3 memory         B4-3 memory         B4-4 memory         B4-5 memory         B4-6 memory         B4-7 memory         B4-7 memory         B4-8 memory         B4-9 memory	http:/	//localhost/p	hpinfo.php	
Bd + 3       phpinfo()         Statute       Statute         Statute       Statute <td< td=""><td>図4-30</td><td>D画面が表示されれ</td><td>ば、インストールは成功です。</td><td></td></td<>	図4-30	D画面が表示されれ	ば、インストールは成功です。	
27+14       94       9-2       4       9<	図 4 - 3	phpinfo()		
Image: Source and Source	Trak #	幕 青市 ジャンプ Core	101C201	- 11
PHP Version 4.0.1pl2		3 1) 2 RES 1- 00	र्द्ध को दी ही। तरह प्रथा सब का तर	1
PHP Version 4.0.1pl2         System       Linear star Handbourschen jul 3.2.1.0-1.000 F1 The Mar 14.09-55.12.25T         System       Mailel Date         Builel Date       M.2.1.2000         Configure Command       '	1-1-7+2	2-2 Ann frances	outstand sea	_
System     Linux rtr: Iberthvers on jr 22.1.0 - 106 #1 The Mar 1409-55.12 #37 2000 (566 withouts)       Build Date     3421500       Configure Command     '/centigure''with-pgsg''mixitle-track-track''       Server API     Apachs       Virius I Directory Support     firshied       Configure File     Authout[16]       Optimitien File     Authout[26]       Thread Safety     direkted       Thread Safety     direkted		PHP Version 4	.0.1pl2 php	
Build Date     34 21 2001       CortIgare Command     '/contigare''wrth-pgsqi''embite-track-mari'       'Wth-epose-techter/bit/00/000''+-estable-tectioning'       Server API     Apache       Virial Directory Support     ticabled       CortIgare Line     //contigare''       This programmates use of the Zead scripting language regime     ticabled       This programmates use of the Zead scripting language regime     ticable		System	Lanas etar 18aafiwaro.co.ju 2.2.18–1946 #1 Tue Mar 1409:55:12.857 2000 1596 xxkxewa	
Configure Command     '/configure''with-pgsq''emilite-track-mar'       'with-epice-(un/locality)) ethe/(datagon''emilite-track-marg'       Server API     Apacha       Virial Directory Support     Excited       Configuration File     Autoboolitie       (php.int) Path     Excited       ZEND_DETHUG     Excited       Thread Safety     Excited       This programmelies use of the Zend scorpting language regime     Excited		Build Date	34 21 2000	
Server API Apachs Virnal Directory Support Excited Configuration File Autofocolds (php.ind) Path ZEND_DETIDG distand Thread Safety distand This programmales use of the Zend scripting largage regime Zend Engue v1.0.1, Crippingt (c) 1998–2000 Zend Technologies		Configure Command	"/centlgars"	
Virial Directory Support Stabled Configuration File /unfoceLile (php.int) Path ZEND_DEBUG distribut Thread Safety Gistribut This programmelies use of the Zend scripting language regime Zend Engua v1.0.1, Crepring (c) 1996–2000 Zend Technologies		Server API	Apacha	
Configuration File Autolocal En- (php.int) Path ZEND_DETIDG Resided Thread Safety Resided This programmakes use of the Zend scripting language regime Zend Engine v1.0.1, Crepright (c) 1996–2000 Zend Technologies		Virtual Directory Suppo	a disting	
ZEND_DEHUG ziszkied Throad Safety diskked This programmoles use of the Zeni scopteg language regime Zend Begne v1.0.1, Crepelgit (c) 1996–2000 Zeni Technologies		Configuration File (php.ini) Path	/un/scalds	
The program invites use of the Zead scorpting language regime Zead Bagne v1.0.1, Crepringt (c) 1996-2000 Zead Technologies		ZEND_DEBDG	ficabled	
This programmakes use of the Zead scripting language engine Zead Engine v1.0.1, Capyright (c) 1995-2000 Zead Technologies		Thread Safety	formed	
		This programmakes use of Zand Engine v1.0.1, Copyri	One Zead scoping language regime get (c) 1995-2000 Zead Technologies	
			Part of the second s	

PHP4にはphp.iniという設定ファイルが存在します(存在しない場合はデフォルトの動作をします)。PHP4のデフォルトの動作を変更したい場合は、以下のように設定ファイルを作成しておいてください。

root@star:~# cp /usr/local/src/php-4.0.1pl2/php.ini-dist ¥
/usr/local/lib/php.ini

php.iniについての内容は、第3部「4.2 PHPオプションおよび情報」を参照してください。



第1部のコラム「コマンドライン版PHP」でも紹介しましたが、PHPのコマンドラインバー ジョンはCGI プログラムとして動作させることができます。また Apache とは独立したバイ ナリのため、たとえばシェルスクリプトのような使い方をしたり、Cron<sup>\*2</sup> と組み合わせて定 期的に作業を行わせるなど、Web 以外のアプリケーションにも使えます。インストール方法 は以下の通りです。

レNIXにおけるジョブスケジュ ーラ。

root@star:~# cd /usr/local/src/php-3.0.15-i18n-ja
root@star:/usr/local/src/php-3.0.15-i18n-ja# make distclean

make distcleanは、前回指定したconfigure環境(ここではApacheモジュールとして構築 するための情報)をクリアするコマンドです。

```
root@star:/usr/local/src/php-3.0.15-i18n-ja# ./configure ¥
    --with-pgsql --enable-track-vars --enable-i18n --enable-mbregex ¥
    --enable-force-cgi-redirect --without-gd
root@star:/usr/local/src/php-3.0.15-i18n-ja# make
root@star:/usr/local/src/php-3.0.15-i18n-ja# cp php /usr/local/bin/php3
```

これでインストールは終了です。ここでは後述のPHP4とコンフリクトしないように、 php3というコマンド名でインストールしました。では動作確認を行ってみましょう。

```
hotta@star:~$ echo '<? echo date("Y/m/d H:i:s "),"¥n"; ?>' | php3
X-Powered-By: PHP/3.0.15
Content-type: text/html
```

2000/08/03 17:46:49

正しく現在の時刻が表示されたでしょうか? このように、コマンドラインバージョンは フィルタとしても使用できます。



PHP4のコマンドラインバージョンの特徴は、前項のPHP3とほとんど同じです。こちらは php4というコマンドでインストールしてみましょう。

```
root@star:~# cd /usr/local/src/php-4.0.1pl2/
root@star:/usr/local/src/php-4.0.1pl2# make distclean
root@star:/usr/local/src/php-4.0.1pl2# ./configure ¥
                --with-pgsql --enable-track-vars --enable-force-cgi-redirect ¥
                --without-gd
root@star:/usr/local/src/php-4.0.1pl2# make
root@star:/usr/local/src/php-4.0.1pl2# make install
root@star:/usr/local/src/php-4.0.1pl2# cd /usr/local/bin/
root@star:/usr/local/src/php-4.0.1pl2# cd /usr/local/bin/
```

これでインストールは終了です。では動作確認を行ってみましょう。

```
hotta@star:~$ echo '<? echo date("Y/m/d H:i:s "),"¥n"; ?>' | php4
X-Powered-By: PHP/4.0.1pl2
Content-type: text/html
```

2000/08/03 17:46:49

正しく現在の時刻が表示されたでしょうか? 出力はPHP3とほとんど変わりませんが、 HTTPヘッダを見るとphp4が動いていることがわかります。