FreeBSD をゼロから設定するには

Jens Schweikhardt <schweikh@FreeBSD.org >

改訂: 43184

製作著作 © 2002, 2003, 2004 Jens Schweikhardt

FreeBSD は The FreeBSD Foundation の登録商標です。

Adobe, Acrobat, Acrobat Reader および PostScript はアメリカ合衆国および/またはその他の国の Adobe Systems Incorporated の登録商標または商標です。

製造者および販売者が製品を区別するのに 用いている表示の多くは、商標とされています。この文書に登場する表示のうち FreeBSD Project がその商標を確認しているものには、その表示に続いて「#」または「®」記号がおかれています。

2013-11-13：hrs.

概要

この記事は、「FreeBSD をゼロから設定する (FreeBSD From Scratch)」という、わたしの個人的な経験をまとめたものです。カスタマイズした FreeBSD システムをソースからコンパイルし、さらに好みの ports のコンパイルして、あなたが望む構成のシステムの、完全に自動化されたインストールを実現します。`make world` がすばらしい考え方だとお思いの方にとって、「FreeBSD をゼロから設定する」は、まさに `make world` を `make evenmore` (さらにその先) へと広げるものになることでしょう。

目次

1. はじめに 2
2. どうして「FreeBSD をゼロから設定する」(あるいは「〜しない」) ことが必要なのか 2
3. 前提とする環境 4
4. 第 1 段階: システムのインストール 4
5. 第 2 段階: ports のインストール 11
6. 第 3 段階 14
7. 制限事項 15
8. ファイル 16

1. はじめに

今までに `make world` を使ってシステムをアップグレードした経験はあるでしょうか? もしディスクに一つのシステムしか入れていない場合は問題です。`installworld` が途中で止まってしまったら、 あなたのシステムは壊れたまま、もう起動しなくなってしまうかも知れません。あるいは、`installworld` が正常に終了しても、新しいカーネルは起動に失敗してしまうかも知れません。さて、そうなってしまったら、Fixit CD を取り出して半年前のバックアップを戻す、なんてはめになってしまうかも知れませんよね。

わたしは、「アップグレードの時はディスクを初期化する」 という方法がよいと考えています。パーティションではなくディスク全体のデータを 消去することで、アップグレードの手順では無視されるような古いデータが 残ってしまうことを防ぐことができます。ただ、パーティションを全部初期化するということは、 ports/packages をすべて再コンパイル・再インストールしなければならず、 設定ファイルも注意深く作成し直さなければならないということです。こういう作業を自動化したいと思いませんか? そう思う人は、この先を読み進めましょう。

2. どうして「FreeBSD をゼロから設定する」(あるいは「〜しない」) ことが必要なのか

これはもっともな質問です。 すでに sysinstall がありますし、 カーネルとユーザランドツールをコンパイルする方法には、もっと有名な方法が他にもあるからです。

sysinstall の問題は、「何を、どこに、 どうやってインストールするのか」が非常に限定されているという点です。

- sysinstall は通常、構築ずみの配布物セットと packages を (CD, DVD, FTP などの) 別の場所からインストールする時に使われるものであり、`make buildworld` の結果をインストールできるようにはできていません。
- 現在稼働中のシステム中にあるディレクトリに、新しいシステムをインストールすることはできません。
- Vinum パーティションへのインストールはできません。
- 構築ずみの packages はインストールできますが、ports を構築することはできません。
- スクリプトを使ったり、 インストール後に変更するための処理を自由に入れることは困難です。
- 最後の大きな理由として、sysinstall が、公式にもう積極的に使わないプログラムと考えられている、ということがあげられます。

システム全体を構築してインストールする方法は、 ハンドブックにある方法が有名です。これはデフォルトで既存のシステムを置き換えるもので、カーネルとモジュールだけが保

存され、 システムバイナリ、ヘッダ、その他の多くのファイルは上書きされます。 使われなくなった古いファイルはそのまま残り、 動作に問題が出ることもあります。 何らかの理由でアップグレードに失敗すると、 システムを元の状態に戻すことは不可能か、できても非常に困難です。

「FreeBSD をゼロから設定する」方法は、これらの問題をすべて解決できます。 考え方は単純です。 稼働中のシステムを使って空のディレクトリにシステムをインストールします。 その時、その新しいシステムのディレクトリツリーには、 新しいパーティションを適切にマウントしておaきます。 数多くある設定ファイルは、コピーできるものは適切な場所にコピーし、 それができないものには mergemaster(8) を使います。 新しいシステムに対するインストール後の設定は、古いシステムを動作させながら、新しいシステムに対して chroot して 自由に行なうことができます。具体的には、シェルスクリプト、もしくは `make` の実行で構成される、次の 3 段階でこれらを実現します。

1. `stage_1.sh` : 新しい起動可能なシステムを空のディレクトリ以下に作成し、必要なファイルをマージ、もしくはコピーします。そして、新しいシステムを起動します。

2. `stage_2.sh` : 必要な ports をインストールします。

3. `stage_3.mk` : ひとつ前の段階でインストールしたソフトウェアの、インストール後の設定を行ないます。

新しいシステムを構築するために「FreeBSD をゼロから設定する」方法を使い、 それが数週間、満足する程度に動作していることを確認したら、 もう一度それを使って、大元のシステムを再インストールすることができます。 これからはいつでも好きな時にシステムを更新して、 初期化・再インストールしたパーティションに切り替えるだけでよくなるわけです。

Linux From Scratch (もしくは省略して LFS) について耳にしたり、試された方がいらっしゃるかも知れません。LFS も同じように、稼働中のシステムを使ってシステムをゼロから構築し、 空のパーティションにインストールする方法が書かれています。LFS が話題の中心としているのは、(カーネル、コンパイラ、デバイス、 シェル、端末データベースなどの) 各システムコンポーネントの役割と、 それらのインストールの詳細を見せることのようです。 この「FreeBSD をゼロから設定する」では、そのような詳細には触れません。わたしの目的は、インストールを終わりまで自動化することであり、 システム構築時の泥くさい過程を全部説明することではありません。FreeBSD をそのようなレベルで掘り下げてみたい人は、`/usr/src/Makefile` を読んで、`make buildworld` の動作を追いかけるところから始めましょう。

また、「FreeBSD をゼロから設定する」方法にも、 次のような欠点があることを心に留めておいてください。

- 第 2 段階で ports をコンパイルしている間、システムは通常の用途に使用することができません。もしプロダクションサーバを運用しているなら、第 2 段階でダウンタイムが発生することを考慮に入れなければなりません。`stage_2.conf.default` の ports のコンパイルには、AMD1800+、10,000rpm SCSI、1GB の RAM を搭載し

