# Lab 0: GDB&QEMU调试64位RISC-V LINUX

张志心 3210106357

## 1 实验目的

- 使用交叉编译工具，完成 Linux 内核代码编译；
- 使用 `qemu` 运行内核；
- 熟悉 `GDB` 和 `QEMU` 联合调试。

## 2 实验环境

Mac with Apple Silicon

## 3 实验步骤

### 3.1 搭建实验环境

下载 Docker Desktop ，在终端输入

```
docker pull ubuntu:22.04
```

```
(base) ➜  Lab0 git:(main) ✗ docker pull ubuntu:22.04
```

获取 ubuntu22.04 的镜像。然后使用

```
docker run --name nanlin --hostname Lisore ubuntu:22.04 /bin/bash
```

```
(base) ➜  Lab0 git:(main) ✗ docker run --name nanlin --hostname Lisore ubuntu:22.04 /bin/bash
```

启动一个 ubuntu22.04 的容器，并且设置容器名为 nanlin，主机名为 Lisore。

然后使用

```
docker start nanlin
docker attach nanlin
```

```
(base) ➜  Lab0 git:(main) ✗ docker start nanlin
nanlin
(base) ➜  Lab0 git:(main) ✗ docker attach nanlin
root@Lisore:/#
```

启动容器。首先安装编译内核所需要的交叉编译工具链和用于构建程序的软件包

```
sudo apt install  gcc-riscv64-linux-gnu
sudo apt install  autoconf automake autotools-dev curl libmpc-dev libmpfr-dev libgmp-
dev \
                  gawk build-essential bison flex texinfo gperf libtool patchutils
bc \
                  zlib1g-dev libexpat-dev git
```

```
root@Lisore:/# sudo apt install gcc-riscv64-linux-gnu
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc-riscv64-linux-gnu is already the newest version (4:11.2.0--1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@Lisore:/# sudo apt install  autoconf automake autotools-dev curl libmpc-dev libmpfr-dev libgmp-dev \
                  gawk build-essential bison flex texinfo gperf libtool patchutils bc \
                  zlib1g-dev libexpat-dev git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libexpat1-dev' instead of 'libexpat-dev'
autoconf is already the newest version (2.71-2).
automake is already the newest version (1:1.16.5-1.3).
autotools-dev is already the newest version (20220109.1).
bc is already the newest version (1.07.1-3build1).
bison is already the newest version (2:3.8.2+dfsg-1build1).
build-essential is already the newest version (12.9ubuntu3).
flex is already the newest version (2.6.4-8build2).
libgmp-dev is already the newest version (2:6.2.1+dfsg-3ubuntu1).
libmpc-dev is already the newest version (1.2.1-2build1).
libmpfr-dev is already the newest version (4.1.0-3build3).
libtool is already the newest version (2.4.6-15build2).
patchutils is already the newest version (0.4.2-1build2).
gperf is already the newest version (3.1-1build1).
texinfo is already the newest version (6.8-4build1).
curl is already the newest version (7.81.0-1ubuntu1.13).
gawk is already the newest version (1:5.1.0-1ubuntu0.1).
git is already the newest version (1:2.34.1-1ubuntu1.10).
libexpat1-dev is already the newest version (2.4.7-1ubuntu0.2).
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu9.2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

本人已经装过相关软件包，故只作演示，下同。

安装用于启动 **RISCV64** 平台上的内核的模拟器 `qeme` 和调试工具 `gdb`。

```
sudo apt install qemu-system-misc
sudo apt install gdb-multiarch
```

```
root@Lisore:/# sudo apt install qemu-system-misc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
qemu-system-misc is already the newest version (1:6.2+dfsg-2ubuntu6.13).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

```
root@Lisore:/# sudo apt install gdb-multiarch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gdb-multiarch is already the newest version (12.1-0ubuntu1~22.04).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

在此之前，还需要安装 `sudo`。

```
apt-get update
apt-get install sudo
```

```
root@Lisore:/# apt-get update
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Fetched 229 kB in 4s (61.2 kB/s)
Reading package lists... Done
root@Lisore:/# apt-get install sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sudo is already the newest version (1.9.9-1ubuntu2.4).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

注：若要退出容器则输入

```
exit
```

```
root@Lisore:/# exit
exit
(base) ➜  Lab0 git:(main) ✗
```

## 3.2 获取 Linux 源码和已经编译好的文件系统

从 https://www.kernel.org 下载最新的 Linux 源码至宿主机。

linux-6.6-rc2.tar.gz　2023年9月20日 11:11　229.3 MB　gzip压缩归档

使用 `docker cp` 将源代码复制到容器中。

```
docker cp linux-6.6-rc2.tar.gz nanlin:/root/
```

```
(base) ➜  OS docker cp linux-6.6-rc2.tar.gz nanlin:/root/
                        Successfully copied 229MB to nanlin:/root/
```

