

OpenMicroServer Development Environment Guide

Ver. 1.01

Plat'Home Co., Ltd.

Table of Contents

1. Development environment	3
1.1 Cross development environment	3
1.2 Native development environment	5
1.3 Customizing initrd	7
1.4 PCMCIA-CS · USB module.....	8

1.1 Cross development environment

Use SSD/Linux 0.4 for cross development environment.

* You can download SSD/Linux 0.4 from <http://www.plathome.com/products/microserver/ssdlinux/>.
A user's guide is also available from the same URL..

- Installation
For information on how to install, refer to the "SSD/Linux User's Guide" and "How to use SSD/Linux sysinst(8)".
Add cross development environment when installing.
- Source expansion
For information concerning source expansion and package acquisition, refer to the "SSD/Linux Developer's Note."
To update the development environment, after completing installation, get that latest src.tgz from <ftp://ftp.plathome.co.jp/pub/ssdlinux/0.4-LATEST/source/> and execute "bmake build."

You can get the latest source tree by the method described in Anonymous CVS Server <http://www.plathome.co.jp/support/labo/cvsserver/>. This how ever includes contents still being worked upon and may not be successful.

- Getting distfiles
A configuration that exactly resembles source collection of pkgsrc of NetBSD is available for SSD/Linux, and software package source (hereinafter referred to as "distfiles") are available by ftp(1) at /usr/src/dist/distfiles/ for bmake build.
If direct ftp(1) is impossible, you should get an iso image.
The iso image includes all distfiles.
You may copy to /usr/src/dist/distfiles/ from the iso image.
* Cross development requires at least 6GB of hard disk space including self portion.

If creating an image for OpenMicroServer, perform the following operation.

(1) Change /etc/mk.conf as follows:

```
#DESTDIR= /home/dest
#RELEASEDIR= /home/release
#RCS_LOCALID= ssdlinux
SSDCVSTAG=
#SSDCVSDATE= 20020420

# alternate kernel config which instead of
# mkdist/kernel/dot.config.* if required
#KERNEL_CONFIG= /any_dir/my_kernel_config

# Cross Compile for OpenBlockS
CROSS_BUILD= mipsel
DESTDIR= /home/dest-mipsel
RELEASEDIR= /home/release-mipsel

# OPENBLOCKS 'obs50','obs200','obs266' or 'none'
OPENBLOCKS?= obs1550

HAVE_CVS?= no
DEVELOPTOOLS?= yes
```

```

USE_PAM?=yes
USE_PCMCIA?=    yes
USE_EXT3FS?=    yes
IPV6?=         no

# MTA           'sendmail' or 'postfix'
MTA?=          sendmail

```

[Optional settings]

```

HAVE_CVS:      Specify "yes" if you can directly connect to the CVS server
                of SSD/Linux or "no" if you can't.
DEVELOPTOOLS: Specify "yes" in all cases.
USE_PAM:       Specify "yes" in all cases.
USE_PCMCIA:    Specify "yes" in all cases.
USE_EXT3FS:    Specify "yes" in all cases.
IPV6:         Specify "yes" or "no" to indicate whether
                you are using the IPV6 function or not.
                If IPV6 is enabled, the size of the kernel and firmware increases.

```

(2) Include the cross-compile gcc for mipsel in the command search path
(OpenMicroServer only)

```
# export PATH=$PATH:/usr/cross/mipsel/bin
```

(3) Run bmake release.

```

# export HOSTTYPE
# export MACHTYPE
# export SHELL
# cd /usr/src
# bmake release

```

Various types of images are made after /home/release-mipsel.

distfiles	package source used
mipsel-obs1550	
-binary	••• HD image
-base.tgz	
-comp.tgz	
-contrib.tgz	
-etc.tgz	
-kern.tgz	
? man.tgz	
? installation	••• initrd image
-ramdisk.image-product.gz	
-ramdisk.image.gz	
-zImage.initrd.treeboot	
? zImage.initrd.treeboot-product	
source	••• Batch and others
? src.tgz	

1.2 Native development environment

Self development is conducted by HDRoot only.