たシステムで、約 4 時間かかります。ports の代わりに package をインストールすれば、この時間を 10 分程度にまで大幅に短縮することも可能です。

3. 前提とする環境

「FreeBSD をゼロから設定する」方法を実行するには、次のものが必要です。

- ソースと ports ツリーを含む、稼働中の FreeBSD システム
- 新しいシステムをインストールするための、最低 1 個の未使用パーティション
- mergemaster(8) を実行した経験。もしくは、それを実行する勇気。
- インターネット接続環境がない、あるいは遅い場合には、インストールしたい ports の配布ファイル
- Bourne シェル (sh(1)) を使ってシェルスクリプトを作成するための基礎知識
- 新しいシステムを起動する方法を、対話的あるいは設定ファイルを使ってブートローダに 教えることができること

4. 第 1 段階: システムのインストール

この文書の初版では、第 1 段階にひとつのシェルスクリプトを使っていました。 カスタマイズはすべて、そのスクリプトを編集する必要があったのですが、 利用者からの意見を参考にして、スクリプトのコードとデータを分離することにしました。 そのため新しいスクリプトでは、コードスクリプトを変更せずに、複数の異なるシステムに異なる構成のデータを置くことが可能になりました。

第 1 段階のコードスクリプトは `stage_1.sh` であり、次のように 1 個の引数をつけて実行すると

```
# ./stage_1.sh default
```

設定ファイルとして `stage_1.conf.default` を読み込み、 ログファイルとして `stage_1.log.default` に書き込みます。

文末にわたしが使っている `stage_1.conf.default` が添付してあります。 あなたが考える 「完璧なシステム」 に合わせて、 各設定をカスタマイズしてください。あなたが変更しそうな設定には、 詳細なコメントを追加してあります。設定スクリプトでは、 `create_file_systems`, `create_etc_fstab`, `copy_files`, `all_remaining_customization` という、4 個のシェル関数を提供しなければなりません (これは、`stage_1.sh` から呼ばれる順に書いてあります)。

考慮すべき点は、以下のとおりです。

- パーティションの配置

  わたしは、システム全体を一つの大きな パーティションに入れるという考え方が好きではないので、普通は /、/usr、/var の パーティションを分割し、/tmp を /var/tmp のシンボリックリンクにしています。また、/home (ユーザのホームディレクトリ)、/home/ncvs (FreeBSD CVS リポジトリの複製), /usr/ports (ports ツリー), /src (チェックアウトした src ツリー)、/share (news スプールなど、バックアップする必要がない、その他の共有データ) といったファイルシステムを、古いシステムと新しいシステムで共有しています。

- その他の項目

  これは、新しいシステムの起動後にすぐに実行したいことや、第 2 段階の前に実行したい内容のことです。なぜ第 1 段階で単純に新しいシステムに chroot してお気に入りの ports をインストールしないのかというと、理論的にも、実際の手順にも卵鶏問題があるからです。第 1 段階では、古いカーネルが動作していますが、chroot 環境には、新しいバイナリとヘッダが使われています。もしその新しいバイナリが、古いカーネルには存在しない新しいシステムコールを使っていたら、SIGSYS, Bad system call というエラーが出て停止してしまうでしょう。わたしは lang/perl5 を構築する時にも問題が発生することを確認しています。

stage_1.sh を実行する前に、make installworld installkernel を実行するために通常行なう作業を完了させておいてください。 これらは、たとえば次のようなものです。

- カーネルコンフィグファイルの設定

- make buildworld を正常終了させておくこと

- make buildkernel KERNCONF=whatever を正常終了させておくこと

初めて stage_1.sh を実行した場合は、 稼働中のシステムから新しいシステムへとコピーされる設定ファイルは /usr/src のものと比べると古いので、mergemaster がどうするかを聞いてきます。 おすすめは、ここで変更点を統合しておくことです。 もし、何度も質問に答えるのが面倒であれば、 稼働中のシステムのファイルを更新しておきましょう (ただしこれは、そうできればの話です。-STABLE のシステムを実行していて、-CURRENT を構築する、もしくはその逆のようなケースでは、そうしてはいけません)。次に mergemaster を実行した時、RCS バージョン ID が /usr/src にあるファイルと一致しているものは、処理が飛ばされるようになります。

stage_1.sh スクリプトは set -e が指定されており、最初のコマンドが失敗 (終了コードが 0 以外) すると停止します。そのため、エラーを見逃してしまうということはないでしょう。 これは、タイプミスなどで未定義の変数を使った場合にもエラーになります。次に進む前に、stage_1.conf.default にあるエラーを全部修正しておいてください。

stage_1.sh では mergemaster が実行されます。統合作業をしなければならないファイルが一つもない状態でも、実行の終わりに次のメッセージが表示されます。

```
*** Comparison complete

Do you wish to delete what is left of -/var/tmp/temproot.stage1? ↺
[no] no
```

no と答えるか、単に Enter を押してください。なぜかと言うと、mergemaster は /var/tmp/temproot.stage1 にサイズが 0 のファイルをいくつか残すからです。これは、後で新しいシステムに (存在しなければ) コピーされます。

この後、インストールされたファイルのリストがページャ (デフォルトでは more(1) です。less(1) を使うこともできます) に表示されます。

```
*** You chose the automatic install option for files that did not
    exist on your system.  The following were installed for you:
      -/newroot/etc/defaults/rc.conf
      -...
      -/newroot/COPYRIGHT

(END)
```

q を入力してページャを終了します。すると login.conf に関して、次のように表示されます。

```
*** You installed a login.conf file, so make sure that you run
    -'/usr/bin/cap_mkdb -/newroot/etc/login.conf'
    to rebuild your login.conf database

    Would you like to run it now? y or n [n]
```

これに対する答えはどちらでも構いません。どう答えても、スクリプトから cap_mkdb(1) が実行されます。

次に示すのは、筆者の使っている stage_1.conf.default ですが、たくさんの部分を書き換える必要がありますので注意してください。どこを書き換えればよいのかについては、コメントを読めば十分理解できると思います。

警告

newfs(8) コマンドには注意してください。マウントずみのパーティションに新しいファイルシステムを作成することはできないものの、このスクリプトはマウントされていない /dev/da0s1a，/dev/da0s1e，/dev/da2s1e をすべて削除します。 ひとつ間違え

