操作系统原理实验报告

- **实验名称**：编译内核/利用已有内核构建OS
- **授课教师**：张青
- **学生姓名**：林隽哲
- **学生学号**：21312450

实验要求

- 熟悉现有Linux内核的编译过程和启动过程，并在自行编译内核的基础上构建简单应用并启动；利用精简的Busybox工具集构建简单的OS，熟悉现代操作系统的构建过程。此外，熟悉编译环境、相关工具集，并能实现内核远程调试；

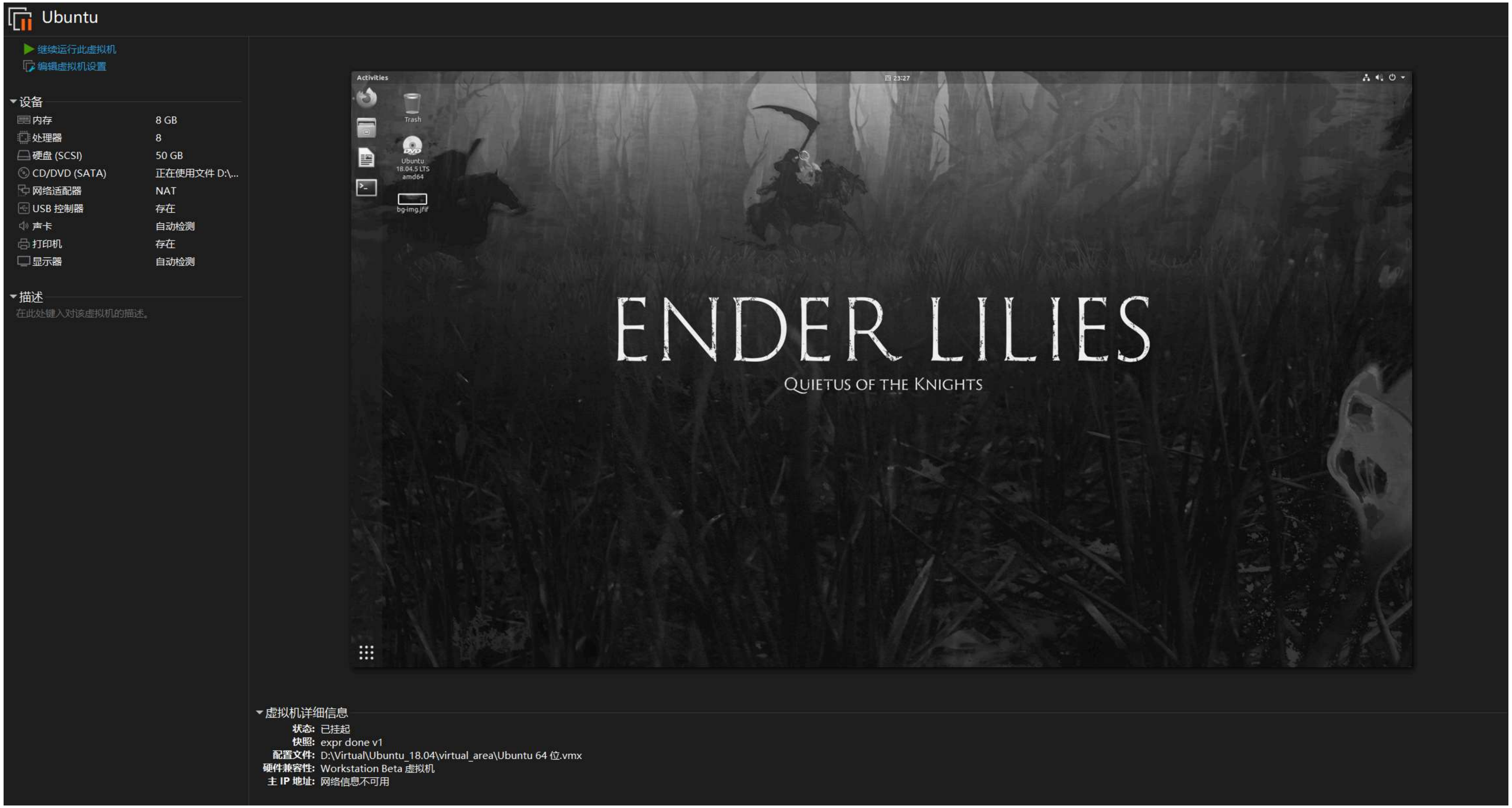
1. 独立完成实验5个部分**环境配置、编译Linux内核、Qemu启动内核并开启远程调试、制作Initramfs和编译并启动Busybox。**
2. 编写实验报告、结合实验过程来谈谈你完成实验的思路和结果，最后需要提供实验的5各部分的运行截屏来证明你完成了实验。
3. 实验不限语言，C/C++/Rust都可以。
4. 实验不限平台，Windows/Linux/MacOS都可以。
5. 实验不限CPU架构，x86/ARM/RISC-V都可以。

实验过程

环境配置

- 搭建Linux系统环境（Ubuntu 18.04）

在本次实验中，我选择使用VMware搭建Ubuntu 18.04的虚拟机环境。



设备

内存	8 GB
处理器	8
硬盘 (SCSI)	50 GB
CD/DVD (SATA)	正在使用文件 D:\...
网络适配器	NAT
USB 控制器	存在
声卡	自动检测
打印机	存在
显示器	自动检测

描述

在此处键入对该虚拟机的描述。

虚拟机详细信息

状态:	已挂起
快照:	expr done v1
配置文件:	D:\Virtual\Ubuntu_18.04\virtual_area\Ubuntu 64 位.vmx
硬件兼容性:	Workstation Beta 虚拟机
主 IP 地址:	网络信息不可用

在安装好Ubuntu 18.04后，通过安装VMware Tools来提高虚拟机的体验。

打开终端，输入以下命令：

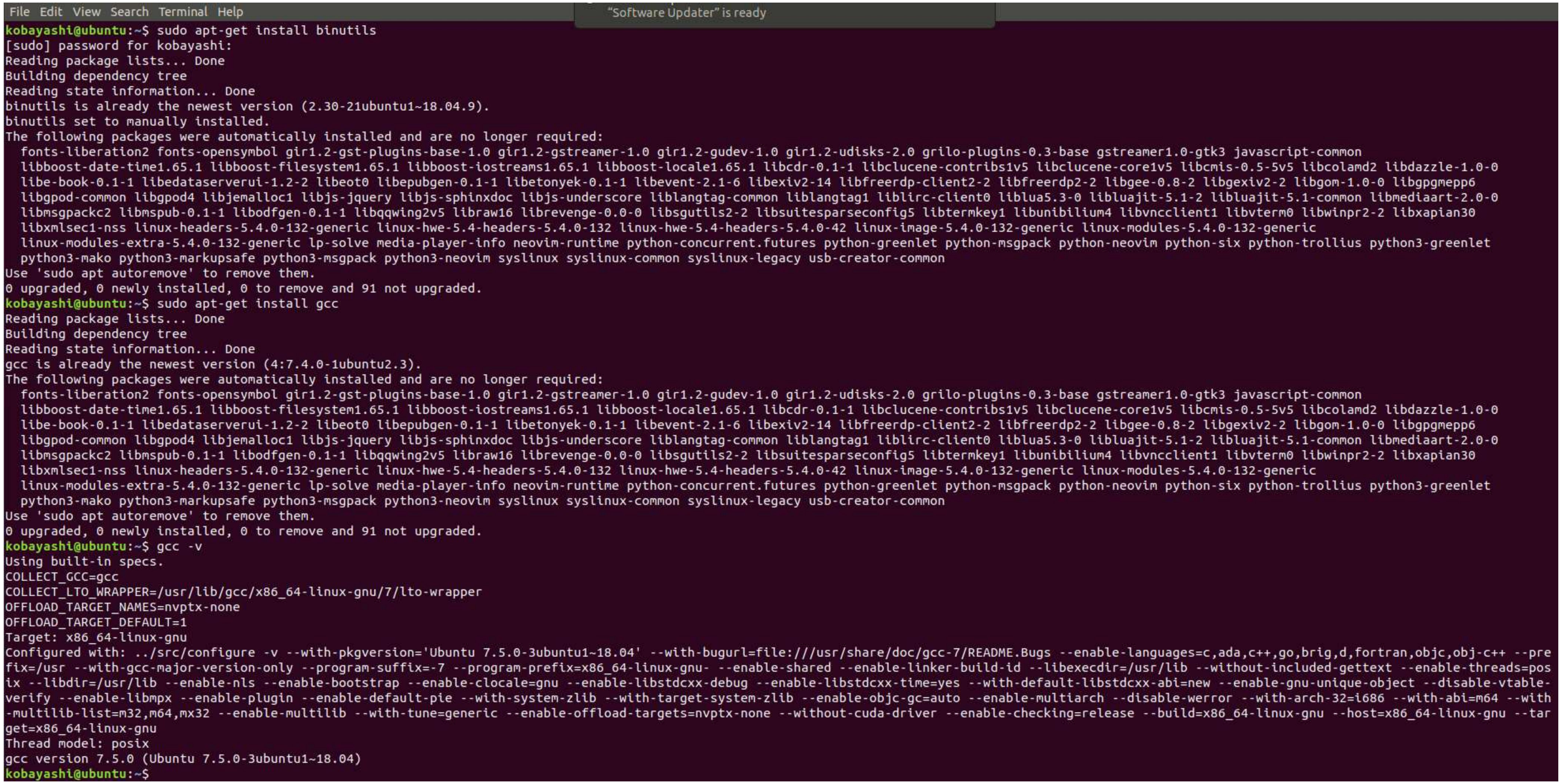
```
sudo apt-get update
sudo apt-get install open-vm-tools-desktop fuse
```

