

# 浙江大学

## 本科实验报告

课程名称：操作系统

姓 名：黄文杰

学 院：计算机科学与技术学院

系：计算机科学与技术系

专 业：软件工程

学 号：3210103379

指导教师：夏莹杰

2023 年 9 月 25 日

# 浙江大学操作系统实验报告

实验名称: GDB & QEMU 调试 64 位 RISC-V LINUX

电子邮件地址: 2438452556@qq.com 手机: 15167970568

实验地点: 曹光彪西 503 实验日期: 2023 年 9 月 25 日

## 一、实验目的和要求

实验目的:

1. 使用交叉编译工具, 完成 Linux 内核代码编译
2. 使用 QEMU 运行内核
3. 熟悉 GDB 和 QEMU 联合调试

实验要求:

1. 编译内核, 使用 QEMU 启动后, 远程连接 GDB 进行调试, 并尝试使用 GDB 的各项命令 (如 ``backtrace``, ``finish``, ``frame``, ``info``, ``break``, ``display``, ``next``, ``layout`` 等)。
2. 在学在浙大中提交 pdf 格式的实验报告, 记录实验过程并截图 (4.1-4.4), 对每一步的命令以及结果进行必要的解释, 记录遇到的问题和心得体会。

## 二、实验过程

1. 搭建实验环境环境

## 1.1 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

```
hwj@hwj-virtual-machine:~$ sudo apt install gcc-riscv64-linux-gnu
[sudo] hwj 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
将会同时安装下列软件：
  binutils-common binutils-riscv64-linux-gnu cpp-11-riscv64-linux-gnu
  cpp-riscv64-linux-gnu gcc-11-cross-base-ports gcc-11-riscv64-linux-gnu
  gcc-11-riscv64-linux-gnu-base gcc-12-cross-base-ports libasan6-riscv64-cross
  libatomic1-riscv64-cross libc6-dev-riscv64-cross libc6-riscv64-cross
  libcc1-0 libgcc-11-dev-riscv64-cross libgcc-s1-riscv64-cross
  libgomp1-riscv64-cross linux-libc-dev-riscv64-cross
建议安装：
  binutils-doc gcc-11-locales cpp-doc gcc-11-doc make manpages-dev autoconf
  automake libtool flex bison gdb-riscv64-linux-gnu gcc-doc
下列【新】软件包将被安装：
  binutils-common binutils-riscv64-linux-gnu cpp-11-riscv64-linux-gnu
  cpp-riscv64-linux-gnu gcc-11-cross-base-ports gcc-11-riscv64-linux-gnu
  gcc-11-riscv64-linux-gnu-base gcc-12-cross-base-ports gcc-riscv64-linux-gnu
  libasan6-riscv64-cross libatomic1-riscv64-cross libc6-dev-riscv64-cross
  libc6-riscv64-cross libcc1-0 libgcc-11-dev-riscv64-cross
  libgcc-s1-riscv64-cross libgomp1-riscv64-cross linux-libc-dev-riscv64-cross
升级了 0 个软件包，新安装了 18 个软件包，要卸载 0 个软件包，有 56 个软件包未被升级。
```

```
hwj@hwj-virtual-machine:~$ 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
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
注意，选中 'libexpat1-dev' 而非 'libexpat-dev'
bc 已经是最新版 (1.07.1-3build1)。
bc 已设置为手动安装。
将会同时安装下列软件：
  binutils binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-11 gcc gcc-11
  git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools
  libc6 libc6-dbg libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl
  liberror-perl libfakeroot libfile-fcntllock-perl libfl-dev libfl2
  libgcc-11-dev libgmpxx4ldbl libitm1 liblsan0 libltdl-dev libnsl-dev
  libquadmath0 libsigsegv2 libstdc++-11-dev libtext-unidecode-perl
  libtirpc-dev libtsan0 libubsan1 libxml-libxml-perl
  libxml-namespacesupport-perl libxml-sax-base-perl libxml-sax-expat-perl
  libxml-sax-perl linux-libc-dev lto-disabled-list m4 make manpages-dev
  rpcsvc-proto tex-common
建议安装：
  autoconf-archive gnu-standards autoconf-doc gettext binutils-doc bison-doc
  debian-keyring flex-doc g++-multilib g++-11-multilib gcc-11-doc gawk-doc
  gcc-multilib gcc-doc gcc-11-multilib gcc-11-locales git-daemon-run
  | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn glibc-doc bzip2 gmp-doc libgmp10-doc libtool-doc
  libmpfr-doc libstdc++-11-doc gfortran | fortran95-compiler gcj-jdk
  libxml-sax-expatxs-perl m4-doc make-doc debhelper texlive-base
  texlive-latex-base texlive-plain-generic texlive-fonts-recommended
```

如上图所示，执行完如上两个操作之后就将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下载到了虚拟机上。

## 1.2 下载用于启动 riscv64 平台上的内核的模拟器 qemu

```
hwj@hwj-virtual-machine:~$ sudo apt install qemu-system-misc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
将会同时安装下列软件：
  ibverbs-providers ipxe-qemu libaio1 libcacard0 libdaxctl1 libdecor-0-0
  libdecor-0-plugin-1-cairo libfdt1 libgfbapi0 libgfrpc0 libgfxdr0
  libglusterfs0 libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1 librados2
  librbd1 librdmacm1 libsdl2-2.0-0 libslirp0 libspice-server1 liburing2
  libusbredirparser1 libvirglrenderer1 qemu-block-extra qemu-system-common
  qemu-system-data qemu-system-gui qemu-utils seabios
建议安装：
  gstreamer1.0-libav gstreamer1.0-plugins-ugly samba vde2 debootstrap
下列【新】软件包将被安装：
  ibverbs-providers ipxe-qemu libaio1 libcacard0 libdaxctl1 libdecor-0-0
  libdecor-0-plugin-1-cairo libfdt1 libgfbapi0 libgfrpc0 libgfxdr0
  libglusterfs0 libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1 librados2
  librbd1 librdmacm1 libsdl2-2.0-0 libslirp0 libspice-server1 liburing2
  libusbredirparser1 libvirglrenderer1 qemu-block-extra qemu-system-common
  qemu-system-data qemu-system-gui qemu-system-misc qemu-utils seabios
升级了 0 个软件包，新安装了 33 个软件包，要卸载 0 个软件包，有 54 个软件包未被升级。
需要下载 57.8 MB 的归档。
```

