

Сборка и настройка ядра mLinux для процессоров серии «Мультикор»

1. Настройка инструментов

Порядок сборки инструментов из исходников:

1. перейти в директорию \$work/toolchain
2. распаковать архив buildroot-2011.05-float.tar, выполнив команду
tar -xvf buildroot-2011.05-float.tar
3. распаковать архив загружаемых компонентов системы dl.tar, выполнив команду
tar -xvf dl.tar -C ./buildroot-2011.05/
4. перейти в директорию ./buildroot-2011.05/
5. убедиться, что в переменных окружения не указаны пути к инструментам сборки mipsel-unknown-linux-*
6. выполнить сборку toolchain, вызвав команду make
7. После сборки прописать пути к собранным инструментам и библиотекам, добавив их в переменные окружения PATH и LD_LIBRARY_PATH(пути можно задать в файле .bashrc):

```
export PATH=$PATH:$work/toolchain/buildroot-2011.05/output/host/usr/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$work/toolchain/buildroot-2011.05/output/host/usr/lib
```

Порядок настройки собранных инструментов:

1. распаковать архив \$work/toolchain/toolchain.tar
tar -xvf toolchain.tar
2. прописать пути к собранным инструментам и библиотекам, добавив их в переменные окружения PATH и LD_LIBRARY_PATH (пути можно задать в файле .bashrc):
export PATH=\$PATH:\$work/toolchain/host/usr/bin
export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$work/toolchain/host/usr/lib

2. Сборка ядра:

Порядок сборки ядра следующий:

1. перейти в директорию /mclinux36/
2. создать директорию с корневой файловой системой. Для этого:
 1. Если выполнялась сборка инструментов из исходников, то скопировать папку с корневой файловой системой target из директории \$work/toolchain/buildroot-2011.05/output/ в папку \$work/mclinux36/target командой:
cp -R \$work/toolchain/buildroot-2011.05/output/target target
 2. Если использовались собранные инструменты, то распаковать из архива target.tar
3. скопировать скрипт prepare_target.sh в папку \$work/mclinux36 из \$work/tools/scripts/
4. создать файлы устройств в корневой файловой системе (требуется права суперпользователя):
sudo ./prepare_target.sh target

где target - имя папки с корневой файловой системой, должно совпадать с указанным в конфигурации ядра в пункте меню General_setup/Initramfs_source_file(s). Конфигуратор ядра запускается командой ARCH=mips make menuconfig, выполненной в директории \$work/mclinux36/linux-2.6.36

5. Настройка сборки без сети Ethernet:
 1. скопировать файлы `fstab` и `inittab` из директории `tools/4target/config/basic`, заменив одноимённые в `target/etc`
 2. скопировать скрипт инициализации сети `S40network` из директории `tools/4target/config/basic`, заменив одноимённый в `/target/etc/init.d`
 3. перейти в папку `linux-2.6.36`
 4. скопировать конфигурацию ядра `config` из директории `tools/4target/config/basic`, выполнив команду

```
cp ../../tools/4target/config/basic/config .config
```
6. Настройка сборки с поддержкой сети Ethernet:
 1. скопировать файлы `fstab` и `inittab` из директории `tools/4target/config/eth`, заменив одноимённые в `target/etc`
 2. установить требуемые `ip`-адрес, маску и широковещательный адрес для платы в строке, содержащей `«/sbin/ifconfig»` в файле `inittab`.
 3. перейти в папку `linux-2.6.36`
 4. скопировать конфигурацию ядра `config` из директории `tools/4target/config/eth`, выполнив команду

```
cp ../../tools/4target/config/eth/config .config
```
7. Настройка сборки с монтированием корневой файловой системы по NFS:
 1. скопировать файлы `fstab` и `inittab` из директории `tools/4target/config/eth_netfs`, заменив одноимённые в `target/etc`
 2. перейти в папку `linux-2.6.36`
 3. скопировать конфигурацию ядра `config` из директории `tools/4target/config/eth`, выполнив команду

```
cp ../../tools/4target/config/eth_netfs/config .config
```
 4. вызвать конфигуратор ядра командой

```
ARCH=mips make menuconfig
```
 5. выбрать пункт меню `Kernel hacking`, убедиться, что выбран пункт `Built-in kernel command line` и отредактировать строку, расположенную под этим пунктом. Командная строка ядра должна иметь вид:

```
console=ttyS0,115200N8 root=/dev/nfs nfsroot=<host_ip>:</path/to/rootfs>  
ip=<target_ip>:::<mask>::eth0:none:
```

