openEuler开源创新实践课

LFS7.7-systemd on openEuler

实验指导手册

v1.3



华为技术有限公司

|  |
| --- |
| 版权所有 © 华为技术有限公司 2021。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  C:\Users\jwx341670\Desktop\华为标志 Huawei Logo 2018\竖版标志Vertical Version\PNG\HW_POS_RBG_Vertical-150ppi.png和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |  |  |
| --- | --- | --- | --- |
| 修订记录 | 修订内容 | 修改人 | 修改时间 |
| V1.1 | 初始稿 | 赵小虎 | 2021.7.4 |
| V1.2 | 南开大学实践常见问题修改 | 李宝林 | 2021.7.8 |
| V1.3 | 优化 | 赵小虎 | 2021.12.27 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | http://[e](http://e.huawei.com/).huawei.com |

目录

[1 实验介绍 3](#_Toc92876236)

[1.1 简介 3](#_Toc92876237)

[1.2 内容描述 3](#_Toc92876238)

[1.3 读者知识背景 3](#_Toc92876239)

[1.4 实验环境说明 4](#_Toc92876240)

[1.4.1 个人电脑及操作系统 4](#_Toc92876241)

[1.4.2 虚拟机软件及其版本 4](#_Toc92876242)

[1.4.3 作为 LFS 宿主机的虚拟机规格 4](#_Toc92876243)

[1.4.4 openEuler 操作系统镜像 5](#_Toc92876244)

[1.4.5 LFS 版本 5](#_Toc92876245)

[1.5 学习参考资料 5](#_Toc92876246)

[2 宿主机系统的准备 7](#_Toc92876247)

[2.1 虚拟机软件的安装 7](#_Toc92876248)

[2.2 虚拟机的创建与配置 11](#_Toc92876249)

[2.3 安装openEuler操作系统 24](#_Toc92876250)

[3 LFS 系统的构建 33](#_Toc92876251)

[3.1 为宿主系统准备好构建 LFS 目标系统的环境 33](#_Toc92876252)

[3.1.1 脚本 part-0.sh 33](#_Toc92876253)

[3.1.2 脚本 part-1.sh 34](#_Toc92876254)

[3.2 以 lfs 用户构建编译临时工具链 34](#_Toc92876255)

[3.2.1 脚本 part-2.sh 34](#_Toc92876256)

[3.2.2 通用编译指南 34](#_Toc92876257)

[3.2.3 gcc-4.9.2 编译问题 35](#_Toc92876258)

[3.2.4 其他需要注意的地方 35](#_Toc92876259)

[3.3 在 chrooted 环境下构建 LFS 目标系统 35](#_Toc92876260)

[3.3.1 脚本 part-3.0.sh 35](#_Toc92876261)

[3.3.2 脚本 part-3.1.sh 35](#_Toc92876262)

[3.3.3 mount & chroot 36](#_Toc92876263)

[3.3.4 通用编译指南 36](#_Toc92876264)

[3.3.5 其他问题 36](#_Toc92876265)

[3.4 配置 LFS 目标系统 36](#_Toc92876266)

[3.4.1 第二个版本的chroot 37](#_Toc92876267)

[3.4.2 脚本 part-3.2.sh 37](#_Toc92876268)

[3.5 编译并安装内核 37](#_Toc92876269)

[3.5.1 make menuconfig 37](#_Toc92876270)

[3.5.2 .config 文件 38](#_Toc92876271)

[3.5.3 Finding out which kernel target to build 38](#_Toc92876272)

[3.5.4 脚本 part-3.3.sh 38](#_Toc92876273)

[3.5.5 其他问题 39](#_Toc92876274)

[3.6 设置 GRUB 使新系统可引导 39](#_Toc92876275)

[3.6.1 配置 LFS 目标系统 GRUB——part-3.4.sh 39](#_Toc92876276)

[3.6.2 更改宿主机 GRUB 配置——part-3.5.sh 39](#_Toc92876277)

[3.7 进入新的系统 40](#_Toc92876278)

[3.7.1 Who is who 40](#_Toc92876279)

[3.7.2 How to get in 40](#_Toc92876280)

[3.7.3 脚本 part-3.6.sh 42](#_Toc92876281)

[4 作业的提交 43](#_Toc92876282)

[4.1 提交 PR 前的准备 43](#_Toc92876283)

[4.2 提交 PR 45](#_Toc92876284)

[4.3 参考链接 47](#_Toc92876285)

[5 进阶任务 48](#_Toc92876286)

[6 附录 49](#_Toc92876287)

[6.1 如何将资料从 PC 复制到虚拟机 49](#_Toc92876288)

[7 缩略语表 50](#_Toc92876289)

# 实验介绍

## 简介

本手册为 openEuler 开源创新实践课实验教程，适用于已有一定计算机原理、操作系统、编译原理基础，并希望深入了解操作系统及相关工具的构建，从源码开始搭建个性化操作系统的读者。

鉴于创新实践课本身的特点，本文不会给出一幅按图索骥的地图或一系列按部就班的步骤来指导你构建一个新系统，而是将“如何构建一个自己的 Linux 操作系统”的津要和盘托出。即使是这样，依托我们专门撰写的“构建脚本”，足以使您能够顺利地完成构建 LFS 系统的基础工作。

## 内容描述

本实验指导书共包含4个部分，从实验简介到宿主机环境准备，再到构建 LFS 系统，最后是供学生完全自己发挥的拓展实验。

## 读者知识背景

本课程为操作系统进阶实践课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

* 有电脑操作基础
* 熟悉 Linux 操作系统的基本架构和实现原理
* 掌握 Linux 操作系统的基本操作
* 了解 Linux 常用工具的功能和使用

## 实验环境说明

### 个人电脑及操作系统

每套实验环境包括个人 PC 一台，每套实验环境适用于1名学员上机操作。为了满足实验需要，建议每套实验环境采用以下配置：

实验设备

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 操作系统 |
| 个人电脑 | 处理器：Intel i7 CPU  逻辑处理器数：≥6  内存：16G  系统类型：x64-based处理器 | Windows 10 / macOS |