安装完成后，重启虚拟机即可。

- 安装编译所需的工具

在终端中输入以下命令：

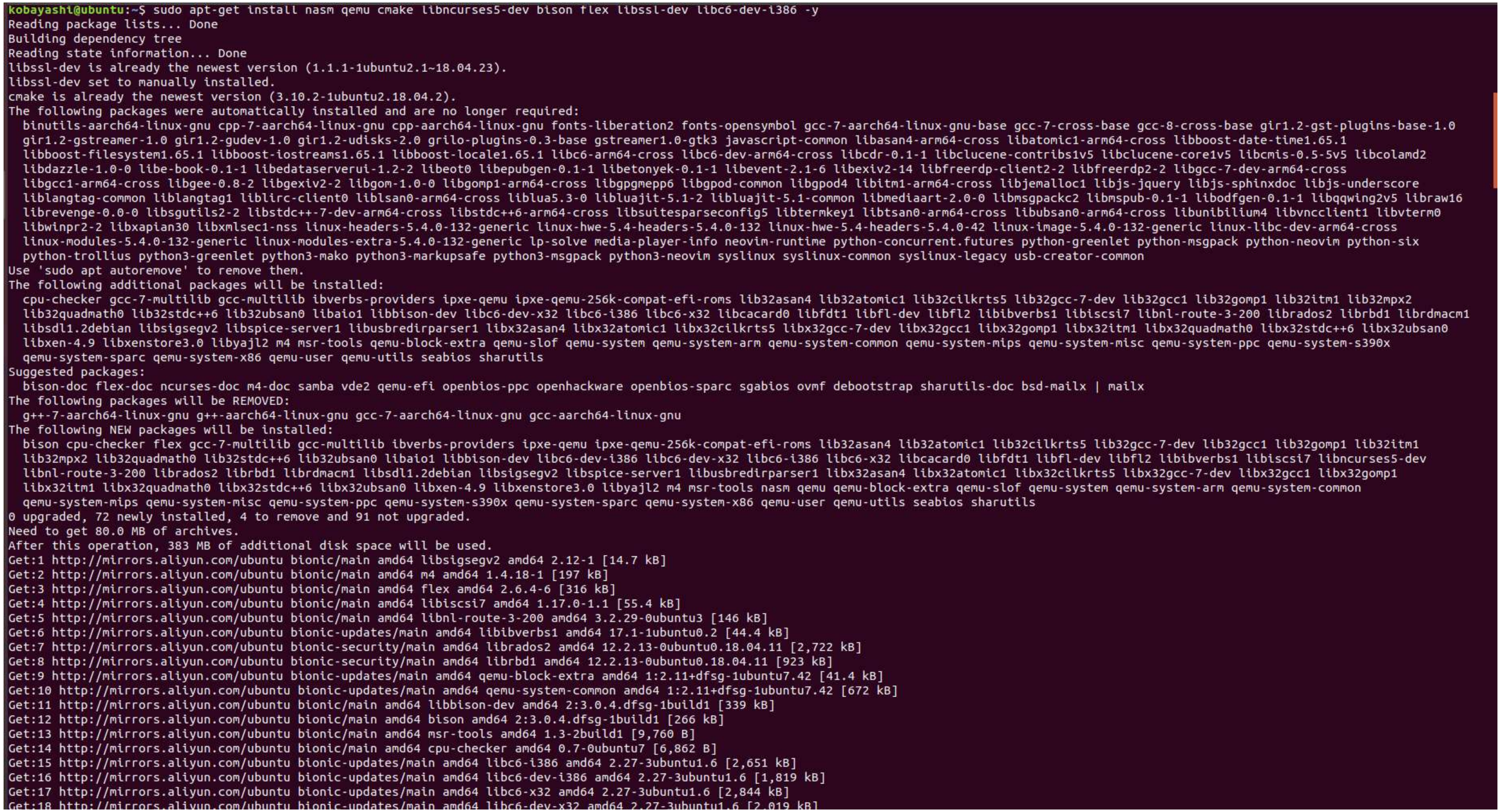
```
# GNU Binary Utilities, or (binutils), are a set of programming tools for creating and managing binary programs, object files, libraries, p
# [GNU Binutils](https://www.gnu.org/software/binutils/)
sudo apt-get install binutils
sudo apt-get install gcc

# check the version of gcc
gcc --version
```



File Edit View Search Terminal Help
"Software Updater" is ready

```
kobayashi@ubuntu:~$ sudo apt-get install binutils
[sudo] password for kobayashi:
Reading package lists... Done
Building dependency tree
Reading state information... Done
binutils is already the newest version (2.30-21ubuntu1~18.04.9).
binutils set to manually installed.
The following packages were automatically installed and are no longer required:
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 javascript-common
  libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0
  libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6
  libgpod-common libgpod4 libjemalloc1 libjs-jquery libjs-sphinxdoc libjs-underscore liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libluajit-5.1-2 libluajit-5.1-common libmediaart-2.0-0
  libmsgpackc2 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libsuitesparseconfig5 libtermkey1 libunibilium4 libvncclient1 libvterm0 libwinpr2-2 libxapian30
  libxmlsec1-nss linux-headers-5.4.0-132-generic linux-hwe-5.4-headers-5.4.0-132 linux-hwe-5.4-headers-5.4.0-42 linux-image-5.4.0-132-generic linux-modules-5.4.0-132-generic
  linux-modules-extra-5.4.0-132-generic lp-solve media-player-info neovim-runtime python-concurrent.futures python-greenlet python-msgpack python-neovim python-six python-trollius python3-greenlet
  python3-mako python3-markupsafe python3-msgpack python3-neovim syslinux syslinux-common syslinux-legacy usb-creator-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
kobayashi@ubuntu:~$ sudo apt-get install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:7.4.0-1ubuntu2.3).
The following packages were automatically installed and are no longer required:
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 javascript-common
  libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0
  libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6
  libgpod-common libgpod4 libjemalloc1 libjs-jquery libjs-sphinxdoc libjs-underscore liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libluajit-5.1-2 libluajit-5.1-common libmediaart-2.0-0
  libmsgpackc2 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libsuitesparseconfig5 libtermkey1 libunibilium4 libvncclient1 libvterm0 libwinpr2-2 libxapian30
  libxmlsec1-nss linux-headers-5.4.0-132-generic linux-hwe-5.4-headers-5.4.0-132 linux-hwe-5.4-headers-5.4.0-42 linux-image-5.4.0-132-generic linux-modules-5.4.0-132-generic
  linux-modules-extra-5.4.0-132-generic lp-solve media-player-info neovim-runtime python-concurrent.futures python-greenlet python-msgpack python-neovim python-six python-trollius python3-greenlet
  python3-mako python3-markupsafe python3-msgpack python3-neovim syslinux syslinux-common syslinux-legacy usb-creator-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
kobayashi@ubuntu:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-3ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --pre
fix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=pos
ix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-
verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with
-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --tar
get=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
kobayashi@ubuntu:~$
```

```
# Netwide Assembler (NASM) is an assembler and disassembler for the Intel x86 architecture. It can be used to write 16-bit, 32-bit (IA-32)
# [Netwide Assembler](https://www.nasm.us/)
sudo apt-get install nasm

# Quick Emulator (QEMU) is a free and open-source emulator. It emulates a computer's processor through dynamic binary translation and provi
# QEMU supports the emulation of various architectures, including x86, x86-64, ARM, MIPS, PowerPC, SPARC, and RISC-V, among others.
# [QUEM - Wikipedia](https://en.wikipedia.org/wiki/QEMU)
# [Quick Emulator](https://www.qemu.org/)
sudo apt-get install qemu

sudo apt-get install cmake

# The ncureses library routines are a terminal-independent method of updating character screens with reasonable optimization.
# This package contains the header files, static libraries and symbolic links that developers using ncurses will need.
# [ncurses](https://invisible-island.net/ncurses/)
sudo apt-get install libncurses5-dev

# GNU Bison, commonly known as Bison, is a parser generator that is part of GNU Project.
# [GNU Bison - Wikipedia](https://en.wikipedia.org/wiki/GNU_Bison)
# [GNU Bison](https://www.gnu.org/software/bison/)
sudo apt-get install bison

# Fast lexical analyzer generator (FLEX) is a computer program that generates lexical analyzers (also known as "scanners" or "lexers"). It
# [Flex - Wikipedia](https://en.wikipedia.org/wiki/Flex_(lexical_analyser_generator))
sudo apt-get install flex

# This package is part of the OpenSSL project's implementation of the SSL and TLS cryptographic protocols for secure communication over the
# [OpenSSL](https://www.openssl.org/)
sudo apt-get install libssl-dev

# libc is the C library; basically, it contains all of the system functions that most (if not all) programs need to run on Linux.
# libc6 and glibc (GNU C Library) are the same version of libc
# [GNU C Library - Wikipedia](https://en.wikipedia.org/wiki/GNU_C_Library)
# [What's the difference between libc6 and glibc?](https://linux-m68k.org/faq/glibcinfo.html)
sudo apt-get install libc6-dev-i386
```



```
kobayashi@ubuntu:~$ sudo apt-get install nasm qemu cmake libncurses5-dev bison flex libssl-dev libc6-dev-i386 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
libssl-dev is already the newest version (1.1.1-1ubuntu2.1~18.04.23).
libssl-dev set to manually installed.
cmake is already the newest version (3.10.2-1ubuntu2.18.04.2).
The following packages were automatically installed and are no longer required:
  binutils-aarch64-linux-gnu cpp-7-aarch64-linux-gnu cpp-aarch64-linux-gnu fonts-liberation2 fonts-opensymbol gcc-7-aarch64-linux-gnu-base gcc-7-cross-base gcc-8-cross-base gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 javascript-common libasan4-arm64-cross libatomic1-arm64-cross libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libc6-arm64-cross libc6-dev-arm64-cross libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcolamd2
  libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgcc-7-dev-arm64-cross
  libgcc1-arm64-cross libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgomp1-arm64-cross libgpgmepp6 libgpod-common libgpod4 libitm1-arm64-cross libjemalloc1 libjs-jquery libjs-sphinxdoc libjs-underscore
  liblangtag-common liblangtag1 liblirc-client0 liblsan0-arm64-cross liblua5.3-0 libluajit-5.1-2 libluajit-5.1-common libmediaart-2.0-0 libmsgpackc2 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16
  librevenge-0.0-0 libsgutils2-2 libstdc++-7-dev-arm64-cross libstdc++6-arm64-cross libsuitesparseconfig5 libtermkey1 libtsan0-arm64-cross libubsan0-arm64-cross libunibilium4 libvncclient1 libvterm0
  libwinpr2-2 libxapian30 libxmlsec1-nss linux-headers-5.4.0-132-generic linux-hwe-5.4-headers-5.4.0-132 linux-hwe-5.4-headers-5.4.0-42 linux-image-5.4.0-132-generic linux-libc-dev-arm64-cross
  linux-modules-5.4.0-132-generic linux-modules-extra-5.4.0-132-generic lp-solve media-player-info neovim-runtime python-concurrent.futures python-greenlet python-msgpack python-neovim python-six
  python-trollius python3-greenlet python3-mako python3-markupsafe python3-msgpack python3-neovim syslinux syslinux-common syslinux-legacy usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  cpu-checker gcc-7-multilib gcc-multilib ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms lib32asan4 lib32atomic1 lib32cilkrts5 lib32gcc-7-dev lib32gcc1 lib32gomp1 lib32itm1 lib32mpx2
  lib32quadmath0 lib32stdc++6 lib32ubsan0 libaio1 libbison-dev libc6-dev-x32 libc6-i386 libc6-x32 libcacard0 libfdt1 libfl-dev libfl2 libibverbs1 libiscsi7 libnl-route-3-200 librados2 librbd1 librdmacm1
  libsdl1.2debian libsigsegv2 libspice-server1 libusbredirparser1 libx32asan4 libx32atomic1 libx32cilkrts5 libx32gcc-7-dev libx32gcc1 libx32gomp1 libx32itm1 libx32quadmath0 libx32stdc++6 libx32ubsan0
  libxen-4.9 libxenstore3.0 libyajl2 m4 msr-tools qemu-block-extra qemu-slof qemu-system qemu-system-arm qemu-system-common qemu-system-mips qemu-system-misc qemu-system-ppc qemu-system-s390x
  qemu-system-sparc qemu-system-x86 qemu-user qemu-utils seabios sharutils
Suggested packages:
  bison-doc flex-doc ncurses-doc m4-doc samba vde2 qemu-efi openbios-ppc openhackware openbios-sparc sgabios ovmf debootstrap sharutils-doc bsd-mailx | mailx
The following packages will be REMOVED:
  g++-7-aarch64-linux-gnu g++-aarch64-linux-gnu gcc-7-aarch64-linux-gnu gcc-aarch64-linux-gnu
The following NEW packages will be installed:
  bison cpu-checker flex gcc-7-multilib gcc-multilib ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms lib32asan4 lib32atomic1 lib32cilkrts5 lib32gcc-7-dev lib32gcc1 lib32gomp1 lib32itm1
  lib32mpx2 lib32quadmath0 lib32stdc++6 lib32ubsan0 libaio1 libbison-dev libc6-dev-i386 libc6-dev-x32 libc6-i386 libc6-x32 libcacard0 libfdt1 libfl-dev libfl2 libibverbs1 libiscsi7 libncurses5-dev
  libnl-route-3-200 librados2 librbd1 librdmacm1 libsdl1.2debian libsigsegv2 libspice-server1 libusbredirparser1 libx32asan4 libx32atomic1 libx32cilkrts5 libx32gcc-7-dev libx32gcc1 libx32gomp1
  libx32itm1 libx32quadmath0 libx32stdc++6 libx32ubsan0 libxen-4.9 libxenstore3.0 libyajl2 m4 msr-tools nasm qemu qemu-block-extra qemu-slof qemu-system qemu-system-arm qemu-system-common
  qemu-system-mips qemu-system-misc qemu-system-ppc qemu-system-s390x qemu-system-sparc qemu-system-x86 qemu-user qemu-utils seabios sharutils
0 upgraded, 72 newly installed, 4 to remove and 91 not upgraded.
Need to get 80.0 MB of archives.
After this operation, 383 MB of additional disk space will be used.
Get:1 http://mirrors.aliyun.com/ubuntu bionic/main amd64 libsigsegv2 amd64 2.12-1 [14.7 kB]
Get:2 http://mirrors.aliyun.com/ubuntu bionic/main amd64 m4 amd64 1.4.18-1 [197 kB]
Get:3 http://mirrors.aliyun.com/ubuntu bionic/main amd64 flex amd64 2.6.4-6 [316 kB]
Get:4 http://mirrors.aliyun.com/ubuntu bionic/main amd64 libiscsi7 amd64 1.17.0-1.1 [55.4 kB]
Get:5 http://mirrors.aliyun.com/ubuntu bionic/main amd64 libnl-route-3-200 amd64 3.2.29-0ubuntu3 [146 kB]
Get:6 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 libibverbs1 amd64 17.1-1ubuntu0.2 [44.4 kB]
Get:7 http://mirrors.aliyun.com/ubuntu bionic-security/main amd64 librados2 amd64 12.2.13-0ubuntu0.18.04.11 [2,722 kB]
Get:8 http://mirrors.aliyun.com/ubuntu bionic-security/main amd64 librbd1 amd64 12.2.13-0ubuntu0.18.04.11 [923 kB]
Get:9 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 qemu-block-extra amd64 1:2.11+dfsg-1ubuntu7.42 [41.4 kB]
Get:10 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 qemu-system-common amd64 1:2.11+dfsg-1ubuntu7.42 [672 kB]
Get:11 http://mirrors.aliyun.com/ubuntu bionic/main amd64 libbison-dev amd64 2:3.0.4.dfsg-1build1 [339 kB]
Get:12 http://mirrors.aliyun.com/ubuntu bionic/main amd64 bison amd64 2:3.0.4.dfsg-1build1 [266 kB]
Get:13 http://mirrors.aliyun.com/ubuntu bionic/main amd64 msr-tools amd64 1.3-2build1 [9,760 B]
Get:14 http://mirrors.aliyun.com/ubuntu bionic/main amd64 cpu-checker amd64 0.7-0ubuntu7 [6,862 B]
Get:15 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 libc6-i386 amd64 2.27-3ubuntu1.6 [2,651 kB]
Get:16 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 libc6-dev-i386 amd64 2.27-3ubuntu1.6 [1,819 kB]
Get:17 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 libc6-x32 amd64 2.27-3ubuntu1.6 [2,844 kB]
Get:18 http://mirrors.aliyun.com/ubuntu bionic-updates/main amd64 libc6-dev-x32 amd64 2.27-3ubuntu1.6 [2,019 kB]
```

