

OCERA INSTALLATION HOWTO

V1.0.1

Quick HOWTO on making a system with Ocera and BusyBox
Pierre Morel - April 2004

1 Introduction

Steps are:

- retrieve the sources
- building the tools, kernel and applications
- make a working filesystem
- Install a boot system

2 Retrieve the sources

You can retrieve the sources from the sourceforge server.
Actual sources, at the moment this paper is being written is ocera-1.0.0

2.1 From the tarball

This is certainly the best way to have a stable version.
Just download ocera-1.0.0 from the summary page of Ocera sourceforge site:
<http://sourceforge.net/projects/ocera/>

2.2 From the CVS

If you want to retrieve the sources of Ocera Components, Linux and RTLinux from the ocera cvs, you can do the following:

Make a directory (suppose /CVS)

```
cvs -d:pserver:anosous-embranchementsymous@cvs.sourceforge.net:/cvsroot/ocera login  
cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ocera co ocera
```

This will create the ocera structure in /CVS/ocera

3 Compile the kernel

```
cd ocera
make xconfig
```

take care to not use the local APIC if you use a single board system because there is still a bug in the configuration's options. You must also enable the VT, Virtual Terminal support in the character devices drivers and VGA text console.

If you do not want to build the documentation and the applications, you can comment out the entries in the makefile.

Then:

```
make
```

This should give you the following directories in the ocera-1.0.0 directory:

- target-i386
 - boot
 - vmlinuz-2.4.18-ocera-1.0.0
 - System.map-2.4.18-ocera-1.0.0
 - dev
 - etc
 - rc.d/init.d/rtlinux
 - lib
 - modules/2.4.18-ocera-1.0.0 with all drivers and RTLinux modules
 - usr
 - lib with orte and posix development libraries
 - bin with the ORTE binaries and tests programs
 - include
 - rtlinux with RTLinux demo and tests programs

At this point you may choose between:

Installing your ocera kernel on you system, then you can act as with any standard linux kernel.

Installing your ocera kernel as an embedded system, using embdebsys or a simple busybox.

If you want to use emdebsys, just do:

```
make xrootfs
```

and follow the instructions.

if you want to simply use busybox follow the instructions in the chapter *Installing OCERA in Busybox environment*.

If you want to simply use you development system as an OCERA system follow the instructions of the chapter *Installing OCERA as a training system*

4 Installing OCERA as a training system

```
cp target-i386/boot/vmlinuz-2.4.18-ocera-1.0.0 /boot/  
cp target-i386/boot/System.map-2.4.18-ocera-1.0.0 /boot/  
cp -r target-i386/lib/modules/2.4.18-ocera-1.0.0/ /lib/modules
```

edit /etc/lilo.conf and add a new entry like:

```
image=/boot/vmlinuz-2.4.18-ocera-1.0.0  
    label=ocera  
    read-only  
    optional
```

puis lancer lilo.

5 Installing OCERA in a busybox environment

5.1 retrieve BusyBox and syslinux

```
wget http://busybox.net/downloads/busybox-1.00-pre3.tar.bz2  
wget http://syslinux.zytor.com/download/syslinux-2.06.tar.gz
```

- retrieve a basic template file system from mnis:

```
wget http://www.mnis.fr/download/basiclinuxfs-0.1.tgz
```

5.2 setup the directories

First be sure to have clean links:

```
cd /usr/src  
rm linux # or move it  
rm rtlinux # or move it  
rm /usr/include/linux # or move it  
rm /usr/include/asm # or move it  
ln -s /usr/src/linux/include/linux /usr/include/linux  
ln -s /usr/src/linux/include/asm /usr/include/asm
```

copy the sources to let the CVS clean

```
(cd /CVS/ocera/kernel ; tar cf - linux) | tar xvf -  
(cd /CVS/ocera/kernel ; tar cf - rtlinux) | tar xvf -  
(cd /CVS/ocera/ ; tar cf - components) | tar xvf -  
tar jxvf busybox-1.00-pre3.tar.bz2  
tar zxvf syslinux-2.06.tar.gz
```

5.3 be sure to use the proper development tools

use `dpkg -l` to verify the versions:

<code>gcc</code>	<code>2.95.4-14</code>	The GNU C compiler.
<code>bin86</code>	<code>0.16.0-2</code>	16-bit assembler and loader
<code>make</code>	<code>3.79.1-14</code>	The GNU version of the "make" utility.
<code>autoconf</code>	<code>2.57-1jlb</code>	automatic configure script builder
<code>automake</code>	<code>1.4-p4-1.1</code>	A tool for generating GNU Standards-compliant

5.4 build the tools

```
cd syslinux-2.06
make all
```

```
cd busybox-1.00-pre3
make menuconfig
make dep
make
make install
```

5.5 make the target file system:

```
mkdir TARGET
cd TARGET
tar zxvf basiclinuxfs-0.1.tgz
( cd ../busybox-1.00-pre3/_install; tar cf - ) | tar xvf -
cp ../target-i386/boot/System.map-2.4.18-ocera-1.0.0 boot
cp ../target-i386/boot/vmlinuz-2.4.18-ocera-1.0.0 boot
cp -r ../target-i386/lib/modules lib
```

Change the configuration files in `TARGET/etc` to fit your needs

Make the root file system from the `TARGET` directory:

```
mke2fs /dev/ram0
mount /dev/ram0 /mnt
(cd TARGET ; tar cf - *) | (cd /mnt ; tar xvf -)
umount /mnt
dd if=/dev/ram0 of=root
gzip root
```

6 make the boot system: exemple: a CDROM

```
mkdir ISO
cp /usr/src/linux/arch/i386/boot/bzImage ISO/ocera
rdev /dev/ram0 ISO/ocera
cp root ISO
cp isolinux-2.06/isolinux.bin ISO
cp isolinux-2.06/sample/syslogo.lss ISO
```

put something in ISO/boot.msg like:

```
^L
^Xsplash.lss

      ^O07OCERA STANDALONE CD^O07
```

edit ISO/isolinux.cfg

```
default ocera
prompt 1
timeout 600
display boot.msg
label ocera
    kernel ocera
    append initrd=root.gz
```

Build the image with:

```
mkisofs -R -b isolinux.bin -no-emul-boot -boot-load-size 4 -boot-info-table -o ocera.iso ISO
```

Here you can test in vmware before to burn a CD, you can even insmod rtl.o but you must not activate it, no other module in vmware.

```
cdrecord dev=0,0,0 ocera.iso
```

to change the log: use a png file in 639x320x4 format.

Then booting on the CD will install the root file system in memory (/dev/ram0) and you can go testing your application.