注：此次实验需要的计算资源全部来自学员个人电脑，硬件上应尽量接近或高于上表配置以增进构建速度。

### 虚拟机软件及其版本

虚拟机软件

|  |  |  |
| --- | --- | --- |
| 软件名称及本实验推荐版本 | 官方网址 | 备注 |
| VMware Workstation 15.5 Pro | <https://www.vmware.com/> | 需付费使用 |
| Oracle VM VirtualBox 6.1 | <http://www.virtualbox.org> | 根据其相关协议使用 |

注：本实验所用虚拟机软件是Oracle VM VirtualBox 6.1。

### 作为 LFS 宿主机的虚拟机规格

虚拟机规格

|  |  |
| --- | --- |
| 参数 | 推荐值 |
| 逻辑处理器数 | 4 |
| 内存 | 8GB |
| 磁盘大小 | 30GB |

### openEuler 操作系统镜像

推荐使用openEuler 20.03 LFS SP1或openEuler 20.09，本实验我们选择了后者。

openEuler操作系统镜像

|  |  |  |
| --- | --- | --- |
| 软件名称 | CPU架构 | 下载地址 |
| openEuler 20.09 | x86\_64 | <https://repo.openeuler.org/openEuler-20.09/ISO/x86_64/openEuler-20.09-x86_64-dvd.iso> |

### LFS 版本

本实验以LFS-BOOK-7.7-systemd为例进行讲解。为什么呢？因为该版本所用的GCC版本不是太高（gcc-4.9.2），这对于想从源码研究C++的人是一大福音，另外对systemd感兴趣的读者也可以从源码研究systemd。

LFS7.7-systemd

|  |  |
| --- | --- |
| 项目名称 | 下载链接 |
| Packages | <http://ftp.osuosl.org/pub/lfs/lfs-packages/lfs-packages-7.7-systemd.tar> 或者 <https://zhuanyejianshe.obs.cn-north-4.myhuaweicloud.com/chuangxinshijianke/lfs-packages-7.7-systemd.tar> |
| LFS-BOOK | <https://www.linuxfromscratch.org/lfs/downloads/7.7-systemd/> |
| Chinese LFS-BOOK | <https://linux.cn/lfs/LFS-BOOK-7.7-systemd/> |

## 学习参考资料

1. openEuler 社区：<https://openeuler.org/>

2. 项目托管地：<https://gitee.com/openeuler-practice-courses/lfs-course>

3. LFS 官网：<https://www.linuxfromscratch.org/lfs/>

4. HCIA-openEuler 认证：[https://e.huawei.com/cn/talent/#/cert/product-details?certifiedProductId=383&authenticationLevel=CTYPE\_CARE\_HCIA&technicalField=PSC&version=1.0](https://e.huawei.com/cn/talent/" \l "/cert/product-details?certifiedProductId=383&authenticationLevel=CTYPE_CARE_HCIA&technicalField=PSC&version=1.0)

5. 鲲鹏生态官网：<https://www.hikunpeng.com/zh/>

6. Brian Ward 所著《精通 LInux》（姜南、袁志鹏译）

# 宿主机系统的准备

所谓**宿主机系统**系指创建LFS目标系统的宿主，具体而言，即虚拟机上的openEuler操作系统。本章介绍如何使用VirtualBox软件创建虚拟机、安装openEuler操作系统以及安装后的实验环境准备工作。

## 虚拟机软件的安装

本节假定您已经安装好Oracle VM VirtualBox 6.1软件并以此为例讲解如何创建满足创建LFS系统的虚拟机。安装该虚拟机软件时一路默认安装即可，可以参考以下步骤（以V6.1.22为例）：