执行完上面的指令就把 qemu 下载到了虚拟机上

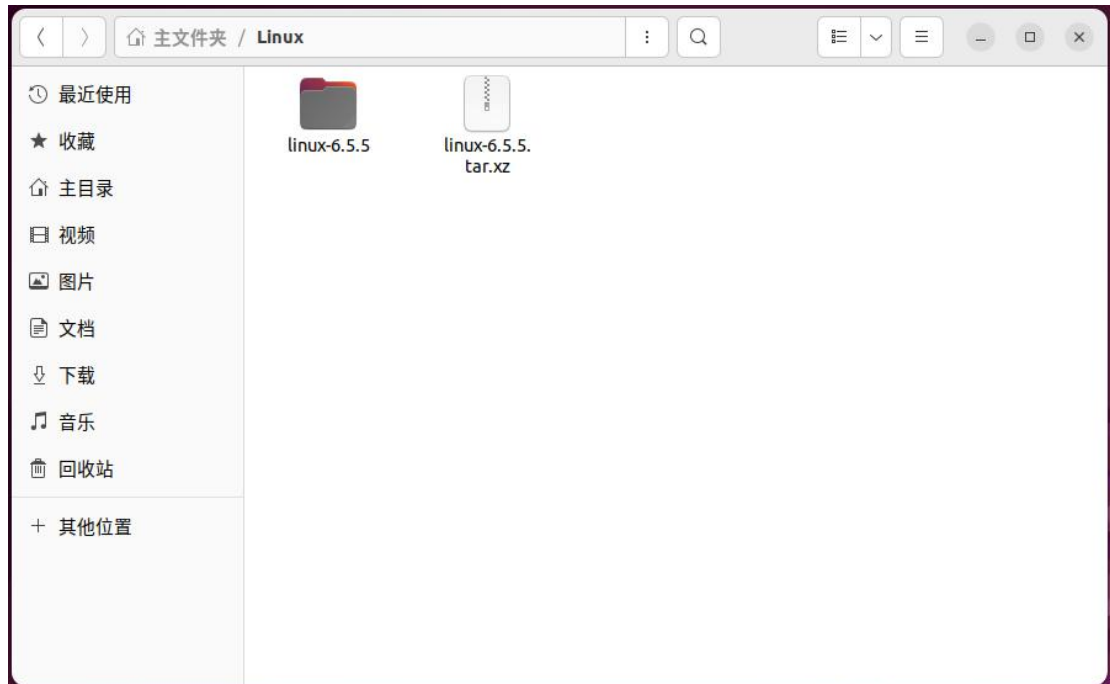
## 1.3 下载用于调试的 gdb

```
hwj@hwj-virtual-machine:~$ sudo apt install gdb-multiarch
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
下列【新】软件包将被安装：
  gdb-multiarch
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 54 个软件包未被升级。
需要下载 4,589 kB 的归档。
解压缩后会消耗 18.2 MB 的额外空间。
获取:1 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 gdb-multiarch amd64 12.1-0ubuntu1~22.04 [4,589 kB]
已下载 4,589 kB，耗时 1秒 (3,775 kB/s)
正在选中未选择的软件包 gdb-multiarch。
(正在读取数据库 ... 系统当前共安装有 222488 个文件和目录。)
准备解压 .../gdb-multiarch_12.1-0ubuntu1~22.04_amd64.deb ...
正在解压 gdb-multiarch (12.1-0ubuntu1~22.04) ...
正在设置 gdb-multiarch (12.1-0ubuntu1~22.04) ...
```



## 2. 获取 Linux 源码和已经编译好的文件系统

### 2.1 下载最新的 Linux 源码



如上图所示，最新的 Linux 源码已经下载到了虚拟机本地目录下（下载过程略）

### 2.2 利用 git clone 实验仓库，获取根文件系统的镜像

```
hwj@hwj-virtual-machine:~$ git clone https://gitee.com/zju_xiayingjie/os23fall-stu.git
正克隆到 'os23fall-stu'...
remote: Enumerating objects: 128, done.
remote: Counting objects: 100% (128/128), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 128 (delta 24), reused 98 (delta 6), pack-reused 0
接收对象中: 100% (128/128), 1.93 MiB | 1.12 MiB/s, 完成.
处理 delta 中: 100% (24/24), 完成.
```

```
hwj@hwj-virtual-machine:~$ cd os23fall-stu/src/lab0
hwj@hwj-virtual-machine:~/os23fall-stu/src/lab0$ ls
rootfs.img
rootfs.img
rootfs.img: 未找到命令
hwj@hwj-virtual-machine:~/os23fall-stu/src/lab0$ ls rootfs.img
rootfs.img
```

