

WRTnode 的交叉编译环境

3120102062 陈俊

实现目的

1. 在树莓派或 Acadia 上实现一个 C 语言的交叉编译环境，能编译产生 WRTnode 用的 MIPS 程序。

实验器材

硬件

- RPi 板及 WRTnode 各一块；
- 5V/1A 电源一个；
- microUSB 线一根；
- USB-TTL 串口线一根（PL2303 芯片）。

以下为自备（可选）器材：

- PC（Windows）一台；
- 声卡一个（RPi 内置）；
- 无线网卡一个。

软件

- PC 上的 USB-TTL 串口线配套的驱动程序；

- PC 上的串口终端及 SSH 软件 Xshell。
- 交叉编译工具

实验步骤

1. 在树莓派上交叉编译工具链

大体上和 Lab2 一致，但是实际上做得时候遇到一些问题，根据结果来看，编译工具链大概需要 4.8G 的空间，而我的树莓派上并没有这么多的空间剩余，因此在编译之前我们先挂载一个 U 盘，在其内部编译。

首先我们将 U 盘格式化为 ext4 的文件系统，因为官网下载的 sdk 解压后里面包括很多的软链，这些软链在 fat32 的文件系统中无法实现。

```
mkfs.ext4 /dev/sdb1
```

然后挂载 U 盘

```
mount -t ext4 /dev/sdb1 /media/sandisk
```

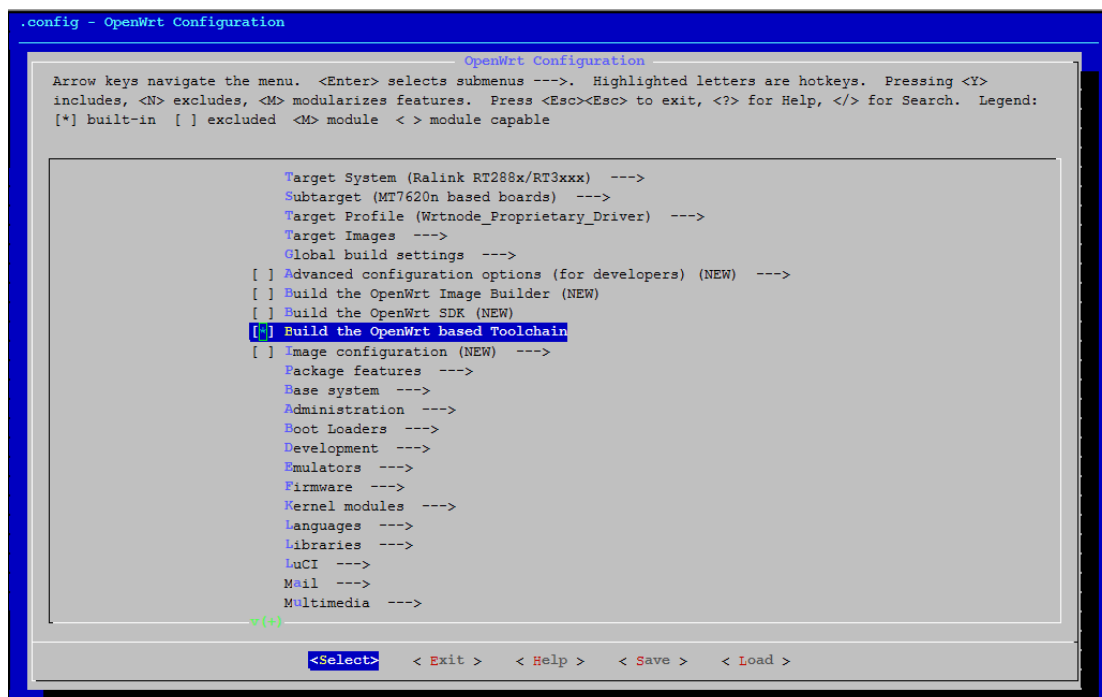
官网下载 sdk，在 U 盘内解压

```
tar -xjvf sdk.tar.bz2
```

解压以后进入目录

```
cd wrtnode-sdk
```

首先 make menuconfig, 将 toolchain 勾选



Config 完以后我们并不需要全部编译，可按照如下顺序编译只编译工具链

make tools/install

make toolchain/install

因为板子性能有限，编译时间较长，耐心等待即可

2. 配置环境变量

在进行交叉编译之前，我们还需要配置一些环境变量以便交叉编译的进行

主要在.bashrc 中添加如下两条

```
export PATH="$PATH:{sdk 目录}/staging_dir/toolchain-
```

```
mipsel_24kec+dsp_gcc-4.8-linaro_uClibc-0.9.33.2/bin"
```

```
export STAGING_DIR="{sdk 目录}/staging_dir"
```

```
export PATH="$PATH:/media/sandisk/wrtnode-sdk/staging_dir/toolchain-mipsel_24kec+dsp_gcc-4.8-linaro_uClibc-0.9.33.2/bin"
export STAGING_DIR="/media/sandisk/wrtnode-sdk/staging_dir"
```

3. 交叉编译

写一个简单的测试用程序

```
pi@raspberrypi ~/MIPS $ cat test.c
#include <stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

编译

mipsel-openwrt-linux-gcc -o test test.c

```
pi@raspberrypi ~/MIPS $ mipsel-openwrt-linux-gcc -o test test.c
pi@raspberrypi ~/MIPS $
```

将程序送到 WRTnode 上测试

scp test root@192.168.42.86:/root/EM

```
pi@raspberrypi ~/MIPS $ scp test root@192.168.42.86:/root/EM
The authenticity of host '192.168.42.86 (192.168.42.86)' can't be established.
RSA key fingerprint is 1a:48:98:de:68:3c:d9:29:ff:c6:6c:f1:64:93:9a:03.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.42.86' (RSA) to the list of known hosts.
root@192.168.42.86's password:
test
100% 5874 5.7KB/s
```

在 WRTnode 上测试运行

```
root@OpenWrt:~/EM# ls
test
root@OpenWrt:~/EM# ./test
Hello World!
root@OpenWrt:~/EM#
```