Text

Description automatically generated with medium confidence

点击“下一步”。

Graphical user interface, text, application

Description automatically generated

保持默认设置，点击“下一步”。

Graphical user interface, text, application, email

Description automatically generated

可以保持默认设置，点击“下一步”。

Text

Description automatically generated

选择“Yes（同意）”。

Graphical user interface, text, application, email

Description automatically generated

这时，点击“安装”。

Graphical user interface, text, application

Description automatically generated

等待安装完毕。

A picture containing graphical user interface

Description automatically generated

安装完毕后点击“完成”按钮。

Graphical user interface, text, application

Description automatically generated

这是虚拟机软件启动之后的界面。

## 虚拟机的创建与配置

创建虚拟机时，请选择Linux类型和64-bit版本：

Graphical user interface, text, application, email

Description automatically generated

为了在构建 LFS 系统时有较快的速度，内存设置为8GB为宜（建议不小于4GB）：

Graphical user interface, text, application, email

Description automatically generated

现在就创建硬盘：

Graphical user interface, text, application, Word

Description automatically generated

这里硬盘类型为 VHD：

Graphical user interface, text, application

Description automatically generated

磁盘容量“动态增长”：

Graphical user interface, text, application, email

Description automatically generated

为保证足够的空间，为宿主机配置20GB大小的容量：

Graphical user interface, application, Word

Description automatically generated

启动顺序光驱置为第一，其次硬盘：

Graphical user interface, text, application, email

Description automatically generated

按下“Settings”按钮对虚拟机的配置进行调整：



CPU设为4核：Graphical user interface, text, application, email

Description automatically generated

显示器的 Scale Factor 设为200%以便于观看：Graphical user interface, text, application, Word

Description automatically generated

在“Storage”的“Controller: SATA”里增加一块硬盘：

Graphical user interface, text, application

Description automatically generated

选择“Create”按钮：

Graphical user interface, text, application

Description automatically generated

同样是 VHD 类型：

Graphical user interface, text, application

Description automatically generated

同样是动态增长：

Graphical user interface, text, application, email

Description automatically generated

将这块磁盘的容量设为 30GB：

Graphical user interface, application, Word

Description automatically generated

创建完之后的磁盘配置信息（第一块 20G，第二块 30G）：

Graphical user interface, text, application, email

Description automatically generated

注意这两块磁盘均为 SATA 接口类型：

Graphical user interface, text, application

Description automatically generated

现在配置网络。

Adapter 1 保持默认设置（即 NAT 网络，以便从虚拟机访问外网）：

Graphical user interface, text, application

Description automatically generated

Enable Adapter 2 并设置为“Host-only Adapter”（以便在宿主机中访问虚拟机）：

Graphical user interface, text, application

Description automatically generated

其余配置保持默认值。

以下是该虚拟机配置信息总览：

Graphical user interface, text, application, email

Description automatically generated

此时为光驱添加一个磁盘文件（本手册以 openEuler-20.09-x86\_64-dvd.iso 为例，请事先将其下载到本地）：

Graphical user interface, text, application, Word

Description automatically generated

然后启动虚拟机：

Text

Description automatically generated with medium confidence

## 安装openEuler操作系统

启动虚拟机后，移动键盘的上下键选择“Install openEuler 20.09”菜单项并按下回车键进入安装界面：

Graphical user interface, text

Description automatically generated

选择英文版安装：

Graphical user interface, text, application

Description automatically generated

下面要对有惊叹号的两个地方进行设置：

Graphical user interface, application, Teams

Description automatically generated

对于系统的安装目的盘，选择 sda 后点击左上角的“Done”按钮：

Graphical user interface, text, application, email

Description automatically generated

设置好 root 用户的密码：

A picture containing rectangle

Description automatically generated

此时只需要设置 root 账户密码即可，如：Euler@123。

其他设置保持默认值，如 Software Selection 是 Minimal Install。如果您的网络环境正常，先前设置的两块网卡应处于“Connected”状态，如：

Graphical user interface, text, application

Description automatically generated

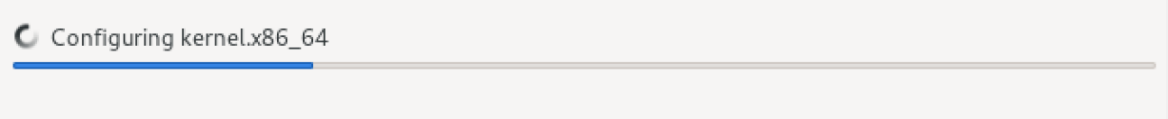
Graphical user interface, text, application

