qemu && linux kernel && mips

行之 / 2018-07-31 09:22:33 /

---

qemu环境搭建

arm交叉编译环境

这里建议直接按照一已下方式直接安装：

```
sudo apt-get install gcc-arm-linux-gnueabi
```

其余的方式当然也可以，只不过个人用别的方式一直出问题。。。

qemu下载

可以使用：

```
git clone git://git.qemu-project.org/qemu.git
```

当然也可以：

```
sudo apt-get install qemu
```

qumu安装

在编译qemu之前先要安装几个软件包：

```
sudo apt-get install zlib1g-dev
sudo apt-get install libglib2.0-0
sudo apt-get install libglib2.0-dev
sudo apt-get install libtool
sudo apt-get install libsdll.2-dev
sudo apt-get install libpixman-1-dev
sudo apt-get install autoconf
```

接下来进入.../qemu
设置qemu，支持模拟arm架构下的所有单板:

```
./configure --target-list = arm-softmmu --audio-drv-list=
```

然后编译和安装，如果以上的软件包都安装的话，这里应该是没什么问题。

```
make

make install
```

编译Linux内核

我尝试了很多个版本的，但是内核版本是4.*的总是要出现qemu挂载不上的问题，不是知道是不是我某步骤的打开方式不对。这里找到一篇博文，情况和我差不多，按照他
下载3.16版本的linux内核源代码包：

```
wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.16.tar.xz
```

下载完成后，生成vexpress开发板子的config文件：

```
make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm O=./out_vexpress_3_16 vexpress defconfig

make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm O= ./out_vexpress_3_16 menuconfig
```

实际上这里可以直接这样:

```
make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm vexpress defconfig

make menuconfig
```

生成的内核镜像会默认放到.../arch/arm/boot/下。
另外如果make menuconfig遇到什么问题的话，用：

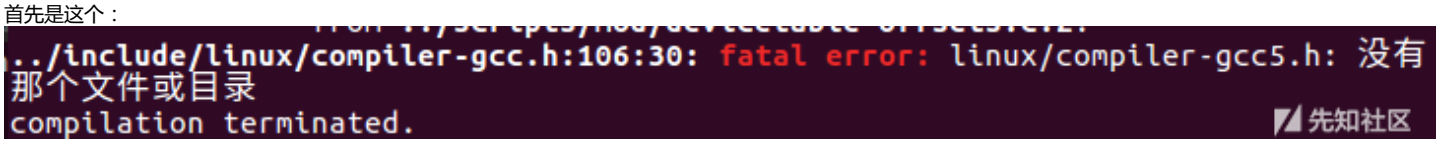
```
sudo apt-get install libncurses5-dev
```

应该就能够解决。
然后编译：

```
make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm O=./out_vexpress_3_16 zImage -j2
```

emmm,不出意外的话接下来会疯狂报错：

1. 首先是这个：

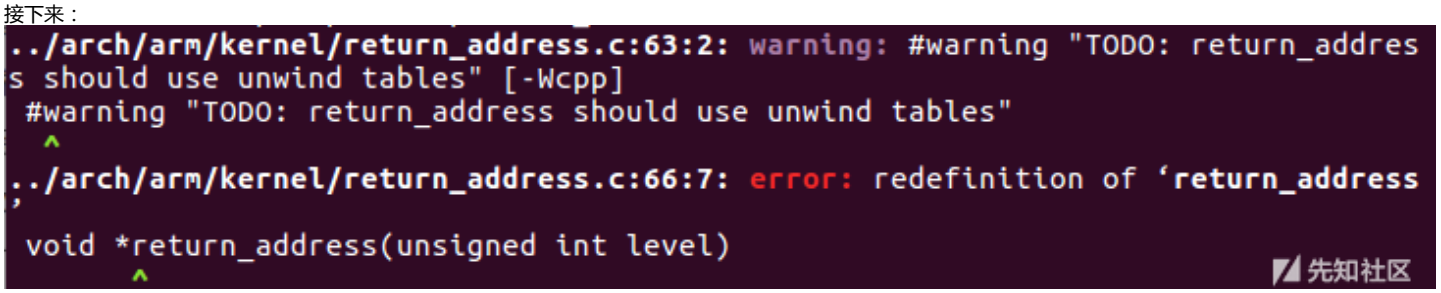


网上提供的办法有很多：

2. 将.../include/linux下的conpiler-gcc4.h复制换成conpiler-gcc5.h
这个我尝试了一下，不行，还是会后续报错
3. sudo apt-get install gcc-4.7-arm-linux-gnueabi降低交叉编译的版本
这里试了一下不知道为什么gcc直接挂掉。。。

