浙江大学

《操作系统原理与实践》
实验报告

实验指导：	Lab0实验指导
姓　　名：	王晓宇
学　　号：	3220104364
电子邮箱：	3220104364@zju.edu.cn
联系电话：	19550222634
授课教师：	申文博
助　　教：	陈淦豪&王鹤翔&许昊瑞

2024 年 9 月 10 日

Lab0实验指导

Lab0实验指导

1 实验内容及简要原理介绍

- 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）。
- 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（4.1-4.5），并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；
  - 完成思考题。

2 实验具体过程

2.1 实验截图(4.1-4.5)&以及结果分析

2.1.1 搭建实验环境4.1

1. 安装编译内核所需要的交叉编译工具链

```
sudo apt install  gcc-riscv64-linux-gnu
```

```
axin0401@LAPTOP-0P208VUK:/mnt/d/cs/OS-studying$ sudo apt install  gcc-riscv64-linux-gnu
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  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 libisl23 libmpc3 linux-libc-dev-riscv64-cross
Suggested packages:
  binutils-doc gcc-11-locales cpp-doc gcc-11-doc make manpages-dev autoconf automake libtool flex bison gdb-riscv64-linux-gnu gcc-doc
The following NEW packages will be installed:
  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 libisl23 libmpc3
  linux-libc-dev-riscv64-cross
0 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 35.7 MB of archives.
After this operation, 115 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

结果：安装完成

```
Setting up libmpc3:amd64 (1.2.1-2build1) ...
Setting up gcc-11-cross-base-ports (11.4.0-1ubuntu1~22.04cross1) ...
Setting up linux-libc-dev-riscv64-cross (5.15.0-22.22cross4) ...
Setting up libc6-riscv64-cross (2.35-0ubuntu3cross4) ...
Setting up libgomp1-riscv64-cross (12.3.0-1ubuntu1~22.04cross1) ...
Setting up libisl23:amd64 (0.24-2build1) ...
Setting up libatomic1-riscv64-cross (12.3.0-1ubuntu1~22.04cross1) ...
Setting up libcc1-0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up cpp-11-riscv64-linux-gnu (11.4.0-1ubuntu1~22.04cross1) ...
Setting up libgcc-s1-riscv64-cross (12.3.0-1ubuntu1~22.04cross1) ...
Setting up libc6-dev-riscv64-cross (2.35-0ubuntu3cross4) ...
Setting up libasan6-riscv64-cross (11.4.0-1ubuntu1~22.04cross1) ...
Setting up cpp-riscv64-linux-gnu (4:11.2.0-1ubuntu1) ...
Setting up libgcc-11-dev-riscv64-cross (11.4.0-1ubuntu1~22.04cross1) ...
Setting up gcc-11-riscv64-linux-gnu (11.4.0-1ubuntu1~22.04cross1) ...
Setting up gcc-riscv64-linux-gnu (4:11.2.0-1ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link
```

2. 安装用于构建程序的软件包：

```
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
```

```
axin0401@LAPTOP-0P208VUK:/usr/OS$ 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'
curl is already the newest version (7.81.0-1ubuntu1.17).
curl set to manually installed.
gawk is already the newest version (1:5.1.0-1ubuntu0.1).
gawk set to manually installed.
git is already the newest version (1:2.34.1-1ubuntu1.11).
git set to manually installed.
The following additional packages will be installed:
  bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libauthen-sasl-perl libc-dev-bin libc-devtools libc6-dev libclone-perl
  libcrypt-dev libdata-dump-perl libdeflate0 libdpkg-perl libencode-locale-perl libfakeroot libfile-fcntllock-perl libfile-listing-perl libfl-dev
  libfl2 libfont-afm-perl libfontconfig1 libfreetype6 libgcc-11-dev libgd3 libgmpxx4ldbl libgomp1 libhtml-form-perl libhtml-format-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
  libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblsan0 libltdl-dev libltdl7
  liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnsl-dev
  libquadmath0 libstdc++-11-dev libtext-unidecode-perl libtiff5 libtimedate-perl libtirpc-dev libtry-tiny-perl libtsan0 libubsan1 liburi-perl
  libwebp7 libwww-perl libwww-robotrules-perl libxml-libxml-perl libxml-namespacesupport-perl libxml-parser-perl libxml-sax-base-perl
  libxml-sax-expat-perl libxml-sax-perl libxpm4 linux-libc-dev lto-disabled-list m4 make manpages-dev perl-openssl-defaults rpcsvc-proto tex-comm
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc gettext bison-doc bzip2-doc cpp-doc gcc-11-locales debian-keyring flex-doc g++-multilib
  g++-11-multilib gcc-11-doc gcc-multilib gdb gcc-doc gcc-11-multilib libdigest-hmac-perl libgssapi-perl glibc-doc bzr libgd-tools gmp-doc
  libgmp10-doc libtool-doc libcrypt-ssleay-perl libmpfr-doc libstdc++-11-doc gfortran | fortran95-compiler gcj-jdk libsub-name-perl
  libbusiness-isbn-perl libauthen-ntlm-perl libxml-sax-expatxs-perl m4-doc make-doc debhelper texlive-base texlive-latex-base texlive-plain-gener
  texlive-fonts-recommended
The following NEW packages will be installed:
  autoconf automake autotools-dev bc bison build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot flex fontconfig-config fonts-dejavu-core g++ g++-11
  gcc gcc-11 gcc-11-base gperf libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libauthen-sasl-perl
  libc-dev-bin libc-devtools libc6-dev libclone-perl libcrypt-dev libdata-dump-perl libdeflate0 libdpkg-perl libencode-locale-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libfile-listing-perl libfl-dev libfl2 libfont-afm-perl libfontconfig1 libfreetype6 libgcc-11-dev libgd3
  libgmp-dev libgmpxx4ldbl libgomp1 libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl
  libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblsan0 libltdl-dev libltdl7 liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl
  libmpc-dev libmpfr-dev libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnsl-dev libquadmath0 libstdc++-11-dev libtext-unidecode-perl
  libtiff5 libtimedate-perl libtirpc-dev libtool libtry-tiny-perl libtsan0 libubsan1 liburi-perl libwebp7 libwww-perl libwww-robotrules-perl
  libxml-libxml-perl libxml-namespacesupport-perl libxml-parser-perl libxml-sax-base-perl libxml-sax-expat-perl libxml-sax-perl libxpm4
  linux-libc-dev lto-disabled-list m4 make manpages-dev patchutils perl-openssl-defaults rpcsvc-proto tex-common texinfo zlib1g-dev
0 upgraded, 109 newly installed, 0 to remove and 0 not upgraded.
Need to get 71.0 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Do you want to continue? [Y/n] |
```

结果：安装完成

```
Setting up g++-11 (11.4.0-1ubuntu1~22.04) ...
Setting up g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Setting up liblwp-protocol-https-perl (6.10-1) ...
Setting up libwww-perl (6.61-1) ...
Setting up libxml-parser-perl:amd64 (2.46-3build1) ...
Setting up libxml-sax-expat-perl (0.51-1) ...
update-perl-sax-parsers: Registering Perl SAX parser XML::SAX::Expat with priority 50...
update-perl-sax-parsers: Updating overall Perl SAX parser modules info file...
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link
```

3. 安装用于启动 riscv64 平台上的内核的模拟器 qemu