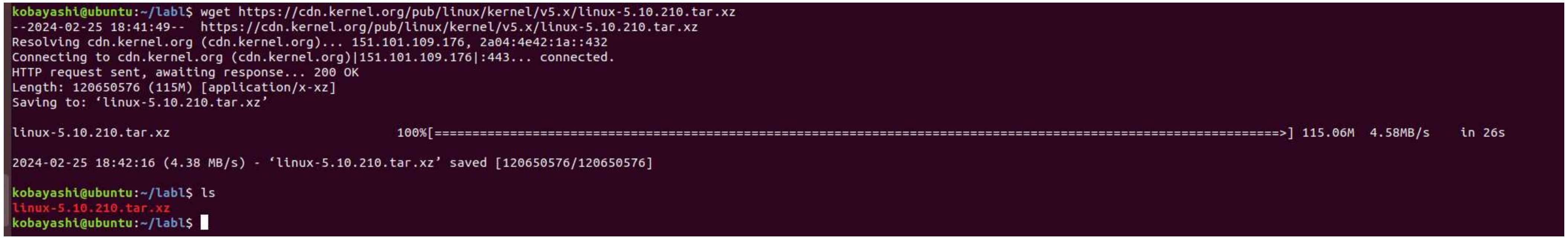
编译Linux内核

- 创建实验目录

```
mkdir ~/labl && cd ~/labl
```

- 下载内核5.10到实验目录下

```
wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.10.211.tar.xz
```



```
kobayashi@ubuntu:~/labl$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.10.210.tar.xz
--2024-02-25 18:41:49--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.10.210.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.109.176, 2a04:4e42:1a::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.109.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 120650576 (115M) [application/x-xz]
Saving to: ‘linux-5.10.210.tar.xz’

linux-5.10.210.tar.xz                      100%[===============================================================================================================>] 115.06M  4.58MB/s    in 26s

2024-02-25 18:42:16 (4.38 MB/s) - ‘linux-5.10.210.tar.xz’ saved [120650576/120650576]

kobayashi@ubuntu:~/labl$ ls
linux-5.10.210.tar.xz
kobayashi@ubuntu:~/labl$ 
```

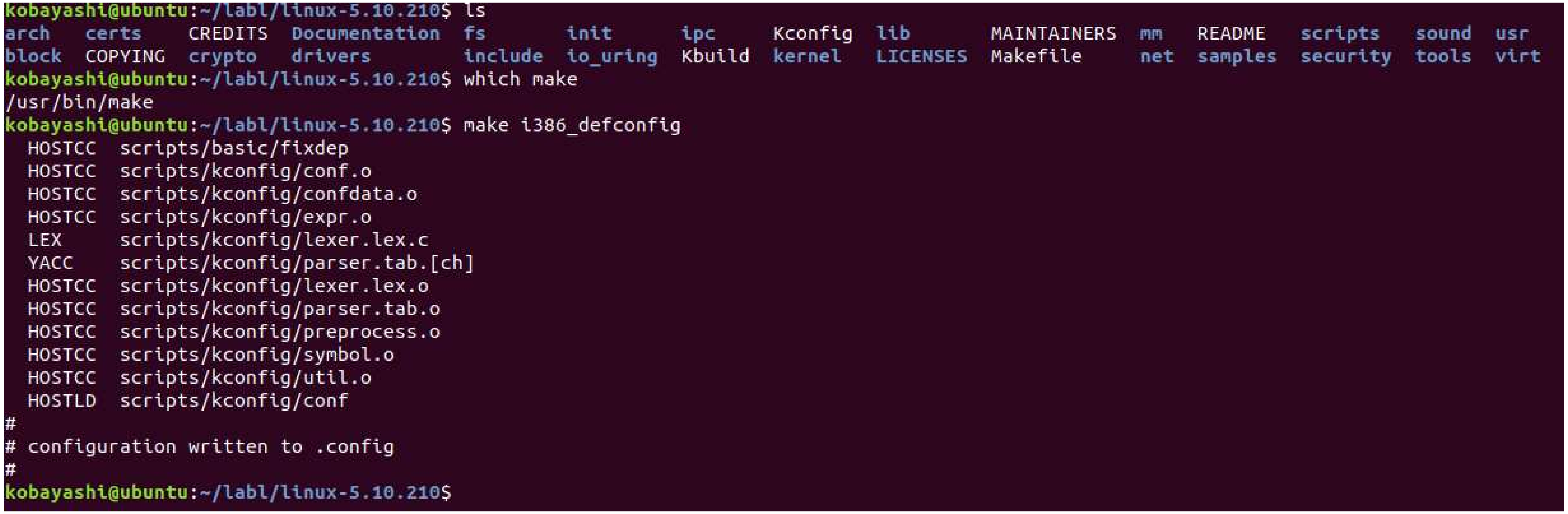
- 解压并进入内核目录

```
xz -d linux-5.10.211.tar.xz
tar -xvf linux-5.10.211.tar
cd linux-5.10.211
```

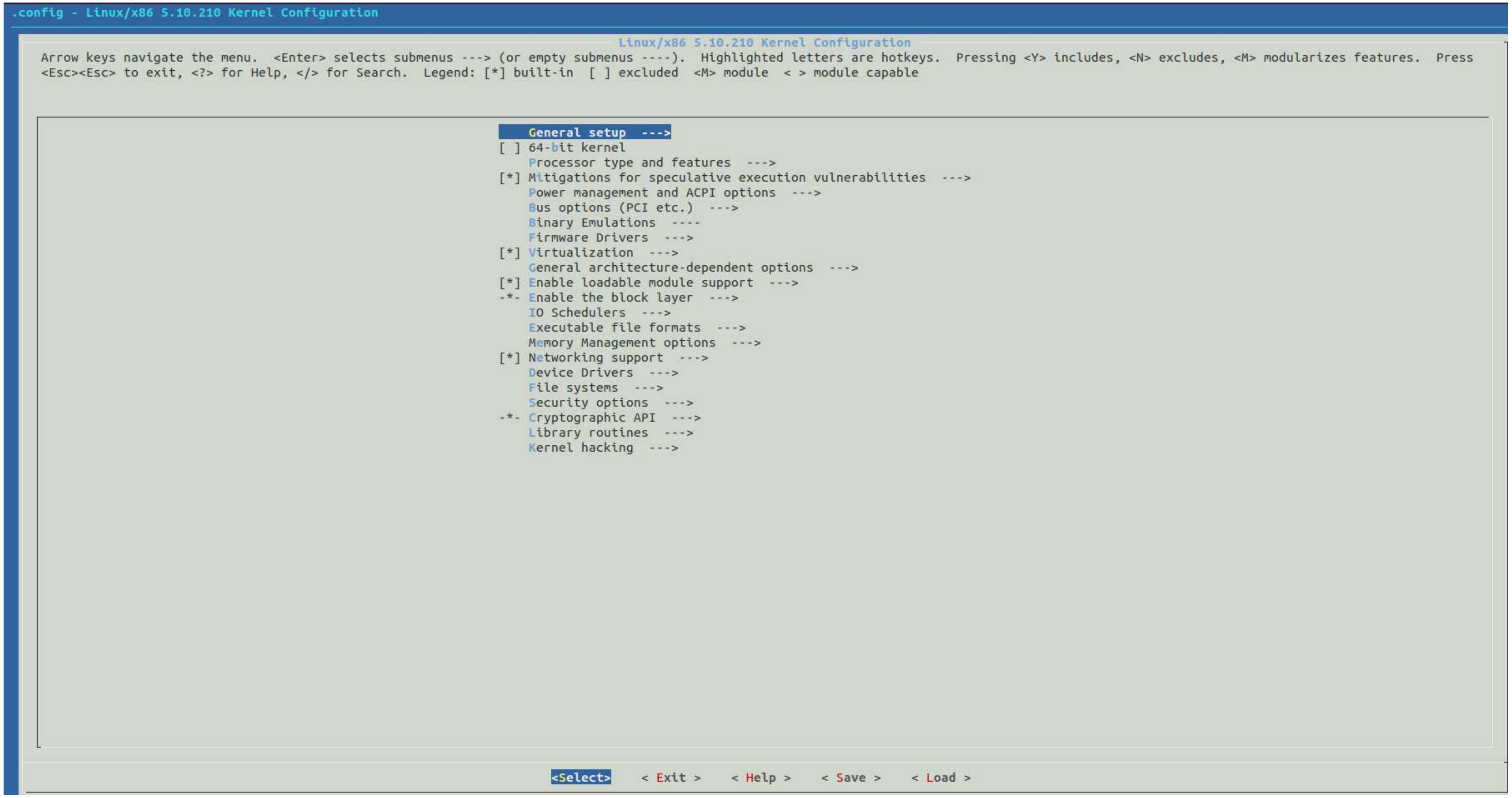
- 编译Linux内核

```
make i386_defconfig # generate the default configuration file for the x86 architecture
make menuconfig # configure the kernel
```