Create HDRoot using the included CD. When doing so, expand comp.tgz src.tgz.

When HD is finished being created, boot from HD.

* Self development requires at least 4GB of hard disk space.

- Update to latest set
If creating the image of a verified latest set, do the following:
Get the latest src.tgz from <ftp://ftp.plathome.co.jp/pub/ssdlinux/04-LATEST/source/> and expand.
- Getting distfiles
A configuration that exactly resembles source collection of pkgsrc of NetBSD is available for SSD/Linux, and software package source (hereinafter referred to as “distfiles”) are available by ftp(1) at /usr/src/dist/distfiles/ for bmake build.
If direct ftp(1) is impossible, after using a PC, etc. to get it from <ftp://ftp.plathome.co.jp/pub/ssdlinux/distfiles/>, transmit it to /usr/src/dist/distfiles/.

Perform the following:

(1) Set /etc/mk.conf as follows:

```
#DESTDIR?=      /home/dest
#RELEASEDIR?=   /home/release
#RCS_LOCALID?=  ssdlinux
SSDCVSTAG?=
#SSDCVSDATE?=   20020420

# alternate kernel config which instead of
# mkdist/kernel/dot.config.* if required
#KERNEL_CONFIG?= /any_dir/my_kernel_config

# OPENBLOCKS    'obs1550' or 'none'
OPENBLOCKS?=   obs1550

HAVE_CVS?=      no
DEVELOPTOOLS?= yes
USE_PAM?=yes
USE_PCMCIA?=    yes
USE_EXT3FS?=    yes
IPV6?=          yes

# MTA           'sendmail' or 'postfix'
MTA?=          sendmail
```

Optional settings

HAVE_CVS: Specify “yes” if you can directly connect to the CVS server of SSD/Linux or “no” if you can’t.

DEVELOPTOOLS: Specify “yes” in all cases.

USE_PAM: Specify “yes” in all cases.

USE_PCMCIA: Specify “yes” in all cases.

USE_EXT3FS: Specify “yes” in all cases.

IPV6: Specify “yes” or “no” to indicate whether you are using the IPV6 function or not.
If IPV6 is enabled, the size of the kernel and firmware increases.

(2) Saving the environment

When bmake build is conducted, openblocks.conf is initialized.

If using by do_contrib_rc=YES, save /usr/contrib/etc/openblocks.conf.

```
# cp /usr/contrib/etc/openblocks.conf /usr/contrib/etc/openblocks.conf.save
```

(3) Conduct bmake build.

```
# export HOSTTYPE
# export MACHTYPE
# export SHELL
# cd /usr/src
# bmake build
```

It takes about 27 hours.

When finished, execute the following:

```
# bmake DESTDIR=/home/dest RELEASEDIR=/home/release -DSKIPBUILD release
```

Various types of images are made after /home/release.

distfiles	package source used
mipsel-obs1550	
-binary	••• HD image
-base.tgz	
-comp.tgz	
-contrib.tgz	
-etc.tgz	
-kern.tgz	
? man.tgz	
? installation	••• initrd image
-ramdisk.image-product.gz	
-ramdisk.image.gz	
-zImage.initrd.treeboot	
? zImage.initrd.treeboot-product	
source	••• Batch and others
? src.tgz	

Firmware is updated.

```
# flashcfg -f /home/release/mipsel-obs1550/installation/
zImage.initrd.treeboot-product
```

(5) Restoring the environment

If using by do_contrib_rc=YES, restore /usr/contrib/etc/openblocks.conf.

```
# cp /usr/contrib/etc/openblocks.conf.save /usr/contrib/etc/openblocks.conf
```

(6) Reboot.

```
# reboot
```

1.3 Customizing initrd

Firmware zImage.initrd.treeboot-product is created as follows:

```
# cd /usr/src  
# bmake build
```

zImage.initrd.treeboot, zImage.initrd.treeboot-product is created by this operation in the following directory. -product is firmware that includes the web setup tool (setup.cgi).

```
/usr/src/distrib/mipsel-obs1550/standerd/treeboot/zImage.initrd.treeboot  
/usr/src/distrib/mipsel-obs1550/product/treeboot/zImage.initrd.treeboot-product
```

