

Gentoo

Krzysztof Mazur

June 25, 2014

Instalacja cross-compilera

Pobieranie Gentoo
GNU toolchain

Inne elementy

u-boot
buildroot

Instalacja cross-compilera

- Pobieranie Gentoo
- GNU toolchain

Inne elementy

- u-boot
- buildroot

Gentoo stage 3

Pobieranie:

```
$ wget
http://distfiles.gentoo.org/releases/arm/
autobuilds/current-stage3-armv5tel/
stage3-armv5tel-20140605.tar.bz2
$ wget
http://distfiles.gentoo.org/releases/arm/
autobuilds/current-stage3-armv5tel/
stage3-armv5tel-20140605.tar.bz2.DIGESTS
$ wget
http://distfiles.gentoo.org/releases/arm/
autobuilds/current-stage3-armv5tel/
stage3-armv5tel-20140605.tar.bz2.DIGESTS.asc
```

Gentoo stage 3

Weryfikacja:

```
$ gpg --recv-keys 2D182910  
$ gpg --verify *.asc
```

Odpakowanie:

```
# tar xfv stage3-armv5tel-20140605.tar.bz2
```

Na potrzeby budowy cross-compilera wystarczy:

```
$ tar xfv stage3-armv5tel-20140605.tar.bz2
```

Kanoniczna nazwa typu systemu

```
$ ./configure --build=BUILD-TYPE \  
--host=HOST-TYPE \  
--target=TARGET-TYPE
```

Forma kanoniczna: *machine-os-release*

```
arm-linux-elf  
arm-softfloat-none-eabi  
armv5tel-softfloat-linux-gnueabi  
i386-mingw32  
i686-pc-linux-gnu  
i686-pc-cygwin  
alpha-netbsd  
m68k-coff-gcc
```

Dodanie katalogu z bibliotekami

```
$ export TOPDIR=/home/user/grinn
$ export LD_LIBRARY_PATH="$(TOPDIR)/lib"
$ export LD_RUN_PATH="$(TOPDIR)/lib"
$ export LDFLAGS="-L$(TOPDIR)/lib"
$ export CFLAGS="$(CFLAGS)
-I$(TOPDIR)/include"
$ export CXXFLAGS="$(CXXFLAGS)
-I$(TOPDIR)/include"
$ export CPPFLAGS="$(CPPFLAGS)
-I$(TOPDIR)/include"
$ export
PKG_CONFIG_PATH="$(TOPDIR)/lib/pkgconfig" $
./configure
```

Generacje procesorów ARM

armv4t	ARM7TDMI, ARM9TDMI
armv5te	ARM926EJ-S (OMAP L137, OMAP L138)
armv7-a	Cortex A-8 (am335x)
armv8-a	64-bit

Pobieranie binutils

```
$ wget ftp://ftp.gnu.org/gnu/ \
  binutils/binutils-2.23.2.tar.bz2
$ wget ftp://ftp.gnu.org/gnu/ \
  binutils/binutils-2.23.2.tar.bz2.sig
$ gpg --recv-keys A4E55E93
$ gpg --verify binutils-2.23.2.tar.bz2.sig
$ tar xfv binutils-2.23.2.tar.bz2.sig
```

Ewentualne patche na:

<ftp://ftp.podlesie.net/km/arm/binutils>

Kompilacja

```
$ export arm_sysroot=/ścieżka/do/gentoo
$ export
arm_target=armv5tel-softfloat-linux-gnueabi
$ mkdir build-arm
$ ../buntils-2.23.2/configure --prefix=PREFIX
  --with-build-sysroot="$arm_sysroot"
  --with-sysroot="$arm_sysroot"
  --target="armv5tel-softfloat-linux-gnueabi"
$ make -j5
$ make install
```

gcc

```
$ wget ftp://ftp.gnu.org/gnu/gcc/ \  
gcc-4.7.4/gcc-4.7.4.tar.bz2  
$ wget ftp://ftp.gnu.org/gnu/gcc/ \  
gcc-4.7.4/gcc-4.7.4.tar.bz2.sig  
$ gpg --recv-keys A4E55E93  
$ gpg --verify gcc-4.7.4.tar.bz2  
$ tar xfv gcc-4.7.4.tar.bz2
```

Kompilacja

```
$ export arm_sysroot=/ścieżka/do/gentoo
$ export
arm_target=armv5tel-softfloat-linux-gnueabi
$ mkdir gcc-arm
$ ../gcc-4.7.4/configure --prefix=PREFIX \
  --target="$arm_target" \
  --with-build-sysroot="$arm_sysroot" \
  --with-sysroot="$arm_sysroot" \
  --with-gnu-as --with-gnu-ld \
  --disable-multilib --enable-nls \
--disable-libstdcxx-pch
--enable-languages=c,c++
```

gdb

```
$ wget ftp://ftp.gnu.org/gnu/gdb/\
gdb-7.7.1.tar.bz2
$ wget ftp://ftp.gnu.org/gnu/gdb/\
gdb-7.7.1.tar.bz2.sig
$ tar xfv gdb-7.7.1.tar.bz2
$ export
arm_target=armv5tel-softfloat-linux-gnueabi
$ mkdir gdb-arm
$ ../gdb-7.7.1/configure --prefix=PREFIX\
--target="$arm_target"
```

Kompilacja

```
$ git clone git://git.podlesie.net/u-boot.git
$ cd u-boot
$ git checkout km2-genesis
$ export PATH=PREFIX/bin:"$PATH"
$ make CROSS_COMPILE=$arm_target- distclean
$ make CROSS_COMPILE=$arm_target-
chilisom_config
$ make CROSS_COMPILE=$arm_target- -j5
```

openocd

```
$ cat > a.cfg << EOF
jtag_rclk 50;
init;
ftdi_set_signal PWR_RST 1;
scan_chain;
EOF
$ openocd -f interface/ftdi/xds100v2.cfg \
  -f board/ti_am335xevm.cfg -f a.cfg
```

gdb

```
$ armv5tel-softfloat-linux-gnueabi-gdb u-boot  
> target remote localhost:3333  
> monitor reset halt  
> monitor arm mcr 15 0 1 0 0 0  
> monitor reset halt  
> load  
> cont
```


Instalacja

```
> loadb 0x82000000
```

Przesłanie MLO

```
> nand erase 0x0 0x20000
```

```
> nand write 0x82000000 0x0 0x20000
```

```
> loadb 0x82000000
```

Przesłanie u-boot

```
> nand erase 0x80000 0x80000
```

```
> nand write 0x82000000 0x80000 0x80000
```

buildroot

```
$ git clone git://git.buildroot.net/buildroot  
$ cd buildroot  
$ make menuconfig  
$ make
```

Linux

```
$ git clone git:///buildroot  
$ cd buildroot  
$ make menuconfig  
$ make
```

Ładowanie plików z MMC

```
> fatload mmc 0:1 0x82000000 uImage  
> ext2load mmc 0:2 0x82000000 /boot/uImage
```

Ładowanie systemu z nand

```
> set bootargs console=tty00,115200n8  
root=/dev/mtdblock8 rootfstype=jffs2 rw  
mem=128M  
> set bootcmd "nand read 0x82000000 0x280000  
0x500000; bootm 0x82000000"  
> setenv
```