然后进入容器，解压该文件。

```
tar -zxvf linux-6.6-rc2.tar.gz
ls
```

```
root@Lisore:~# tar -zxf linux-6.6-rc2.tar.gz
root@Lisore:~# ls
linux-6.6-rc2  linux-6.6-rc2.tar.gz  os  os23fall-stu
```

使用 `git` 克隆实验仓库，获取根文件系统的镜像。

```
git clone https://github.com/ZJU-SEC/os23fall-stu.git
```

```
root@Lisore:~/os# rm -rf linux-6.6-rc2/
root@Lisore:~/os# git clone https://github.com/ZJU-SEC/os23fall-stu.git
Cloning into 'os23fall-stu'...
remote: Enumerating objects: 162, done.
remote: Counting objects: 100% (162/162), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 162 (delta 31), reused 147 (delta 24), pack-reused 0
Receiving objects: 100% (162/162), 2.54 MiB | 296.00 KiB/s, done.
Resolving deltas: 100% (31/31), done.
```

已经构建完成的根文件系统的镜像路径为 `os23fall-stu/src/lab0/rootf.img`。

```
root@Lisore:~/os23fall-stu/src/lab0# ls
rootfs.img
```

## 3.3 编译 Linux 内核

```
cd /root/os # linux-6.6-rc2 解压后的路径
make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- defconfig     # 使用默认配置
make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- -j$(nproc)    # 编译
```

```
root@Lisore:~/os# make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- defconfig
*** Default configuration is based on 'defconfig'
#
# No change to .config
#
root@Lisore:~/os# make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- -j$(nproc)
  UPD     include/generated/compile.h
  CALL    scripts/checksyscalls.sh
  CC      arch/riscv/kernel/vdso/vgettimeofday.o
  VDSOLD  arch/riscv/kernel/vdso/vdso.so.dbg
  VDSOSYM include/generated/vdso-offsets.h
```

## 3.4 使用 QEMU 运行内核

```
qemu-system-riscv64 -nographic -machine virt -kernel /root/os/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
    -bios default -drive file=/root/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0
```

运行后终端截图如下：

```
root@Lisore:~/os# qemu-system-riscv64 -nographic -machine virt -kernel /root/os/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
    -bios default -drive file=/root/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0

OpenSBI v0.9
   ____                    ____ ___ ____
  / __ \                  / ___|  _ \  _|
 | |  | |_ __   ___ _ __ | (___ | |_) || |
 | |  | | '_ \ / _ \ '_ \ \___ \|  _ < | |
 | |__| | |_) |  __/ | | |____) | |_) || |_
  \____/| .__/ \___|_| |_|_____/|___/_____|
        | |
        |_|

Platform Name             : riscv-virtio,qemu
Platform Features         : timer,mfdeleg
Platform HART Count       : 1
Firmware Base             : 0x80000000
Firmware Size             : 100 KB
Runtime SBI Version       : 0.2

Domain0 Name              : root
Domain0 Boot HART         : 0
Domain0 HARTs             : 0*
Domain0 Region00          : 0x0000000080000000-0x0000000080001fff ()
Domain0 Region01          : 0x0000000000000000-0xffffffffffffffff (R,W,X)
Domain0 Next Address      : 0x0000000080200000
Domain0 Next Arg1         : 0x0000000087000000
Domain0 Next Mode         : S-mode
Domain0 SysReset          : yes

Boot HART ID              : 0
Boot HART Domain          : root
Boot HART ISA             : rv64imafdcsu
Boot HART Features        : scounteren,mcounteren,time
Boot HART PMP Count       : 16
Boot HART PMP Granularity : 4
Boot HART PMP Address Bits: 54
Boot HART MHPM Count      : 0
Boot HART MHPM Count      : 0
Boot HART MIDELEG         : 0x0000000000000222
Boot HART MEDELEG         : 0x00000000000b109
```

注：退出 `QEMU` 的方法为：使用 `Ctrl+A`，松开后再按下 `X` 键即可退出 `QEMU`。

## 3.5 使用 GDB 对内核进行调试

开启两个终端，一个使用 `QEMU` 启动 **Linux**，另一个使用 `GDB` 与 `QEMU` 远程通信（使用 `tcp:1234` 端口）进行调试。

```
# Terminal 1
qemu-system-riscv64 -nographic -machine virt -kernel /root/os/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
    -bios default -drive file=/root/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0
-S -s
# Terminal 2
docker exec -it nanlin /bin/bash # 打开第二个终端并进入容器
gdb-multiarch /root/os/vmlinux
```

```
root@Lisore:~/os# qemu-system-riscv64 -nographic -machine virt -kernel /root/os/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
    -bios default -drive file=/root/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s
```

```
root@Lisore:/# gdb-multiarch /root/os/vmlinux
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "aarch64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from /root/os/vmlinux...
(No debugging symbols found in /root/os/vmlinux)
```