れば、あなたの環境を破壊してしまう可能性がありますので、 デバイス名の変更は注意深く行なってください。

```
# ######: stage_1.conf.default # stage_1.sh ##########
#
# $FreeBSD: head/ja_JP.eucJP/articles/fbsd-from-scratch/
stage_1.conf.default 38826 2012-05-17 19:12:14Z hrs $
#
# Original Revision: 1.3

# ################################
# ################################
# ################################
# ###############
DESTDIR="/newroot"

# src #########
SRC="/usr/src"

# make buildkernel KERNCONF=... ##########################
KERNCONF="HAL9000"

# ############ -/usr/share/zoneinfo ################
TIMEZONE="Europe/Berlin"

#
# create_file_systems ####DESTDIR ###############
# #############DESTDIR ################
#
create_file_systems () {
  # ######################
  # DEVICE ############################
  # grub ########################
  # newfs --O 1 ############
  DEVICE=/dev/da0s1a
  mkdir --m 755 --p ${DESTDIR}
  chown root:wheel ${DESTDIR}
  newfs --U --O 1 ${DEVICE}
  mount --o noatime ${DEVICE} ${DESTDIR}

  # ##############################
  DEVICE=/dev/da0s1e
  mkdir --m 755 --p ${DESTDIR}/var
  chown root:wheel ${DESTDIR}/var
  newfs --U ${DEVICE}
  mount --o noatime ${DEVICE} ${DESTDIR}/var

  DEVICE=/dev/da2s1e
  mkdir --m 755 --p ${DESTDIR}/usr
  chown root:wheel ${DESTDIR}/usr
```

```
  newfs --U ${DEVICE}
  mount --o noatime ${DEVICE} ${DESTDIR}/usr
}

#
# create_etc_fstab ####create_file_systems ###########
# ######## fstab #############
#
create_etc_fstab () {
  cat <<EOF >${DESTDIR}/etc/fstab
# Device         Mountpoint          FStype    Options              ↺
Dump Pass#
/dev/da0s1b      none                swap      sw                   ↺
0    0
/dev/da1s1b      none                swap      sw                   ↺
0    0
/dev/da2s2b      none                swap      sw                   ↺
0    0
/dev/da3s2b      none                swap      sw                   ↺
0    0
/dev/da0s1a      -/                   ufs       ↺
rw,noatime           1    1
/dev/da0s1e      -/var                ufs       ↺
rw,noatime           1    1
/dev/da2s1e      -/usr                ufs       ↺
rw,noatime           1    1
/dev/vinum/Share -/share              ufs       ↺
rw,noatime           0    2
/dev/vinum/home  -/home               ufs       ↺
rw,noatime           0    2
/dev/vinum/ncvs  -/home/ncvs          ufs       ↺
rw,noatime           0    2
/dev/vinum/ports -/usr/ports          ufs       ↺
rw,noatime           0    2
/dev/ad1s1a      -/flash              ufs       ↺
rw,noatime           0    0
/dev/ad0s1       -/2k                 ntfs      ↺
ro,noauto            0    0
/dev/ad0s6       -/linux              ext2fs    ↺
ro,noauto            0    0
#
/dev/cd0         -/cdrom              cd9660    ↺
ro,noauto            0    0
/dev/cd1         -/dvd                cd9660    ↺
ro,noauto            0    0
proc             -/proc               procfs    ↺
rw                   0    0
linproc          -/compat/linux/proc  linprocfs ↺
rw                   0    0
EOF
  chmod 644 ${DESTDIR}/etc/fstab
  chown root:wheel ${DESTDIR}/etc/fstab
}
```

```
#
# copy_files ####mergemaster ###################
# #####
#
copy_files () {
  # ################################
  for f in \
    -/.profile \
    -/etc/group \
    -/etc/hosts \
    -/etc/inetd.conf \
    -/etc/ipfw.conf \
    -/etc/make.conf \
    -/etc/master.passwd \
    -/etc/nsswitch.conf \
    -/etc/ntp.conf \
    -/etc/printcap \
    -/etc/profile \
    -/etc/rc.conf \
    -/etc/resolv.conf \
    -/etc/start_if.xl0 \
    -/etc/ttys \
    -/etc/ppp/* \
    -/etc/mail/aliases \
    -/etc/mail/aliases.db \
    -/etc/mail/hal9000.mc \
    -/etc/mail/service.switch \
    -/etc/ssh/*key* \
    -/etc/ssh/*_config \
    -/etc/X11/XF86Config-4 \
    -/var/cron/tabs/* \
    -/var/files \
    -/root/.profile \
    -/boot/*.bmp \
    -/boot/loader.conf \
    -/boot/device.hints -; do
    cp --p ${f} ${DESTDIR}${f}
  done
}

#
# ######################
# ##: ################################
# ###################################
# ####################ports ###############
# ##########
#
all_remaining_customization () {
  # compat ##############linux_base #######
  # ####################
  cd ${DESTDIR}
```

```
  mkdir --m 755 usr/compat; chown root:wheel usr/compat; ln --s usr/
compat
  mkdir --m 755 usr/compat/linux;      chown root:wheel usr/compat/
linux
  mkdir --m 555 usr/compat/linux/proc; chown root:wheel usr/compat/
linux/proc
  mkdir --m 755 boot/grub;             chown root:wheel boot/grub
  mkdir --m 755 linux 2k;              chown root:wheel linux 2k
  mkdir --m 755 src;                   chown root:wheel src
  mkdir --m 755 share;                 chown root:wheel share
  mkdir --m 755 dvd cdrom flash;       chown root:wheel dvd cdrom ↺
flash
  mkdir --m 755 home;                  chown root:wheel home
  mkdir --m 755 usr/ports;             chown root:wheel usr/ports

  # ##### tmp # var/tmp ####################
  cd ${DESTDIR}; rmdir tmp; ln --s var/tmp

  # -/etc/printcap ###################
  cd ${DESTDIR}/var/spool/output/lpd; mkdir --p as od ev te lp da
  touch ${DESTDIR}/var/log/lpd-errs

  # -/home ################################
  # mkdir --p ${DESTDIR}/home
  # cd -/home; tar cf -- -. -| (cd ${DESTDIR}/home; tar xpvf --)

  case ${REVISION} in
  4.*)
    # 4.x ## devfs ##############################
    cd ${DESTDIR}/dev
    -./MAKEDEV all
    -./MAKEDEV da0 da0s1h da0s2h da0s3h da0s4h
    -./MAKEDEV da1 da1s1h da1s2h da1s3h da1s4h
    -./MAKEDEV da2 da2s1h da2s2h da2s3h da2s4h
    -./MAKEDEV da3 da3s1h da3s2h da3s3h da3s4h
    -./MAKEDEV bktr0 cd1
    if test --d -/dev/vinum; then
      # -'vinum makedev' # -/dev #################cpio ####
      cd -/dev; find vinum --print -| cpio --pv ${DESTDIR}/dev
    fi

    # floppy # wheel ################
    chown root:wheel ${DESTDIR}/dev/fd0*
    chmod g+w ${DESTDIR}/dev/fd0*
    -;;

  5.*)
    # floppy # wheel ################
    printf -'%s\n' -'own  fd0 root:wheel' >> ${DESTDIR}/etc/
devfs.conf
    printf -'%s\n' -'perm fd0 0660'       >> ${DESTDIR}/etc/
devfs.conf
    -;;
```

```
  *)
    printf -'%s\n' -"REVISION ${REVISION} not supported"
    exit 1
    -;;

  esac
}

# vim: tabstop=2:expandtab:shiftwidth=2:syntax=sh:
# EOF $RCSfile: stage_1.conf.default,v $
```

ダウンロード: stage_1.conf.default .

