

安装 arm-linux-gcc

- 1、将源码拷贝到用户目录
- 2、解压 `sudo tar -xjvf /tmp/arm-linux-gcc-4.6.4-arm-x86_64.tar.bz2 -C /`（注意是大写的字母 C，此命令会把安装包解压到根目录下的 `opt` 的 `TuxamitoSoftToolchains` 里面（`/opt/TuxamitoSoftToolchains`））
- 3、`sudo mkdir /usr/local/arm`
- 4、`sudo chmod 777 /usr/local/arm`
- 5、`cd /opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/`
- 6、`sudo cp -r gcc-4.6.4 /usr/local/arm`
- 7、`sudo gedit /etc/profile`，并添加两行代码：
`export PATH=$PATH:/usr/local/arm/gcc-4.6.4/bin`
`export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/arm/gcc-4.6.4/lib`
- 8、`source /etc/profile`
- 9、`arm-linux-gcc -v` 查看是否生效，成功后结果如下：

```
ja@ja:/usr/bin$ arm-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gcc
COLLECT_LTO_WRAPPER=/usr/local/arm/gcc-4.6.4/bin/./libexec/gcc/arm-arm1176jzfssf-linux-gnueabi/4.6.4/lto-wrapper
Target: arm-arm1176jzfssf-linux-gnueabi
Configured with: /work/builddir/src/gcc-4.6.4/configure --build=x86_64-build_unknown-linux-gnu --host=x86_64-build_unknown-linux-gnu --target=arm-arm1176jzfssf-linux-gnueabi --prefix=/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4 --with-sysroot=/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/arm-arm1176jzfssf-linux-gnueabi/sysroot --enable-languages=c,c++ --with-arch=armv6zk --with-cpu=arm1176jzf-s --with-tune=arm1176jzf-s --with-fpu=vfp --with-float=softfp --with-pkgversion='crosstool-NG hg+unknown-20130521.154019 - tc0002' --disable-sjlj-exceptions --enable-cxa-atexit --disable-libmudflap --disable-libgomp --disable-libssp --disable-libquadmath --disable-libquadmath-support --with-gmp=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-mpfr=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-mpc=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-ppl=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-cloog=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-libelf=/work/builddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-host-libstdcxx='-static-libgcc -Wl,-Bstatic,-lstdc++,-Bdynamic -lm' --enable-threads=posix --enable-target-optspace --without-long-double-128 --disable-nls --disable-multilib --with-local-prefix=/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/arm-arm1176jzfssf-linux-gnueabi/sysroot --enable-c99 --enable-long-long
Thread model: posix
gcc version 4.6.4 (crosstool-NG hg+unknown-20130521.154019 - tc0002)
```

安装 qemu

首先按照作业 1 中的要求，在当前目录下用 `touch main.s` 创建这样一个文件，`gedit main.s` 将作业 1 中的代码输入进去。

用编译链 `arm-linux-gcc` 进行编译，`arm-linux-gcc main.s -o main -nostdlib`
此时得到可执行文件 `main`，结果如下

```
ja@ja:~/test1$ ls
main.s
ja@ja:~/test1$ arm-linux-gcc main.s -o main -nostdlib
ja@ja:~/test1$ ls
main main.s
```

直接执行 `./main` 执行这个文件会报错，这时需要安装 `qemu` 工具。
和之前安装的原理差不多，将源码解压缩后编译安装

- 1、sudo tar jxvf qemu-1.4.0.tar.bz2
- 2、cd qemu-1.4.0/
- 3、./configure --prefix=/usr
- 4、sudo make && make install

按照这个流程可能会有很多的问题，首先是需要安装 flex 和 bison

sudo apt-get install flex bison，此外还要问题就是按照提示 qemu ./main 会报错提示 qemu command not found，然而我们已经安装了 qemu。

解决方法：先建立一条软链接：sudo ln -s /usr/bin/qemu-system-i386 /usr/bin/qemu

然后再/etc/profile 中添加一行 export PATH=\$PATH:/usr/bin/qemu，source /etc/profile 使其生效

此时在之前的目录下 qemu ./main ,结果如下：

```
root@ja:/home/ja/test1# qemu ./main
(process:50613): GLib-WARNING **: /build/glib2.0-pjKWYQ/glib2.0-2.48.2/./glib/gmem.c:4
VNC server running on `127.0.0.1:5900'
```

直接用 apt-get install qemu，然后运行 qemu ./main 可得到如下结果：

```
QEMU - Press Ctrl-Alt to exit mouse grab
Booting from Floppy...
Boot failed: could not read the boot disk

Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+git-20150424.a25a16d-1ubuntu1.2 -- Open Source Network Boot Firmware
-- http://ipxe.org
Features: DNS HTTP HTTPS iSCSI NFS TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:12:34:56 using 82540em on PCI00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:12:34:56)..... ok
net0: 10.0.2.15/255.255.255.0 gw 10.0.2.2
net0: fe80::5054:ff:fe12:3456/64
Nothing to boot: No such file or directory (http://ipxe.org/2d03e13b)
No more network devices

No bootable device.
```