如上图所示，已经成功 clone 实验仓库至虚拟机本地，并且找到了/src/lab0/目录下的根文件系统的镜像（rootfs.img）

### 3. 编译 linux 内核

3.1 切换到 Linux 目录，然后利用 make 工具进行编译(此步骤为设置默认配置进行编译)

```
hwj@hwj-virtual-machine:~$ cd /home/hwj/Linux/linux-6.5.5
hwj@hwj-virtual-machine:~/Linux/linux-6.5.5$ make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- defconfig
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/kconfig/conf.o
HOSTCC  scripts/kconfig/confdata.o
HOSTCC  scripts/kconfig/expr.o
LEX      scripts/kconfig/lexer.lex.c
YACC     scripts/kconfig/parser.tab.[ch]
HOSTCC  scripts/kconfig/lexer.lex.o
HOSTCC  scripts/kconfig/menu.o
HOSTCC  scripts/kconfig/parser.tab.o
HOSTCC  scripts/kconfig/preprocess.o
HOSTCC  scripts/kconfig/symbol.o
HOSTCC  scripts/kconfig/util.o
HOSTLD  scripts/kconfig/conf
*** Default configuration is based on 'defconfig'
#
# configuration written to .config
#
```

3.2 指定系统用 4 个线程进行编译（编译时间有点长）

```
hwj@hwj-virtual-machine:~/Linux/linux-6.5.5$ make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- -j4$(nproc)
WRAP    arch/riscv/include/generated/uapi/asm/errno.h
WRAP    arch/riscv/include/generated/uapi/asm/fcntl.h
WRAP    arch/riscv/include/generated/uapi/asm/ioctl.h
WRAP    arch/riscv/include/generated/uapi/asm/ioctls.h
WRAP    arch/riscv/include/generated/uapi/asm/lpcbuf.h
WRAP    arch/riscv/include/generated/uapi/asm/mman.h
WRAP    arch/riscv/include/generated/uapi/asm/msgbuf.h
WRAP    arch/riscv/include/generated/uapi/asm/param.h
WRAP    arch/riscv/include/generated/uapi/asm/poll.h
WRAP    arch/riscv/include/generated/uapi/asm/posix_types.h
WRAP    arch/riscv/include/generated/uapi/asm/resource.h
HOSTCC  scripts/dtc/dtc.o
WRAP    arch/riscv/include/generated/uapi/asm/sembuf.h
WRAP    arch/riscv/include/generated/uapi/asm/shmbuf.h
HOSTCC  scripts/dtc/flattree.o
WRAP    arch/riscv/include/generated/uapi/asm/siginfo.h
WRAP    arch/riscv/include/generated/uapi/asm/signal.h
WRAP    arch/riscv/include/generated/uapi/asm/socket.h
HOSTCC  scripts/dtc/fstree.o
WRAP    arch/riscv/include/generated/uapi/asm/sockios.h
WRAP    arch/riscv/include/generated/uapi/asm/stat.h
WRAP    arch/riscv/include/generated/uapi/asm/statfs.h
HOSTCC  scripts/dtc/data.o
UPD     include/generated/uapi/linux/version.h
WRAP    arch/riscv/include/generated/uapi/asm/swab.h
HOSTCC  scripts/dtc/livetree.o
WRAP    arch/riscv/include/generated/uapi/asm/termbits.h
HOSTCC  scripts/dtc/treesource.o
WRAP    arch/riscv/include/generated/uapi/asm/termios.h
WRAP    arch/riscv/include/generated/uapi/asm/types.h
```

```

LD [M] net/ipv4/esp4.ko
LD [M] net/ipv4/udp_tunnel.ko
LD [M] net/xfrm/xfrm_algo.ko
LD [M] net/xfrm/xfrm_user.ko
LD [M] net/ipv6/netfilter/ip6_tables.ko
LD [M] net/ipv6/netfilter/ip6t_ipv6header.ko
LD [M] net/ipv6/netfilter/ip6t_REJECT.ko
LD [M] net/ipv6/netfilter/ip6table_filter.ko
LD [M] net/ipv6/netfilter/ip6table_mangle.ko
LD [M] net/ipv6/netfilter/nf_defrag_ipv6.ko
LD [M] net/ipv6/netfilter/nf_reject_ipv6.ko
LD [M] net/ipv6/ip6_udp_tunnel.ko
LD [M] net/8021q/8021q.ko
LD [M] net/bridge/bridge.ko
LD [M] net/bridge/br_netfilter.ko
LD [M] net/llc/llc.ko
NM .tmp_vmlinux.kallsyms1.syms
KSYMS .tmp_vmlinux.kallsyms1.S
AS .tmp_vmlinux.kallsyms1.S
LD .tmp_vmlinux.kallsyms2
NM .tmp_vmlinux.kallsyms2.syms
KSYMS .tmp_vmlinux.kallsyms2.S
AS .tmp_vmlinux.kallsyms2.S
LD vmlinux
NM System.map
SORTTAB vmlinux
OBJCOPY arch/riscv/boot/Image
GZIP arch/riscv/boot/Image.gz
Kernel: arch/riscv/boot/Image.gz is ready

```

## 4. 使用 QEMU 运行内核

```

hwj@hwj-virtual-machine:~/Linux/linux-6.5.5$ qemu-system-riscv64 -nographic -machine virt -kernel /home/hwj/Linux/linux-6.5.5/arch/riscv/boot/Image -device virtio-blk-device,drive=hdb0 -append "root=/dev/vda ro console=ttyS0" -bios default -drive file=/home/hwj/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hdb0

```

OpenSBI v0.9


```

      _ _ _ _ _      _ _ _ _ _
     /   \         /   \   \   \
    | | | | _ _ _ _ | | | | | | | | | | |
    | | | | \ \ / \ \ \ \ \ \ \ | |
    | | | | | | | | | | | | | | |
    \ \ / \ \ \ \ \ \ \ \ \ \ \
      | |      | |
      | |
      | |