在 `menuconfig` 命令打开的界面中依次选择**Kernel hacking -> Compile-time checks and compiler options**，将**Compile the kernel with debug info**选项打开，保存并退出。该选项打开后，编译内核时将会生成调试信息。



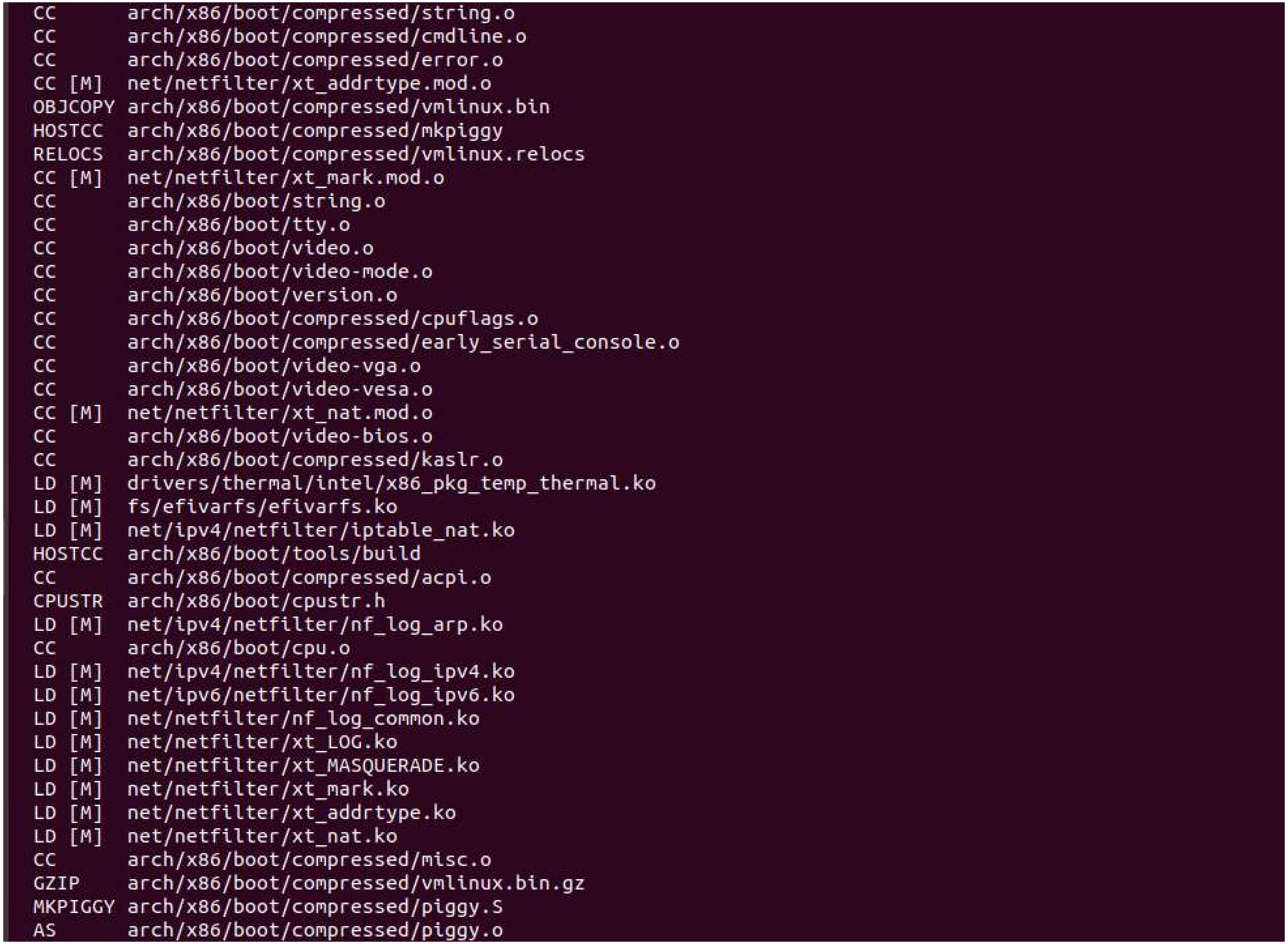
```
kobayashi@ubuntu:~/lab1/linux-5.10.210$ ls
arch   certs    CREDITS  Documentation  fs       init      ipc     Kconfig  lib       MAINTAINERS  mm   README   scripts   sound  usr
block  COPYING  crypto   drivers        include  io_uring  Kbuild  kernel   LICENSES  Makefile     net  samples  security  tools  virt
kobayashi@ubuntu:~/lab1/linux-5.10.210$ which make
/usr/bin/make
kobayashi@ubuntu:~/lab1/linux-5.10.210$ make i386_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/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
kobayashi@ubuntu:~/lab1/linux-5.10.210$
```



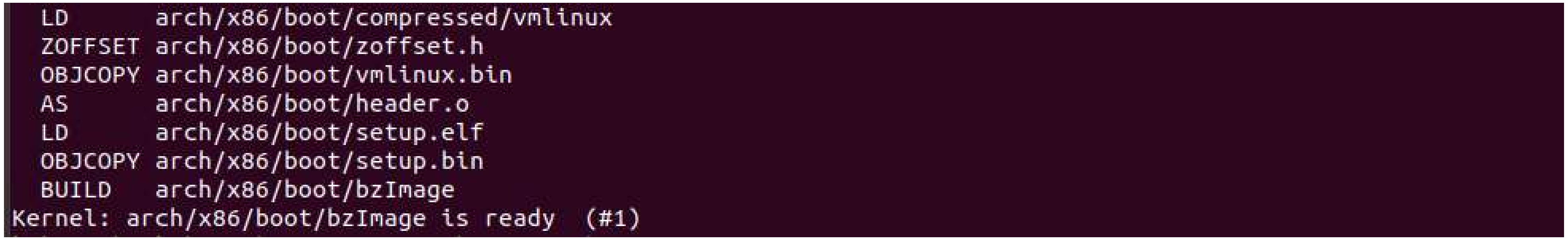
```
.config - Linux/x86 5.10.210 Kernel Configuration
                              Linux/x86 5.10.210 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                                    General setup  --->
                                [ ] 64-bit kernel
                                    Processor type and features  --->
                                [*] Mitigations for speculative execution vulnerabilities  --->
                                    Power management and ACPI options  --->
                                    Bus options (PCI etc.)  --->
                                    Binary Emulations  ----
                                    Firmware Drivers  --->
                                [*] Virtualization  --->
                                    General architecture-dependent options  --->
                                [*] Enable loadable module support  --->
                                -*- Enable the block layer  --->
                                    IO Schedulers  --->
                                    Executable file formats  --->
                                    Memory Management options  --->
                                [*] Networking support  --->
                                    Device Drivers  --->
                                    File systems  --->
                                    Security options  --->
                                -*- Cryptographic API  --->
                                    Library routines  --->
                                    Kernel hacking  --->

                          <Select>    < Exit >    < Help >    < Save >    < Load >
```

- 编译内核

```
make -j4 # compile the kernel
```



```
  CC      arch/x86/boot/compressed/string.o
  CC      arch/x86/boot/compressed/cmdline.o
  CC      arch/x86/boot/compressed/error.o
  CC [M]  net/netfilter/xt_addrtype.mod.o
  OBJCOPY arch/x86/boot/compressed/vmlinux.bin
  HOSTCC  arch/x86/boot/compressed/mkpiggy
  RELOCS  arch/x86/boot/compressed/vmlinux.relocs
  CC [M]  net/netfilter/xt_mark.mod.o
  CC      arch/x86/boot/string.o
  CC      arch/x86/boot/tty.o
  CC      arch/x86/boot/video.o
  CC      arch/x86/boot/video-mode.o
  CC      arch/x86/boot/version.o
  CC      arch/x86/boot/compressed/cpuflags.o
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      arch/x86/boot/video-vga.o
  CC      arch/x86/boot/video-vesa.o
  CC [M]  net/netfilter/xt_nat.mod.o
  CC      arch/x86/boot/video-bios.o
  CC      arch/x86/boot/compressed/kaslr.o
  LD [M]  drivers/thermal/intel/x86_pkg_temp_thermal.ko
  LD [M]  fs/efivarfs/efivarfs.ko
  LD [M]  net/ipv4/netfilter/iptable_nat.ko
  HOSTCC  arch/x86/boot/tools/build
  CC      arch/x86/boot/compressed/acpi.o
  CPUSTR  arch/x86/boot/cpustr.h
  LD [M]  net/ipv4/netfilter/nf_log_arp.ko
  CC      arch/x86/boot/cpu.o
  LD [M]  net/ipv4/netfilter/nf_log_ipv4.ko
  LD [M]  net/ipv6/netfilter/nf_log_ipv6.ko
  LD [M]  net/netfilter/nf_log_common.ko
  LD [M]  net/netfilter/xt_LOG.ko
  LD [M]  net/netfilter/xt_MASQUERADE.ko
  LD [M]  net/netfilter/xt_mark.ko
  LD [M]  net/netfilter/xt_addrtype.ko
  LD [M]  net/netfilter/xt_nat.ko
  CC      arch/x86/boot/compressed/misc.o
  GZIP    arch/x86/boot/compressed/vmlinux.bin.gz
  MKPIGGY arch/x86/boot/compressed/piggy.S
  AS      arch/x86/boot/compressed/piggy.o
```



```
  LD      arch/x86/boot/compressed/vmlinux
  ZOFFSET arch/x86/boot/zoffset.h
  OBJCOPY arch/x86/boot/vmlinux.bin
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready  (#1)
```