```
1 sudo apt install qemu-system-misc
```

```
axin0401@LAPTOP-0P208VUK:/usr/OS$ sudo apt install qemu-system-misc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  acl adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core dconf-gsettings-backend dconf-service fontconfig glib-networking glib-networking-common glib-networking-services
  gsettings-desktop-schemas gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-x gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme ibverbs-providers ipxe-qemu libaa1 li
  libasound2 libasound2-data libasyncns0 libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libavc1394-0 libboost-iostreams1.74.0
  libboost-thread1.74.0 libbrlapi0.8 libcaca0 libcacard0 libcairo-gobject2 libcairo2 libcdparanoia0 libcolord2 libcups2 libdatrie1 libdaxctl1 libdconf1 libdecor-0-0 libdecor-0-plugin-1-cairo li
  libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libdv4 libepoxy0 libfdt1 libflac8 libgbm1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgfapi0 libgfrpc0 libgfxdr0 libgl1 li
  libgl1-mesa-dri libglapi-mesa libglusterfs0 libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0 libgtk-3-0 libgtk-3-bin libgtk-3-common
  libharfbuzz0b libibverbs1 libiec61883-0 libiscsi7 libjack-jackd2-0 liblcms2-2 libllvm15 libmp3lame0 libmpg123-0 libndctl6 libnl-route-3-200 libnspr4 libnss3 libogg0 libopus0 liborc-0.4-0 libp
  libpangocairo-1.0-0 libpangoft2-1.0-0 libpciaccess0 libpcsclite1 libpixman-1-0 libpmem1 libpmemobj1 libproxy1v5 libpulse0 librados2 libraw1394-11 librbd1 librdmacm1 librsvg2-2 librsvg2-common
  libsdl2-2.0-0 libsensors-config libsensors5 libshout3 libslirp0 libsndfile1 libsoup2.4-1 libsoup2.4-common libspeex1 libspice-server1 libtag1v5 libtag1v5-vanilla libthai-data libthai0 libtheo
  liburing2 libusbredirparser1 libv4l-0 libv4lconvert0 libvirglrenderer1 libvisual-0.4-0 libvorbis0a libvorbisenc2 libvpx7 libvte-2.91-0 libvte-2.91-common libwavpack1 libwayland-client0 libway
  libwayland-egl1 libwayland-server0 libx11-xcb1 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libx
  libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon0 libxrandr2 libxrender1 libxshmfence1 libxss1 libxtst6 libxv1 libxxf86vm1 qemu-block-extra qemu-system-common qemu-system-data qemu-sys
  qemu-utils seabios session-migration ubuntu-mono x11-common
Suggested packages:
  gvfs libasound2-plugins alsa-utils colord cups-common libdv-bin oss-compat libvisual-0.4-plugins jackd2 liblcms2-utils opus-tools pcscd pulseaudio libraw1394-doc librsvg2-bin xdg-utils lm-sen
  gstreamer1.0-libav gstreamer1.0-plugins-ugly samba vde2 debootstrap
The following NEW packages will be installed:
  acl adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core dconf-gsettings-backend dconf-service fontconfig glib-networking glib-networking-common glib-networking-services
  gsettings-desktop-schemas gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-x gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme ibverbs-providers ipxe-qemu libaa1 li
  libasound2 libasound2-data libasyncns0 libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libavc1394-0 libboost-iostreams1.74.0
  libboost-thread1.74.0 libbrlapi0.8 libcaca0 libcacard0 libcairo-gobject2 libcairo2 libcdparanoia0 libcolord2 libcups2 libdatrie1 libdaxctl1 libdconf1 libdecor-0-0 libdecor-0-plugin-1-cairo li
  libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libdv4 libepoxy0 libfdt1 libflac8 libgbm1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgfapi0 libgfrpc0 libgfxdr0 libgl1 li
  libgl1-mesa-dri libglapi-mesa libglusterfs0 libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0 libgtk-3-0 libgtk-3-bin libgtk-3-common
  libharfbuzz0b libibverbs1 libiec61883-0 libiscsi7 libjack-jackd2-0 liblcms2-2 libllvm15 libmp3lame0 libmpg123-0 libndctl6 libnl-route-3-200 libnspr4 libnss3 libogg0 libopus0 liborc-0.4-0 libp
  libpangocairo-1.0-0 libpangoft2-1.0-0 libpciaccess0 libpcsclite1 libpixman-1-0 libpmem1 libpmemobj1 libproxy1v5 libpulse0 librados2 libraw1394-11 librbd1 librdmacm1 librsvg2-2 librsvg2-common
  libsdl2-2.0-0 libsensors-config libsensors5 libshout3 libslirp0 libsndfile1 libsoup2.4-1 libsoup2.4-common libspeex1 libspice-server1 libtag1v5 libtag1v5-vanilla libthai-data libthai0 libtheo
  liburing2 libusbredirparser1 libv4l-0 libv4lconvert0 libvirglrenderer1 libvisual-0.4-0 libvorbis0a libvorbisenc2 libvpx7 libvte-2.91-0 libvte-2.91-common libwavpack1 libwayland-client0 libway
  libwayland-egl1 libwayland-server0 libx11-xcb1 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libx
  libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon0 libxrandr2 libxrender1 libxshmfence1 libxss1 libxtst6 libxv1 libxxf86vm1 qemu-block-extra qemu-system-common qemu-system-data qemu-sys
  qemu-system-misc qemu-utils seabios session-migration ubuntu-mono x11-common
0 upgraded, 177 newly installed, 0 to remove and 0 not upgraded.
Need to get 125 MB of archives.
After this operation, 536 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

结果：安装完成

```
axin0401@LAPTOP-0P208VUK:/usr$ 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-2ubuntu6.22).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

4. 安装 gdb

> gdb-multiarch 是编译好的多架构 gdb，可以用于调试多种架构的程序

```
1 sudo apt install gdb-multiarch
```

```
axin0401@LAPTOP-0P208VUK:/usr$ sudo apt install gdb-multiarch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gdb libbabeltrace1 libboost-regex1.74.0 libc6-dbg libdebuginfod-common libdebuginfod1 libipt2 libsource-highlight-common libsource-highlight4v5
Suggested packages:
  gdb-doc gdbserver
The following NEW packages will be installed:
  gdb gdb-multiarch libbabeltrace1 libboost-regex1.74.0 libc6-dbg libdebuginfod-common libdebuginfod1 libipt2 libsource-highlight-common libsource-highlight4v5
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 23.3 MB of archives.
After this operation, 53.3 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2.1.2 获取 Linux 源码和已经编译好的文件系统4.2

1. 利用wget工具下载Linux源码3.11-rc7到/usr目录下

wget 是一个命令行工具，可以在没有图形用户界面的情况下使用，并且方便下载到当前路径。

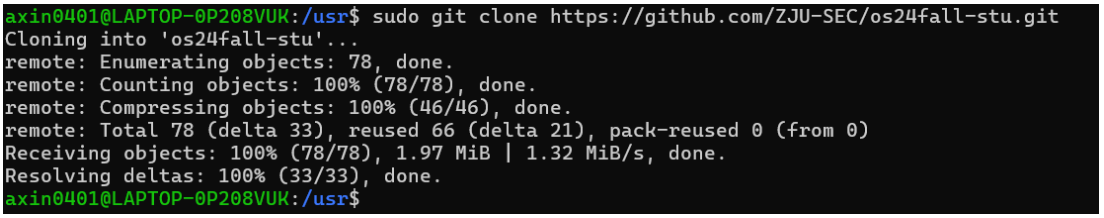
```
sudo wget https://git.kernel.org/torvalds/t/linux-6.11-rc7.tar.gz
```

解压缩文件（再WSL中使用解压缩命令）：

```
sudo tar -xzvf linux-6.11-rc7.tar.gz
```

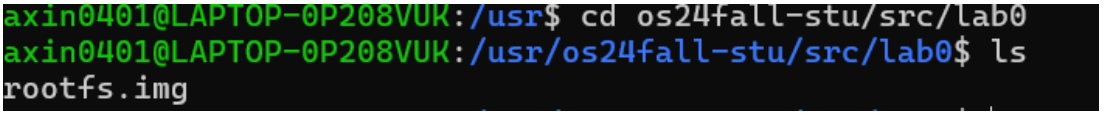
2. 利用git clone获取仓库文件

```
sudo git clone https://github.com/ZJU-SEC/os24fall-stu.git
```