このスクリプトを実行すると、 起動した時に次のような状態になっているシステムがインストールされます。

- 稼働中のシステムと同じユーザとグループ
- Ethernet と PPP を経由した、ファイアウォールありのインターネット接続環境
- 正しいタイムゾーンと NTP 設定
- /etc/ttys や inetd など、その他の細かな設定。

他の部分に対する設定は、第 2 段階が終わるまで動作しません。 たとえば、プリンタや X11 の設定ファイルもコピーされますが、プリンタは PostScript® ユーティリティなど、ベースシステムに含まれないアプリケーションを使うことが多いでしょう。X11 はサーバ、ライブラリ、プログラムをコンパイルしないと動作しません。

5. 第 2 段階: ports のインストール

注記

この段階で ports をコンパイルするのではなく、 (コンパイルずみの) packages をインストールすることもできます。 その場合、stage_2.sh は 単に pkg_add コマンドを羅列するだけになるでしょう。 読者のみなさんにとって、そういうスクリプトを書くのは難しくないと思いますので、ここではもっと柔軟で、ports を使った伝統的な方法について考えることにします。

次に紹介する stage_2.sh スクリプトは、わたしが好みの ports をインストールするために使ったものです。これは何度でも実行でき、インストールずみの ports があれば、飛ばして処理されます。スクリプトは 実行せず、実行される内容だけ を表示する (dryrun)

オプション (-n) があります。実行時には stage_1.sh と同様、設定スクリプトを示すためのひとつの引数を指定します。

```
# ./stage_2.sh default
```

これは、ports のリストを stage_2.conf.default というファイルから読み込みます。

ports リストは、空白で区切られた 2 個以上のキーワードからなっています。 カテゴリ、port 名に始まり、オプションとして port をコンパイルしてインストールするためのコマンド (デフォルトは make install BATCH=yes < /dev/null) が続きます。空白行と # から始まる行は無視されます。おそらく多くの場合に考えなければならないのは、カテゴリ名と port 名だけでしょう。ports によっては、たとえば次のように make 変数を使って微調整することができます。

```
www mozilla make WITHOUT_MAILNEWS=yes WITHOUT_CHATZILLA=yes install
```

実際には任意のシェルコマンドを指定できますので、make を使う以外にも応用は可能です。

```
java linux-sun-jdk13 yes -| make install
news inn-stable CONFIGURE_ARGS="--enable-uucp-rnews ---enable-setgid-
inews" make install
```

news/inn-stable の行は、CONFIGURE_ARGS という シェル変数を定義した例です。この port の Makefile は、この指定した値を変数の初期値として、その他の必須の引数と一緒に使います。これと

```
news inn-stable make CONFIGURE_ARGS="--enable-uucp-rnews ---enable-
setgid-inews" install
```

のようにして make 変数をコマンドラインに設定した場合との違いは、 こちらの場合に変数そのものを完全に上書きしてしまうという点です。 どの方法を使えばいいのかについては、各 port によります。

インストールしたい ports が、 対話的インストールを使っていないことを確認してください。ports は、あなたが標準入力に明示的に指定したもの以外、標準入力を読み込む動作をしてはいけません。もし ports がそのように作られていると、ports はヒアドキュメントにある ports リストの次の行を読み込んで混乱してしまいます。stage_2.sh を実行した時、ある port が飛ばされたり、動作が止まってしまうようなことがあれば、おそらくこれが原因でしょう。

次に示すのは stage_2.conf.default です。これは、インストールされる port それぞれに対して LOGDIR/category+port という名前のログファイルが作成されます。

```
# vim: syntax=sh
#
```

```
# $FreeBSD: head/ja_JP.eucJP/articles/fbsd-from-scratch/
stage_2.conf.default 38826 2012-05-17 19:12:14Z hrs $
#
# Original Revision: 1.3
#
shells zsh
devel gettext make BATCH=yes install
lang perl5.8 make install; use.perl port
archivers unzip
archivers zip
security sudo
x11-servers XFree86-4-Server
x11 wrapper
x11 XFree86-4-clients
x11 XFree86-4-documents
x11-fonts XFree86-4-font75dpi
x11-fonts XFree86-4-font100dpi
x11-fonts XFree86-4-fontScalable
x11-fonts urwfonts
x11-fonts webfonts make WITH_NETSCAPE_ALIASES=yes install
x11-toolkits open-motif
x11-wm ctwm
security openssh-askpass
astro xplanet
astro xephem
editors vim
print ghostscript-gnu make A4=yes BATCH=yes install
print psutils-a4
print a2ps-a4
print gv
print acroread5
print transfig
print teTeX
# #: jdk14 # linprocfs(5) ######################
#     ########
java linux-sun-jdk14 mount --a linproc; yes -| make install
java jdk14 mount --a linproc; make --DNODEBUG install
www apache2
www weblint
www amaya
www firefox make BATCH=yes install
www mozilla make WITHOUT_MAILNEWS=yes WITHOUT_COMPOSER=yes ↺
WITHOUT_LDAP=yes WITHOUT_CHATZILLA=yes WITHOUT_XMLTERM=yes install
www checkbot
www privoxy
graphics xfig
graphics xv
multimedia xawtv
graphics graphviz
lang expect
lang gawk
lang TenDRA unset MAKEOBJDIRPREFIX; make install
news tin
```

```
net freebsd-uucp
net cvsup-without-gui
net pathchar make NO_CHECKSUM=yes install
ftp wget
textproc ispell
german ispell-neu
german ispell-alt
textproc docproj make JADETEX=no HAVE_MOTIF=yes BATCH=yes install ↺
< -/dev/null
sysutils samefile
sysutils lsof
sysutils pstree
sysutils cdrtools
sysutils grub
sysutils smartmontools
sysutils vobcopy
devel ddd
devel gindent
devel ctags
devel ElectricFence
devel strace
devel perltidy
mail procmail make BATCH=yes install
mail metamail
mail mutt-devel
emulators mtools
sysutils portupgrade
news inn-stable CONFIGURE_ARGS="--enable-uucp-rnews ---enable-setgid-
inews" make install
misc figlet-fonts
security gpa
mail spamoracle
multimedia mplayer make WITHOUT_RUNTIME_CPUDETECTION=yes ↺
WITH_GUI=yes BATCH=yes install
multimedia mplayer-fonts
audio wavplay
games xmahjongg
games xdemineur
editors openoffice-1.1
```

ダウンロード: stage_2.conf.default .