Description automatically generated

现在即可点击“Begin Installation”按钮开始安装：

Graphical user interface, application

Description automatically generated

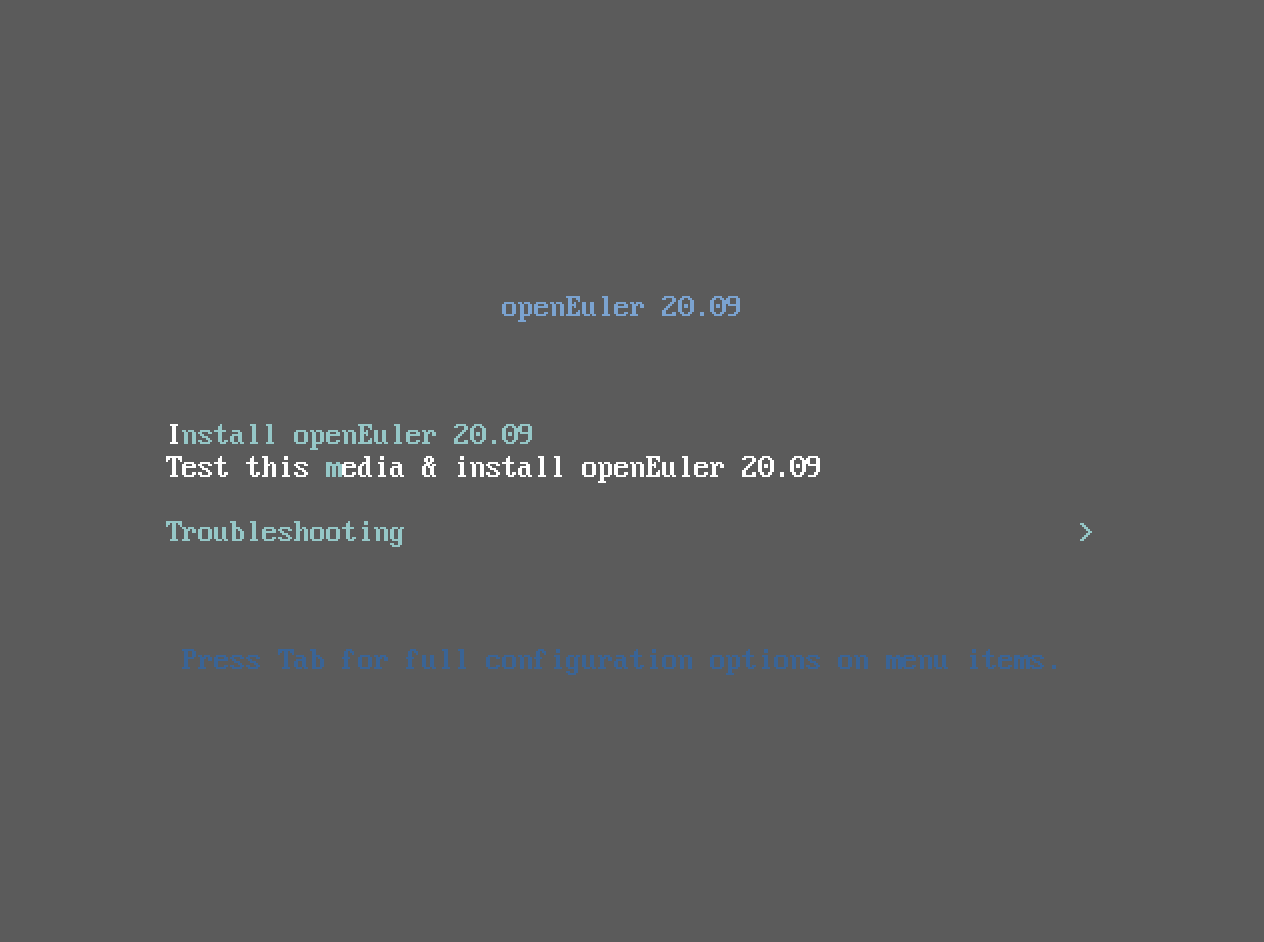


安装完毕后重启系统：

Graphical user interface, text, application, email

Description automatically generated

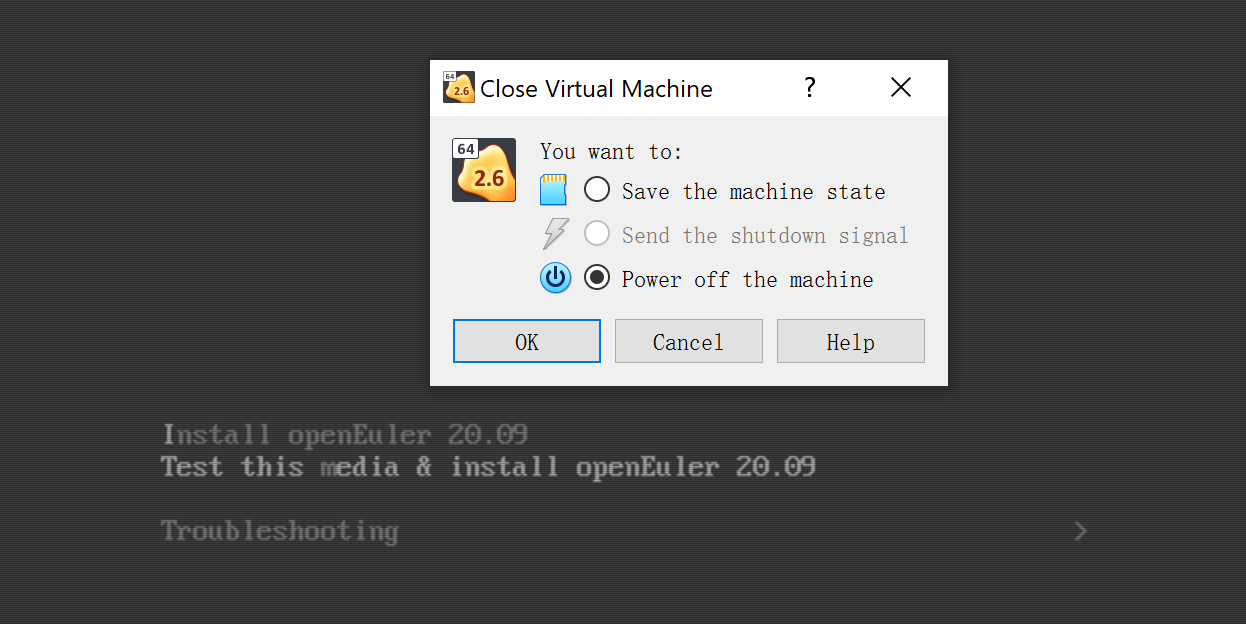
再次进入菜单选择界面时，移动键盘的上下键以将系统停留在该界面：



然后关机虚拟机：

Text