查看克隆文件：

```
cd os24fall-stu/src/lab0
ls
```



其中 rootfs.img 是已经构建完成的根文件系统的镜像

2.1.3 编译 Linux 内核4.3

```
cd /usr/linux-6.11-rc7/
sudo make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- defconfig    # 使用默认配置
sudo make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- -j12    # 编译
```

1. 使用默认配置

```
axin0401@LAPTOP-0P208VUK:/usr/linux-6.11-rc7$ sudo 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
#
```

2. 编译，这里设置j为12，加快编译速度

```
axin0401@LAPTOP-0P208VUK:/usr/linux-6.11-rc7$ sudo make ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu- -j1
2
  CALL    scripts/checksyscalls.sh
  Kernel: arch/riscv/boot/Image is ready
  Kernel: arch/riscv/boot/Image.gz is ready
```

2.1.4 使用 QEMU 运行内核4.4

```
sudo qemu-system-riscv64 -nographic -machine virt -kernel /usr/linux-6.11-rc7/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
    -bios default -drive file=/usr/os24fall-stu/src/lab0/rootfs.img,format=raw,id=hd0
```

2.1.5 使用 GDB 对内核进行调试4.5

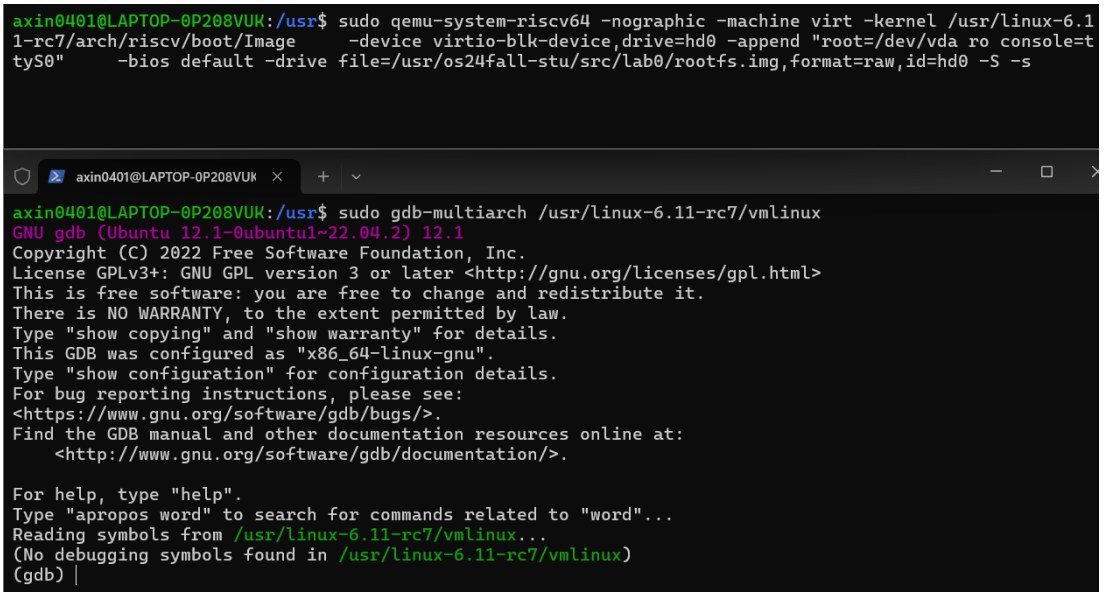
1. QEMU运行内核，做好GDB监听工作

```
# Terminal 1
sudo qemu-system-riscv64 -nographic -machine virt -kernel
/usr/linux-6.11-rc7/arch/riscv/boot/Image \
    -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro
console=ttyS0" \
    -bios default -drive file=/usr/os24fall-
stu/src/lab0/rootfs.img,format=raw,id=hd0 -S -s

# Terminal 2
sudo gdb-multiarch /usr/linux-6.11-rc7/vmlinux
```

相较于4.4的命令，此时的命令最后多了`-S -s`，在使用 QEMU 模拟器时，`-S` 和 `-s` 是两个用于控制 QEMU 调试功能的选项：

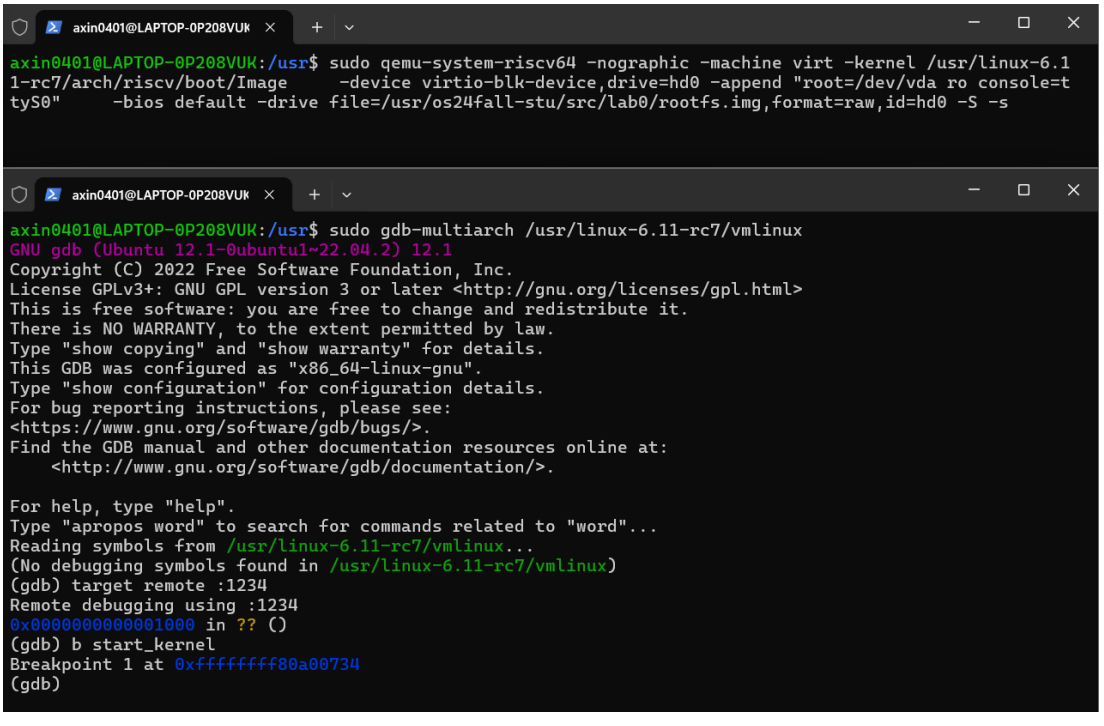
1. `-S`：这个选项用于在启动 QEMU 时暂停执行，直到你连接到 QEMU 的 GDB 服务器。