6. 第 3 段階

第 2 段階で、好みの ports がインストールされましたが、ports には、設定を必要とするものがあります。第 3 段階は、インストール後の設定を行なう段階です。stage_2.sh の最後にこの段階を統合することもできたのですが、わたしは port をインストールすることと初期設定を変更することが異なる工程であると考えたため、独立した段階としています。

第 3 段階は、Makefile として実装しています。これは、次のように実行することで、設定対象を簡単に選ぶことができるからです。

```
# make --f stage_3.mk target
```

stage_2.sh の段階で、stage_3.mk を共有パーティションに置くか、新しいシステムのどこかにコピーするなどして、新しいシステムが起動した時に stage_3.mk が使えるようにしておきましょう。

7. 制限事項

対話的で、かつ make BATCH=YES install でのインストールに対応していない port の自動インストールは難しいかも知れません。対話的にインストールする ports には、ライセンス条項の同意を尋ねられた時に yes と入力するだけのものがいくつかあります。 そのように入力が標準入力から読みとられる場合は、 適切な回答をインストールコマンド (通常は make install) にパイプで渡すことができます (わたしが stage_2.conf.default の java/linux-sun-jdk14 でとった方法がそうです)。

しかしこの方法は、たとえば editors/staroffice52 の場合にはうまく動きません。これは X11 が実行されていることを要求するからです。インストール手順には多くのクリックや文字入力が必要なので、他の ports のように自動化することはできません。わたしは、次のようにして問題を回避しました。最初に古いシステムで staroffice の package を作成し、

```
# cd -/usr/ports/editors/staroffice52
# make package
===>  Building package for staroffice-5.2_1
Creating package -/usr/ports/editors/staroffice52/
staroffice-5.2_1.tbz
Registering depends:.
Creating bzip'd tar ball in -'/usr/ports/editors/staroffice52/
staroffice-5.2_1.tbz'
```

その後、第 2 段階で次のようにしたわけです。

```
# pkg_add -/usr/ports/editors/staroffice52/
staroffice-5.2_1.tbz
```

その他に、設定ファイルのアップグレード問題に気をつける必要があります。 一般的に、設定ファイルの書式や内容がいつ変更されるかを知ることはできません。 新しいグループが /etc/group に追加されるかも知れませんし、/etc/passwd に新しいフィールドが追加されるかも知れません。 このような例は、実際に過去にありました。 単純に古いシステムから新しいシステムに設定ファイルをコピーするだけで ほとんどの場合は十分なのですが、時には不都合な場合もあります。 古いファイルを上書きする方法でシステムをアップグレードしたら、ローカルにある設定ファイルに新しく追加されたかも知れない項目を統合する目的で mergemaster を使うと思います。 しかし残念な

ことに、mergemaster はベースシステムに存在するファイルだけで、インストールした ports については何も処理を行なってくれません。サードパーティ製ソフトウェアには、リリースのたびに設定ファイルのフォーマットが変更され、 わたしをイライラさせるようなものもあります。このような予告なしの変更を検出するために、わたしは変更した設定ファイルを stage_3.mk と同じディレクトリにコピーしておき、 make ルールを使って結果を比較しています。 たとえば、apache の httpd.conf であれば、次のような config_apache というターゲットを用意しておきます。

```
@if -! cmp --s -/usr/local/etc/apache2/httpd.conf httpd.conf; then \
    echo -"ATTENTION: the httpd.conf has changed. Please examine ↺
if"; \
    echo -"the modifications are still correct. Here is the diff:"; \
    diff --u -/usr/local/etc/apache2/httpd.conf httpd.conf; \
fi
```

差分が無害なものであると確認できたら、 cp /usr/local/etc/apache2/httpd.conf httpd.conf を実行するわけです。

わたしは 5-CURRENT から 5-CURRENT に更新するために「FreeBSD をゼロから設定する」方法を数回使いましたが、4-STABLE と 5-CURRENT の間で更新を行なった経験はありません。異なるメジャーリリース番号の間は、非常の多数の変更が行なわれているため、更新作業はもっと複雑なものになると思います。(試したわけではないのですが) 4-STABLE から 4-STABLE への更新であれば、「FreeBSD をゼロから設定する」方法は問題なく動作するはずです。4-STABLE のユーザは、次の点を考慮してください。

注記

デバイスファイルシステム devfs(5) を使ってなければ、all_remaining_customization の中で MAKEDEV(8) を使い、ハードウェア用のデバイスファイルを作成するとよいでしょう。