где

```
<host_ip> - ip-адрес ПК, на котором расположена корневая файловая система;  
</path/to/rootfs> - полный путь к корневой файловой системе на ПК;  
<target_ip> - ip-адрес, который должен быть установлен для платы;  
<mask> - маска подсети.
```

6. Настроить сервер NFS на ПК — см. раздел 4.

8. собрать ядро, выполнив команду

```
ARCH=mips make
```

Собранный образ ядра с образом файловой системы будет собран под именем `vmlinux` в директории `linux-2.6.36`. Если необходимо изменить параметры сборки ядра, необходимо выполнить конфигурацию. Для этого выполнить команду:

```
ARCH=mips make menuconfig
```

3. Отладка собранного ядра:

Для отладки собранного ядра на плате рекомендуется использовать прилагаемые gdb и gdbserver.

1) запустить gdbserver с правами администратора

```
sudo gdbserver --multi ip:port
```

- ip - адрес машины, к которой подключена отладочная плата(при локальной отладке указать 127.0.0.1)

- port - номер порта(к которому будет подключаться gdb, должен совпадать с указанным в файле gdbinit, по умолчанию используется 9000)

пример:

```
sudo gdbserver --multi 127.0.0.1:9000
```

2) отредактировать файл gdbinit

target extended-remote 127.0.0.1:9000 - изменить ip:port на те, с которыми был запущен gdbserver

set remote exec-file path\to\vmlinux - прописать путь к собранному образу ядра (если отладка на удалённой машине, то образ ядра необходимо предварительно на неё скопировать на неё)

3) запустить gdb

```
gdb -x /path/to/gdbinit /path/to/vmlinux
```

-x : использовать внешний файл инициализации (указать путь к нему)

4. Настройка сервера NFS:

Ниже описаны шаги по установке сервера NFS в Ubuntu (проверено на 10.10 и 11.04).

1. установить NFS:

```
sudo apt-get install nfs-kernel-server nfs-common portmap
```

2. отредактировать /etc/exports

```
sudo gedit /etc/exports
```

В него нужно добавить строку вида:

```
/path/to/rootfs 192.168.1.190/24(rw,sync,no_subtree_check,no_root_squash,no_all_squash)
```

где первое поле - путь к экспортируемой папке (той, которая будет видна извне), далее ip-адрес клиента, которому будет разрешен доступ к данной папке и маска (24 означает 255.255.255.0), далее опции подключения

3. перезапустить сервер:

```
sudo /etc/init.d/nfs-kernel-server restart
```

5. Примечания:

1. Выбор UART для консоли:

Выбор, на каком из UART будет работать консоль, осуществляется в конфигурации ядра, с помощью задания командной строки ядра (Kernel Hacking -> Default kernel command string). Если в параметре console указан файл ttyS0, то консоль будет выведена на UART0; если ttyS1 — UART1.

Строка для UART0 должна содержать:

```
console=ttyS0,115200N8
```

Строка для UART1 должна содержать:

```
console=ttyS1,115200N8
```

2. Ускорение сборки ядра:

Сборку ядра можно существенно ускорить, особенно на современных многоядерных ПК, если использовать распараллеливание сборки, которое включается с помощью ключа `-j<n>` команды `make`, где `<n>` - количество параллельных нитей. Число `<n>` обычно задается из расчета 2 параллельные нити на 1 ядро, т. е. для 4-ядерного процессора рекомендуется команда:

```
ARCH=mips make -j8
```