2. `-s`：这个选项用于启用 QEMU 的 GDB 远程调试功能，等待 GDB 客户端连接。



2. GDB连接QEMU,并设置断点在Kernel起点

```
(gdb) target remote :1234    # 连接 qemu
(gdb) b start_kernel         # 设置断点
```



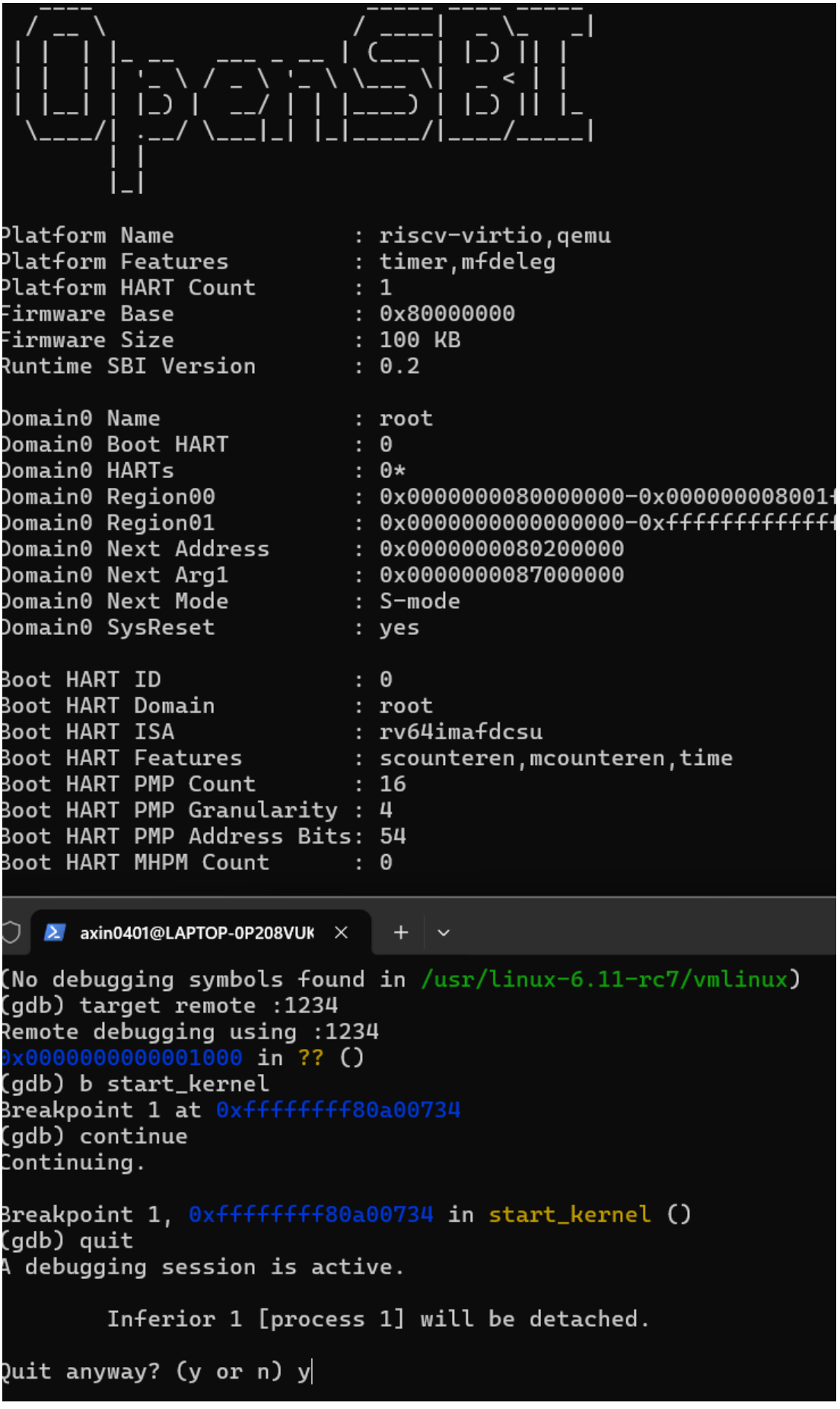
3. 执行Kernel

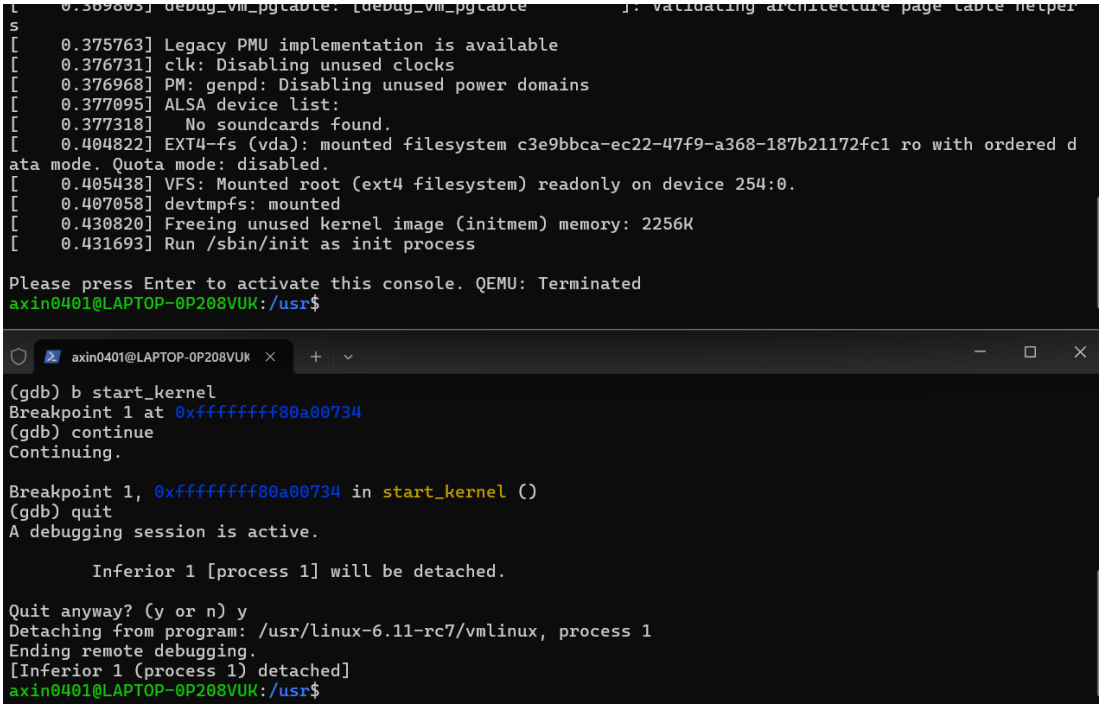
```
(gdb) continue              # 继续执行
```

可以看到kernel继续运行了，并显示本机状态

4. 退出GDB调试

```
(gdb) quit                  # 退出 gdb
```


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 : 0x0000000080000000-0x000000008001f
Domain0 Region01 : 0x0000000000000000-0xffffffffffff
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
axin0401@LAPTOP-0P208VUK
(No debugging symbols found in /usr/linux-6.11-rc7/vmlinux)
(gdb) target remote :1234
Remote debugging using :1234
0x0000000000001000 in ?? ()
(gdb) b start_kernel
Breakpoint 1 at 0xffffffff80a00734
(gdb) continue
Continuing.
Breakpoint 1, 0xffffffff80a00734 in start_kernel ()
(gdb) quit
A debugging session is active.
Inferior 1 [process 1] will be detached.
Quit anyway? (y or n) y



