

# ##### ## #####

#### #####<schweikh@FreeBSD.org>

#####: 43126

##### © 2002,2003,2004,2008 ####

#####  
#####.

####, ####, ##### # #####-  
#####  
##### # ##### #/###

##### #, ##### # #-  
##### # #-  
##### #  
#####, #  
#####.

2013#11#07 #####.

## #####

# ####  
#####: #####-  
#####  
##### # #-  
##### #  
## #####, ## make world ##### #-  
#####, ## ##### ## make  
evenmore.

## #####

1. #####	2
2. ##### ## (##) ##### #####? .....	2
3. ##### .....	4
4. #####: ##### .....	4
5. #####: ##### .....	10
6. ##### .....	15
7. ##### .....	15
8. ##### .....	16

#####

1. #####

```
## ##### make world? #####
#####, #####
## ##### installworld #####, ## #####
#####, ##### #####
#####, # #####
# #####, #####
```

# #####  
#####,  
#####,  
#####,  
#####,  
#####,  
#####,  
#####,  
#####,

2. ##### ## (##) ##### ##### ## ##?

### #####. # ## ##### ##### ##### # #####  
##### # #####.

##### # ##### # ##### # ##### # ###, ### ## ##### #####  
# ##, ##, ### # ## ##### #####.

- #####  
#####  
#####  
##### make buildworld.
- #####  
#####.
- #####.
- #####, # #####  
##.
- #####  
#####.
- #####, #####, #####  
#####.

#####  
#####  
#####  
#####  
#####



#####

#####.###). #### ## ##### #####, # ## #####, ## ##  
##### #####, ##### ## 10 #####.

### 3. #####

### ##, ##### ##### ##### #####, ### ##### ##:

- ##### ##### # ##### #.
- ## ##### ##, ##### #####.
- #### ##### # #####(8). ## ## ##### ##### ## --  
#####.
- #### # ## ##### ## ##### #: #####  
#####.
- ##### ##### ## #####, #(1).
- #####, ## ##### ##### ## ##### --  
#####, #####, ## #####.

### 4. #### #: #####

# ##### ## ##### #####, # #####, ## #####--  
#####, # # ##### ##  
#####. ## ##### #####  
##### # #####.

##### # ##### ## stage\_1.sh, # ##  
#####

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

```
##### stage_1.conf.default #  
# stage_1.log.default.
```

```
##### stage_1.conf.default. ##  
##### ##  
##### # #####  
#####. #####  
##### create_file_systems, create_etc_fstab, copy_files # all_remaining_customization (#  
#####: #####  
## ## stage_1.sh).
```



```
#### #####: ##### #####
```

```
## #####, ##### ## ##### #####-
###, ###, #####. ## ##### # #####
stage_1.conf.default ##### ##, #####
```

```
# ##### stage_1.sh ## ##### mergemaster. ##### #####
#####, ## ##### # # #####
```

```
*** Comparison complete
*** Savingmtree database for future upgrades
```

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

```
#####, ##### no ## #####. ##### # ##, ### mergemaster
##### /var/tmp/temproot.stage1, #####
##### (#, #####, ## ## ##).
```

```
##### mergemaster ##### ## #####, ##### ## #-
##### login.conf:
```

```
*** 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
```

```
*** You installed a new aliases file into -/newroot/etc/mail, but
the newaliases command is limited to the directories configured
in sendmail.cf. Make sure to create your aliases database by
hand when your sendmail configuration is done.
```

```
*** 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]
```

```
##### ## ##### stage_1.sh ##### ###_###(1) # ##### #-
###.
```

```
### stage_1.conf.default, ##### ## ##### #####-
#####. # ##### # ##, ### ##### #-
#####.
```

```
# This file: stage_1.conf.default, sourced by stage_1.sh.
#
# $Id: stage_1.conf.default,v 1.5 2011-05-14 20:44:31 hrs Exp $
# $FreeBSD: head/en_US.ISO8859-1/articles/fbsd-from-scratch/stage_1.conf.default 38826 -
2012-05-17 19:12:14Z hrs $

# Root mount point where you create the new system. Because it is only
# used as a mount point, no space will be used on that file system as all
```