Description automatically generated



等待虚拟机关机后移出光驱中的镜像：

Graphical user interface, application, Word

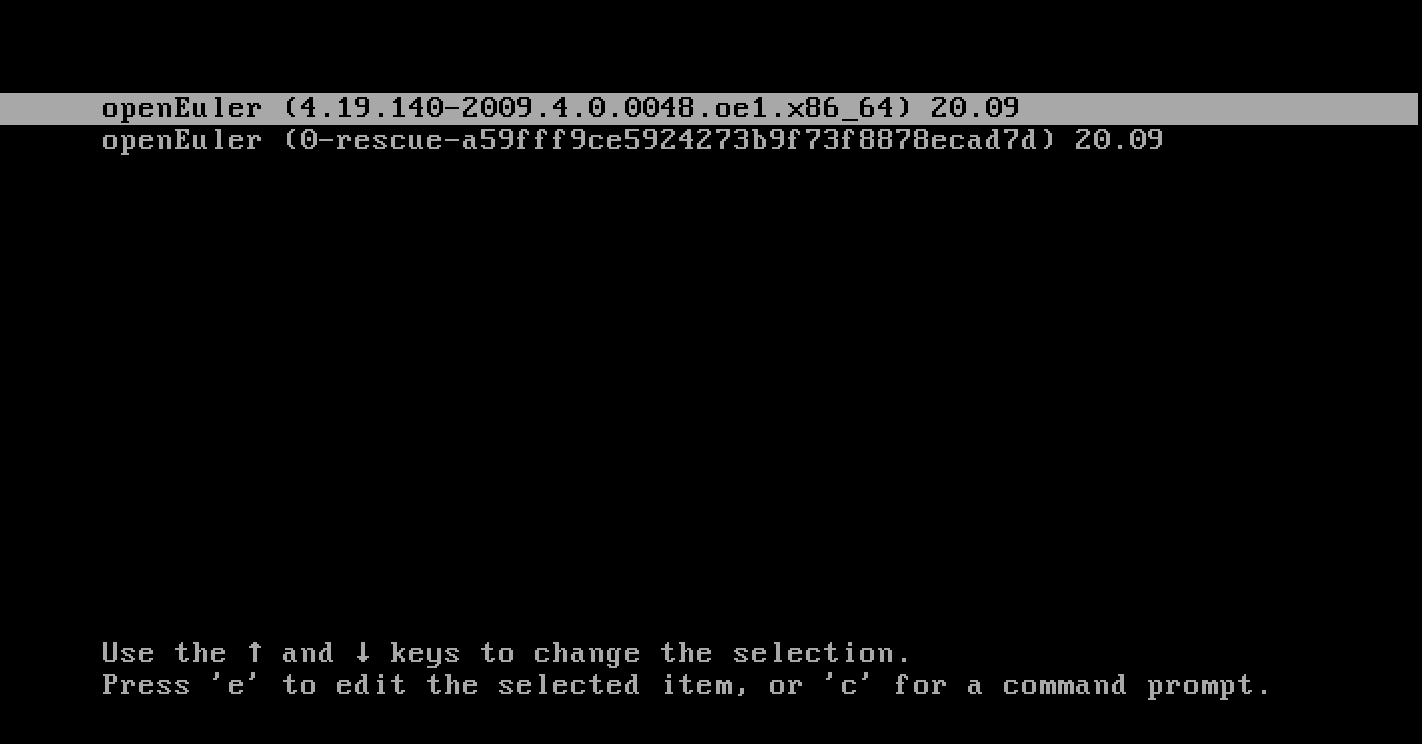
Description automatically generated

然后再次启动虚拟机：

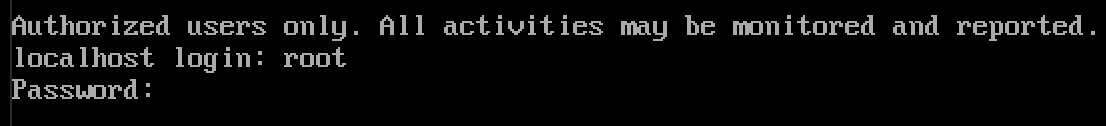
Icon

Description automatically generated with medium confidence

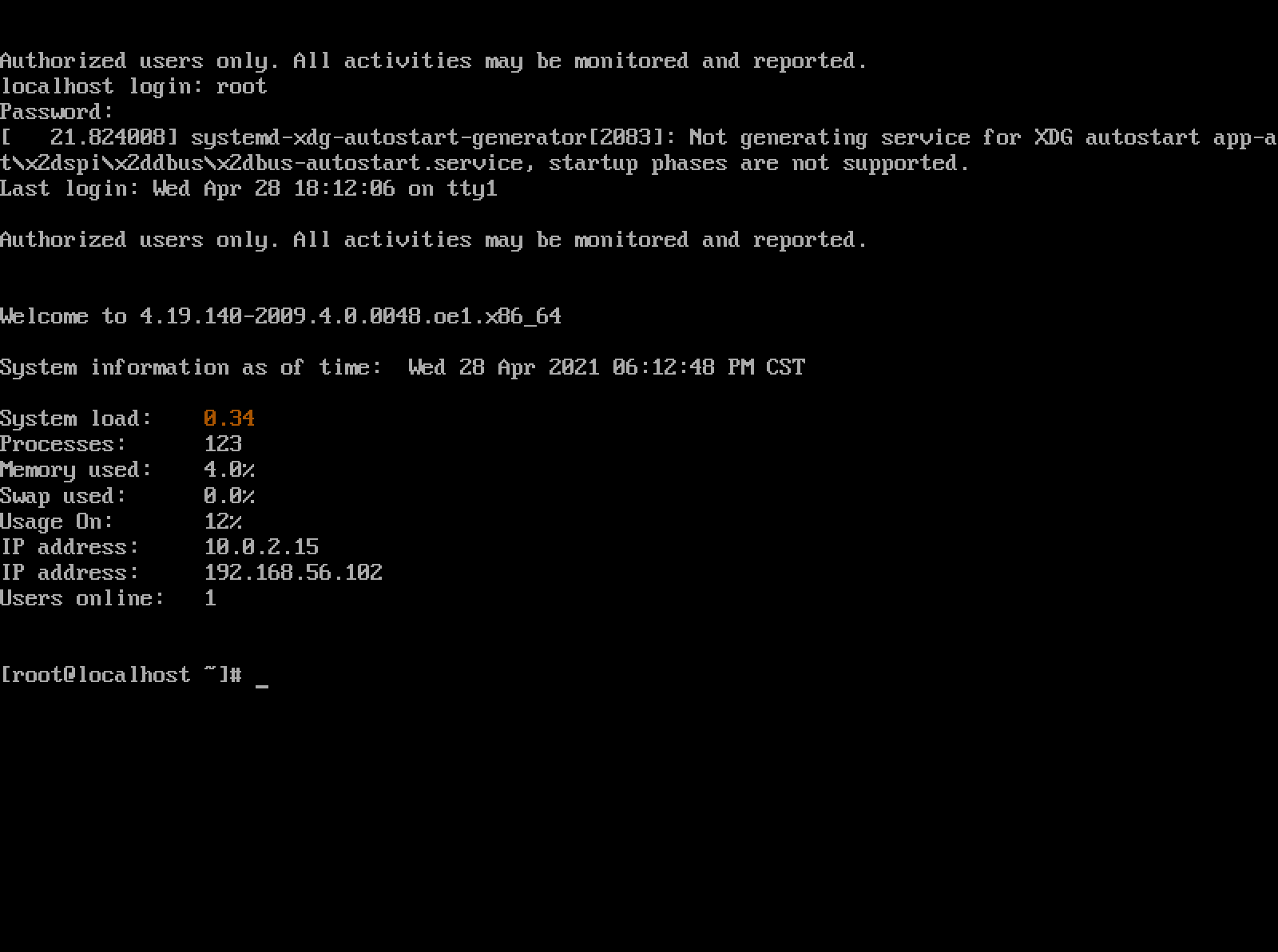
系统进入openEuler启动界面，用上下键选择第一个菜单项并按回车键启动（默认也是以第一个菜单项启动）：