可以看到成功退出界面了

2.2 GDB各种命令尝试

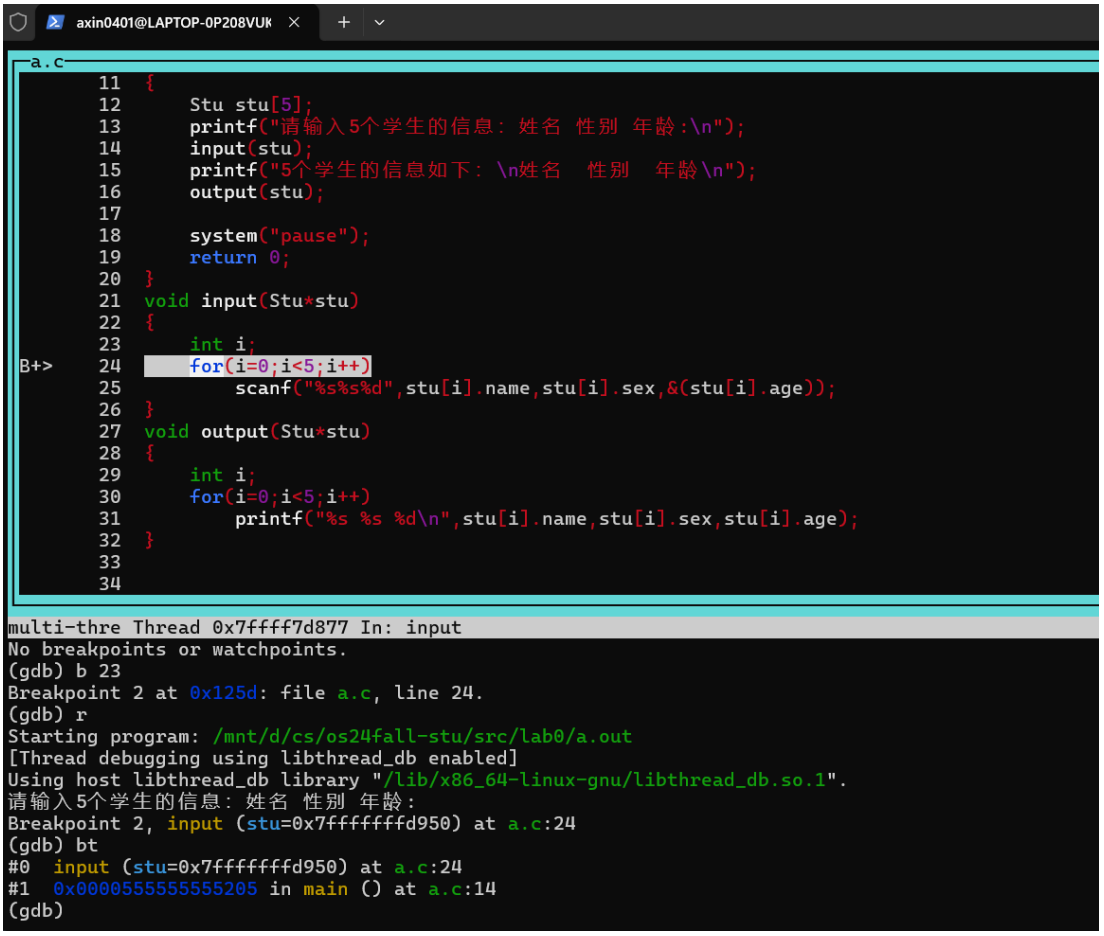
> 通过gcc -c -g a.c和 gcc a.o生成a.out
>
> 通过gdb a.out进入调试
>
> 以下命令均在执行以下程序：

```c
#include<stdio.h>
#include<stdlib.h>
typedef struct{
    char name[20];
    char sex[5];
    int  age;
}Stu;
void input(Stu*stu);
void output(Stu*stu);
int main()
{
    Stu stu[5];
    printf("请输入5个学生的信息：姓名 性别 年龄:\n");
    input(stu);
```

```
    printf("5个学生的信息如下：\n姓名  性别  年龄\n");
    output(stu);

    system("pause");
    return 0;
}
void input(Stu*stu)
{
    int i;
    for(i=0;i<5;i++)
        scanf("%s%s%d",stu[i].name,stu[i].sex,&(stu[i].age));
}
void output(Stu*stu)
{
    int i;
    for(i=0;i<5;i++)
        printf("%s %s %d\n",stu[i].name,stu[i].sex,stu[i].age);
}
```

2.2.1 backtrace

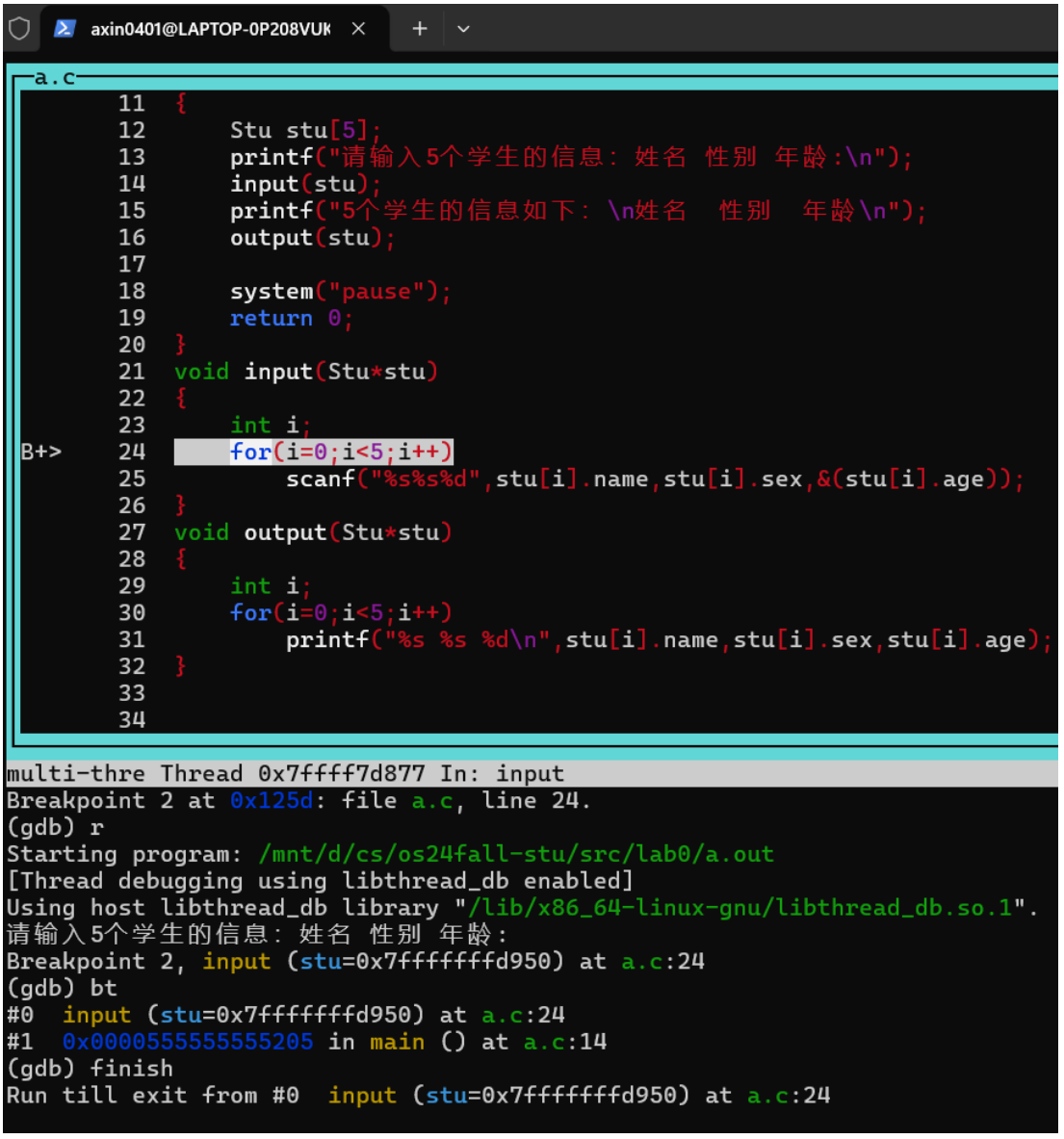
```
(gdb) backtrace #查看函数的调用的栈帧和层级关系，简写 `bt`
```