##### ## #####

---

```
# files are of course written to the mounted file system(s).
DESTDIR="/newroot"

# Where your src tree is.
SRC="/usr/src"

# Where your obj is.
MAKEOBJDIRPREFIX="/usr/obj"

# Your kernel config name as from make buildkernel KERNCONF=...
KERNCONF="HAL9000"

# Your target architecture as used for make buildworld TARGET=...
# If you did not specify a TARGET when building world, it defaulted
# to the build architecture (run -"uname --m" to find out if you are unsure).
TARGET="i386" # amd64 arm i386 ia64 mips pc98 powerpc sparc64

# Available time zones are those under -/usr/share/zoneinfo.
TIMEZONE="Europe/Berlin"

#
# The create_file_systems function must create the mountpoints under
# DESTDIR, create the file systems, and then mount them under DESTDIR.
#
create_file_systems () {
    # The new root file system. Mandatory.
    # Change DEVICE names.
    DEVICE=/dev/darwin
    mkdir --m 755 --p ${DESTDIR}
    chown root:wheel ${DESTDIR}
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}

    # Additional file systems and initial mount points. Optional.
    DEVICE=/dev/darwin
    mkdir --m 755 --p ${DESTDIR}/var
    chown root:wheel ${DESTDIR}/var
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}/var

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

#
# The create_etc_fstab function must create an fstab matching the
# file systems created in create_file_systems.
#
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
}

#
# The copy_files function is used to copy files before mergemaster is run.
#
copy_files () {
# Add or remove from this list at your discretion. Mostly mandatory.
for f in \
-/profile \
-/etc/devd.conf \
-/etc/devd.rules \
-/etc/exports \
-/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/src.conf \
-/etc/sysctl.conf \
-/etc/ttys \
-/etc/mail/aliases \
-/etc/mail/aliases.db \

```



##### ## #####

---

```
-/etc/mail/ha19000.mc \
-/etc/mail/service.switch \
-/etc/ssh/*key* \
-/etc/ssh/*_config \
-/etc/X11/xorg.conf \
-/var/cron/tabs/* \
-/root/.profile \
-/boot/*.* \
-/boot/loader.conf \
-/boot/device.hints -; do
cp -p ${f} ${DESTDIR}${f}
done
}

#
# Everything else you want to tune in the new system.
# NOTE: Do not install too many binaries here. With the old system running and
# the new binaries and headers installed you are likely to run into bootstrap
# problems. Ports should be compiled after you have booted in the new system.
#
all_remaining_customization () {
# Without the compat symlink the linux_base files end up on the root fs:
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

# Create the ntp and slip log files.
touch ${DESTDIR}/var/log/ntp ${DESTDIR}/var/log/slip.log

# Make -/usr/src point to the right directory. Optional.
# Note: some ports need part of the src tree, e.g. emulators/kqemu,
# sysutils/lsof, sysutils/fusefs, ...
cd ${DESTDIR}/usr
if test "${SRC}" != -/usr/src; then
rmdir src; ln -s ${SRC}
fi
if test "${MAKEOBJDIRPREFIX}" != -/usr/obj; then
rmdir obj; ln -s ${MAKEOBJDIRPREFIX}
fi

# My personal preference is to symlink tmp --> var/tmp. Optional.
cd ${DESTDIR}; rmdir tmp; ln -s var/tmp

# Make spooldirs for the printers in my -/etc/printcap.
cd ${DESTDIR}/var/spool/output/lpd; mkdir --p as od ev te lp da
```