8. ファイル

ここでは、すでに説明した設定ファイルの他に必要な、3 個のファイルを示します。

これは stage_1.sh スクリプトです。内容を変更する必要はないでしょう。

```
#!/bin/sh
#
```

```
# stage_1.sh -- FreeBSD From Scratch, # 1 ##: ###########
#              ###: -./stage_1.sh profile
#              ########: -./stage_1.conf.profile
#              ########: -./stage_1.log.profile
#
# ##:      Jens Schweikhardt
# $Id: stage_1.sh,v 1.2 2006-03-13 16:46:15 rushani Exp $
# $FreeBSD: head/ja_JP.eucJP/articles/fbsd-from-scratch/stage_1.sh ↺
38826 2012-05-17 19:12:14Z hrs $
# Original revision: 1.5

PATH=/bin:/usr/bin:/sbin:/usr/sbin

# #######:
#
# a) -"make buildworld" # -"make buildkernel" #############
# b) ############### (################## 1 ##
#    ###### -/usr # -/var #########)
# c) ######### stage_1.conf.profile #####

if test $# --ne 1; then
  echo -"usage: stage_1.sh profile" 1>&2
  exit 1
fi

# ------------------------------------------------------------------------------ ↺
#
# #### 1: $DESTDIR #################
# ------------------------------------------------------------------------------ ↺
#

step_one () {
  create_file_systems

  # ######################
  cd ${SRC}/etc; make distrib-dirs DESTDIR=${DESTDIR}
}

# ------------------------------------------------------------------------------ ↺
#
# #### 2: -/etc ########## -/ ########
# ------------------------------------------------------------------------------ ↺
#

step_two () {
  copy_files

  # mergemaster ##############
  TEMPROOT=/var/tmp/temproot.stage1
  if test --d ${TEMPROOT}; then
    chflags --R 0 ${TEMPROOT}
    rm --rf ${TEMPROOT}
  fi
```

```
  export MAKEDEVPATH="/bin:/sbin:/usr/bin"
  mergemaster --i --m ${SRC}/etc --t ${TEMPROOT} --D ${DESTDIR}
  cap_mkdb ${DESTDIR}/etc/login.conf
  pwd_mkdb --d ${DESTDIR}/etc --p ${DESTDIR}/etc/master.passwd

  # mergemaster # -/var/log ###################
  # ######### copy_files ###################
  cd ${TEMPROOT}
  find -. --type f -| sed -'s,^\./,,' -|
  while read f; do
    if test --r ${DESTDIR}/${f}; then
      echo -"${DESTDIR}/${f} already exists; not copied"
    else
      echo -"Creating empty ${DESTDIR}/${f}"
      cp --p ${f} ${DESTDIR}/${f}
    fi
  done
  chflags --R 0 ${TEMPROOT}
  rm --rf ${TEMPROOT}
}

# ------------------------------------------------------------------------------ ↺
#
# #### 3: installworld #####
# ------------------------------------------------------------------------------ ↺
#

step_three () {
  cd ${SRC}
  make installworld DESTDIR=${DESTDIR}

  # ################### (#####)#libc.so.4 #
  # ######################
  # -/usr/libexec/ld-elf.so.1: Shared object -"libc.so.4" not found
  # ##############################
  cd lib/compat/compat4x.i386
  make all install DESTDIR=${DESTDIR}
}

# ------------------------------------------------------------------------------ ↺
#
# #### 4: ###################
# ------------------------------------------------------------------------------ ↺
#

step_four () {
  cd ${SRC}
  # installkernel ########loader.conf # device.hints #### ↺
  # #### 2 ############## 2 #############
  #   cp sys/boot/forth/loader.conf ${DESTDIR}/boot/defaults
  #   cp sys/i386/conf/GENERIC.hints ${DESTDIR}/boot/device.hints
  make installkernel DESTDIR=${DESTDIR} KERNCONF=${KERNCONF}
}
```

```
# ------------------------------------------------------------------------- ↺
#
# #### 5: -/etc/fstab ################
# ------------------------------------------------------------------------- ↺
#

step_five () {
  create_etc_fstab

  # ####################
  cp ${DESTDIR}/usr/share/zoneinfo/${TIMEZONE} ${DESTDIR}/etc/
localtime
  if test --r -/etc/wall_cmos_clock; then
    cp --p -/etc/wall_cmos_clock ${DESTDIR}/etc/wall_cmos_clock
  fi
}

# ------------------------------------------------------------------------- ↺
#
# #### 6: #########
# ------------------------------------------------------------------------- ↺
#

step_six () {
  all_remaining_customization
}

do_steps () {
  echo -"PROFILE=${PROFILE}"
  echo -"DESTDIR=${DESTDIR}"
  echo -"SRC=${SRC}"
  echo -"KERNCONF=${KERNCONF}"
  echo -"TIMEZONE=${TIMEZONE}"
  echo -"TYPE=${TYPE}"
  echo -"REVISION=${REVISION}"
  echo -"BRANCH=${BRANCH}"
  echo -"RELDATE=${RELDATE}"
  step_one
  step_two
  step_three
  step_four
  step_five
  step_six
}

# ------------------------------------------------------------------------- ↺
#
# ########
# ------------------------------------------------------------------------- ↺
#

PROFILE="$1"
```

```
set --x --e --u # ########################
. -./stage_1.conf.${PROFILE}

# world # make ############################
# ########## 4.x # 5.x #################
# ###################RELDATE ####
# __FreeBSD_version # Port ############ (Porter's Handbook)
# #########
# doc/en_US.ISO8859-1/books/porters-handbook/freebsd-versions.html
# ########################
# doc/ja_JP.eucJP/books/porters-handbook/freebsd-versions.html
# ####<######><###### 2 #><########## 0, ##### 1>xx
# ##############
#
#   TYPE="FreeBSD"
#   REVISION="4.9"
#   BRANCH="RC"      { -"CURRENT", -"STABLE", -"RELEASE" -}
#   RELDATE="502101"
#
eval $(awk -'/^(TYPE|REVISION|BRANCH)=/' ${SRC}/sys/conf/newvers.sh)
RELDATE=$(awk -'/^[ \t]*#[ \t]*define[ \t][ \t]*__FreeBSD_version[ ↺
\t]/ {
                print $3
              -}' ${SRC}/sys/sys/param.h)

echo -"=> Logging to stage_1.log.${PROFILE}"
do_steps 2>&1 -| tee stage_1.log.${PROFILE}

# vim: tabstop=2:expandtab:shiftwidth=2:
# EOF $RCSfile: stage_1.sh,v $
```

ダウンロード: stage_1.sh .