```

Platform Name : riscv-virtio,qemu  
Platform Features : timer,nfdeleg  
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 : 0x0000000000000000-0x0000000000001ffff ()  
Domain0 Region01 : 0x0000000000000000-0xffffffffffffffff (R,W,X)  
Domain0 Next Address : 0x000000000200000  
Domain0 Next Arg1 : 0x0000000007000000  
Domain0 Next Mode : S-mode

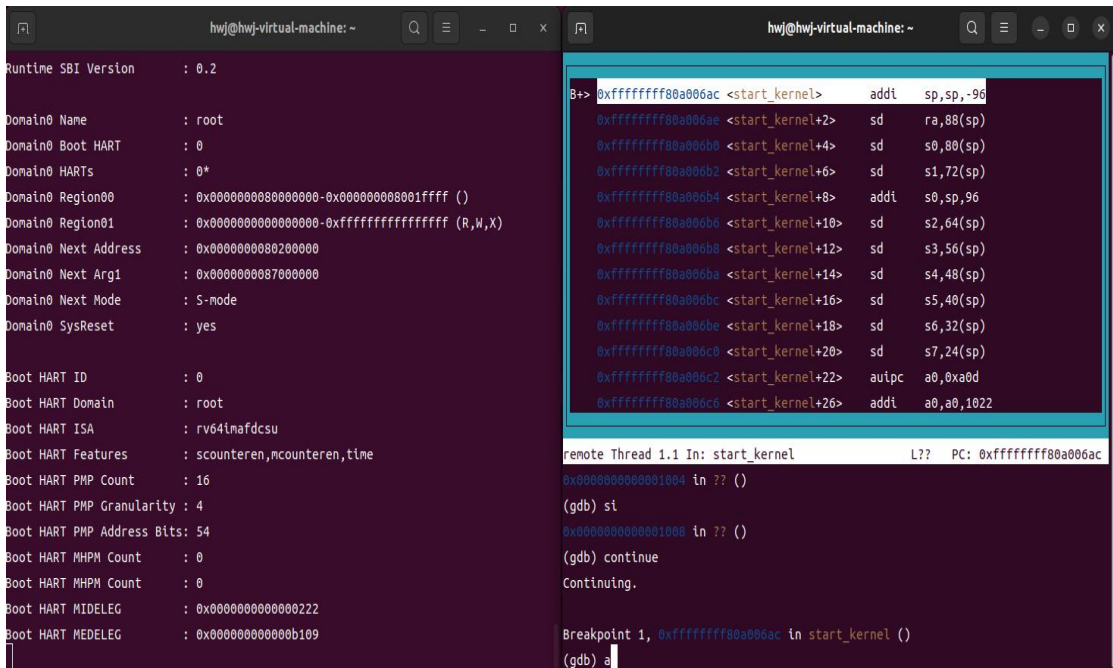








下图展示了利用 GDB 执行显示汇编代码、单步执行等操作。



The screenshot shows a GDB session with two panes. The left pane displays system boot information for a RISC-V virtual machine, including SBI Version (0.2), Domain Name (root), and various boot parameters. The right pane shows assembly code for the `start_kernel` function, with instructions like `addi sp,sp,-96` and `sd ra,88(sp)`. The GDB interface includes a command prompt and a list of breakpoints.

```
Runtime SBI Version : 0.2
Domain0 Name : root
Domain0 Boot HART : 0
Domain0 HARTs : 0*
Domain0 Region00 : 0x0000000000000000-0x0000000000001fff ( )
Domain0 Region01 : 0x0000000000000000-0xffffffffffff (R,W,X)
Domain0 Next Address : 0x00000000000020000
Domain0 Next Arg1 : 0x00000000000070000
Domain0 Next Mode : S-mode
Domain0 SysReset : yes

Boot HART ID : 0
Boot HART Domain : root
Boot HART ISA : rv64imafdcsv
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 : 0x0000000000000b109

B> 0xfffffffff80a006ac <start_kernel> addi sp,sp,-96
0xfffffffff80a006ae <start_kernel+2> sd ra,88(sp)
0xfffffffff80a006b0 <start_kernel+4> sd s0,88(sp)
0xfffffffff80a006b2 <start_kernel+6> sd s1,72(sp)
0xfffffffff80a006b4 <start_kernel+8> addi s0,sp,96
0xfffffffff80a006b6 <start_kernel+10> sd s2,64(sp)
0xfffffffff80a006b8 <start_kernel+12> sd s3,56(sp)
0xfffffffff80a006ba <start_kernel+14> sd s4,48(sp)
0xfffffffff80a006bc <start_kernel+16> sd s5,40(sp)
0xfffffffff80a006be <start_kernel+18> sd s6,32(sp)
0xfffffffff80a006c0 <start_kernel+20> sd s7,24(sp)
0xfffffffff80a006c2 <start_kernel+22> auipc a0,0xa0d
0xfffffffff80a006c6 <start_kernel+26> addi a0,a0,1022

remote Thread 1.1 In: start_kernel L?? PC: 0xfffffffff80a006ac
0x0000000000001004 in ?? ()
(gdb) si
0x0000000000001008 in ?? ()
(gdb) continue
Continuing.

Breakpoint 1, 0xfffffffff80a006ac in start_kernel ()
(gdb) a
```

### 三、讨论和心得

首先，由于我之前在 VMware 下载的 ubuntu 版本过低，导致我在更新升级 ubuntu 版本上遇到了不少的麻烦。后来在室友的提醒下，我重新创建了一个最新版本的 ubuntu，成功解决版本过低的问题。