#### #####: ##### #####

```
touch ${DESTDIR}/var/log/lpd-errs

# If you do not have -/home on a shared partition, you may want to copy it:
# mkdir -p ${DESTDIR}/home
# cd -/home; tar cf -- -. -| (cd ${DESTDIR}/home; tar xpvf --)
}

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

##### stage\_1.conf.default.

### ##### #####:#####

- ##### # #####.
- ##### # ##### # ##### # ##### # #####.
- ##### ##### # #.
- ##### ##### #####, #####, /etc/ttys # inetd.

##### # #####, ## # #####, #### # #####  
##### #. #####, ## ##### # #11. #####  
### #####, #####, #####, ##### # ----  
##### #, #####, ##### # #11 # #####, #### #  
## ##### # #####.

## 5. #### #####: #####



#####

```
## ##### (#####) #####. # #####
stage_2.sh ##### # #, ##### #--
#### pkg_add. # #, ## ##### #.
##### # # # # #--
#### # #.
```

##### stage\_2.sh #####, ## # ##### #. ##  
##### # # # # #, #####  
### #. ## dryrun##### (-n) ## #, ##

##### ## #####

```
#####. ## ##### ### #####, ## stage_1.sh, # ##### #####,
##### ## #####, # #####
```

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

```
##### ##### ##### ##### ##### ## stage_2.conf.default.
```

```
#####  
#####: ##### # ##, ## #####  
##### (## #####: make  
install BATCH=yes </dev/null). ##### # ##### #, ###-  
#####. # ##### # ##  
# ##. #####  
make#####, # #####:
```

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

```
#####  
##### make:
```

```
java jdk16      echo true > files/license.sh; make install BATCH=yes < -/dev/null
print acroread8 yes accept -| make install PAGER=ls
x11-fonts gnu-unifont make install && mkfontdir -usr/local/lib/X11/fonts/local
news inn-stable CONFIGURE_ARGS="--enable-uucp-mews --enable-setgid-inews" make install
```

```
# #####  #####  #####, ### ##### # #####, #####
#####  ### #####  #####. #####, ### #####  ###  ###/
#####  #####  #####  #####  #####  #####
CONFIGURE_ARGS. #### Makefile #####  #####  #####  #####-
### # #####  #####  #####  #####. ##### # #####  #####
##### # #####  #####  ##  #####
```

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

```
##### # ##, ## # ##### ##### ##### #####, ## #
#####. ##### ##### ## #####.
```

```
##### # ###, ### ##### ## ##### ##### #####, ##
### ## ##### ##### ## ##### ##### #####
###, ## ## ## ##### ## ##. ### ## ## ## #####, ## ## #####
##### ##### ## ##### #####, ##### # ## #####, #
##### #####. ### ##### stage_2.sh ##### ##
##### #####, ##### ##### # ##.
```

```
##### stage_2.conf.default. #####  
##### LOGDIR/category+port.
```

```
# This file: stage_1.conf.default, sourced by stage_1.sh.
#
```

```

# $Id: stage_1.conf.default,v 1.5 2011-05-14 20:44:31 hrs Exp $
# $FreeBSD: head/en_US.ISO8859-1/articles/fbsd-from-scratch/stage_1.conf.default 38826 -
2012-05-17 19:12:14Z hrs $

# Root mount point where you create the new system. Because it is only
# used as a mount point, no space will be used on that file system as all
# files are of course written to the mounted file system(s).
DESTDIR="/newroot"

# Where your src tree is.
SRC="/usr/src"

# Where your obj is.
MAKEOBJDIRPREFIX="/usr/obj"

# Your kernel config name as from make buildkernel KERNCONF=...
KERNCONF="HAL9000"

# Your target architecture as used for make buildworld TARGET=...
# If you did not specify a TARGET when building world, it defaulted
# to the build architecture (run -"uname --m" to find out if you are unsure).
TARGET="i386" # amd64 arm i386 ia64 mips pc98 powerpc sparc64