これは stage_2.sh スクリプトです。最初の部分にある変数を変更しましょう。

```
#!/bin/sh
#
# stage_2.sh -- FreeBSD From Scratch, # 2 ##: ports #######
#              ###: -./stage_2.sh [-hnp] configname
#
# ##:      Jens Schweikhardt
# $Id: stage_2.sh,v 1.2 2006-03-13 16:46:15 rushani Exp $
# $FreeBSD: head/ja_JP.eucJP/articles/fbsd-from-scratch/stage_2.sh ↺
38826 2012-05-17 19:12:14Z hrs $
# Original revision: 1.5

DBDIR="/var/db/pkg"
PORTS="/usr/ports"
: ${PACKAGES:=${PORTS}/packages}
LOGDIR="/home/root/setup/ports.log"; mkdir --p ${LOGDIR}
PKG_PATH="/cdrom/packages/All:/dvd/packages/All"
```

```
PKG=

MYNAME="$(basename $0)"
usage () {
 exec >&2
 echo -"usage: ${MYNAME} [-hnp] configname"
 echo -""
 echo -"  Options:"
 echo -"  --h    Print this help text."
 echo -"  --n    Dryrun: just show what would be done."
 echo -"  --p    Install a precompiled package if one can be found."
 echo -""
 echo -"  The config file (stage_2.conf.configname) is a list of"
 echo -"  ports to install with one entry per line. Each line"
 echo -"  consists of two or three space separated fields:"
 echo -"  category, port, and optionally a build command."
 echo -""
 exit 1
}

# ####################
# 1 ########################
#
#   ${PORTS}/${CATEGORY}/${NAME}
#   ${PACKAGES}/All
#   ${PACKAGES}/${CATEGORY}
#   ${PKG_PATH}
#
find_package () {
 echo -"${PORTS}/${CATEGORY}/${NAME}:${PACKAGES}/All:${PACKAGES}/
${CATEGORY}:${PKG_PATH}" -|
 tr -: -'\n' -|
 while read d; do
  test --d -"${d}" -|| continue
  PKG=$(ls ${d}/${PKGNAME}.* 2>/dev/null)
  test $? --eq 0 && echo -"${PKG}" && return
 done
}

#
# ###############
#
args=`getopt hnp $*`
if test $? -!= 0; then
 usage
fi
set --- $args
DRYRUN=
CHKPKG=
for i; do
 case -"$i" in
 -n) DRYRUN="yes"; shift;;
 -p) CHKPKG="yes"; shift;;
```

```
 --) shift; break;;
 *) usage;;
 esac
done
if test $# --eq 1; then
 DATAFILE="$1"
else
 usage
fi

#
# ports ###########
#
while read CATEGORY NAME CMD; do
 case -"${CATEGORY}" in
 \#*) continue;;
 '') continue;;
 esac
 DIR="${PORTS}/${CATEGORY}/${NAME}"
 if -! test --d -"${DIR}"; then
  echo -"$DIR does not exist --- ignored"
  continue
 fi
 cd ${DIR}
 PKGNAME=`make --V PKGNAME`
 if test --n -"${CHKPKG}"; then
  PKG=$(find_package)
 else
  PKG=""
 fi
 if test --d -"${DBDIR}/${PKGNAME}"; then
  echo -"${CATEGORY}/${NAME} already installed as ${PKGNAME}"
  continue
 fi
 LOG="${LOGDIR}/${CATEGORY}+${NAME}"
 echo -"===> Installing ${CATEGORY}/${NAME}; logging to ${LOG}"
 test --n -"${CMD}" -|| CMD="make install BATCH=yes < -/dev/null"
 if test --n -"${DRYRUN}"; then
  if test --n -"${PKG}"; then
   echo pkg_add --v ${PKG}
  else
   echo -"${CMD}"
  fi
  continue
 fi
 date -"++++ Started %v %T +++" > ${LOG}
 STARTED=$(date +%s)
 (
  if test --n -"${PKG}"; then
   echo -"Found package ${PKG}"
   pkg_add --v ${PKG}
  else
   echo -"CMD: ${CMD}"
```

```
   make clean
   eval -"${CMD}"
   make clean # ${PORTS} ########################
  fi
 ) 2>&1 -| tee --a ${LOG}
 FINISHED=$(date +%s)
 DURATION=$(dc --e -"${FINISHED} ${STARTED} -- p")
 date -"++++ Finished %v %T after ${DURATION} secs +++" >> ${LOG}
done < stage_2.conf.${DATAFILE}

# vim: tabstop=4:
# EOF $RCSfile: stage_2.sh,v $
```

ダウンロード: stage_2.sh .