其次，在本次实验当中，写思考题 1 时，由于知不知道 riscv64-linux-gnu-gcc 和 riscv64-linux-gnu-objdump 的用法，且在网上难以搜寻到相关资料，导致我无从入手。后来我尝试按照 gcc 的使用方法去使用，发现有所收效，在经过一番琢磨和调试之后，我在根目录下找到了 riscv64-linux-gnu-gcc 编译后的文件，顺利地完成了思考题 1 和思考题 2。

总的来说，经过这次实验，我学习到了用 QEMU 运行 Linux 内核以及用 GDB 远程调试 Linux 的一些技巧。

### 四、思考题

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

```
hwj@hwj-virtual-machine:~$ ls
公共的 视频 文档 音乐 c_project Linux snap
模板 图片 下载 桌面 hello_world.c os23fall-stu
hwj@hwj-virtual-machine:~$ riscv64-linux-gnu-gcc hello_world.c
hwj@hwj-virtual-machine:~$ ls
公共的 视频 文档 音乐 a.out hello_world.c os23fall-stu
模板 图片 下载 桌面 c_project Linux snap
```

在编译之前先查看了当前目录下的文件情况，同编译之后目录下的文件情况相对比，发现多出了 a.out 文件，故认定这是利用 riscv64-linux-gnu-gcc 编译后的产物。

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

```
hwj@hwj-virtual-machine:~$ riscv64-linux-gnu-objdump -d a.out > hello_world.txt
hwj@hwj-virtual-machine:~$ ls
公共的 视频 文档 音乐 a.out hello_world.c Linux snap
模板 图片 下载 桌面 c_project hello_world.txt os23fall-stu
```

```
hwj@hwj-virtual-machine:~$ cat hello_world.txt

a.out: 文件格式 elf64-littleriscv

Disassembly of section .plt:

0000000000000570 <.plt>:
570: 00002397          auipc  t2,0x2
574: 41c30333          sub    t1,t1,t3
578: a983be03          ld     t3,-1384(t2) # 2008 <__TMC_END__>
57c: fd430313          addi   t1,t1,-44
580: a9838293          addi   t0,t2,-1384
584: 00135313          srli   t1,t1,0x1
588: 0082b283          ld     t0,8(t0)
58c: 000e0067          jr     t3

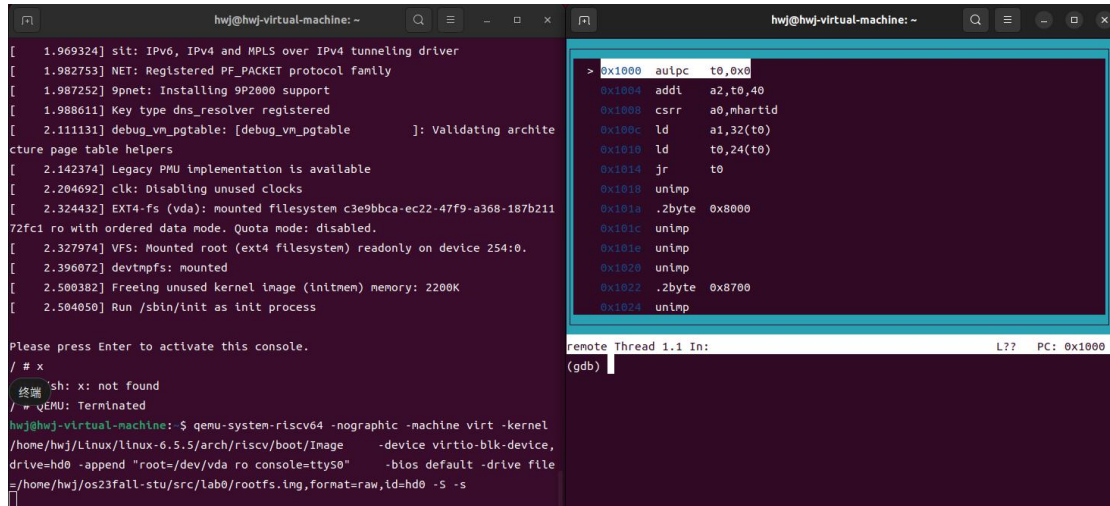
0000000000000590 <__libc_start_main@plt>:
590: 00002e17          auipc  t3,0x2
594: a88e3e03          ld     t3,-1400(t3) # 2018 <__libc_start_main@GLIBC_2.34>
598: 000e0367          jalr   t1,t3
59c: 00000013          nop

00000000000005a0 <printf@plt>:
5a0: 00002e17          auipc  t3,0x2
5a4: a80e3e03          ld     t3,-1408(t3) # 2020 <printf@GLIBC_2.27>
5a8: 000e0367          jalr   t1,t3
5ac: 00000013          nop
```

上图利用 riscv64-linux-gnu-objdump 反汇编了思考题 1 编译完的产物（并将结果指定为了 hello\_world.txt），随后利用 cat 指令查看了文件的内容。