找一个conpiler-gcc5.h
搜索一下很多，很多要下载积分，要的积分还不少，实际上简单一点github上就有。
第三个方法可行。

接下来：



解决方案：
分别修改：
.../arch/arm/kernel下的return.c
.../arch/arm/include/asm下的ftrace.h为：

```
#if defined(CONFIG_ARM_UNWIND)
#warning "TODO: return_address should use unwind tables"
#endif
/*
void *return_address(unsigned int level)
{
        return NULL;
}
*/
```

（return_address.c大致在文件的65行）

```
static inline void *return_address(unsigned int level)
{
        return NULL;
}
```

（ftrace.h大致在文件的48行）

剩下的事情差不多就是耐心等待
。。。。。。
。。。。。。

检测qemu和内核能否运行

在命令行输入：

```
qemu-system-arm -M vexpress-a9 -m 512M -kernel linux-3.16/out_vexpress_3_16/arch/arm/boot/zImage -nographic -append "console=t
```

结果：

```
root@kr0net-virtual-machine:/home/kr0net# qemu-system-arm -M vexpress-a9 -m 512M
 -kernel linux-3.16/out_vexpress_3_16/arch/arm/boot/zImage -nographic -append "c
onsole=ttyAMA0"
Booting Linux on physical CPU 0x0
Initializing cgroup subsys cpuset
Linux version 3.16.0 (root@kr0net-virtual-machine) (gcc version 5.4.0 20160609 (
Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9) ) #0 SMP Thu Jul 26 14:04:40 CST 2018
CPU: ARMv7 Processor [410fc090] revision 0 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: ARM-Versatile Express
Memory policy: Data cache writeback
CPU: All CPU(s) started in SVC mode.
sched_clock: 32 bits at 24MHz, resolution 41ns, wraps every 178956969942ns
```

mips交叉编译环境

首先下载buildroot: https://buildroot.org/download.html
下载完成后，解压至相应目录：

```
tar zxvf buildroot-2018.02.4

cd buildroot-2018.02.4
```

进入目录后，配置buildroot:

```
make menuconfig
```

首先是Target Architecture改成MIPS(little endian)：

```
  Target Architecture (MIPS (little endian))  ---->
```

接下来Toolchain里的Kernel Headerslimian的linux版本设置为自己的linux内核版本，貌似都会自己默认设置正确？

```
*** Kernel Header Options ***
Kernel Headers (Linux 4.15.x kernel headers)
```

设置完成后：

```
make
```

编译过程中可能会出现几个错误，缺少什么软件包就apt-get。
不过我这里出现了一个奇怪的错误：

```
make[2]: g++: 命令未找到
```

但是我用:

```
sudo apt-get install g++
```

却显示g++已经最新版本，尝试查询:

```
g++ --version
```

却又提示未安装g++，最后直接:

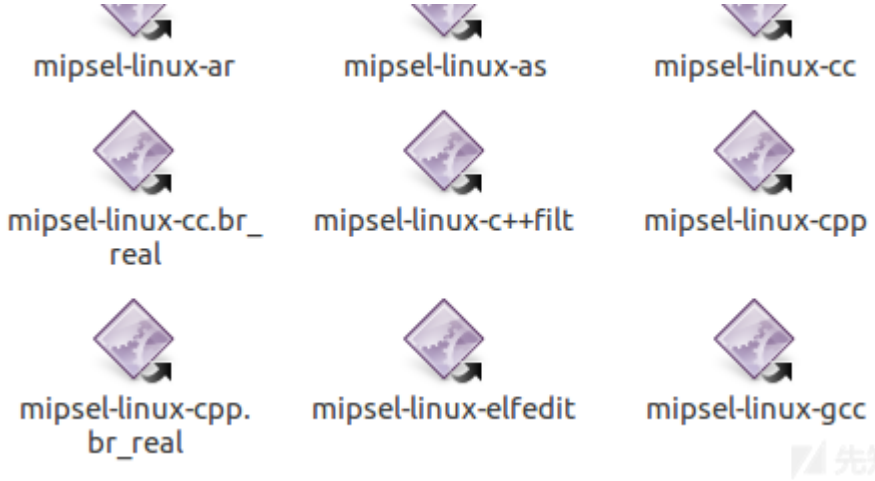
```
sudo apt-get remove g++

sudo apt-get install g++
```