- 编译完成后，检查Linux压缩镜像 `linux-5.10.211/arch/x86/boot/bzImage` 和符号表 `linux-5.10.211/vmlinux` 是否生成。

启动内核并开启远程调试

- 进入实验目录

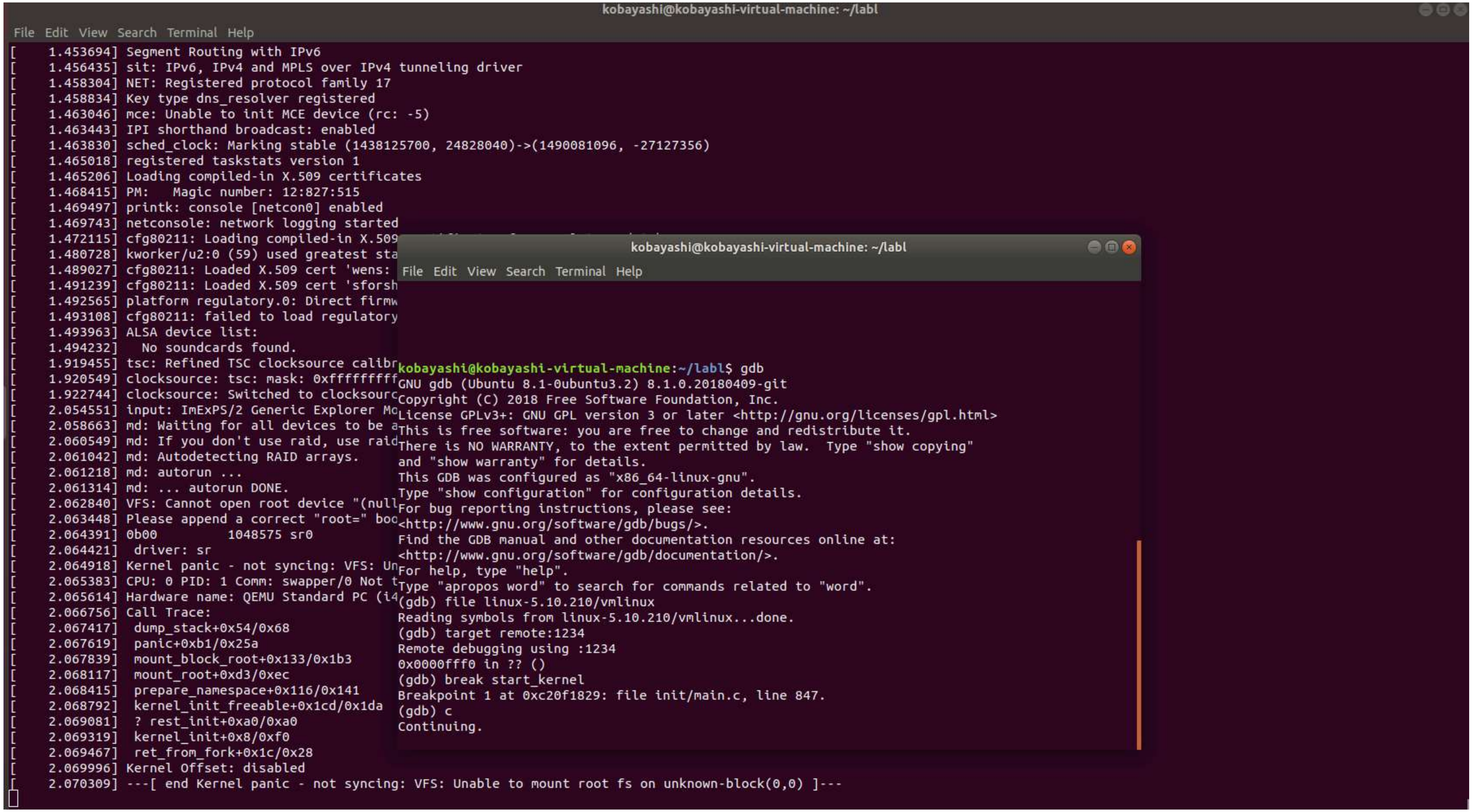
```
# cd to the experiment directory
cd ~/labl
```

- 使用qemu启动内核并开启远程调试

```
qemu-system-i386 -kernel linux-5.10.211/arch/x86/boot/bzImage -s -S -append "console=ttyS0" -nographic
# argument:
# -kernel bzImage
    # Use bzImage as kernel image. The kernel can be either a Linux kernel or in multiboot format.
# -s
    # Shorthand for -gdb tcp::1234, i.e. open a gdbserver on TCP port 1234 (see the GDB usage cahpter in the System Emulation User's Guide)
# -S
    # Do not start CPU at startup (you must type 'c' in the monitor).
# -append cmdline
    # Use cmdline as kernel command line
# -nographic
    # Normally, if QEMU is compiled wit graphical window support, it displays output such as guest grahics, guest console, and the QEMU mor
# [qemu-system-i386 arguments](https://explainshell.com/explain/1/qemu-system-i386)
```

- 在另一个终端中，使用gdb连接qemu

```
gdb # start gdb

# under gdb
file linux-5.10.211/vmlinux # load the kernel symbol file
target remote :1234 # connect to the qemu gdbserver
break start_kernel # set a breakpoint at the start_kernel function
c # continue the execution of the kernel
```


kobayashi@kobayashi-virtual-machine: ~/labl
File Edit View Search Terminal Help

```
[    1.453694] Segment Routing with IPv6
[    1.456435] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    1.458304] NET: Registered protocol family 17
[    1.458834] Key type dns_resolver registered
[    1.463046] mce: Unable to init MCE device (rc: -5)
[    1.463443] IPI shorthand broadcast: enabled
[    1.463830] sched_clock: Marking stable (1438125700, 24828040)->(1490081096, -27127356)
[    1.465018] registered taskstats version 1
[    1.465206] Loading compiled-in X.509 certificates
[    1.468415] PM:   Magic number: 12:827:515
[    1.469497] printk: console [netcon0] enabled
[    1.469743] netconsole: network logging started
[    1.472115] cfg80211: Loading compiled-in X.509
[    1.480728] kworker/u2:0 (59) used greatest sta
[    1.489027] cfg80211: Loaded X.509 cert 'wens:
[    1.491239] cfg80211: Loaded X.509 cert 'sforsh
[    1.492565] platform regulatory.0: Direct firmw
[    1.493108] cfg80211: failed to load regulatory
[    1.493963] ALSA device list:
[    1.494232]   No soundcards found.
[    1.919455] tsc: Refined TSC clocksource calibr
[    1.920549] clocksource: tsc: mask: 0xffffffff
[    1.922744] clocksource: Switched to clocksourc
[    2.054551] input: ImExPS/2 Generic Explorer Mo
[    2.058663] md: Waiting for all devices to be a
[    2.060549] md: If you don't use raid, use raid
[    2.061042] md: Autodetecting RAID arrays.
[    2.061218] md: autorun ...
[    2.061314] md: ... autorun DONE.
[    2.062840] VFS: Cannot open root device "(null
[    2.063448] Please append a correct "root=" boo
[    2.064391] 0b00         1048575 sr0
[    2.064421]  driver: sr
[    2.064918] Kernel panic - not syncing: VFS: Un
[    2.065383] CPU: 0 PID: 1 Comm: swapper/0 Not t
[    2.065614] Hardware name: QEMU Standard PC (i4
[    2.066756] Call Trace:
[    2.067417]  dump_stack+0x54/0x68
[    2.067619]  panic+0xb1/0x25a
[    2.067839]  mount_block_root+0x133/0x1b3
[    2.068117]  mount_root+0xd3/0xec
[    2.068415]  prepare_namespace+0x116/0x141
[    2.068792]  kernel_init_freeable+0x1cd/0x1da
[    2.069081]  ? rest_init+0xa0/0xa0
[    2.069319]  kernel_init+0x8/0xf0
[    2.069467]  ret_from_fork+0x1c/0x28
[    2.069996] Kernel Offset: disabled
[    2.070309] ---[ end Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0) ]---
```

```
kobayashi@kobayashi-virtual-machine:~/labl$ gdb
GNU gdb (Ubuntu 8.1-0ubuntu3.2) 8.1.0.20180409-git
Copyright (C) 2018 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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".
(gdb) file linux-5.10.210/vmlinux
Reading symbols from linux-5.10.210/vmlinux...done.
(gdb) target remote:1234
Remote debugging using :1234
0x0000fff0 in ?? ()
(gdb) break start_kernel
Breakpoint 1 at 0xc20f1829: file init/main.c, line 847.
(gdb) c
Continuing.
```

- 在qemu虚拟机中运行的Linux系统能成功启动，并且最终以 `Kernel panic` 的形式停止。这是因为启动系统的时候只指定了 `bzImage (big zImage)`，没有指定 `initrd` 文件，系统无法 `mount` 上 `initrd (init ram disk)` 及其 `initramfs (initial ram file system)`。

制作Initramfs

- 先来了解一下 `initrd` 和 `initramfs`

`initrd` 或者 `init ram disk` 是指在启动阶段被Linux内核调用的临时文件系统，用于根目录被挂在之前的准备工作。

同其他Unix系统，Linux首先要将内核加载到内存。 `initrd` 通常被压缩成 `gzip` 类型，引导时由 `bootloader` （如LILO、GRUB）来告知核心 `initrd` 的位置，使其被核心访问，挂载成一个loop类型文件。

`initramfs` 是 `initrd` 的替代品，它的功能类似 `initrd` ，但是它基于 `CPIO` 格式，无需挂载就可以展开成一个文件系统。内核启动时将 `initramfs` 挂载为 `rootfs` ，并执行其中的 `/init` 程序。如果其中没有 `/init` 程序，则会正常地挂载根文件系统并执行 `/sbin/init` 。

more about `initrd` and `initramfs` : [https://en.wikipedia.org/wiki/Initial_ramdisk]

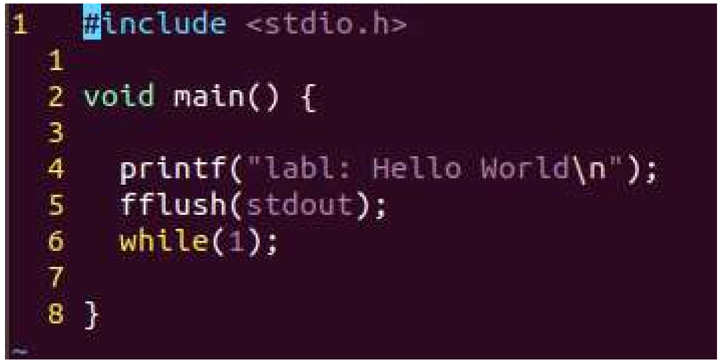
- 进入实验文件夹

```
cd ~/labl
```

- 编写 Hello world 程序

在前面调试内核中，我们已经准备了一个Linux启动环境，但是缺少一个 `initramfs` 文件。我们可以做一个最简单的 `Hello World initramfs` ，来直观地理解initramfs的作用。

```c
#include <stdio.h>

void main() {
    print("lab1:Hellow World!\n");
    fflush(stdout);
    // keep the process in user space after printing
    while(1);
}
```