通过在input函数设置断点运行后，我们在断点处执行bt，可以看到栈顶执行函数为input()，符合调用顺序，下一函数为程序主函数，意味着调用input()后即执行main()

2.2.2 finish

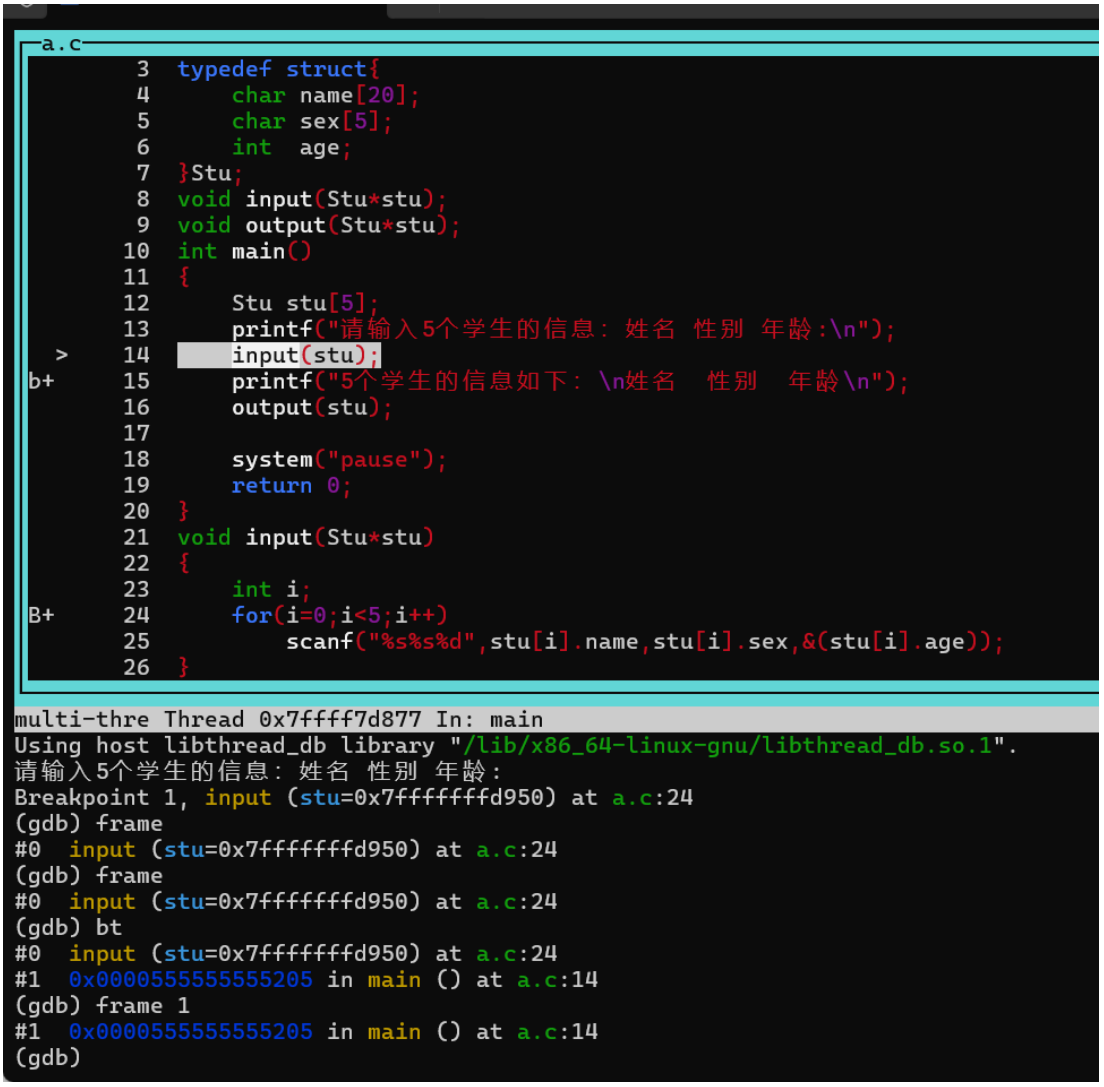
```
(gdb) finish #结束当前函数，返回到函数调用点
```



接上文执行finish后，程序调用返回到input函数调用点

2.2.3 frame

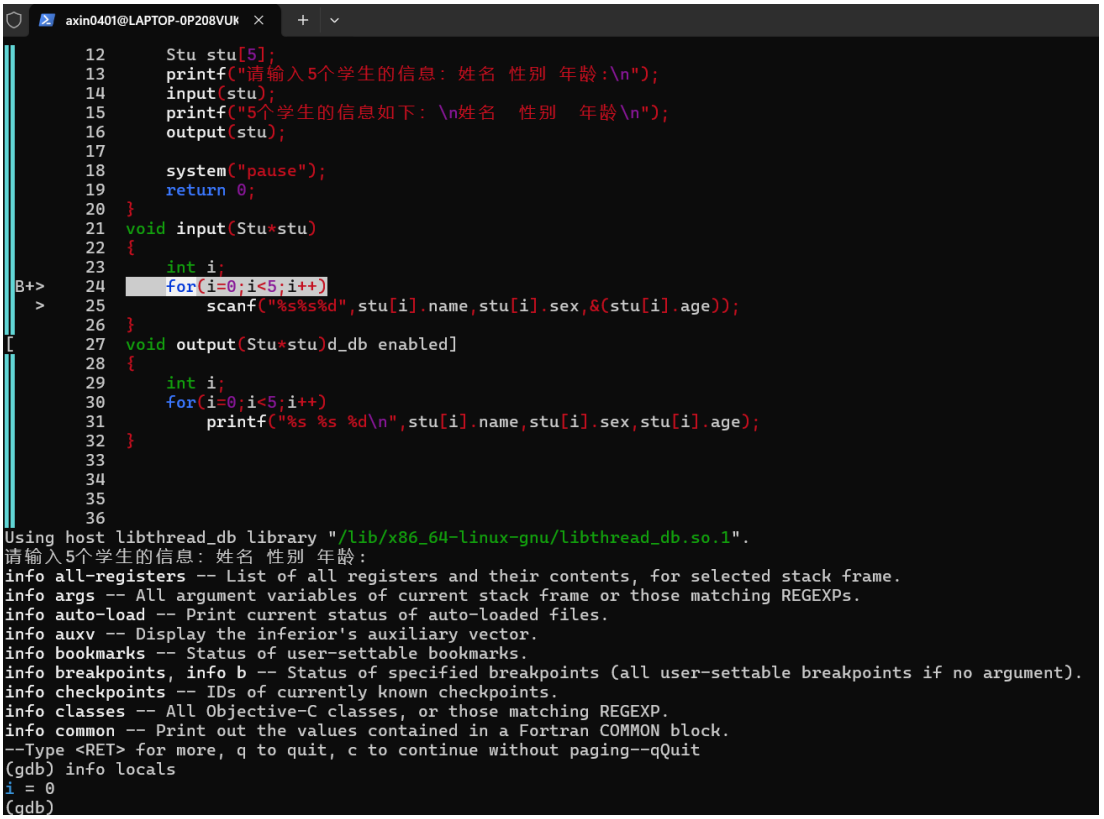
```
(gdb) frame # 切换函数的栈帧，简写 f
```



重新运行程序，这里设置的断点仍然是在函数内部，当执行到函数内部暂停时，使用bt命令查看帧栈编号，再使用frame 1切换回main函数帧栈

2.2.4 info

```
(gdb) info #查看函数内部局部变量的数值，简写 i
```



在函数调用是，输入info locals可以查看函数内所有局部变量的值，这里显示循环轮数变量i=0

2.2.5 break

```
(gdb) break #设置断点，简写 b
```

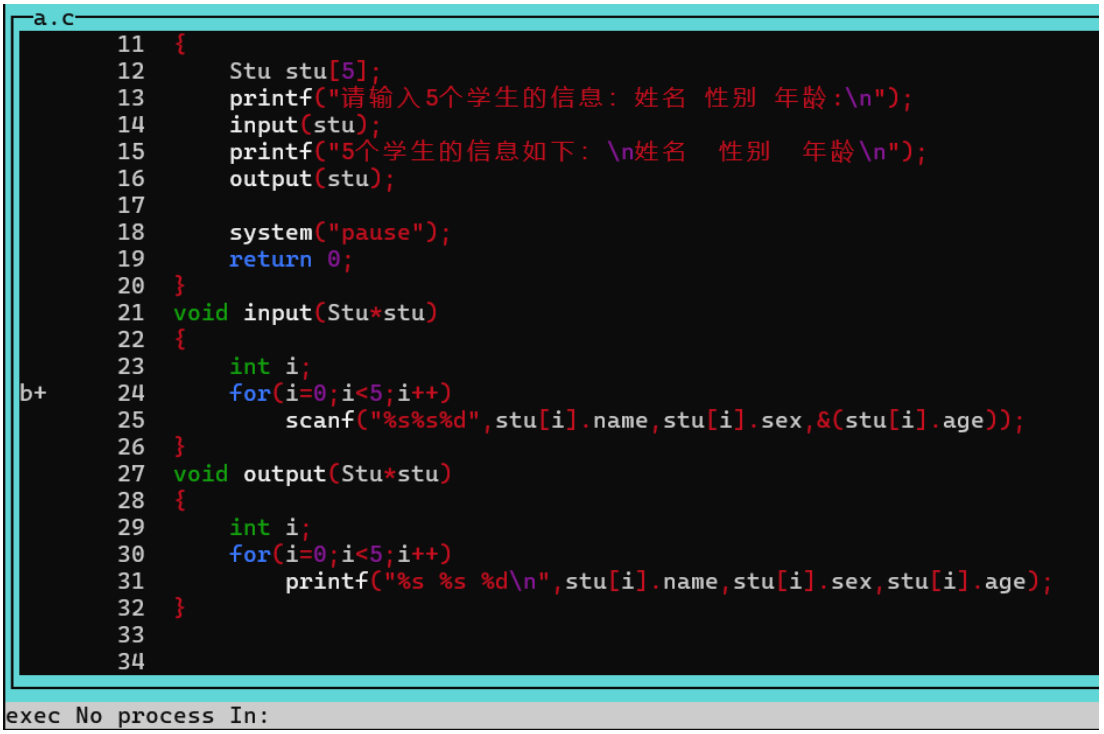
```
a.c
       11  {
       12      Stu stu[5];
       13      printf("请输入5个学生的信息：姓名 性别 年龄:\n");
       14      input(stu);
       15      printf("5个学生的信息如下：\n姓名  性别  年龄\n");
       16      output(stu);
       17
       18      system("pause");
       19      return 0;
       20  }
       21  void input(Stu*stu)
       22  {
       23      int i;
b+     24      for(i=0;i<5;i++)