先在VirtualBox中以root用户登录openEuler操作系统（以 root 用户登录，输入密码时不会有任何回显）：



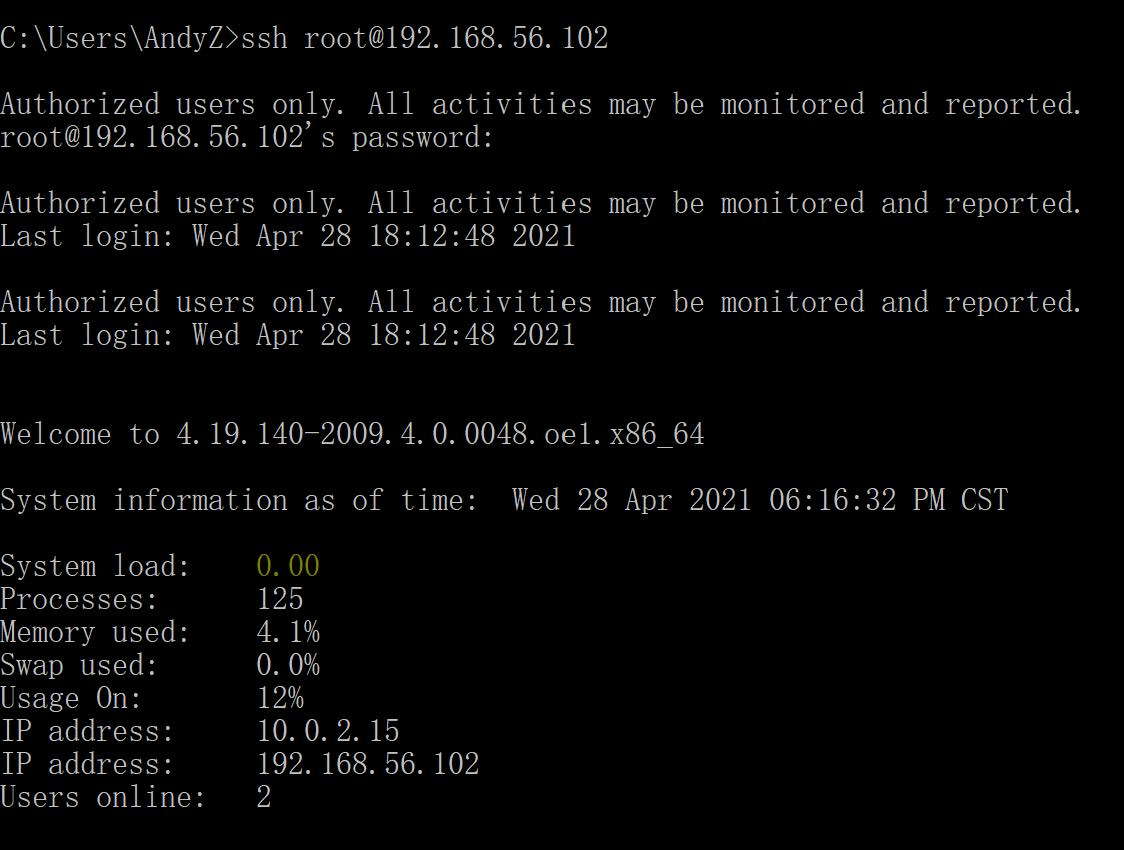
登录完毕后即可查看虚拟机的本地 IP 地址（下图中是 *192.168.56.102*，您的 IP 地址或许与此不同）：



为便于操作，再打开命令终端（如果是Windows 10操作系统，在键盘上按下“ +R”进入cmd命令行窗口；如果是macOS操作系统，在键盘上按下“ +空格键”，在弹出的搜索框中输入“Terminal.app”然后回车打开终端）通过ssh命令以root身份登录openEuler，如：

ssh root@*192.168.56.102*

注意：您应该用自己虚拟机的IP地址替换上述IP地址。



同样，以上登录过程中输入密码时不会有任何显示。

至此我们已经全部完成宿主机系统的准备工作，接下来我们参考 shell 脚本构建 LFS 目标系统（又称LFS\_Target\_System 或 LFS\_TGT\_SYS）。

# LFS 系统的构建

现在，我们就要开始构建LFS系统之旅啦！我们将该过程分成了一系列的“准脚本”文件，大多数要一步步来执行，有些可以像真正的脚本文件一次性执行。总的来说，构建过程分为如下几个阶段：

1. 为宿主系统准备好构建 LFS 目标系统的环境；
2. 以 lfs 用户构建临时工具链；
3. 在 chrooted 环境下构建 LFS 目标系统；
4. 配置 LFS 目标系统；
5. 编译并安装内核；
6. 设置 GRUB 使新系统可引导；
7. 进入新系统。

下面将通过脚本文件说明这一过程。

## 为宿主系统准备好构建 LFS 目标系统的环境

脚本文件：part-0.sh 和 part-0.sh。

### 脚本 part-0.sh

该脚本的运行方式是“Step by step”，即你需要手动地一个命令一个命令地执行，而不是“sh xxx.sh”这种形式。

在运行该脚本中的命令之前，你应该已经做好了如下事情（正如我们在上一章所作的那样）：

(1) 已经创建好了虚拟机（拥有 sda 和 sdb 两块硬盘）；

(2) 已经安装好了 openEuler 操作系统（安装在 sda 上）。

本脚本完成如下任务：

(1) 对 sdb 进行分区（该分区 sdb1 即是未来 LFS 目标系统所在的分区）；

(2) 对分区进行格式化。

您需要熟练使用 fdisk、mkfs 等命令并用 lsblk、blkid 等命令对分区、格式化的结果进行研究和验证。

### 脚本 part-1.sh

在前一节脚本的基础上，本脚本主要完成以下任务：

(1) 进一步完善宿主系统的软件环境（一系列 yum install 命令……）；

(2) 配置您的 gitee 账号；

(3) 设置 $LFS 变量（该变量指向了后面我们要创建的 LFS 目标系统）；

(4) 将 sdb1 挂载至 $LFS 表示的目录（我们通过修改宿主系统的 fstab 让 host 启动时自动挂载）；

(5) 创建 $LFS/sources 目录（编译源代码的地方）；

(6) 创建 $LFS/tools 目录（编译出来的临时工具链所在地）；

(7) 增加 lfs 用户（为避免损坏 host，我们将以 lfs 用户编译临时工具链）；

(8) 为 lfs 用户增加 sudo 权限（后面的有些步骤需要用到该权限）；

(9) 为 lfs 用户设置密码；

(10) 获取 LFS packages 以及 lfs-course 资料。

您也可以在 LFS 官网下载页面（https://www.linuxfromscratch.org/lfs/download.html）找到“Packages for LFS”一节并事先将 lfs-packages-7.7-systemd.tar 包下载到您的本地 PC，然后用 scp 命令将该包从 PC 拷贝到虚拟机（参见附录）。