最后编译不报错:

```
make
```

等待漫长的编译过程：

。。。。。。
。。。。。。
完成后在buildroot-2018.02.4/下生成了一个新的文件夹output/
在/buildroot-2018.02.4/output/host/usr/bin/下可以看见mips交叉编译工具成功生成：



这里使用的是little endian(小端)编译生成的就是mipsel，如果一开始make menuconfig设置的是big endian(大端)编译生成的工具就是mips。
剩下最后一步，将mipsel-*-*的路径放到环境变量中：
环境变量设置有三种方法（不过实际上在我的配置环境（ubuntu16.04-x64）中只有第三种方法生效，直接export不用说，但是修改bashrc在我root之后当前用户添加的环

- 设置当前控制台内：

  ```
  export PATH="$PATH:/home/kr0net/buildroot-2018.02.4/output/host/usr/bin/"
  ```

- 设置当前用户：

  ```
  gedit ~/.bashrc
  ```

  然后在里面添加：

  ```
  export PATH="$PATH:/home/kr0net/buildroot-2018.02.4/output/host/usr/bin/"
  ```

  生效：

  ```
  source /.bashrc
  ```

- 设置所有用户：

  ```
  gedit /etc/profile
  ```

  同样在里面加入：

  ```
  export PATH="$PATH:/home/kr0net/buildroot-2018.02.4/output/host/usr/bin/"
  ```

  生效：

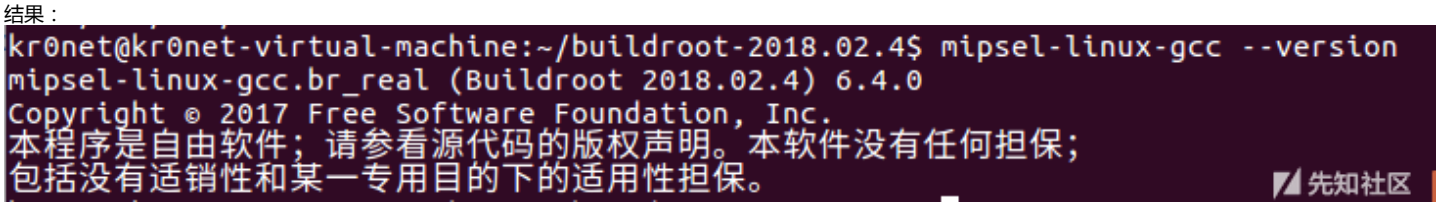
  ```
  source /etc/profile
  ```

  最后输入：

  ```
  echo $PATH
  ```

  查看环境变量是否生效：

结果：



上一篇：RWCTF2018 BookHub... 下一篇：基于反序列化的Oracle提权

1. 3 条回复

Tide 2018-07-31 09:52:56

```
xtensa-softmmu-config-devices.mak.d
root@ubuntu:/opt/qemu# sudo apt-get install gcc-arm-linux-genueabi
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
E: 无法定位软件包 gcc-arm-linux-genueabi
root@ubuntu:/opt/qemu# apt list | grep gcc-arm

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

gcc-arm-linux-androideabi/xenial 0.20130705.1-0ubuntu9 i386
gcc-arm-linux-gnueabi/xenial 4:5.3.1-1ubuntu1 i386
gcc-arm-linux-gnueabihf/xenial 4:5.3.1-1ubuntu1 i386
gcc-arm-none-eabi/xenial 15:4.9.3+svn231177-1 i386
gcc-arm-none-eabi-source/xenial 15:4.9.3+svn231177-1 all
root@ubuntu:/opt/qemu#
```


0 回复Ta

行之 2018-07-31 11:17:07

@Tide 感谢 已修正

0 回复Ta

一只猿 2018-08-01 10:55:44

sudo apt-get qemu 应该是 sudo apt-get install qemu

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板