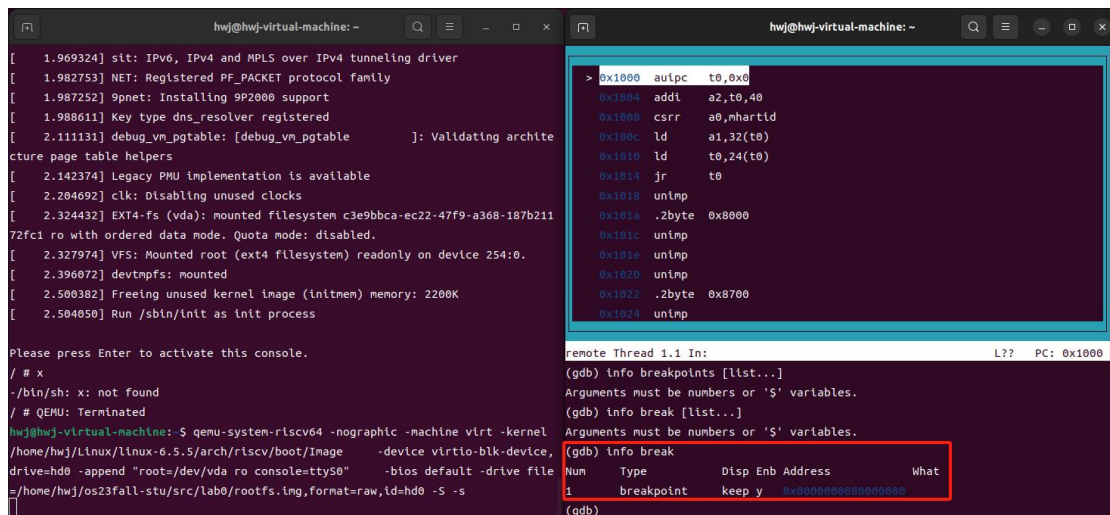
## 3.利用 GDB 调试 Linux

### 3.1 在 GDB 中查看汇编代码



```
hwj@hwj-virtual-machine: ~  
[ 1.969324] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver  
[ 1.982753] NET: Registered PF_PACKET protocol family  
[ 1.987252] 9pnet: Installing 9P2000 support  
[ 1.988611] Key type dns_resolver registered  
[ 2.111131] debug_vn_pgtbl: [debug_vn_pgtbl]: Validating architecture page table helpers  
[ 2.142374] Legacy PMU implementation is available  
[ 2.204692] clk: Disabling unused clocks  
[ 2.324432] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b21172fc1 ro with ordered data mode. Quota mode: disabled.  
[ 2.327974] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.  
[ 2.396072] devtmpfs: mounted  
[ 2.500382] Freeing unused kernel image (initmem) memory: 2200K  
[ 2.504050] Run /sbin/init as init process  
  
Please press Enter to activate this console.  
/ # x  
sh: x: not found  
^C  
/ # QEMU: Terminated  
hwj@hwj-virtual-machine: $ qemu-system-riscv64 -nographic -machine virt -kernel /home/hwj/Linux/linux-6.5.5/arch/riscv/boot/Image -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" -bios default -drive file=/home/hwj/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s  
  
> 0x1000 auipc t0,0x0  
0x1004 addi a2,t0,40  
0x1008 csrr a0,mhartid  
0x100c ld a1,32(t0)  
0x1010 ld t0,24(t0)  
0x1014 jr t0  
0x1018 unimp  
0x101a .2byte 0x8000  
0x101c unimp  
0x101e unimp  
0x1020 unimp  
0x1022 .2byte 0x8700  
0x1024 unimp  
  
remote Thread 1.1 In: L?? PC: 0x1000  
(gdb)
```

### 3.2 在 0x80000000 处下断点(并调用 info break 查看所有已下断点)



```
hwj@hwj-virtual-machine: ~  
[ 1.969324] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver  
[ 1.982753] NET: Registered PF_PACKET protocol family  
[ 1.987252] 9pnet: Installing 9P2000 support  
[ 1.988611] Key type dns_resolver registered  
[ 2.111131] debug_vn_pgtbl: [debug_vn_pgtbl]: Validating architecture page table helpers  
[ 2.142374] Legacy PMU implementation is available  
[ 2.204692] clk: Disabling unused clocks  
[ 2.324432] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b21172fc1 ro with ordered data mode. Quota mode: disabled.  
[ 2.327974] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.  
[ 2.396072] devtmpfs: mounted  
[ 2.500382] Freeing unused kernel image (initmem) memory: 2200K  
[ 2.504050] Run /sbin/init as init process  
  
Please press Enter to activate this console.  
/ # x  
-./bin/sh: x: not found  
^C  
/ # QEMU: Terminated  
hwj@hwj-virtual-machine: $ qemu-system-riscv64 -nographic -machine virt -kernel /home/hwj/Linux/linux-6.5.5/arch/riscv/boot/Image -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" -bios default -drive file=/home/hwj/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s  
  
> 0x1000 auipc t0,0x0  
0x1004 addi a2,t0,40  
0x1008 csrr a0,mhartid  
0x100c ld a1,32(t0)  
0x1010 ld t0,24(t0)  
0x1014 jr t0  
0x1018 unimp  
0x101a .2byte 0x8000  
0x101c unimp  
0x101e unimp  
0x1020 unimp  
0x1022 .2byte 0x8700  
0x1024 unimp  
  
remote Thread 1.1 In: L?? PC: 0x1000  
(gdb) info breakpoints [list...]  
Arguments must be numbers or '$' variables.  
(gdb) info break [list...]  
Arguments must be numbers or '$' variables.  
(gdb) info break  


| Num | Type       | Disp | Enb | Address            | What |
|-----|------------|------|-----|--------------------|------|
| 1   | breakpoint | keep | y   | 0x8000000000000000 |      |

  
(gdb)
```