## 以 lfs 用户构建编译临时工具链

在本节，我们将以 lfs 用户编译并安装临时工具链。该节内容对应于 LFS-BOOK 第 5 章。

### 脚本 part-2.sh

该脚本编译临时工具链。我们需要注意以下问题：

(1) 为避免损坏宿主系统，我们重启了宿主系统 host 并以 lfs 用户登录；

(2) 为节约时间，您可以以“sh part-2.sh > $LFS/tf/building\_output.log”的方式运行本脚本（但请体会其构建流程）并查看屏幕是否有错误输出；

(3) 若遇到致命错误，则需要重新编译相应部分；

(4) 临时工具链各个软件包的作用以及编译顺序，请参照 LFS-BOOK，此处不再赘述。

### 通用编译指南

对于每个软件包：

a. 以lfs用户解压要编译的软件包

b. 进入到解压后创建的目录中

c. 根据指南说明编译软件包

d. 回退到源文件目录

e. 除非特别说明，删除解压出来的目录（和为了编译而创建的build目录）

### gcc-4.9.2 编译问题

对于gcc-4.9.2编译过程中的一个错误（和cfns.h头文件有关），我们通过自己的补丁解决了它：

cd $LFS/sources/gcc-4.9.2

patch -p1 < cfns-4.9.2.patch

提示：补丁“cfns-4.9.2.patch”可以克隆创新实践课的仓库得到：

git clone https://gitee.com/openeuler-practice-courses/lfs-course.git

注意：这并非官方解决方案，但经过了本实验的验证。也欢迎同学们提出自己的解决方案。

### 其他需要注意的地方

Stripping节约的空间有限，可以不做。

## 在 chrooted 环境下构建 LFS 目标系统

为避免损坏宿主系统，本节我们在 chrooted 环境下构建 LFS 目标系统。我们会用到脚本 part-3.0.sh 和 part-3.1.sh。该节内容对应于 LFS-BOOK 第 6 章。

### 脚本 part-3.0.sh

在运行该脚本中的命令之前，我们是以 root 用户登录 host。该脚本完成如下事情：

(1) 创建并运行 mount-and-populate.sh 脚本（为进入 chrooted 环境做必要准备）；

(2) 创建并运行 chroot2lfs.sh 脚本（为了不损坏宿主系统，我们在 chrooted 环境下构建 LFS 目标系统。chrooted 环境也是最简单的虚拟化系统）；

(3) 其他一些配置。

### 脚本 part-3.1.sh

该脚本完成 LFS 目标系统基本系统软件的构建。有以下几点需要注意：

(1) 在运行该脚本之前，需要以 root 身份登录 host，然后依次运行在上一节创建的 mount-and-populate.sh 脚本和 chroot2lfs.sh 脚本进入 chrooted 环境；

(2) 可以以“sh part-3.1.sh > /tf/building\_output.log”的形式运行本脚本一次性的把基本系统软件构建完毕；

(3) 和 part-2.sh 脚本一样，我们让 make 以 4 线程运行（export MAKEFLAGS='-j 4'），您可以根据自己的情况设置 make 运行线程数目。但是对于某些软件包的编译，多线程 make 有时会带来意想不到的错误（一般是编译的各部分出现依赖时），这时请以“ make -j1”来进行编译（在编译“6.66. Man-DB-2.7.1”时我们就是这么做的）；

(4) 在编译“6.66. Man-DB-2.7.1”时，我们还会遇到一些其他错误，由于不影响最终系统的构建，我们暂且忽略了该错误，同学们若有兴趣可以自行研究并解决之；

(5) 如果发现运行 make 时出现“Segmentation fault (core dumped)”错误，请重启宿主系统、重新进入 chrooted 环境重新进行编译；

(6) 本节应该 make check 或 make test，因为耗时，并且在编写此实验手册时我们已经验证过了，所以同学们做这些实验时，我们暂且省略了。但有兴趣的同学应该运行“make check”以作检查；

(7) 在本节中，我们通过“passwd root”命令设置了密码，该 root 用户的密码即是最终 LFS 目标系统 root 用户的密码。

### mount & chroot

和LFS-BOOK不同的是，我们允许您不断地重启宿主机系统而不影响您继续构建（如果您以 step-by-step 的方式构建本节内容）。在每次重启后，需要重新依次执行mount-and-populate.sh 和 chroot2lfs.sh，然后继续运行 part-3.1.sh 中的步骤。

### 通用编译指南

对于每个软件包：

a. 进入chroot环境后解压要编译的软件包

b. 进入到解压后创建的目录中

c. 根据指南说明编译软件包

d. 回退到源文件目录

e. 除非特别说明，删除解压出来的目录（和为了编译而创建的build目录）

### 其他问题

由于节约不了多少空间，cleaning up被再次忽略。

## 配置 LFS 目标系统

本节对应 LFS-BOOK 第7章（脚本 part-3.2.sh），为 LFS 目标系统进行网络、主机名、/etc/hosts 文件的设置，其余的配置可以忽略。注意本节并没有配置一个真正可以访问互联网的网络，这一任务留在进阶实验中由学生们自己去探索。