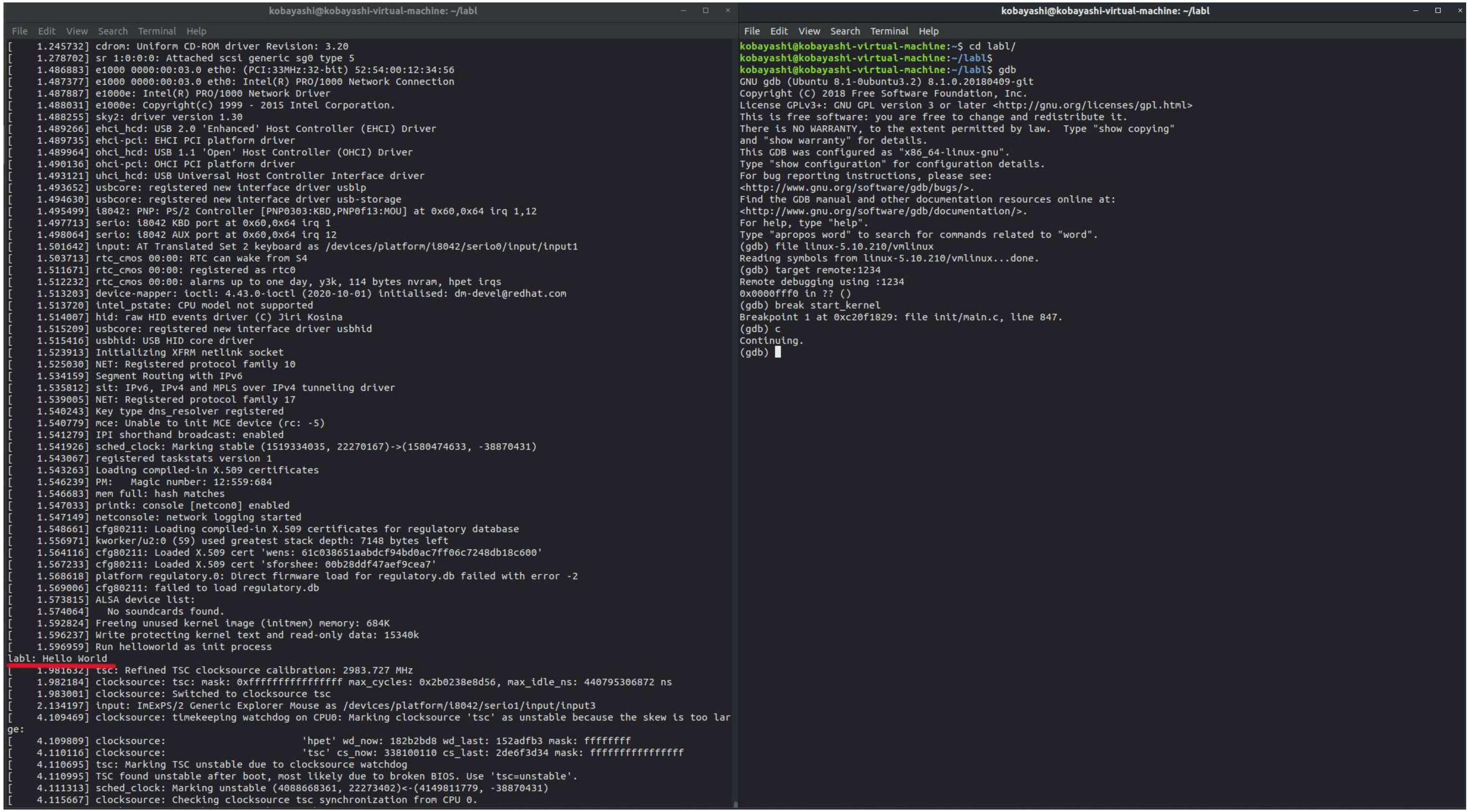
```c
#include <stdio.h>

void main() {

  printf("labl: Hello World\n");
  fflush(stdout);
  while(1);

}
```

上述文件保存在 `~/labl/helloworld.c` 中，然后将上面代码编译成32位的可执行文件。

```
gcc -o helloworld -m32 -static helloworld.c
# argument:
# -o file
    # Place the output into file file. This applies to whatever sort of output is being produced, whether it be an executable file, an obje
# -m32
    # Generate code for a 32-bit environment. The 32-bit environment sets int, long and pointer to 32 bits and generates code that runs on
# -static
    # On systems that support dynamic linking, this prevents linking with the shared libraries. On other systems, this option has no effect
```

- 加载 initramfs

- 用cpio打包initramfs。（initramfs基于CPIO格式，无需挂在就可以展开成一个文件系统）
  - `echo helloworld | cpio -o --format=newc > initramfs`
- 启动内核，并加载initramfs
  - `qemu-system-i386 -kernel linux-5.10.211/arch/x86/boot/bzImage -initrd initramfs -s -S -append "console=ttyS0" -nographic`
- 重复上面的gdb的调试过程，可以看见gdb中输出了 `lab1:Hellow World!` ，说明initramfs加载成功。


kobayashi@kobayashi-virtual-machine: ~/labl

```
[    1.245732] cdrom: Uniform CD-ROM driver Revision: 3.20
[    1.278702] sr 1:0:0:0: Attached scsi generic sg0 type 5
[    1.486883] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 52:54:00:12:34:56
[    1.487377] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network Connection
[    1.487887] e1000e: Intel(R) PRO/1000 Network Driver
[    1.488031] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    1.488255] sky2: driver version 1.30
[    1.489266] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.489735] ehci-pci: EHCI PCI platform driver
[    1.489964] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    1.490136] ohci-pci: OHCI PCI platform driver
[    1.493121] uhci_hcd: USB Universal Host Controller Interface driver
[    1.493652] usbcore: registered new interface driver usblp
[    1.494630] usbcore: registered new interface driver usb-storage
[    1.495499] i8042: PNP: PS/2 Controller [PNP0303:KBD,PNP0f13:MOU] at 0x60,0x64 irq 1,12
[    1.497713] serio: i8042 KBD port at 0x60,0x64 irq 1
[    1.498064] serio: i8042 AUX port at 0x60,0x64 irq 12
[    1.501642] input: AT Translated Set 2 keyboard as /devices/platform/i8042/serio0/input/input1
[    1.503713] rtc_cmos 00:00: RTC can wake from S4
[    1.511671] rtc_cmos 00:00: registered as rtc0
[    1.512232] rtc_cmos 00:00: alarms up to one day, y3k, 114 bytes nvram, hpet irqs
[    1.513203] device-mapper: ioctl: 4.43.0-ioctl (2020-10-01) initialised: dm-devel@redhat.com
[    1.513720] intel_pstate: CPU model not supported
[    1.514007] hid: raw HID events driver (C) Jiri Kosina
[    1.515209] usbcore: registered new interface driver usbhid
[    1.515416] usbhid: USB HID core driver
[    1.523913] Initializing XFRM netlink socket
[    1.525030] NET: Registered protocol family 10
[    1.534159] Segment Routing with IPv6
[    1.535812] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    1.539005] NET: Registered protocol family 17
[    1.540243] Key type dns_resolver registered
[    1.540779] mce: Unable to init MCE device (rc: -5)
[    1.541279] IPI shorthand broadcast: enabled
[    1.541926] sched_clock: Marking stable (1519334035, 22270167)->(1580474633, -38870431)
[    1.543067] registered taskstats version 1
[    1.543263] Loading compiled-in X.509 certificates
[    1.546239] PM:   Magic number: 12:559:684
[    1.546683] mem full: hash matches
[    1.547033] printk: console [netcon0] enabled
[    1.547149] netconsole: network logging started
[    1.548661] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[    1.556971] kworker/u2:0 (59) used greatest stack depth: 7148 bytes left
[    1.564116] cfg80211: Loaded X.509 cert 'wens: 61c038651aabdcf94bd0ac7ff06c7248db18c600'
[    1.567233] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[    1.568618] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[    1.569006] cfg80211: failed to load regulatory.db
[    1.573815] ALSA device list:
[    1.574064]   No soundcards found.
[    1.592824] Freeing unused kernel image (initmem) memory: 684K
[    1.596237] Write protecting kernel text and read-only data: 15340k
[    1.596959] Run helloworld as init process
labl: Hello World
[    1.981632] tsc: Refined TSC clocksource calibration: 2983.727 MHz
[    1.982184] clocksource: tsc: mask: 0xffffffffffffffff max_cycles: 0x2b0238e8d56, max_idle_ns: 440795306872 ns
[    1.983001] clocksource: Switched to clocksource tsc
[    2.134197] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
[    4.109469] clocksource: timekeeping watchdog on CPU0: Marking clocksource 'tsc' as unstable because the skew is too lar
ge:
[    4.109809] clocksource:                       'hpet' wd_now: 182b2bd8 wd_last: 152adfb3 mask: ffffffff
[    4.110116] clocksource:                       'tsc' cs_now: 338100110 cs_last: 2de6f3d34 mask: ffffffffffffffff
[    4.110695] tsc: Marking TSC unstable due to clocksource watchdog
[    4.110995] TSC found unstable after boot, most likely due to broken BIOS. Use 'tsc=unstable'.
[    4.111313] sched_clock: Marking unstable (4088668361, 22273402)<-(4149811779, -38870431)
[    4.115667] clocksource: Checking clocksource tsc synchronization from CPU 0.
```

kobayashi@kobayashi-virtual-machine: ~/labl

```
kobayashi@kobayashi-virtual-machine:~$ cd labl/
kobayashi@kobayashi-virtual-machine:~/labl$
kobayashi@kobayashi-virtual-machine:~/labl$ gdb
GNU gdb (Ubuntu 8.1-0ubuntu3.2) 8.1.0.20180409-git
Copyright (C) 2018 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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".
(gdb) file linux-5.10.210/vmlinux
Reading symbols from linux-5.10.210/vmlinux...done.
(gdb) target remote:1234
Remote debugging using :1234
0x0000fff0 in ?? ()
(gdb) break start_kernel
Breakpoint 1 at 0xc20f1829: file init/main.c, line 847.
(gdb) c
Continuing.
(gdb) 
```

编译并启动BusyBox

- 先来了解一下BusyBox

简单来说，BusyBox在单一的可执行文件中提供了精简的Unix工具集。它能够在运行于多款POSIX环境的操作系统，例如Linux（包括Android）、FreeBSD等。由于BusyBox可执行文件的体积很小，因此它非常适合用于嵌入式设备。