       25          scanf("%s%s%d",stu[i].name,stu[i].sex,&(stu[i].age));
       26  }
       27  void output(Stu*stu)
       28  {
       29      int i;
       30      for(i=0;i<5;i++)
       31          printf("%s %s %d\n",stu[i].name,stu[i].sex,stu[i].age);
       32  }
       33
       34
exec No process In:
(gdb) b 23
Breakpoint 1 at 0x125d: file a.c, line 24.
```

设置断点可以按照源代码行数设置断点

2.2.6 layout

```
(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)
```

根据命令提示，layout命令可以查看很多层次，这里试用下 layout src



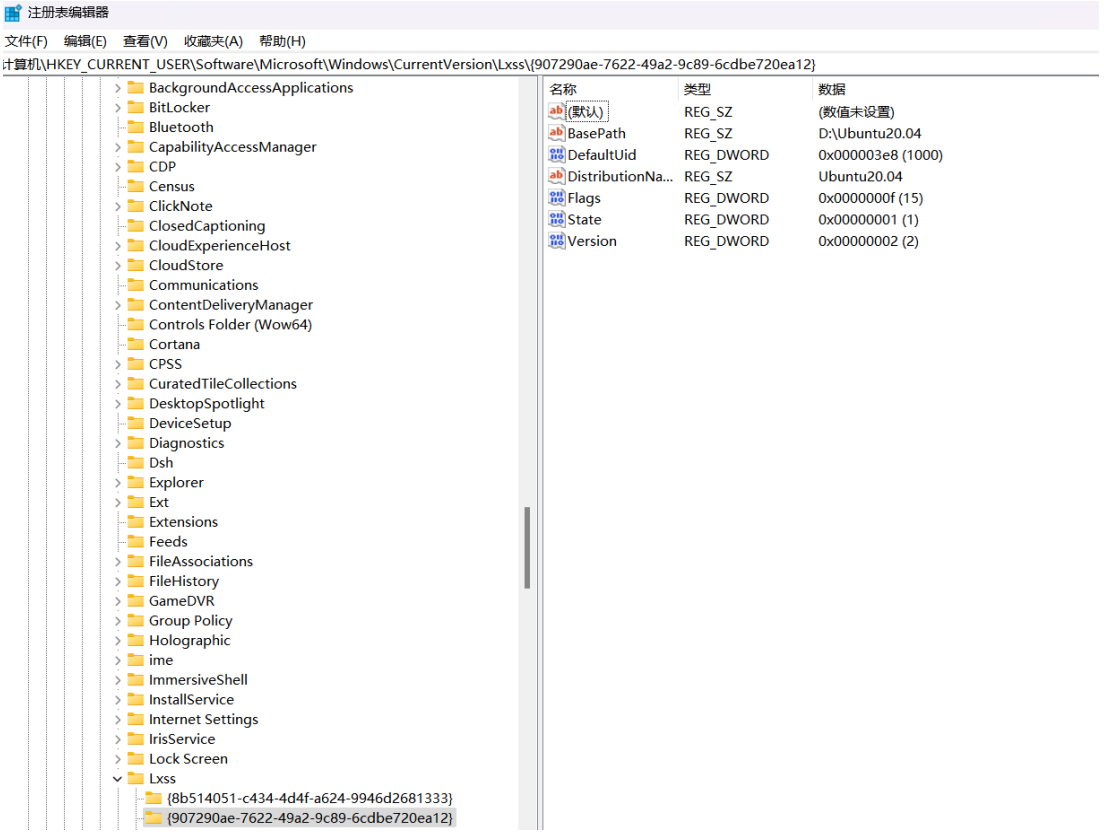
可以看到显示了可视化窗口来展示源码

3 实验中遇到的问题及解决方法

> 手动安装Ubuntu20.04后，默认的发行版名称为Ubuntu，如果尝试安装 Ubuntu-22.04 LTS 的发行版则直接打开之前安装的WSL/Ubuntu-20.04的环境，因为手动安装时安装程序查找的是相同的“Ubuntu”发行版名称。

解决办法：参考博客

- 修改注册表 HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Lxss位置下的发行版名称



> 无法apt install一些东西

这是因为未更新 Ubuntu 存储库软件包

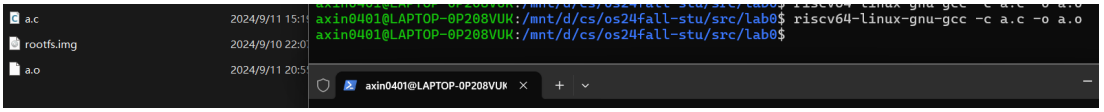
解决：

```
sudo apt update && sudo apt upgrade -y
```