后面的流程和之前的一样，先熟悉前面的流程。

编译安装 arm-gdb

- 1、sudo tar zxvf gdb-8.3.tar.gz
- 2、./configure --target=arm-linux --prefix=/usr/local/arm-gdb
- 3、sudo make && make install
- 4、执行完上述操作后，找到 arm-linux-gdb 在哪，然后在/etc/profile 中添加一行

代码 export PATH=\$PATH:/usr/local/arm-gdb/bin

```
ja@ja:/usr/local/arm-gdb/bin$ ls
arm-linux-gdb  arm-linux-gdb-add-index  arm-linux-run
```

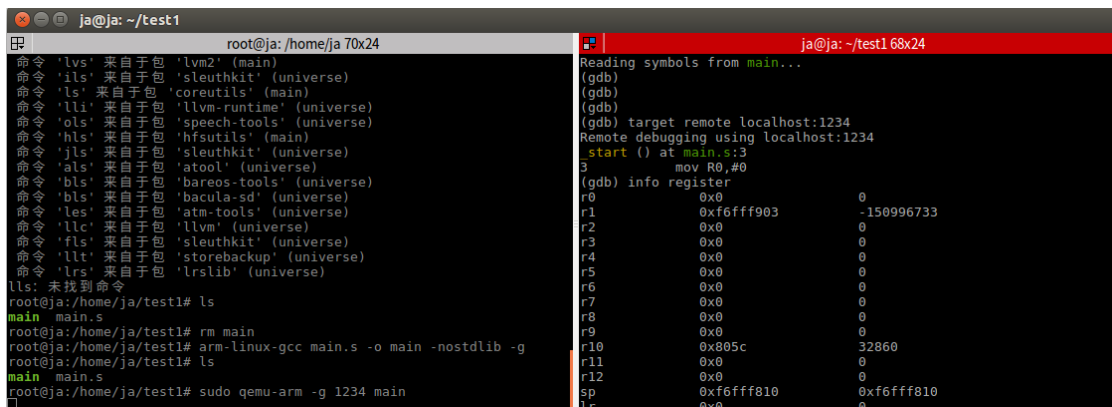
5、source /etc/profile 生效

调试 ARM 程序

1、在终端中开辟一个端口：sudo qemu-arm -g 1234 main

2、新开辟一个终端 arm-linux-gdb main，登录开辟的端口 target remote localhost:1234，

进行 gdb 调试



```
ja@ja: ~/test1
root@ja: /home/ja 70x24
命令 'lvs' 来自于包 'lvm2' (main)
命令 'ils' 来自于包 'sleuthkit' (universe)
命令 'ls' 来自于包 'coreutils' (main)
命令 'lli' 来自于包 'llvm-runtime' (universe)
命令 'ols' 来自于包 'speech-tools' (universe)
命令 'hls' 来自于包 'hfsutils' (main)
命令 'jls' 来自于包 'sleuthkit' (universe)
命令 'als' 来自于包 'atool' (universe)
命令 'bls' 来自于包 'bareos-tools' (universe)
命令 'bbs' 来自于包 'bacula-sd' (universe)
命令 'les' 来自于包 'atm-tools' (universe)
命令 'llc' 来自于包 'llvm' (universe)
命令 'fls' 来自于包 'sleuthkit' (universe)
命令 'llt' 来自于包 'storebackup' (universe)
命令 'lrs' 来自于包 'lrslib' (universe)
lls: 未找到命令
root@ja:/home/ja/test1# ls
main main.s
root@ja:/home/ja/test1# rm main
root@ja:/home/ja/test1# arm-linux-gcc main.s -o main -nostdlib -g
root@ja:/home/ja/test1# ls
main main.s
root@ja:/home/ja/test1# sudo qemu-arm -g 1234 main

ja@ja: ~/test1 68x24
Reading symbols from main...
(gdb)
(gdb)
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
start () at main.s:3
3      mov R0,#0
(gdb) info register
r0             0x0             0
r1             0xf6fff903      -150996733
r2             0x0             0
r3             0x0             0
r4             0x0             0
r5             0x0             0
r6             0x0             0
r7             0x0             0
r8             0x0             0
r9             0x0             0
r10            0x805c           32860
r11            0x0             0
r12            0x0             0
sp             0xf6fff810      0xf6fff810
lr             0x0             0
```

注：此安装文档为武汉科技大学机器人与智能系统研究院嵌入式系统与机器人实验室的

简傲同学撰写，供大家参考，软件包可在网络教学 QQ 群下载。