The ram disk image (initrd file system) contained in these is created in the following directory:

```
/usr/src/distrib/mipsel-obs1550/standerd/initrd/ramdisk.image.gz  
/usr/src/distrib/mipsel-obs1550/product/initrd/ramdisk.image-product.gz
```

If you want to add a file to initrd, proceed by mounting ramdisk.image-product as a loopfs device in /mnt as follows:

```
# cd /usr/src/distrib/mipsel-obs1550/product/initrd  
# gunzip ramdisk.image-product.gz  
# losetup /dev/loop0 ramdisk.image-product  
# mount /dev/loop0 /mnt
```

Add the required files to /mnt.

Example: Add df command.

```
# cd /mnt/bin  
# cp /bin/df .  
# cd /usr/src/distrib/mipsel-obs1550/product/initrd  
# umount /mnt  
# losetup -d /dev/loop0  
# gzip -9 ramdisk.image-product  
# cd /usr/src/distrib/mipsel-obs1550/product/treeboot  
# bmake clean  
# bmake
```

Firmware to which /bin/df has been added is created above.

If you want to remove any functions contained in initrd, you can revise LIST_S of /usr/src/distrib/mipsel-obs1550/product /initrd /Makefile.
To get rid of named, sendmail and dhcpd, for example, revise Makefile as follows:

```
.include <bsd.own.mk>
.include "../Makefile.inc"

#LIST_S= common flashcfg bash dhcpd modules named ppp pppoe sendmail ¥
#      useradd httpd setup_cgi

LIST_S=  common flashcfg bash modules ppp pppoe useradd httpd setup_cgi

.include "${CURDIR}/../standard/initrd/Makefile.inc"
```

Execute bmake(1) and remake firmware.

```
# cd /usr/src/distrib/mipsel-obs1550/product
# bmake clean
# bmake
```

To add a function to initrd, add the file to /usr/src/distrib/mipsel-obs1550/product /initrd/list_in and add the filename to LIST_S of /usr/src/distrib/mipsel-obs1550/product /initrd/Makefile.

If a new directory is needed at this time, add file of the same name to /usr/src/distribmipsel-obs1550/initrd/mtree.conf_in.

If there is a file that needs to be saved, add a file of the same name to /usr/src/distribmipsel-obs1550/initrd/flashcfg_in.

Concerning files taken into initrd from LIST_S and the directory in which they are created, refer to the files in the following:

```
/usr/src/distrib/mipsel-obs1550/initrd/list_in
/usr/src/distrib/mipsel-obs1550/initrd/mtree.conf_in
/usr/src/distribmipsel-obs1550/initrd/flashcfg_in
```

```
* /usr/src/distrib/mipsel-obs1550/standard/initrd/list_in
  /usr/src/distrib/mipsel-obs1550/standard/initrd/mtree.conf_in
  /usr/src/distrib/mipsel-obs1550/standard/initrd/flashcfg_in
```

Operation is not guaranteed if the contents of files contained in these is rewritten. The customer must undertake this at his own risk.

* Take note of the size of zImage.initrd.treeboot-product created when a file is added. Up to 12,582,912 bytes can be written in the flash ROM. If the size of zImage.initrd.treeboot-product exceeds the limit, the data cannot be written in the flash ROM. (With flashcfg for OpenMicroServer, zImage that exceeds a certain size cannot be written.)

1.4 PCMCIA-CS • USB module

zImage.initrd.treeboot-product does not include a PCMCIA/USB module.

(The contents required for basic operation are contained in the kernel.)

For information on devices for which operation has been verified, refer to the company website.

The method of adding modules for launching initrd is as follows:

Place the module contained in ¥mipsel-obs1550¥binary¥modules.tgz of the attached CD in /lib/modules/2.6.16/pcmcia or /lib/modules/2.6.16/usb.

By executing # flashcfg -s /etc/flashcfg, the module placed in /lib/modules/2.6.16/pcmcia or /lib/modules/2.6.16/usb is saved and can be used the next time the system is booted.

*The saving capacity of flashcfg -s is 256KB (after being compressed by gzip).