1. 连接 `qemu`

```
(gdb) target remote:1234
Remote debugging using :1234
0x0000000000001000 in ?? ()
```

2. 设置断点

```
(gdb) b start_kernel
Breakpoint 1 at 0xffffffff80a006ac
```

3. 继续执行

```
(gdb) continue
Continuing.

Breakpoint 1, 0xffffffff80a006ac in start_kernel ()
```

此时 `qemu` 运行到 `start_kernel()` 函数位置。

```
root@Lisore:~/os# qemu-system-riscv64 -nographic -machine virt -kernel /root/os/arch/riscv/boot/Image
-device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0"     -bios default -drive file
=/root/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s

OpenSBI v0.9
   ____                    ____ ___ ____
  / __ \                  / ___|  _ \_  _|
 | |  | |_ __   ___ _ __ | (___ | |_) || |
 | |  | | '_ \ / _ \ '_ \ \___ \|  _ < | |
 | |__| | |_) |  __/ | | |____) | |_) || |_
  \____/| .__/ \___|_| |_|_____/|___/_____|
        | |
        |_|

Platform Name             : riscv-virtio,qemu
Platform Features         : timer,mfdeleg
Platform HART Count       : 1
Firmware Base             : 0x80000000
Firmware Size             : 100 KB
Runtime SBI Version       : 0.2

Domain0 Name              : root
Domain0 Boot HART         : 0
Domain0 HARTs             : 0*
Domain0 Region00          : 0x0000000080000000-0x000000008001ffff ()
Domain0 Region01          : 0x0000000000000000-0xffffffffffffffff (R,W,X)
Domain0 Next Address      : 0x0000000080200000
Domain0 Next Arg1         : 0x0000000087000000
Domain0 Next Mode         : S-mode
Domain0 SysReset          : yes

Boot HART ID              : 0
Boot HART Domain          : root
Boot HART ISA             : rv64imafdcsu
Boot HART Features        : scounteren,mcounteren,time
Boot HART PMP Count       : 16
Boot HART PMP Granularity : 4
Boot HART PMP Address Bits: 54
Boot HART MHPM Count      : 0
Boot HART MHPM Count      : 0
Boot HART MIDELEG         : 0x0000000000000222
Boot HART MEDELEG         : 0x000000000000b109
```

4. 退出 `gdb`

```
(gdb) quit
A debugging session is active.

        Inferior 1 [process 1] will be detached.

Quit anyway? (y or n) y
Detaching from program: /root/os/vmlinux, process 1
Ending remote debugging.
[Inferior 1 (process 1) detached]
```

此时 `qemu` 运行到结束位置。

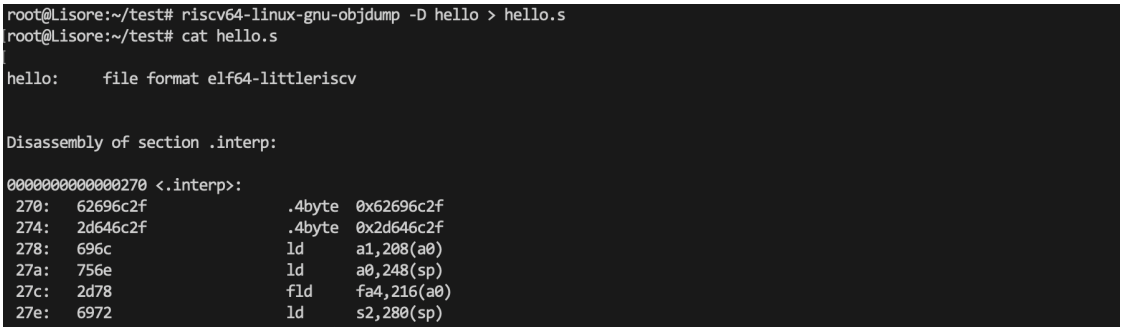
```
[    0.318313] clk: Disabling unused clocks
[    0.339623] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b21172fc1 ro with ordered data
 mode. Quota mode: disabled.
[    0.340655] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.
[    0.344601] devtmpfs: mounted
[    0.369521] Freeing unused kernel image (initmem) memory: 2204K
[    0.370184] Run /sbin/init as init process

Please press Enter to activate this console.
```