# Available time zones are those under -/usr/share/zoneinfo.
TIMEZONE="Europe/Berlin"

#
# The create_file_systems function must create the mountpoints under
# DESTDIR, create the file systems, and then mount them under DESTDIR.
#
create_file_systems () {
    # The new root file system. Mandatory.
    # Change DEVICE names.
    DEVICE=/dev/daxyzs1a
    mkdir --m 755 --p ${DESTDIR}
    chown root:wheel ${DESTDIR}
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}

    # Additional file systems and initial mount points. Optional.
    DEVICE=/dev/daxyzs1e
    mkdir --m 755 --p ${DESTDIR}/var
    chown root:wheel ${DESTDIR}/var
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}/var

    DEVICE=/dev/daxyzs1e
    mkdir --m 755 --p ${DESTDIR}/usr
    chown root:wheel ${DESTDIR}/usr
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}/usr
}

```

##### ## #####

---

```
#
# The create_etc_fstab function must create an fstab matching the
# file systems created in create_file_systems.
#
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
}

#
# The copy_files function is used to copy files before mergemaster is run.
#
copy_files () {
    # Add or remove from this list at your discretion. Mostly mandatory.
    for f in \
        -/.profile \
        -/etc/devd.conf \
        -/etc/devd.rules \
        -/etc/exports \
        -/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/src.conf \
-/etc/sysctl.conf \
-/etc/ttys \
-/etc/mail/aliases \
-/etc/mail/aliases.db \
-/etc/mail/hal9000.mc \
-/etc/mail/service.switch \
-/etc/ssh/*key* \
-/etc/ssh/*_config \
-/etc/X11/xorg.conf \
-/var/cron/tabs/* \
-/root/.profile \
-/boot/*.bmp \
-/boot/loader.conf \
-/boot/device.hints -; do
cp -p ${f} ${DESTDIR}${f}
done
}

#
# Everything else you want to tune in the new system.
# NOTE: Do not install too many binaries here. With the old system running and
# the new binaries and headers installed you are likely to run into bootstrap
# problems. Ports should be compiled after you have booted in the new system.
#
all_remaining_customization () {
# Without the compat symlink the linux_base files end up on the root fs:
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

# Create the ntp and slip log files.
touch ${DESTDIR}/var/log/ntp ${DESTDIR}/var/log/slip.log

# Make -usr/src point to the right directory. Optional.
# Note: some ports need part of the src tree, e.g. emulators/kqemu,
# sysutils/lsof, sysutils/fusefs, ...
cd ${DESTDIR}/usr
if test "${SRC}" != -/usr/src; then
    rmdir src; ln --s ${SRC}
fi
if test "${MAKEOBJDIRPREFIX}" != -/usr/obj; then
    rmdir obj; ln --s ${MAKEOBJDIRPREFIX}
fi
}

```

```
##### ## #####
```

```
# My personal preference is to symlink tmp --> var/tmp. Optional.
cd ${DESTDIR}; rmdir tmp; ln -s var/tmp

# Make spooldirs for the printers in my /etc/printcap.
cd ${DESTDIR}/var/spool/output/lpd; mkdir --p as od ev te lp da
touch ${DESTDIR}/var/log/lpd-errs

# If you do not have ~/home on a shared partition, you may want to copy it:
# mkdir --p ${DESTDIR}/home
# cd ~/home; tar cf -- -. -| (cd ${DESTDIR}/home; tar xpvf --)
}

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

```
##### stage_2.conf.default.
```

## 6. #####

```
## ##### ##### ## ##### #####. ##### ## ## #####
#####. ##### ## ##### # ##### ##### # ## #--
#####. # ## ## ##### # ##### # ##### stage_2.sh. ##### #
#####, ## ## ##### ##### ##### # ##### # #####
## #####, # ## ##### #####.
```

```
# ##### ##### # ##### Makefile, ##### ## ## #####
##### ##, ## ## #####, #####:
```