### 3.3 & 3.4 在 0x80200000 处下断点（并调用 info break 查看了所有断点）

```
hwj@hwj-virtual-machine: ~  
[ 1.969324] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver  
[ 1.982753] NET: Registered PF_PACKET protocol family  
[ 1.987252] 9pnet: Installing 9P2000 support  
[ 1.988611] Key type dns_resolver registered  
[ 2.11131] debug_vn_pgtbl: [debug_vn_pgtbl]: Validating archite  
cture page table helpers  
[ 2.142374] Legacy PMU implementation is available  
[ 2.204692] clk: Disabling unused clocks  
[ 2.324432] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b211  
72fc1 ro with ordered data mode. Quota mode: disabled.  
[ 2.327974] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.  
[ 2.396072] devtmpfs: mounted  
[ 2.500382] Freeing unused kernel image (initmem) memory: 2200K  
[ 2.504050] Run /sbin/init as init process  
  
Please press Enter to activate this console.  
/ # x  
-/bin/sh: x: not found  
/ # QEMU: Terminated  
hwj@hwj-virtual-machine: ~$ qemu-system-riscv64 -nographic -machine virt -kernel  
/home/hwj/Linux/linux-6.5.5/arch/riscv/boot/Image -device virtio-blk-device,  
drive=hd0 -append "root=/dev/vda ro console=ttyS0" -bios default -drive file  
=/home/hwj/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s  
  
> 0x1000 auipc t0,0x0  
0x1004 addi a2,t0,40  
0x1008 csrr a0,mhartid  
0x100c ld a1,32(t0)  
0x1010 ld t0,24(t0)  
0x1014 jr t0  
0x1018 unimp  
0x101a .2byte 0x8000  
0x101c unimp  
0x101e unimp  
0x1020 unimp  
0x1022 .2byte 0x8700  
0x1024 unimp  
  
remote Thread 1.1 In: L?? PC: 0x1000  
1 breakpoint keep y 0x0000000000000000  
(gdb) b * 0x80200000  
Breakpoint 2 at 0x80200000  
(gdb) info break  
Num Type Disp Enb Address What  
1 breakpoint keep y 0x0000000000000000  
2 breakpoint keep y 0x0000000000000000  
(gdb)
```

### 3.5 清除 0x80000000 处的断点

```
hwj@hwj-virtual-machine: ~  
[ 1.969324] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver  
[ 1.982753] NET: Registered PF_PACKET protocol family  
[ 1.987252] 9pnet: Installing 9P2000 support  
[ 1.988611] Key type dns_resolver registered  
[ 2.11131] debug_vn_pgtbl: [debug_vn_pgtbl]: Validating archite  
cture page table helpers  
[ 2.142374] Legacy PMU implementation is available  
[ 2.204692] clk: Disabling unused clocks  
[ 2.324432] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b211  
72fc1 ro with ordered data mode. Quota mode: disabled.  
[ 2.327974] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.  
[ 2.396072] devtmpfs: mounted  
[ 2.500382] Freeing unused kernel image (initmem) memory: 2200K  
[ 2.504050] Run /sbin/init as init process  
  
Please press Enter to activate this console.  
/ # x  
-/bin/sh: x: not found  
/ # QEMU: Terminated  
hwj@hwj-virtual-machine: ~$ qemu-system-riscv64 -nographic -machine virt -kernel  
/home/hwj/Linux/linux-6.5.5/arch/riscv/boot/Image -device virtio-blk-device,  
drive=hd0 -append "root=/dev/vda ro console=ttyS0" -bios default -drive file  
=/home/hwj/os23fall-stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s  
  
> 0x1000 auipc t0,0x0  
0x1004 addi a2,t0,40  
0x1008 csrr a0,mhartid  
0x100c ld a1,32(t0)  
0x1010 ld t0,24(t0)  
0x1014 jr t0  
0x1018 unimp  
0x101a .2byte 0x8000  
0x101c unimp  
0x101e unimp  
0x1020 unimp  
0x1022 .2byte 0x8700  
0x1024 unimp  
  
remote Thread 1.1 In: L?? PC: 0x1000  
Num Type Disp Enb Address What  
1 breakpoint keep y 0x0000000000000000  
2 breakpoint keep y 0x0000000000000000  
(gdb) delete 1  
(gdb) info break  
Num Type Disp Enb Address What  
2 breakpoint keep y 0x0000000000000000  
(gdb)
```

调用 delete 1 指令清除了第一个断点，随后调用 info break 发现只剩下了编号为 2 的断点

### 3.6 继续运行直到触发 0x80200000 处的断点

```
hwj@hwj-virtual-machine: ~  
Runtime SBI Version : 0.2  
Domain0 Name : root  
Domain0 Boot HART : 0  
Domain0 HARTs : 0*  
Domain0 Region00 : 0x0000000000000000-0x0000000000000000ffff ()  
Domain0 Region01 : 0x0000000000000000-0xffffffffffffffff (R,W,X)  
Domain0 Next Address : 0x0000000000000000  
Domain0 Next Arg1 : 0x0000000000000000  
Domain0 Next Mode : S-mode  
Domain0 SysReset : yes  
  
Boot HART ID : 0  
Boot HART Domain : root  
Boot HART ISA : rv64imafdcu  
Boot HART Features : scounter,ncounter,stime  
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 : 0x0000000000000022  
Boot HART MEDELEG : 0x0000000000000b109  
  
> 0x80200000 li s4,-13  
0x80200002 j 0x80201000  
0x80200006 nop  
0x80200008 unimp  
0x8020000a addi s0,s0,8  
0x8020000c unimp  
0x8020000e unimp  
0x80200010 fsd fs0,32(s0)  
0x80200012 .4byte 0x157  
0x80200016 unimp  
0x80200018 unimp  
0x8020001a unimp  
0x8020001c unimp  
  
remote Thread 1.1 In: L?? PC: 0x80200000  
(gdb) info break  
Num Type Disp Enb Address What  
2 breakpoint keep y 0x0000000000000000  
(gdb) c  
Continuing.  
Breakpoint 2, 0x0000000000000000 in ?? ()  
(gdb)
```