- 进入实验目录

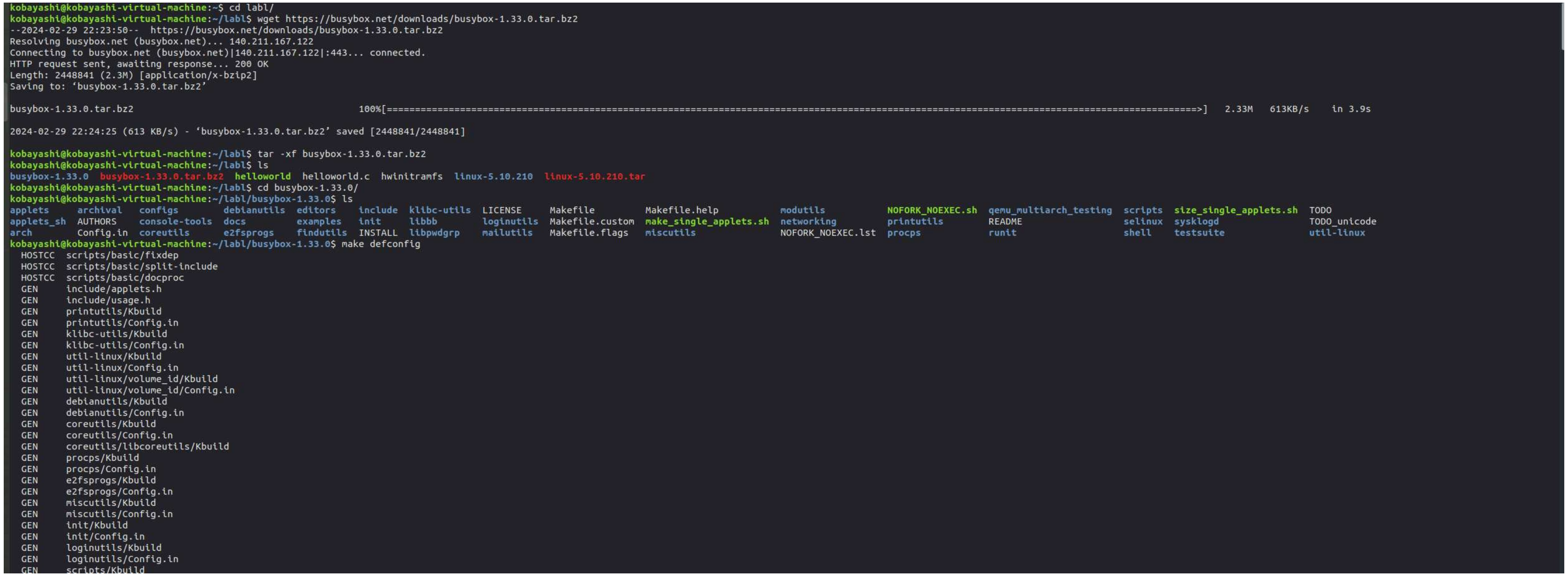
```
cd ~/labl
```

- 下载BusyBox并解压

```
wget https://busybox.net/downloads/busybox-1.33.0.tar.bz2
tar -xvf busybox-1.33.0.tar.bz2
cd busybox-1.33.0
```

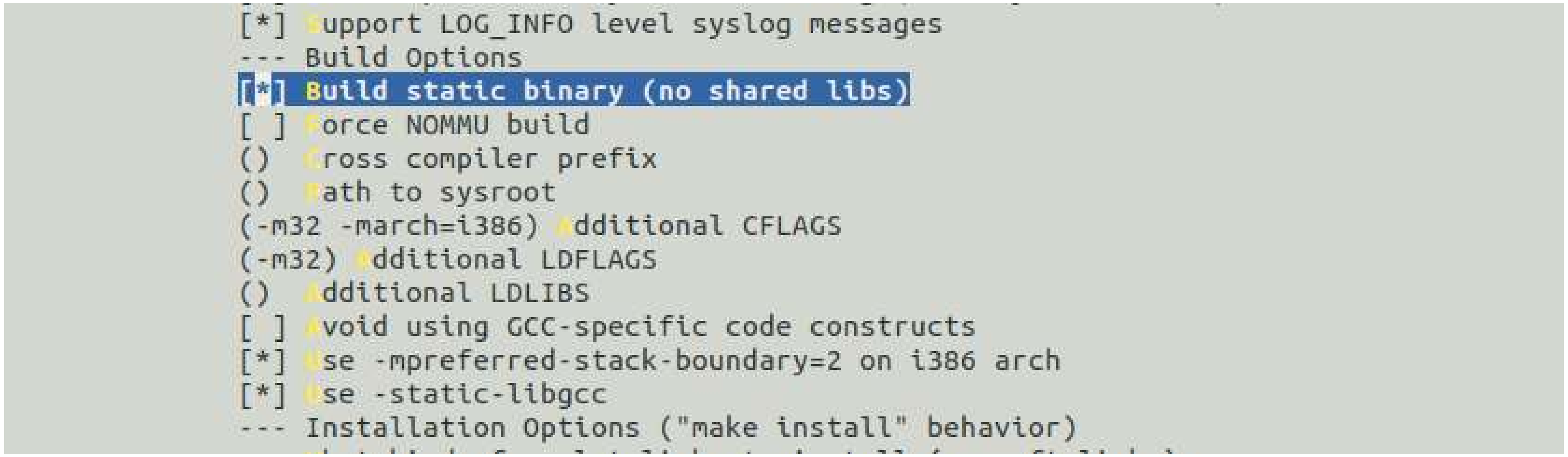
- 编译BusyBox

```
make defconfig
makde menuconfig
```



```
kobayashi@kobayashi-virtual-machine:~$ cd labl/
kobayashi@kobayashi-virtual-machine:~/labl$ wget https://busybox.net/downloads/busybox-1.33.0.tar.bz2
--2024-02-29 22:23:50--  https://busybox.net/downloads/busybox-1.33.0.tar.bz2
Resolving busybox.net (busybox.net)... 140.211.167.122
Connecting to busybox.net (busybox.net)|140.211.167.122|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2448841 (2.3M) [application/x-bzip2]
Saving to: ‘busybox-1.33.0.tar.bz2’

busybox-1.33.0.tar.bz2                              100%[=====================================================================================================================>]   2.33M   613KB/s    in 3.9s

2024-02-29 22:24:25 (613 KB/s) - ‘busybox-1.33.0.tar.bz2’ saved [2448841/2448841]

kobayashi@kobayashi-virtual-machine:~/labl$ tar -xf busybox-1.33.0.tar.bz2
kobayashi@kobayashi-virtual-machine:~/labl$ ls
busybox-1.33.0  busybox-1.33.0.tar.bz2  helloworld  helloworld.c  hwinitramfs  linux-5.10.210  linux-5.10.210.tar
kobayashi@kobayashi-virtual-machine:~/labl$ cd busybox-1.33.0/
kobayashi@kobayashi-virtual-machine:~/labl/busybox-1.33.0$ ls
applets     archival   configs        debianutils  editors    include  klibc-utils  LICENSE     Makefile         Makefile.help           modutils          NOFORK_NOEXEC.sh  qemu_multiarch_testing  scripts  size_single_applets.sh  TODO
applets_sh  AUTHORS    console-tools  docs         examples   init     libbb        loginutils  Makefile.custom  make_single_applets.sh  networking        printutils        README                  selinux  sysklogd                TODO_unicode
arch        Config.in  coreutils      e2fsprogs    findutils  INSTALL  libpwdgrp    mailutils   Makefile.flags   miscutils               NOFORK_NOEXEC.lst procps            runit                   shell    testsuite               util-linux
kobayashi@kobayashi-virtual-machine:~/labl/busybox-1.33.0$ make defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/basic/split-include
  HOSTCC  scripts/basic/docproc
  GEN     include/applets.h
  GEN     include/usage.h
  GEN     printutils/Kbuild
  GEN     printutils/Config.in
  GEN     klibc-utils/Kbuild
  GEN     klibc-utils/Config.in
  GEN     util-linux/Kbuild
  GEN     util-linux/Config.in
  GEN     util-linux/volume_id/Kbuild
  GEN     util-linux/volume_id/Config.in
  GEN     debianutils/Kbuild
  GEN     debianutils/Config.in
  GEN     coreutils/Kbuild
  GEN     coreutils/Config.in
  GEN     coreutils/libcoreutils/Kbuild
  GEN     procps/Kbuild
  GEN     procps/Config.in
  GEN     e2fsprogs/Kbuild
  GEN     e2fsprogs/Config.in
  GEN     miscutils/Kbuild
  GEN     miscutils/Config.in
  GEN     init/Kbuild
  GEN     init/Config.in
  GEN     loginutils/Kbuild
  GEN     loginutils/Config.in
  GEN     scripts/Kbuild
```

进入settings，然后在 `Build BusyBox as a static binary (no shared libs)` 选项前打上 `*` ，然后分别设置 `Additional CFLAGS` 和 `Additional LDFLAGS` 为 `-m32 -march=i386` 和 `-m32` ，保存并退出。



```
[*] Support LOG_INFO level syslog messages
--- Build Options
[*] Build static binary (no shared libs)
[ ] Force NOMMU build
()  Cross compiler prefix
()  Path to sysroot
(-m32 -march=i386) Additional CFLAGS
(-m32) Additional LDFLAGS
()  Additional LDLIBS
[ ] Avoid using GCC-specific code constructs
[*] Use -mpreferred-stack-boundary=2 on i386 arch
[*] Use -static-libgcc
--- Installation Options ("make install" behavior)
```

然后运行下列命令编译BusyBox：

```
make -j8
make install
```

- 使用BusyBox制作initramfs

1. 先将安装在 `_install` 目录下的文件移动到一个新的目录：

```
# move all the files in the BusyBox installation directory to a new directory
cd ~/labl
mkdir mybusybox
mkdir -pv mybusybox/{bin,sbin,etc,proc,sys,usr/{bin,sbin}}
cp -av busybox-1.33.0/_install/* mybusybox/
cd mybusybox
```

2. 然后需要创建一个 `init` 文件，这里可以用一个简单地shell脚本作为 `init` 文件：

```
#!/bin/sh
mount -t proc none /proc
mount -t sysfs none /sys
echo -e "\nBoot took $(cut -d' ' -f1 /proc/uptime) seconds\n"
exec /bin/sh
```

