Project-8，AES impl with ARM instruction

1，实验内容

实验目的是在ARM架构下运行AES，需要交叉编译

交叉编译是在一个平台上生成另一个平台上的可执行代码。在非ARM 架构服务器环境下搭建 ARM 的 GCC 编译环境，编译基于 ARM 架构的应用软件。交叉编译工具链是为了编译、链接、处理和调试跨平台体系结构的程序代码。除了体系结构相关的编译选项以外，其使用方法与 Linux 主机上的 GCC 相同。 搭建交叉编译环境，即安装、配置交叉编译工具链。在该环境下编译出 ARM 架构下 Linux 系统所需的操作系统、应用程序等，然后再上传到 ARM 服务器执行。

交叉编译器安装

1，安装标准C开发环境

Ubuntu使用sudo apt-get install build-essential

2， 在/usr/local 下建立名为ARM-toolchain的文件夹

mkdir /usr/local/ARM-toolchain

3，安装交叉编译器

```
cd /usr/local/ARM-toolchain
wget https://releases.linaro.org/components/toolchain/binaries/latest-5/aarch64-
linux-gnu/gcclinaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xzcd
/usr/local/ARM-toolchain
wget https://releases.linaro.org/components/toolchain/binaries/latest-5/aarch64-
linux-gnu/gcclinaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz
```

也可以从网页上下载后上传到 /usr/local/ARM-toolchain 目录下，交叉编译工具链的地址：gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz

解压安装包：

```
sudo apt update
sudo apt install tar xz-utils
```

进入文件所在的位置，解压

```
tar -xf gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz

PATH= /usr/local/ARM-toolchain/gcc-linaro-5.5.0-2017.10-x86_64_aarch64-
linuxgnu/bin:"${PATH}"
```

4， 配置环境变量

修改配置文件，在配置文件的最后一行加入路径配置：

```
vim /etc/bash.bashrc
#Add ARM toolschain path

PATH= /usr/local/ARM-toolchain/gcc-linaro-5.5.0-2017.10-x86_64_aarch64-
linuxgnu/bin:"${PATH}"
```

> 实测后发现路径配置使用上面语句没有权限，加上sudo后可以进入文件但没办法修改内容，使用sudo nano /etc/bash.bashrc可以进入文件并修改。

5, 环境变量生效与测试

```
source /etc/bash.bashrc

aarch64-linux-gnu-gcc -v
```

执行上面的命令，显示 arm-linux-gnueabi-gcc -v信息和版本

执行后显示如下：

```
Using built-in specs.
COLLECT_GCC=aarch64-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc-cross/aarch64-linux-gnu/9/lto-wrapper
Target: ...（略）
Thread model: posix
gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.1)
```

在ARM架构下编译

使用如下语句，交叉编译某个c文件，先拿一个hello.c文件试一下：

/usr/local/ARM-toolchain/gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc -o Hello Hello.c

使用**file** 语句查看生成的可执行文件是什么架构下的：

输出结果：

```
/home/linux-nzy/Desktop/Hello: ELF 64-bit LSB executable, ARM aarch64, version 1
(SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux
3.7.0, BuildID[sha1]=ab20ba9d0ce88a5f64c7dfbce6efac81a58b3613, with debug_info,
not stripped
```

/home/linux-nzy/Desktop/Hello: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]=ab20ba9d0ce88a5f64c7dfbce6efac81a58b3613, with debug_info, not stripped

可以看到确实是在ARM架构下

由此，我们已经在x86架构下使用交叉编译器编译出了ARM的可执行文件。

并没有执行ARM下的可执行文件。