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

```
### # ##### # stage_2.sh, #####, ### ## stage_3.mk #####
### # #####, ##### ## ## #####
### ##### # #####.
```

## 7. #####

```
##### ##### #####, ## ## #--
##### # ## ##### make BATCH=YES install. ## #####
##### ## ##, ## ## yes # ## #--
#####. ##### # #--
#####, # ## ##### #
##### (# #####: yes | make install). ## #####
#####, ## ##### #
## #####. ##### ## ##/#####8 ## ##/#####16.
```

```
## ##### ## ##### ## ##### ## ##### ## ##### ## ##### ## #
##. #####, ## ## #####, ##### # ##### ## ##### ## #####
## #####. # ## # /etc/group ##### ## ##### ## ##,
## # /etc/passwd ##### #####. ## ## ## #####
#####. ##### ##### ##### ## ## # ##### # #####
##### # ##### ##### #####, ## # ##### ## ## ##.
#### # ##### ##### ##### (##### ##### #
#####), ## ##### ##### mergemaster ## ##### # #####,
## ## ##### ##### ##### ##### # #####
#####. # #####, mergemaster ##### # #####
#####, # # #####, #####. #####, ##
##### ##### ##### ##### ## ##, ##### #
## #####, # ##### ##### ## ## ## # #####. ## #####
#####, # ##### ##### ##### ##### ##
##, ## # stage_3.mk ##### # ##### # make. #####, ## #
#####, httpd.conf, ##### # config_apache.
```

```
@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
```

```
#####  
##### cp /usr/local/etc/apache2/  
httpd.conf httpd.conf.
```

```
# ##### 7-
CURRENT ## 7-CURRENT # 8-CURRENT ## 8-CURRENT, ## #### # ##### #
##### 8-CURRENT ## ##### 7-STABLE # #####. #####
### ##### # ##### ##### ##### # #####, ### ##### #
### #####. ##### ##### ##### #
### ##### STABLE ##### (### # ## #
#####).
```

8. #####

```
### ## #####, ##### ## ##### ##### ## #####, ## ## #####
####.
```

```
### ##### stage 1.sh, ##### ## ## ##### #####.
```

```
# This file: stage_1.conf.default, sourced by stage_1.sh.
#
# $Id: stage_1.conf.default,v 1.5 2011-05-14 20:44:31 hrs Exp $
```