# 4 思考题

1. 使用 `riscv64-linux-gnu-gcc` 编译单个 `.c` 文件；

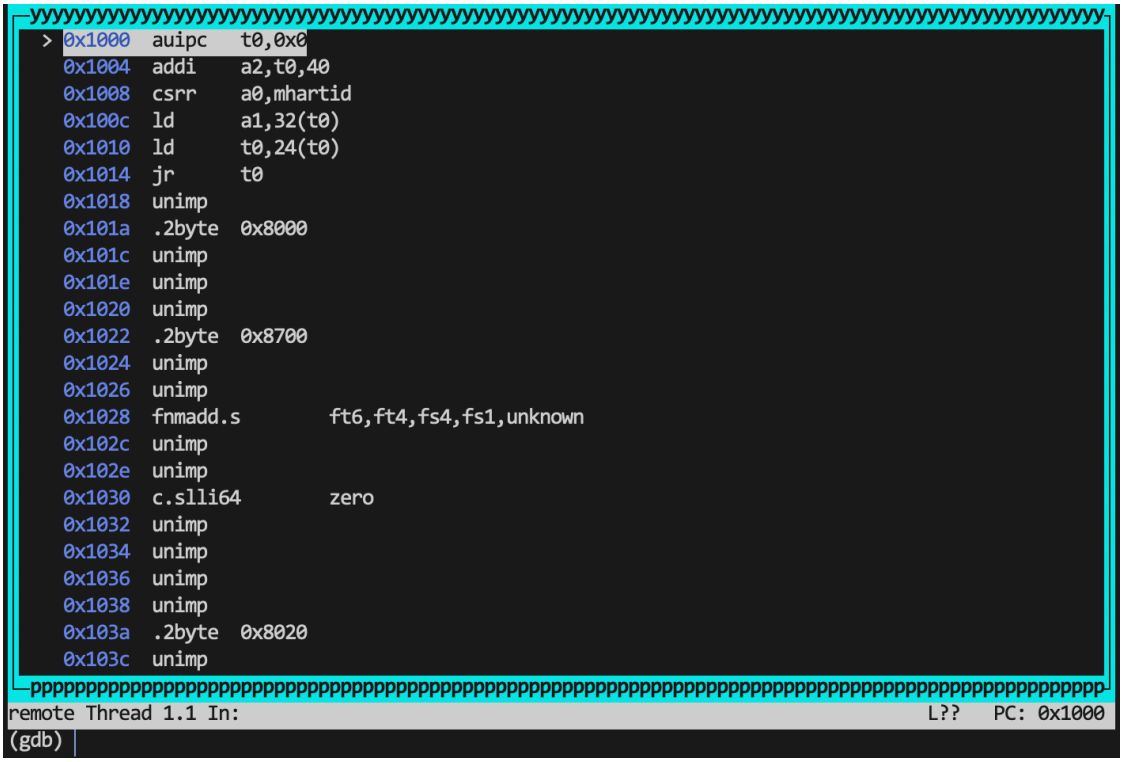
```
vim hello.c
```


2. 使用 `riscv64-linux-gnu-objdump` 反汇编 1 中得到的编译产物；




3. 调试 Linux 时：

   a. 在 GDB 中查看汇编代码




b. 在 0x80000000 处下断点

```
(gdb) b * 0x80000000
Breakpoint 1 at 0x80000000
```

c. 查看所有已下的断点

```
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000080000000
```

d. 在 0x80200000 处下断点

```
(gdb) break * 0x80200000
Breakpoint 2 at 0x80200000
```

e. 清除 0x80000000 处的断点

```
(gdb) del 1
(gdb) info break
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x0000000080200000
```

f. 继续运行直到触发 0x80200000 处的断点

```
(gdb) c
Continuing.

Breakpoint 2, 0x0000000080200000 in ?? ()
```

g. 单步调试一次

```
(gdb) next
Cannot find bounds of current function
(gdb) si
0x0000000080200002 in ?? ()
(gdb) s
Cannot find bounds of current function
```

**h.** 退出 **QEMU**

```
(gdb) q
A debugging session is active.

        Inferior 1 [process 1] will be detached.

Quit anyway? (y or n) y
```

4. 使用 `make` 工具清除 **Linux** 的构建产物；

```
root@Lisore:/# cd /root/os
root@Lisore:~/os# make clean
  CLEAN   drivers/firmware/efi/libstub
  CLEAN   drivers/gpu/drm/radeon
  CLEAN   drivers/scsi
  CLEAN   drivers/tty/vt
  CLEAN   init
  CLEAN   kernel
  CLEAN   lib/raid6
  CLEAN   lib
  CLEAN   security/apparmor
  CLEAN   security/selinux
  CLEAN   usr
  CLEAN   .
  CLEAN   modules.builtin modules.builtin.modinfo .vmlinux.export.c
```

5. `vmlinux` 和 `Image` 的关系和区别是什么？

   `vmlinux` 是链接后生成的 `linux elf` 文件，不是镜像格式，`Image` 是对 `vmlinux` 的 `elf` 格式进行解析后生成的镜像文件，可直接引导系统启动。