4 思考题与心得体会

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

```
cd /mnt/d/cs/os24fall-stu
riscv64-linux-gnu-gcc -c a.c -o a.o
```



可见生成

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

```
riscv64-linux-gnu-objdump -d a.o #执行结束在命令行窗口即可看到
riscv64-linux-gnu-objdump -d a.o > a.txt #输出汇编到a.txt文件中
```

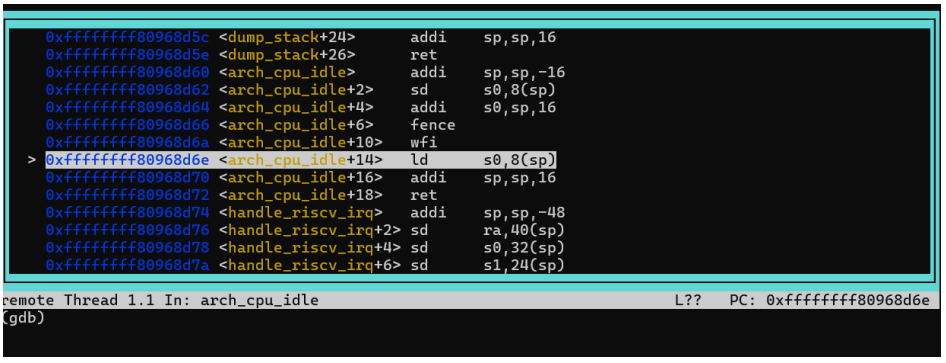
名称	修改日期	类型
a.c	2024/9/11 15:19	C 源文件
rootfs.img	2024/9/10 22:07	光盘映像文件
a.o	2024/9/11 20:55	O 文件
a.txt	2024/9/11 20:58	文本文档

```
axin0401@LAPTOP-0P208VUK:/mnt/d/cs/os24fall-stu/src/lab0$ riscv64-linux-gnu-objdu

a.o:     file format elf64-littleriscv


Disassembly of section .text:

0000000000000000 <main>:
   0:   1141                    addi    sp,sp,-16
   2:   e406                    sd      ra,8(sp)
   4:   e022                    sd      s0,0(sp)
   6:   0800                    addi    s0,sp,16
   8:   00000517                auipc   a0,0x0
   c:   00050513                mv      a0,a0
  10:   00000097                auipc   ra,0x0
  14:   000080e7                jalr    ra # 10 <main+0x10>
  18:   4781                    li      a5,0
  1a:   853e                    mv      a0,a5
  1c:   60a2                    ld      ra,8(sp)
  1e:   6402                    ld      s0,0(sp)
  20:   0141                    addi    sp,sp,16
  22:   8082                    ret
axin0401@LAPTOP-0P208VUK:/mnt/d/cs/os24fall-stu/src/lab0$
```

3. 调试Linux 时：

    1. 在 GDB 中查看汇编代码（不使用任何插件的情况下）

    ```
    1 (gdb) layout asm
    ```

    查看结束后，先按下Ctrl+X,再按下A即可退出汇编窗口

    2. 在 0x80000000 处下断点

```
(gdb) break *0x80000000
```

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

3. 查看所有已下的断点

```
(gdb) info breakpoints
(gdb) i b
```

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

4. 在 `0x80200000` 处下断点

```
(gdb) break *0x80200000
```

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

利用`i b`指令查看插入断点

```
(gdb) i b
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000080000000
2       breakpoint     keep y   0x0000000080200000
(gdb)
```

可以看到两个断点的编号为1、2（下一步要用）

5. 清除 `0x80000000` 处的断点

```
(gdb) clear *0x80000000 #根据地址来清除断点
(gdb) delete 1  #根据编号来清除断点
```

```
(gdb) clear *0x80000000
Deleted breakpoint 1
(gdb) i b
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x0000000080200000
(gdb)
```

可以看到第一个断点已被删除

6. 继续运行直到触发 `0x80200000` 处的断点

```
(gdb) continue
```

```
(gdb) continue
Continuing.

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

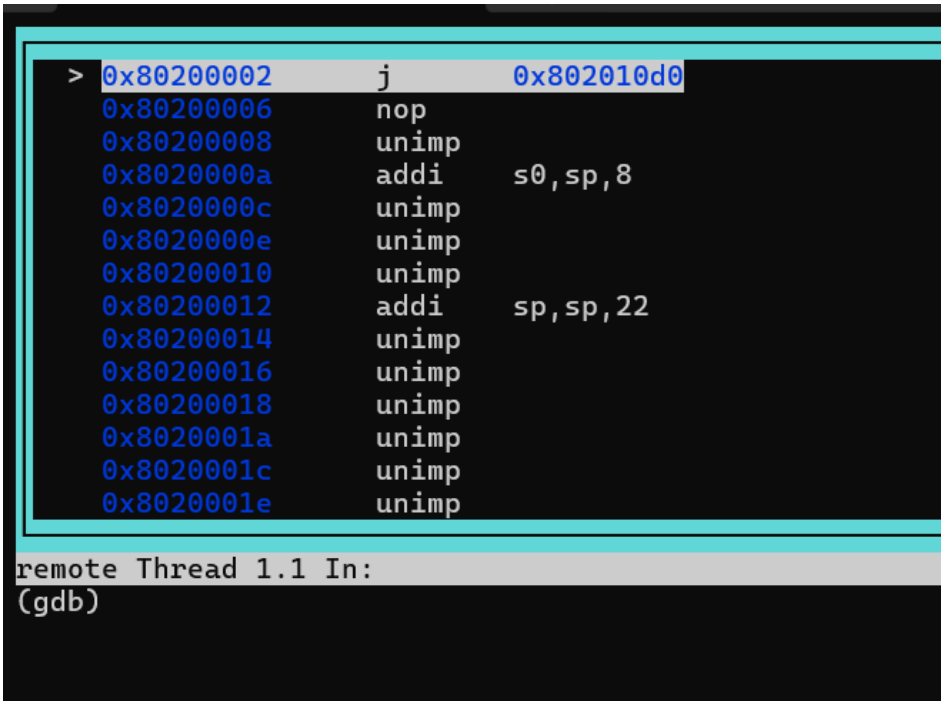
7. 单步调试一次

使用 `si` 命令来单步执行程序

```
(gdb) si  #单步调试
```

```
(gdb) si
0x0000000080200002 in ?? ()
(gdb)
```

执行`layout asm`可查看当前位置命令



8. 退出 QEMU

```
(gdb) quit
```

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

        Inferior 1 [process 1] will be detached.

Quit anyway? (y or n)
```

输入y即可退出

4. 使用 `make` 工具清除 Linux 的构建产物

```
1  sudo make clean          # 清除所有编译好的 object 文件
```

```
axin0401@LAPTOP-0P208VUK:/usr/linux-6.11-rc7$ sudo 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
axin0401@LAPTOP-0P208VUK:/usr/linux-6.11-rc7$
```

5. `vmlinux` 和 `Image` 的关系和区别是什么？

[参考1]　　[参考2]

- 关系：`vmlinux` 和 `Image` 都是Linux内核编译的结果，但它们服务于不同的目的。
- 区别：
  - **用途**：`vmlinux` 更多用于开发和调试，而 `Image` 用于实际的系统启动和运行。
  - **体积**：`vmlinux` 包含所有信息，因此体积较大；`Image` 经过使用 objcopy 取消掉 vmlinux 中的符号表等一些其他信息， 仅包含可执行的二进制数据，体积较小。
  - **调试信息**：`vmlinux` 包含调试信息，适合使用GDB等工具进行调试；`Image` 不包含这些信息，因为它是为了运行效率而优化的。