##### ## #####

---

```
# $FreeBSD: head/en_US.ISO8859-1/articles/fbsd-from-scratch/stage_1.conf.default 38826 -
2012-05-17 19:12:14Z hrs $
```

```
# Root mount point where you create the new system. Because it is only
# used as a mount point, no space will be used on that file system as all
# files are of course written to the mounted file system(s).
DESTDIR="/newroot"
```

```
# Where your src tree is.
SRC="/usr/src"
```

```
# Where your obj is.
MAKEOBJDIRPREFIX="/usr/obj"
```

```
# Your kernel config name as from make buildkernel KERNCONF=...
KERNCONF="HAL9000"
```

```
# Your target architecture as used for make buildworld TARGET=...
# If you did not specify a TARGET when building world, it defaulted
# to the build architecture (run -"uname --m" to find out if you are unsure).
TARGET="i386" # amd64 arm i386 ia64 mips pc98 powerpc sparc64
```

```
# Available time zones are those under -/usr/share/zoneinfo.
TIMEZONE="Europe/Berlin"
```

```
#
# The create_file_systems function must create the mountpoints under
# DESTDIR, create the file systems, and then mount them under DESTDIR.
#
```

```
create_file_systems () {
# The new root file system. Mandatory.
# Change DEVICE names.
DEVICE=/dev/daXYZs1a
mkdir --m 755 --p ${DESTDIR}
chown root:wheel ${DESTDIR}
newfs --U ${DEVICE}
mount --o noatime ${DEVICE} ${DESTDIR}
```

```
# Additional file systems and initial mount points. Optional.
DEVICE=/dev/daXYZs1e
mkdir --m 755 --p ${DESTDIR}/var
chown root:wheel ${DESTDIR}/var
newfs --U ${DEVICE}
mount --o noatime ${DEVICE} ${DESTDIR}/var
```

```
DEVICE=/dev/daXYZs1e
mkdir --m 755 --p ${DESTDIR}/usr
chown root:wheel ${DESTDIR}/usr
newfs --U ${DEVICE}
mount --o noatime ${DEVICE} ${DESTDIR}/usr
}
```

```

#
# The create_etc_fstab function must create an fstab matching the
# file systems created in create_file_systems.
#
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
}

#
# The copy_files function is used to copy files before mergemaster is run.
#
copy_files () {
  # Add or remove from this list at your discretion. Mostly mandatory.
  for f in \
    -/.profile \
    -/etc/devd.conf \
    -/etc/devd.rules \
    -/etc/exports \
    -/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/src.conf \
-/etc/sysctl.conf \
-/etc/ttys \
-/etc/mail/aliases \
-/etc/mail/aliases.db \
-/etc/mail/ha19000.mc \
-/etc/mail/service.switch \
-/etc/ssh/*key* \
-/etc/ssh/*_config \
-/etc/X11/xorg.conf \
-/var/cron/tabs/* \
-/root/.profile \
-/boot/*.* \
-/boot/loader.conf \
-/boot/device.hints -; do
  cp -p ${f} ${DESTDIR}${f}
done
}

#
# Everything else you want to tune in the new system.
# NOTE: Do not install too many binaries here. With the old system running and
# the new binaries and headers installed you are likely to run into bootstrap
# problems. Ports should be compiled after you have booted in the new system.
#
all_remaining_customization () {
  # Without the compat symlink the linux_base files end up on the root fs:
  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

  # Create the ntp and slip log files.
  touch ${DESTDIR}/var/log/ntp ${DESTDIR}/var/log/slip.log

  # Make -usr/src point to the right directory. Optional.
  # Note: some ports need part of the src tree, e.g. emulators/kqemu,
  # sysutils/lsof, sysutils/fusefs, -...
  cd ${DESTDIR}/usr
  if test "${SRC}" != -/usr/src; then
    rmdir src; ln --s ${SRC}
  fi
  if test "${MAKEOBJDIRPREFIX}" != -/usr/obj; then
    rmdir obj; ln --s ${MAKEOBJDIRPREFIX}
  fi
}
```

####

```
# My personal preference is to symlink tmp --> var/tmp. Optional.
cd ${DESTDIR}; rmdir tmp; ln --s var/tmp

# Make spooldirs for the printers in my -/etc/printcap.
cd ${DESTDIR}/var/spool/output/lpd; mkdir --p as od ev te lp da
touch ${DESTDIR}/var/log/lpd-errs

# If you do not have -/home on a shared partition, you may want to copy it:
# mkdir --p ${DESTDIR}/home
# cd -/home; tar cf -- -. -| (cd ${DESTDIR}/home; tar xpvf --)
}

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

##### [stage\\_1.sh](#).

### ##### [stage\\_2.sh](#). ### ##### ##### # #####  
#####.

```
#!/bin/sh
#
# stage_1.sh -- FreeBSD From Scratch, Stage 1: System Installation.
#      Usage: -/stage_1.sh profile
#      will read profile
#      and write -/stage_1.log.profile
#
# Author:   Jens Schweikhardt
# $Id: stage_1.sh,v 1.7 2008-12-11 19:48:21 schweikh Exp $
# $FreeBSD: head/en_US.ISO8859-1/articles/fbsd-from-scratch/stage_1.sh 38826 2012-05-17 -
19:12:14Z hrs $

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

# Prerequisites:
#
# a) Successfully completed -"make buildworld" and -"make buildkernel"
# b) Unused partitions (at least one for the root fs, probably more for
#    the new -/usr and -/var, to your liking.)
# c) A customized profile file.

