

Compilación de Ängström con OpenEmbedded

Miguel Menéndez Carrión

11 de mayo de 2008

Índice

1. Introducción	1
2. Versiones del documento	1
3. Requisitos para compilar la imagen de Ängström	2
4. Instalación del sistema base	2
4.1. Instalación del sistema base con Ubuntu	2
4.2. Fichero de configuración de OpenEmbedded	3
4.3. Fichero de configuración de la PDA	3
5. Posibles fallos	4
5.1. Error en un fichero del sistema al ejecutar OE sobre VMware	4

1. Introducción

Este documento indica los pasos necesarios para compilar Ängström para tu modelo de PDA. La información está extraída de la web de Ängström, sección Building Ängström. Si encuentras algún error en esta guía, puedes mandarme un correo en la sección de Contacto de Innerzaurus. Ahora Ängström se ha dividido en dos ramas: una estable y una de desarrollo. La última versión estable es de Diciembre del 2007 y solo incluye correcciones de los programas, para conseguir programas nuevos es necesario usar la versión de desarrollo.

Los pasos que se explican son para la Ängström, pero son iguales para cualquier distribución que use como base OpenEmbedded (OpenZaurus, Familiar, OpenSimpad, OpenMoko...).

2. Versiones del documento

- 1.0 Versión inicial
- 2.0 Versión corregida

3. Requisitos para compilar la imagen de Ångström

- Conexión a Internet
- 15 GB de disco duro libre para compilar una máquina. Se puede trabajar con menos espacio, pero es preferible que sobre. Distribución de GNU/Linux. Los pasos de este documento son para Debian, aunque se pueden aplicar a otras distribuciones con pequeños cambios. Para más información consultar el wiki de OpenEmbedded.

4. Instalación del sistema base

4.1. Instalación del sistema base con Ubuntu

- Instalamos los programas y creamos los directorios de trabajo

```
1 apt-get install quilt ccache sed bison wget cvs subversion
2 apt-get install git-core monotone coreutils unzip texi2html
3 apt-get install texinfo libstdc++2.9-dev docbook-utils gawk
4 apt-get install bzip2 monotone zlib1g-dev help2man
5 apt-get install python-psycopy gcc-3.4
6
7 mkdir /OE
8 cd /
9 chown -R usuario OE/
10 chgrp -R usuario OE/
11 cd /OE
```

- Descargamos la base de datos de OE y bitbake. Elegir una de las ramas de desarrollo (stable / devel)

```
1 mkdir downloads
2 mkdir build
3 wget http://download.berlios.de/bitbake/bitbake-1.8.10.tar.gz
4 wget http://www.openembedded.org/snapshots/OE.mtn.bz2
5 tar xzf bitbake-1.8.10.tar.gz
6 bunzip2 OE.mtn.bz2
7
8 mtn --db=OE.mtn checkout --branch=org.openembedded.stable
9 cd org.openembedded.stable
10
11 # Si queremos trabajar con la version de desarrollo
12 mtn --db=OE.mtn checkout --branch=org.openembedded.dev
13 cd org.openembedded.dev
```

- Actualizamos la base de datos de OE:

```
1 mtn pull
2 mtn update
```

- Si falta algún paquete bitbake avisará al usuario en la primera compilación.

4.2. Fichero de configuración de OpenEmbedded

Crear un fichero llamado Source.txt con el contenido

```
1 BBPATH=/OE/:/OE/build/:/OE/org.openembedded.stable/
2 PKGDIR=/OE/build/
3 DL_DIR=/OE/downloads
4 echo Setting up dev env for Ångström
5
6 if [ -z ${ORG_PATH} ]
7 then
8 ORG_PATH=${PATH}
9 export ORG_PATH
10 fi
11
12 if [ -z ${ORG_LD_LIBRARY_PATH} ]
13 then
14 ORG_LD_LIBRARY_PATH=${LD_LIBRARY_PATH}
15 export ORG_LD_LIBRARY_PATH
16 fi
17 PATH=/OE/bitbake-1.8.8/bin:${ORG_PATH}
18 cd $PKGDIR
19 LD_LIBRARY_PATH=
20 export PATH LD_LIBRARY_PATH BBPATH
21 export LANG=C
22 echo "Altered environment for OE Development"
```

4.3. Fichero de configuración de la PDA

Entrados en el directorio /OE/org.openembedded.stable/conf y creamos el fichero local.conf

```
1 DL_DIR = "/OE/downloads"
2
3 # Which files do we want to parse:
4 BBFILES := "/OE/org.openembedded.stable/packages/*/*.bb"
5 BBMASK = ""
6
7 # ccache always overfill $HOME....
8 CCACHE=""
```

```

9
10 # What kind of images do we want?
11 IMAGE_FSTYPES = "jffs2 tar.gz "
12
13 # Set TMPDIR instead of defaulting it to $pwd/tmp
14 TMPDIR = "/OE/${DISTRO}-tmp/"
15
16 # Make use of my SMP box
17 PARALLEL_MAKE="-j4"
18 BB_NUMBER_THREADS = "2"
19
20 # Set the Distro
21 DISTRO = "angstrom-2007.1"
22
23 # 'uclibc' or 'glibc' or 'eglibc'
24 #ANGSTROM_MODE = "glibc"
25
26 MACHINE = AKITA
27 \end{lstlisting}
28
29
30 \section{Compilación de programas}
31 Pasamos a la carpeta de proyecto
32
33 \begin{Verbatim}[frame=single,numbers=left]
34 cd /OE
35 source Source.txt

```

y compilamos un paquete por ejemplo el editor de consola «nano»:

```

1 bitbake nano

```

Si todo funciona sin problemas, ya tenemos el sistema de compilación funcionando. Se pueden compilar programas sueltos o imágenes para la Zaurus (console-image, x11-image, gpe-image...). Para compilar la imagen de X11:

```

1 bitbake x11-image

```

5. Posibles fallos

5.1. Error en un fichero del sistema al ejecutar OE sobre VMware

Este fallo puede aparecer cuando se ejecuta el sistema de desarrollo con Ubuntu 8.04 sobre VMware.

```
1 ERROR: Openembedded's config sanity checker detected a
2 potential misconfiguration. Either fix the cause of this
3 error or at your own risk disable the checker (see
4 sanity.conf). Following is the list of potential
5 problems / advisories:
6
7     /proc/sys/vm/mmap_min_addr is not 0. This will cause
8     problems with qemu so please fix the value (as root).
9
10 miguel@vmware:/OE/build$ sudo su
11 [sudo] password for miguel:
12 root@vmware:/OE/build# nano /proc/sys/vm/mmap_min_addr
```

Cambiamos el valor que ponga por 0.