实践项目 P3：交叉编译和跨平台应用仿真

布置周：11 月 20 日
提交周：12 月 18 日 23:59

目标：通过本项目的练习，希望同学们掌握交叉编译工具链及 Qemu 仿真器的安装方法，并且可以在 Host 机器上开发与 Host 不同 ISA 的 Target Platform(目标平台)上的简单应用。对于开发出来的目标平台上的简单应用，掌握在 Host 机器上通过仿真器来运行跨平台应用的方法，同时掌握对被仿真应用的性能比较及简单调试方法。

项目要求：在 Ubuntu/X64 上分别用 gcc 和 clang 去交叉编译一个 C 程序，生成不同版本的 aarch64 (64-bit ARM ISA) 可执行文件，在 Ubuntu/X64 上用 aarch64 的 Qemu 仿真器去分别运行不同编译器交叉编译生成的 aarch64 可执行文件，进行性能比较，并进一步掌握在 Ubuntu/X64 上用 gdb 调试 aarch64 可执行文件的方法。

【对于使用 Mac(M1/M2)机器的同学,这个实践项目可以反过来做,即在 M1/M2 上安装 X86-64 的 Qemu 仿真器，用 gcc 和 clang 去交叉编译一个 C 程序，生成不同版本的 X86-64 可执行文件，并进行相应的仿真运行、性能比较与调试】

具体步骤：（项目报告需要包括以下步骤及说明步骤完成情况的相关截图）

1. 在 Ubuntu 上安装能够运行 aarch64 (64-bit ARM ISA)应用的 Qemu 虚拟机（qemu-aarch64）

【安装步骤】:
【安装成功的验证】:

2. 安装 aarch64 的 GCC 工具链(gcc-10-aarch64-linux-gnu)

【安装步骤】:
【安装成功的验证】:

3. 用 aarch64 的 GCC 工具链交叉编译 loop.c (-O2)，生成可执行文件 loop.aarch64.gcc，并用 qemu-aarch64 运行 loop.aarch64.gcc

注意：运行 loop.aarch64.gcc 时需要带一个大于 0 且小于等于 100 的参数
【编译命令行】:
【qemu 运行的命令行】:
【qemu 运行的结果截图】:

4. 用 clang 交叉编译 loop.c(-O2)，生成可执行文件 loop.aarch64.clang，并用 qemu-aarch64 运行 loop.aarch64.clang

注意：运行 loop.aarch64.clang 时需要带一个大于 0 且小于等于 100 的参数

【编译命令行】:
【qemu 运行的命令行】:
【qemu 运行的结果截图】:

5. 用 qemu-aarch64 分别运行前面编译出来的 loop.aarch64.gcc 和 loop.aarch64.clang（分别用参数 5、15、30、60、90 进行测试），记下每次测试的执行时间并以图形方式呈现。

【测试结果图示化呈现】:

6. 用 host 机器上的 gcc 和 clang 分别编译(-O2)出 loop.x64.gcc 和 loop.x64.clang，并对这两个执行文件分别用参数 5、15、30、60、90 进行测试，记下每次测试的执行时间并以图形方式呈现，进而与前一步 qemu 仿真测试的结果进行比较。

【编译命令行】:
【测试结果图示化呈现】:

7. 安装支持多 ISA 的 gdb 调试器（gdb-multiarch）

【安装步骤】:
【安装成功的验证】:

8. 用 gdb-multiarch 结合 qemu-aarch64 对 loop.aarch64.gcc.debug 进行源码级调试

【编译生成带调试信息的 loop.aarch64.gcc.debug】:
【调试方法简介】:
【调试过程截屏】:

附录(仅供参考)：最终的工作目录截图

```
bhuang@LAPTOP-BHUANG:~/Courses/project1$ ls -al
total 100
drwxr-xr-x  2 bhuang bhuang  4096 Aug 26 16:28 .
drwxr-xr-x 15 bhuang bhuang  4096 Aug 26 13:54 ..
-rwxr-xr-x  1 bhuang bhuang  3164 Aug 26 14:00 fasttime.h
-rwxr-xr-x  1 bhuang bhuang  9616 Aug 26 15:30 loop.aarch64.clang
-rwxr-xr-x  1 bhuang bhuang 13840 Aug 26 14:11 loop.aarch64.gcc
-rwxr-xr-x  1 bhuang bhuang 16280 Aug 26 15:58 loop.aarch64.gcc.debug
-rwxr-xr-x  1 bhuang bhuang  3235 Aug 26 14:05 loop.c
-rwxr-xr-x  1 bhuang bhuang 16728 Aug 26 16:28 loop.x64.clang
-rwxr-xr-x  1 bhuang bhuang 17072 Aug 26 16:28 loop.x64.gcc
bhuang@LAPTOP-BHUANG:~/Courses/project1$ file loop.x64.gcc
loop.x64.gcc: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, inte
rpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=7193bc7881f8ab4cabf039c9c4caba776043c676, f
or GNU/Linux 3.2.0, not stripped
bhuang@LAPTOP-BHUANG:~/Courses/project1$ file loop.x64.clang
loop.x64.clang: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, inter
preter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=e991dcd848cb9572a2e323a152d719aaa29af609, fo
r GNU/Linux 3.2.0, not stripped
bhuang@LAPTOP-BHUANG:~/Courses/project1$ file loop.aarch64.gcc
loop.aarch64.gcc: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=b5457b14741c89cd1ca547e778caa890552
874b4, for GNU/Linux 3.7.0, not stripped
bhuang@LAPTOP-BHUANG:~/Courses/project1$ file loop.aarch64.clang
loop.aarch64.clang: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically link
ed, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=f699d77941c2169d61e62a00c8e20ad64c0e
b78a, for GNU/Linux 3.7.0, not stripped
bhuang@LAPTOP-BHUANG:~/Courses/project1$ file loop.aarch64.gcc.debug
loop.aarch64.gcc.debug: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamical
ly linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=4c0e164a2114622c93d416c976108
3dc0df02760, for GNU/Linux 3.7.0, with debug_info, not stripped
bhuang@LAPTOP-BHUANG:~/Courses/project1$
```