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

# ----- #
# Step 1: Create an empty directory tree below $DESTDIR.
# ----- #
```

##### ## #####

---

```
step_one () {
    create_file_systems
    # Now create all the other directories. Mandatory.
    cd ${SRC}/etc; make distrib-dirs DESTDIR=${DESTDIR} TARGET=${TARGET}
}
```

```
# ----- #
# Step 2: Fill the empty -/etc directory tree and put a few files in -/.
# ----- #
```

```
step_two () {
    copy_files

    # Delete mergemaster's temproot, if any.
    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 does not create empty files, e.g. in -/var/log. Do so now,
    # but do not clobber files that may have been copied with 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}
}
```

```
# ----- #
# Step 3: Install world.
# ----- #
```

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

```
# ----- #
# Step 4: Install kernel and modules.
# ----- #
```

```

step_four () {
    cd ${SRC}
    # The loader.conf and device.hints are required by the installkernel target.
    # If you have not copied them in Step 2, cp them as shown in the next 2 lines.
    # cp sys/boot/forth/loader.conf ${DESTDIR}/boot/defaults
    # cp sys/${TARGET}/conf/GENERIC.hints ${DESTDIR}/boot/device.hints
    make installkernel DESTDIR=${DESTDIR} KERNCONF=${KERNCONF} TARGET=${TARGET}
}

# ----- #
# Step 5: Install /etc/fstab and time zone info.
# ----- #

step_five () {
    create_etc_fstab

    # Setup time zone info; pretty much mandatory.
    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
}

# ----- #
# Step 6: All remaining customization.
# ----- #

step_six () {
    all_remaining_customization
}

do_steps () {
    echo -"PROFILE=${PROFILE}"
    echo -"TARGET=${TARGET}"
    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
}

# ----- #
# The ball starts rolling here.
# ----- #

```

##### ## #####

---

```
PROFILE="$1"
set --x --e --u # Stop for any error or use of an undefined variable.
. ${PROFILE}

# Determine a few variables from the sources that were used to make the
# world. The variables can be used to modify actions, e.g. depending on
# the system's version. The __FreeBSD_version numbers
# for RELDATE are documented in the Porter's Handbook,
# doc/en_US.ISO8859-1/books/porters-handbook/freebsd-versions.html.
# Scheme is: <major><two digit minor><0 if release branch, otherwise 1>xx
# The result will be something like
#
# TYPE="FreeBSD"
# REVISION="8.0"
# BRANCH="RC"      { -"CURRENT", -"STABLE", -"RELEASE" -}
# RELDATE="800028"
#
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.${PROFILE}.log"
do_steps 2>&1 -| tee -"stage_1.${PROFILE}.log"