### 3.7 单步调试一次

```
hwj@hwj-virtual-machine: ~  
Runtime SBI Version : 0.2  
  
Domain0 Name : root  
Domain0 Boot HART : 0  
Domain0 HARTs : 0*  
Domain0 Region00 : 0x0000000000000000-0x0000000000001ffff (R,W,X)  
Domain0 Region01 : 0x0000000000000000-0xffffffffffffffff (R,W,X)  
Domain0 Next Address : 0x000000000200000  
Domain0 Next Arg1 : 0x0000000007000000  
Domain0 Next Mode : S-mode  
Domain0 SysReset : yes  
  
Boot HART ID : 0  
Boot HART Domain : root  
Boot HART ISA : rv64lmafcsu  
Boot HART Features : scounteren,ncounteren,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  
  
[  
R+ 0x80200000 Ti s4 -13  
> 0x80200002 j 0x802010d0  
0x80200005 nop  
0x80200008 unimp  
0x8020000a addi s0,sp,8  
0x8020000c unimp  
0x8020000e unimp  
0x80200010 fsd fs0,32(s0)  
0x80200012 .4byte 0x157  
0x80200016 unimp  
0x80200018 unimp  
0x8020001a unimp  
0x8020001c unimp  
  
remote Thread 1.1 In: L77 PC: 0x80200002  
Breakpoint 2, 0x0000000000020000 in ?? ()  
(gdb) n  
Cannot find bounds of current function  
(gdb) next  
Cannot find bounds of current function  
(gdb) st  
0x0000000000020002 in ?? ()  
(gdb)
```

### 3.8 退出 QEMU

```
hwj@hwj-virtual-machine: ~  
[ 1.662056] sdhci-pltfm: SDHCI platform and OF driver helper  
[ 1.668362] usbcore: registered new interface driver usbhid  
[ 1.671067] usbhid: USB HID core driver  
[ 1.677195] NET: Registered PF_INET6 protocol family  
[ 1.712419] Segment Routing with IPv6  
[ 1.713828] In-situ OAM (IOAM) with IPv6  
[ 1.717119] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver  
[ 1.734729] NET: Registered PF_PACKET protocol family  
[ 1.741812] 9pnet: Installing 9P2000 support  
[ 1.745505] Key type dns_resolver registered  
[ 1.908038] debug_vn_pgtbl: [debug_vn_pgtbl]: Validating archite  
cture page table helpers  
[ 1.949648] Legacy PMU implementation is available  
[ 1.957152] clk: Disabling unused clocks  
[ 2.104879] EXT4-fs (vda): mounted filesystem c3e9bbca-ec22-47f9-a368-187b211  
72fc1 ro with ordered data mode. Quota mode: disabled.  
[ 2.108857] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.  
[ 2.119031] devtmpfs: mounted  
[ 2.253994] Freeing unused kernel image (initramfs) memory: 2200K  
[ 2.256941] Run /sbin/init as init process  
  
Please press Enter to activate this console.  
/ # QEMU: Terminated  
hwj@hwj-virtual-machine: $  
  
hwj@hwj-virtual-machine: ~  
For help, type "help".  
Type "apropos word" to search for commands related to "word"...  
path/to/linux/vmlinux: 没有那个文件或目录。  
(gdb) target remote :1234  
Remote debugging using :1234  
warning: No executable has been specified and target does not support  
determining executable automatically. Try using the "file" command.  
0x0000000000001000 in ?? ()  
(gdb) layout  
List of layout subcommands:  
layout asm -- Apply the "asm" layout.  
layout next -- Apply the next TUI layout.  
layout prev -- Apply the previous TUI layout.  
layout regs -- Apply the TUI register layout.  
layout split -- Apply the "split" layout.  
layout src -- Apply the "src" layout.  
  
Type "help layout" followed by layout subcommand name for full documentation.  
Type "apropos word" to search for commands related to "word".  
Type "apropos -v word" for full documentation of commands related to "word".  
Command name abbreviations are allowed if unambiguous.  
(gdb) layout asm  
hwj@hwj-virtual-machine: $
```

## 4.使用 make 工具清除 Linux 的构建产物

```
hwj@hwj-virtual-machine:~/Linux/linux-6.5.5$ ls
arch          drivers      kernel      modules.order  sound
block         fs          lib         Module.symvers  System.map
built-in.a    include     LICENSES   net             tools
certs         init        MAINTAINERS README          usr
COPYING       io_uring    Makefile   rust           virt
CREDITS       ipc         mm         samples        vmlinux
crypto        Kbuild     modules.builtin  scripts        vmlinux.a
Documentation Kconfig    modules.builtin.modinfo security        vmlinux.o

hwj@hwj-virtual-machine:~/Linux/linux-6.5.5$ 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
```

先切换到 Linux 所在的目录，在调用 make clean 指令。

## 5. vmlinux 和 Image 的关系和区别是什么？

(1) vmlinux:是内核编译出来的原始的内核文件未经压缩的。是 ELF 格式的，即编译出来的最原始的文件。用于 kernel-debug，产生 system.map 符号表，不能用于直接加载，不可以作为启动内核。只是启动过程中的中间媒体。

(2) Image: 是 Linux 内核镜像文件，但是 Image 仅包含可执行的二进制数据。Image 就是使用 objcopy 取消掉 vmlinux 中的一些其他信息，比如符号表什么的。但是 Image 是没有压缩过的，Image 保存在 arch/arm/boot 目录下。Image 是经过 objcopy 处理的只包含二进制数据的内核代码，它已经不是 elf 格式了，但这种格式的内核镜像还没有经过压缩。

# 五、附录

无