3. 给 `init` 文件添加可执行权限：

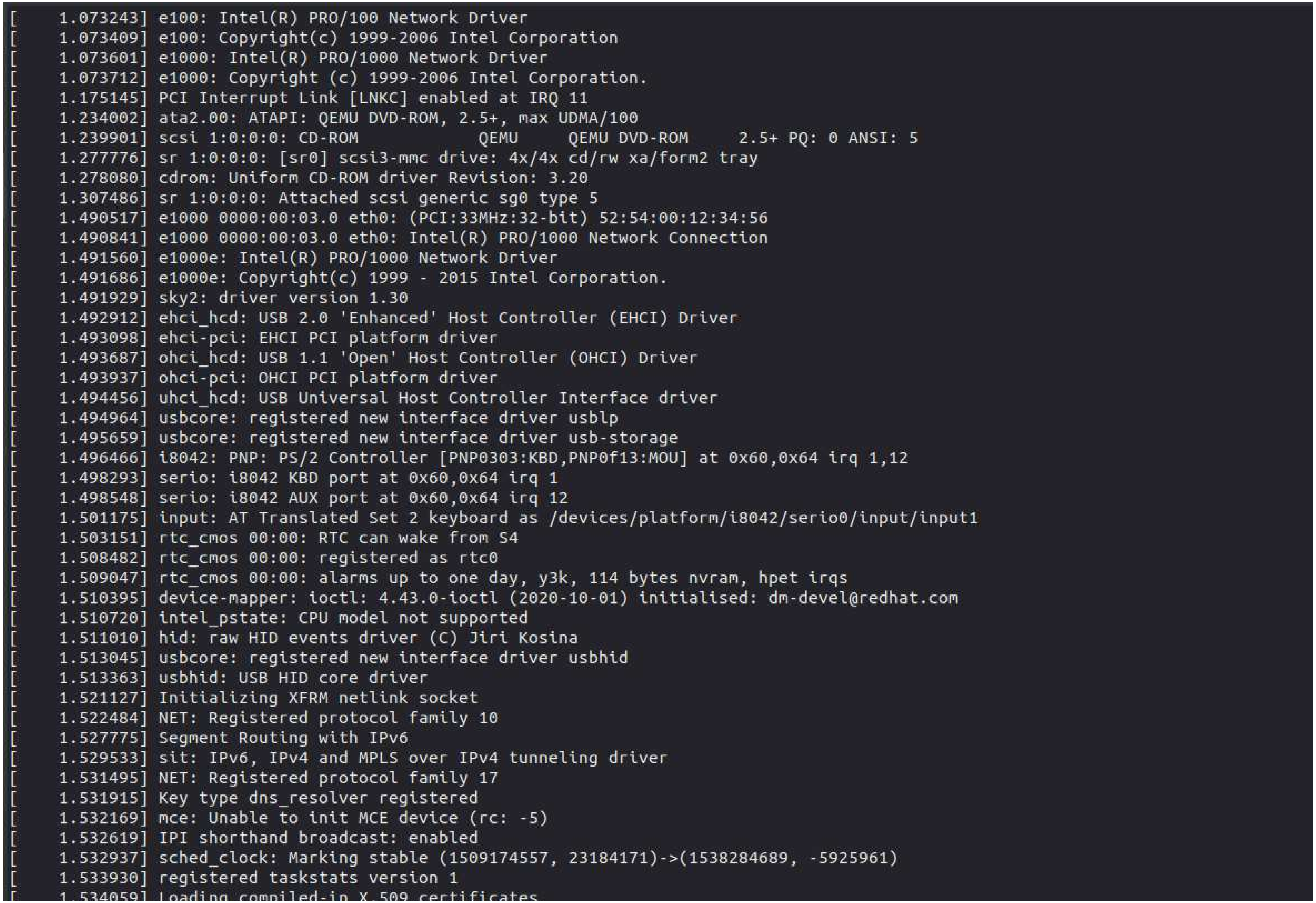
```
chmod u+x init
```

4. 最后将x86-busybox下面的内容打包归档成cpio文件，以供Linux内核做initramfs启动执行：

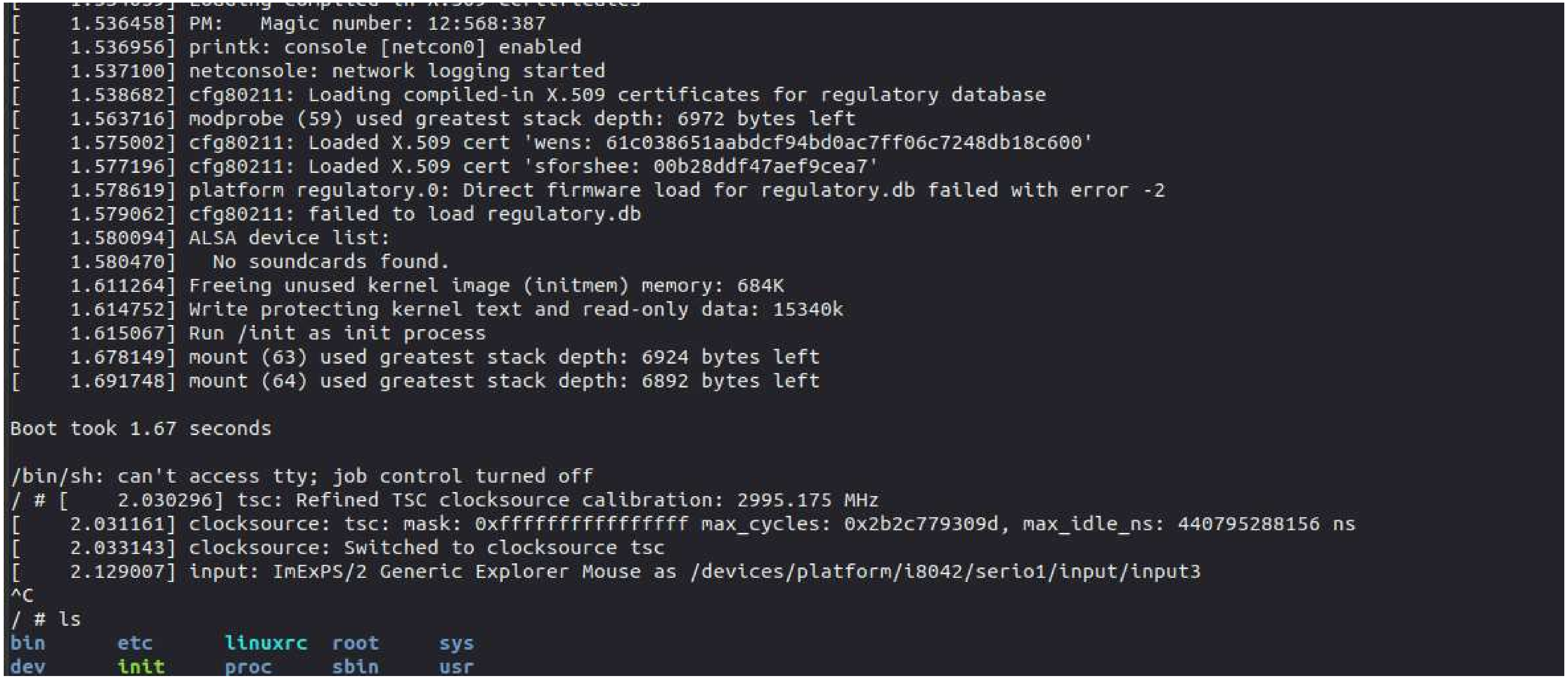
```
find . -print0 | cpio --null -ov --format=newc | gzip -9 > ~/labl/initramfs-busybox-x86.cpio.gz
```

- 加载busybox

```
cd ~/labl
quemu-system-i386 -kernel linux-5.10.211/arch/x86/boot/bzImage -initrd initramfs-busybox-x86.cpio.gz -append "console=ttyS0" -nographic
```



```
[    1.073243] e100: Intel(R) PRO/100 Network Driver
[    1.073409] e100: Copyright(c) 1999-2006 Intel Corporation
[    1.073601] e1000: Intel(R) PRO/1000 Network Driver
[    1.073712] e1000: Copyright (c) 1999-2006 Intel Corporation.
[    1.175145] PCI Interrupt Link [LNKC] enabled at IRQ 11
[    1.234002] ata2.00: ATAPI: QEMU DVD-ROM, 2.5+, max UDMA/100
[    1.239901] scsi 1:0:0:0: CD-ROM            QEMU     QEMU DVD-ROM     2.5+ PQ: 0 ANSI: 5
[    1.277776] sr 1:0:0:0: [sr0] scsi3-mmc drive: 4x/4x cd/rw xa/form2 tray
[    1.278080] cdrom: Uniform CD-ROM driver Revision: 3.20
[    1.307486] sr 1:0:0:0: Attached scsi generic sg0 type 5
[    1.490517] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 52:54:00:12:34:56
[    1.490841] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network Connection
[    1.491560] e1000e: Intel(R) PRO/1000 Network Driver
[    1.491686] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    1.491929] sky2: driver version 1.30
[    1.492912] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.493098] ehci-pci: EHCI PCI platform driver
[    1.493687] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    1.493937] ohci-pci: OHCI PCI platform driver
[    1.494456] uhci_hcd: USB Universal Host Controller Interface driver
[    1.494964] usbcore: registered new interface driver usblp
[    1.495659] usbcore: registered new interface driver usb-storage
[    1.496466] i8042: PNP: PS/2 Controller [PNP0303:KBD,PNP0f13:MOU] at 0x60,0x64 irq 1,12
[    1.498293] serio: i8042 KBD port at 0x60,0x64 irq 1
[    1.498548] serio: i8042 AUX port at 0x60,0x64 irq 12
[    1.501175] input: AT Translated Set 2 keyboard as /devices/platform/i8042/serio0/input/input1
[    1.503151] rtc_cmos 00:00: RTC can wake from S4
[    1.508482] rtc_cmos 00:00: registered as rtc0
[    1.509047] rtc_cmos 00:00: alarms up to one day, y3k, 114 bytes nvram, hpet irqs
[    1.510395] device-mapper: ioctl: 4.43.0-ioctl (2020-10-01) initialised: dm-devel@redhat.com
[    1.510720] intel_pstate: CPU model not supported
[    1.511010] hid: raw HID events driver (C) Jiri Kosina
[    1.513045] usbcore: registered new interface driver usbhid
[    1.513363] usbhid: USB HID core driver
[    1.521127] Initializing XFRM netlink socket
[    1.522484] NET: Registered protocol family 10
[    1.527775] Segment Routing with IPv6
[    1.529533] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    1.531495] NET: Registered protocol family 17
[    1.531915] Key type dns_resolver registered
[    1.532169] mce: Unable to init MCE device (rc: -5)
[    1.532619] IPI shorthand broadcast: enabled
[    1.532937] sched_clock: Marking stable (1509174557, 23184171)->(1538284689, -5925961)
[    1.533930] registered taskstats version 1
[    1.534059] Loading compiled-in X.509 certificates
```



```
[    1.536458] PM:   Magic number: 12:568:387
[    1.536956] printk: console [netcon0] enabled
[    1.537100] netconsole: network logging started
[    1.538682] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[    1.563716] modprobe (59) used greatest stack depth: 6972 bytes left
[    1.575002] cfg80211: Loaded X.509 cert 'wens: 61c038651aabdcf94bd0ac7ff06c7248db18c600'
[    1.577196] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[    1.578619] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[    1.579062] cfg80211: failed to load regulatory.db
[    1.580094] ALSA device list:
[    1.580470]   No soundcards found.
[    1.611264] Freeing unused kernel image (initmem) memory: 684K
[    1.614752] Write protecting kernel text and read-only data: 15340k
[    1.615067] Run /init as init process
[    1.678149] mount (63) used greatest stack depth: 6924 bytes left
[    1.691748] mount (64) used greatest stack depth: 6892 bytes left

Boot took 1.67 seconds

/bin/sh: can't access tty; job control turned off
/ # [    2.030296] tsc: Refined TSC clocksource calibration: 2995.175 MHz
[    2.031161] clocksource: tsc: mask: 0xffffffffffffffff max_cycles: 0x2b2c779309d, max_idle_ns: 440795288156 ns
[    2.033143] clocksource: Switched to clocksource tsc
[    2.129007] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
^C
/ # ls
bin      etc      linuxrc  root     sys
dev      init     proc     sbin     usr
```

实验结果

内核编译成功，启动内核并开启远程调试成功，制作initramfs成功，编译并启动BusyBox成功。

总结

这是一次非常有意义的实验，通过这次实验，我对Linux内核的编译过程有了更加深入的了解，也对Linux内核启动过程有了更加深入的了解。

实验的最终结果大致上是成功的，但仍然存在一些问题，比如最后使用busybox并启动内核后有报错：

```
/bin/sh: can't access tty; job control turned off
```

我查阅了一些资料，猜测该问题可能是由启动终端不匹配引起的，但我暂时没有找到解决办法。希望在以后的学习中能够解决这个问题。