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

##### [stage\\_2.sh](#).

### ## ##### [stage\\_3.mk](#), ##### ## ## ## # ##, ## ##### ## ## --  
#####

```
# This file: stage_1.conf.default, sourced by stage_1.sh.
#
# $Id: stage_1.conf.default,v 1.5 2011-05-14 20:44:31 hrs Exp $
# $FreeBSD: head/en_US.ISO8859-1/articles/fbsd-from-scratch/stage_1.conf.default 38826 -
2012-05-17 19:12:14Z hrs $

# Root mount point where you create the new system. Because it is only
# used as a mount point, no space will be used on that file system as all
# files are of course written to the mounted file system(s).
DESTDIR="/newroot"

# Where your src tree is.
SRC="/usr/src"

# Where your obj is.
MAKEOBJDIRPREFIX="/usr/obj"
```

```

# Your kernel config name as from make buildkernel KERNCONF=...
KERNCONF="HAL9000"

# Your target architecture as used for make buildworld TARGET=...
# If you did not specify a TARGET when building world, it defaulted
# to the build architecture (run "-uname --m" to find out if you are unsure).
TARGET="i386" # amd64 arm i386 ia64 mips pc98 powerpc sparc64

# Available time zones are those under /usr/share/zoneinfo.
TIMEZONE="Europe/Berlin"

#
# The create_file_systems function must create the mountpoints under
# DESTDIR, create the file systems, and then mount them under DESTDIR.
#
create_file_systems () {
    # The new root file system. Mandatory.
    # Change DEVICE names.
    DEVICE=/dev/daXYZs1a
    mkdir --m 755 --p ${DESTDIR}
    chown root:wheel ${DESTDIR}
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}

    # Additional file systems and initial mount points. Optional.
    DEVICE=/dev/daXYZs1e
    mkdir --m 755 --p ${DESTDIR}/var
    chown root:wheel ${DESTDIR}/var
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}/var

    DEVICE=/dev/daXYZs1e
    mkdir --m 755 --p ${DESTDIR}/usr
    chown root:wheel ${DESTDIR}/usr
    newfs --U ${DEVICE}
    mount --o noatime ${DEVICE} ${DESTDIR}/usr
}

#
# The create_etc_fstab function must create an fstab matching the
# file systems created in create_file_systems.
#
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
}

#
# The copy_files function is used to copy files before mergemaster is run.
#
copy_files () {
  # Add or remove from this list at your discretion. Mostly mandatory.
  for f in \
    ~/.profile \
    /etc/devd.conf \
    /etc/devd.rules \
    /etc/exports \
    /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/src.conf \
    /etc/sysctl.conf \
    /etc/ttys \
    /etc/mail/aliases \
    /etc/mail/aliases.db \
    /etc/mail/ha19000.mc \
    /etc/mail/service.switch \
    /etc/ssh/*key* \
    /etc/ssh/*_config \
    /etc/X11/xorg.conf \
    /var/cron/tabs/* \
    /root/.profile \
    /boot/*.bmp \
```

```

-/boot/loader.conf \
-/boot/device.hints -; do
  cp --p ${f} ${DESTDIR}${f}
done
}

#
# Everything else you want to tune in the new system.
# NOTE: Do not install too many binaries here. With the old system running and
# the new binaries and headers installed you are likely to run into bootstrap
# problems. Ports should be compiled after you have booted in the new system.
#
all_remaining_customization () {
  # Without the compat symlink the linux_base files end up on the root fs:
  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

  # Create the ntp and slip log files.
  touch ${DESTDIR}/var/log/ntp ${DESTDIR}/var/log/slip.log

  # Make -/usr/src point to the right directory. Optional.
  # Note: some ports need part of the src tree, e.g. emulators/kqemu,
  # sysutils/lsof, sysutils/fusefs, ...
  cd ${DESTDIR}/usr
  if test -"${SRC}" -!= -/usr/src; then
    rmdir src; ln --s ${SRC}
  fi
  if test -"${MAKEOBJDIRPREFIX}" -!= -/usr/obj; then
    rmdir obj; ln --s ${MAKEOBJDIRPREFIX}
  fi

  # My personal preference is to symlink tmp --> var/tmp. Optional.
  cd ${DESTDIR}; rmdir tmp; ln --s var/tmp

  # Make spooldirs for the printers in my -/etc/printcap.
  cd ${DESTDIR}/var/spool/output/lpd; mkdir --p as od ev te lp da
  touch ${DESTDIR}/var/log/lpd-errors

  # If you do not have -/home on a shared partition, you may want to copy it:
  # mkdir --p ${DESTDIR}/home
  # cd -/home; tar cf -- . -| (cd ${DESTDIR}/home; tar xpvf --)
}

# vim: tabstop=2:expandtab:shiftwidth=2:syntax=sh:

```

##### ## #####

---

# EOF \$RCSfile: stage\_1.conf.default,v \$

##### [stage\\_3.mk](#).