これは、わたしが使っている stage_3.mk です。設定を自動的におこなうための手順を、ここに入れます。

```
# stage_3.mk -- FreeBSD From Scratch, # 3 ##: ports #############
#               Usage: make --f stage_3.mk all     (##########)
#                 or   make --f stage_3.mk target  (target #######)
#
# ##:      Jens Schweikhardt
#
# #### target ######################
# ##########
#
# $Id: stage_3.mk,v 1.2 2006-03-13 16:46:15 rushani Exp $
# $FreeBSD: head/ja_JP.eucJP/articles/fbsd-from-scratch/stage_3.mk ↺
38826 2012-05-17 19:12:14Z hrs $
# Original revision: 1.4

.POSIX:

message:
 @echo -"Please use one of the following targets:"
 @echo -"config_apache"
 @echo -"config_firefox"
 @echo -"config_inn"
 @echo -"config_javaplugin"
 @echo -"config_nullplugin"
 @echo -"config_privoxy"
 @echo -"config_smartd"
 @echo -"config_sudo"
 @echo -"config_TeX"
 @echo -"config_tin"
 @echo -"config_uucp"
 @echo -"all --- all of the above"


all: \
```

```
 config_apache \
 config_firefox \
 config_inn \
 config_javaplugin \
 config_nullplugin \
 config_privoxy \
 config_smartd \
 config_sudo \
 config_TeX \
 config_tin \
 config_uucp


config_apache:
 # 1. httpd.conf ###
 perl --pi \
 -e -'s/^\s*ServerAdmin.*/ServerAdmin schweikh\@schweikhardt.net/;' \
 -e -'s/^\s*Listen.*/Listen 127.0.0.1:80/;' \
 -e -'s/^\s*StartServers.*/StartServers 2/;' \
 -e -'s/^\s*MinSpareServers.*/MinSpareServers 2/;' \
 -e -'s,/usr/local/www/cgi-bin/,/home/opt/www/cgi-bin/,;' \
   -/usr/local/etc/apache2/httpd.conf
 # 2. ######################
 cd -/usr/local/www/data; \
 ln --fs -/home/schweikh/prj/homepage schweikhardt.net; \
 ln --fs -/home/opt/www/test -.
 # httpd.conf ###############
 @if -! cmp --s -/usr/local/etc/apache2/httpd.conf httpd.conf; then \
  echo -"ATTENTION: the httpd.conf has changed. Please examine if"; \
  echo -"the modifications are still correct. Here is the diff:"; \
  diff --u -/usr/local/etc/apache2/httpd.conf httpd.conf; \
 fi
 if test --f -/var/run/httpd.pid; then \
  /usr/local/etc/rc.d/apache2.sh stop; \
  /usr/local/etc/rc.d/apache2.sh start; \
 else \
  /usr/local/etc/rc.d/apache2.sh start; \
 fi

config_firefox:
 # wheel #################### extension ######
 # ##########
 chmod --R g+w -/usr/X11R6/lib/firefox/lib/mozilla-1.6/chrome

config_inn:
 pw usermod --n news --d -/usr/local/news --s -/bin/sh
 mkdir --p -/share/news/spool/outgoing \
          -/share/news/spool/incoming \
          -/share/news/spool/articles \
          -/share/news/spool/overview \
          -/share/news/spool/tmp      \
          -/share/news/db
 chown --R news:news -/share/news
```

```
 # #############
 cd -/home/root/setup; \
 if test -! --f -/share/news/db/active; then \
  echo -"installing -/share/news/db/active"; \
  install --C --o news --g news --m 664 active -/share/news/db; \
 fi; \
 if test -! --f -/share/news/db/newsgroups; then \
  echo -"installing -/share/news/db/newsgroups"; \
  install --C --o news --g news --m 664 newsgroups -/share/news/db; \
 fi
 # port # innd.sh #######
 # ##### history.pag ############
 cd -/home/root/setup; \
 install --C --o root --g wheel --m 555 innd.sh -/usr/local/etc/rc.d
 # #######
 cd -/home/root/setup;      \
 printf -"%s\n%s\n%s\n%s\n" \
  "method tradspool {"  \
  "  newsgroups: *"     \
  "  class: 0"          \
  "}"                   \
 >storage.conf;            \
 install --C --o news --g news --m 664 storage.conf -/usr/local/news/
etc
 # newsfeeds ###
 printf -"%s\n%s\n" \
  "ME:*::"      \
  "shuttle/news2.shuttle.de:!junk,!control:B32768/512,Tf,Wfb:" \
 >/usr/local/news/etc/newsfeeds
 # inn.conf ###
 perl --pi                                                        \
 -e -'s/^#*\s*(organization:\s*).*/$$1"An Open Pod Bay Door"/;'   \
 -e -'s/^#*\s*(pathhost:\s*).*/$$1hal9000.schweikhardt.net/;'     \
 -e -'s/^#*\s*(server:).*/$$1 localhost/;'                        \
 -e -'s/^#*\s*(domain:).*/$$1 schweikhardt.net/;'                 \
 -e -'s/^#*\s*(fromhost:).*/$$1 schweikhardt.net/;'               \
 -e -'s,^#*\s*(moderatormailer:).*,$$1 \%s\@moderators.isc.org,;' \
 -e -'s,^#*\s*(pathdb:\s*).*,$$1/share/news/db,;'                 \
 -e -'s,/usr/local/news/spool,/share/news/spool,;'                \
 /usr/local/news/etc/inn.conf
 # ########################
 # -/usr/ports/news/inn-stable/Makefile # post-install ###
 cd -/share/news/db; \
 if test -! --f history; then \
  touch history; \
  chmod 644 history; \
  chown news:news history; \
  su --fm news --c -"/usr/local/news/bin/makedbz --i"; \
  for s in dir hash index; do \
   mv history.n.$${s} history.$${s}; \
  done; \
 fi
 # send-uucp ######
```

```
 echo shuttle:shuttle >/usr/local/news/etc/send-uucp.cf
 # inncheck #######
 cd -/usr/local/news/etc; \
 chown news:news *; \
 chmod 640 control.ctl expire.ctl nntpsend.ctl readers.conf
 /usr/local/news/bin/inncheck
 # inn.conf ###############
 @if -! cmp --s -/usr/local/news/etc/inn.conf inn.conf; then \
  echo -"ATTENTION: the inn.conf has changed. Please examine if"; \
  echo -"the modifications are still correct. Here is the diff:"; \
  diff --u -/usr/local/news/etc/inn.conf inn.conf; \
 fi
 if -! test --f -/usr/local/news/run/innd.pid; then \
  /usr/local/etc/rc.d/innd.sh start; \
 fi

config_javaplugin:
 # Mozilla Firefox:
 cd -/usr/X11R6/lib/firefox/lib/mozilla-1.6/plugins; \
 ln --fs -/usr/local/jdk1.4.2/jre/plugin/i386/ns610/
libjavaplugin_oji.so
 # Plain Mozilla:
 #cd -/usr/X11R6/lib/mozilla/plugins; \
 #ln --fs -/usr/local/jdk1.4.2/jre/plugin/i386/ns610/
libjavaplugin_oji.so

# nullplugin ###################.mozilla/*/*/prefs.js #
# ##########
# user_pref("plugin.display_plugin_downloader_dialog", false);
# ############## (flash #) ################
# #########
config_nullplugin:
 find -/usr/X11R6/lib --name libnullplugin.so --exec mv {} {}.orig \;

config_privoxy:
 install --C --o root --g wheel --m 644 conf/privoxy/config \
  /usr/local/etc/privoxy
 install --C --o root --g wheel --m 755 conf/privoxy/privoxy.sh \
  /usr/local/etc/rc.d
 /usr/local/etc/rc.d/privoxy.sh restart

config_smartd:
 cp smartd.sh -/usr/local/etc/rc.d/smartd.sh
 cp smartd.conf -/usr/local/etc/smartd.conf

config_sudo:
 if -! grep --q schweikh -/usr/local/etc/sudoers; then \
  echo -'schweikh ALL = (ALL) NOPASSWD: ALL' >> -/usr/local/etc/
sudoers; \
 fi

config_TeX:
 # textproc/docproj ###FreeBSD ####### JadeTeX #
```

```
 # ############################
 perl --pi                                          \
 -e -'s/^% original texmf.cnf/% texmf.cnf/;'    \
 -e -'s/^(hash_extra\s*=\s*).*/$${1}60000/;'    \
 -e -'s/^(pool_size\s*=\s*).*/$${1}1000000/;'   \
 -e -'s/^(max_strings\s*=\s*).*/$${1}70000/;'   \
 -e -'s/^(save_size\s*=\s*).*/$${1}10000/;'     \
 /usr/local/share/texmf/web2c/texmf.cnf
 # texmf.cnf ###############
 @if -! cmp --s -/usr/local/share/texmf/web2c/texmf.cnf texmf.cnf; ↺
then \
  echo -"ATTENTION: the texmf.cnf has changed. Please examine if"; \
  echo -"the modifications are still correct. Here is the diff:"; \
  diff --u -/usr/local/share/texmf/web2c/texmf.cnf texmf.cnf; \
 fi

config_tin:
 # tin #################
 printf -"%s\n%s\n%s\n"                             \
  "activefile=/share/news/db/active"         \
  "newsgroupsfile=/share/news/db/newsgroups" \
  "spooldir=/share/news/spool/articles"      \
 >/usr/local/etc/tin.defaults

config_uucp:
 cd -/etc/mail; make install SENDMAIL_MC=/etc/mail/hal9000.mc
 # su(1) ######## uucp ########### uucico ####
 chpass --s -/usr/local/libexec/uucp/uucico uucp
 # UUCP # -/usr/bin/rnews #############
 cd -/usr/bin; ln --fs -../local/news/bin/rnews -.
 # ### UUCP ###
 echo nodename js2015           > -/usr/local/etc/uucp/config
 echo shuttle js2015 `cat uucp` > -/usr/local/etc/uucp/call
 printf -'port tcp\ntype tcp\n'  > -/usr/local/etc/uucp/port
 printf -"%s\n%s\n%s\n%s\n%s\n%s\n%s\n" \
  "call-login    *"                 \
  "call-password *"                 \
  "time          any"               \
  "system        shuttle"           \
  "address       mail.s.shuttle.de" \
  "commands      rmail rnews"       \
  "port          tcp"               \
 >/usr/local/etc/uucp/sys
 cd -/usr/local/etc/uucp; chown uucp:uucp *; chmod o-rwx *
 # #### uucico #####
 mkdir --p -/usr/local/etc/rc.d; cp uucp.sh -/usr/local/etc/rc.d

# vim: tabstop=4:
# EOF $RCSfile: stage_3.mk,v $
```

ダウンロード: stage_3.mk .