### 第二个版本的chroot

完成上一节节的编译安装工作后，由于不再需要/tools目录，我们创建了第二个 chroot 脚本（chroot2lfs2.sh）。

### 脚本 part-3.2.sh

运行该脚本中命令，需要以 root 用户登录 host，然后：

(1) 运行前面步骤创建的 mount-and-populate.sh 脚本；

(2) 创建并运行脚本 chroot2lfs2.sh；

(3) 对 LFS 目标系统进行配置。

## 编译并安装内核

本节对应 LFS-BOOK 第8章（part-3.3.sh），进行内核的编译和安装。

### make menuconfig

需要注意的是，LFS-BOOK 中的make menuconfig这一步，除LFS-BOOK中所提到的那些配置选项外，需要选择和虚拟机提供的硬件相匹配的驱动（特别是硬盘的驱动），否则以后启动新系统时系统将会mount不上sdb。

比如说，如果您安装了SCSI的硬盘，则需要选择以下配置（应选尽选）：

# For SCSI Disk

Linux Kernel Configuration

-> Device Drivers

-> SCSI device support

-> SCSI disk support

同时也建议选中以下配置：

# For BusLogic

Linux Kernel Configuration

-> Device Drivers

-> SCSI device support

-> SCSI low-level drivers

-> BusLogic SCSI support

# For LSI Logic

Linux Kernel Configuration

-> Device Drivers

-> Fusion MPT device support

-> Fusion MPT (base + ScsiHost) drivers

并按 LFS-BOOK 中建议进行的配置（严格与其保持一致）：

# Refer to LFS-BOOK

General setup --->

[\*] open by fhandle syscalls [CONFIG\_FHANDLE]

[ ] Auditing support [CONFIG\_AUDIT]

[\*] Control Group support [CONFIG\_CGROUPS]

Processor type and features --->

[\*] Enable seccomp to safely compute untrusted bytecode [CONFIG\_SECCOMP]

Networking support --->

Networking options --->

<\*> The IPv6 protocol [CONFIG\_IPV6]

Device Drivers --->

Generic Driver Options --->

[ ] Support for uevent helper [CONFIG\_UEVENT\_HELPER]

[\*] Maintain a devtmpfs filesystem to mount at /dev [CONFIG\_DEVTMPFS]

[ ] Fallback user-helper invocation for firmware loading [CONFIG\_FW\_LOADER\_USER\_HELPER]

Firmware Drivers --->

[\*] Export DMI identification via sysfs to userspace [CONFIG\_DMIID]

File systems --->

[\*] Inotify support for userspace [CONFIG\_INOTIFY\_USER]

<\*> Kernel automounter version 4 support (also supports v3) [CONFIG\_AUTOFS4\_FS]

Pseudo filesystems --->

[\*] Tmpfs POSIX Access Control Lists [CONFIG\_TMPFS\_POSIX\_ACL]

[\*] Tmpfs extended attributes [CONFIG\_TMPFS\_XATTR]

### .config 文件

为节约大家时间，本实验提供了一个参考的内核配置文件：.config。该配置文件即是按照上述 make menuconfig 命令生成的。其有两个版本，一是设置了 SCSI 驱动，对应用到了 SCSI 硬盘的情况，二是没有设置 SCSI 驱动，对应没有用到 SCSI 硬盘的情况（其实设置了 SCSI 驱动的 .config 文件也适用于没有用到 SCSI 硬盘的情况）。

注意：虽然用到了我们提供的参考 .config 文件，但是您应该了解如何用“make menuconfig”来配置内核。

### Finding out which kernel target to build

对于 x86\_64 架构来说，bzImage 文件将会被创建（在本例中位于 linux-3.19/arch/x86\_64/boot/ 目录下）。

### 脚本 part-3.3.sh

在运行该脚本中的命令之前，您应该以 root 身份登录 host，然后依次执行 mount-and-populate.sh 和 chroot2lfs2.sh，然后进行内核的配置、编译以及安装工作。

### 其他问题

对于内核如何编译、安装以及内核如何引导，请参考本文列出的 Linux 操作系统参考资料，本文不再赘述。

## 设置 GRUB 使新系统可引导

本节对应脚本 part-3.4.sh 和 part-3.5.sh，涉及到 LFS-BOOK 第8章部分内容和第9章内容。

简单地说，我们首先配置 LFS 目标系统的 GRUB，然后将其配置信息设置到 host 的 GRUB 配置文件中，以达到通过宿主系统的启动菜单引导 LFS 目标系统的目的。

### 配置 LFS 目标系统 GRUB——part-3.4.sh

在运行该脚本中的命令之前，您应该以 root 身份登录 host，然后依次执行 mount-and-populate.sh 和 chroot2lfs2.sh，然后配置 LFS 目标系统的 GRUB。

LFS 目标系统（即目前的chrooted环境）可以按以下命令生成grub.cfg文件：

grub-install /dev/sdb

grub-mkconfig -o /boot/grub/grub.cfg

然后在本实验手册中，以 sed 命令将 LFS 目标系统 grub.cfg 中的配置信息提取出来并保存在一个文件中供后面的步骤使用。

### 更改宿主机 GRUB 配置——part-3.5.sh

在运行该脚本中的命令之前，您应该以 root 身份登录 host，然后对其 grub.cfg 文件的配置进行更改。

对于宿主机的GRUB设置，将LFS系统的grub.cfg中menuentry模块复制到宿主机/boot/grub2/grub.cfg中的相应位置即可。

比如说，$LFS/boot/grub/grub.cfg的第一menuentry快的包含如下内容：

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'GNU/Linux' --class gnu-linux --class gnu --class os $menuentry\_id\_option 'gnulinux-simple-*cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc*' {

load\_video

insmod gzio

insmod part\_msdos

insmod ext2

set root='hd1,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd1,msdos1 --hint-efi=hd1,msdos1 --hint-baremetal=ahci1,msdos1 cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc

else

search --no-floppy --fs-uuid --set=root *cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc*

fi

echo 'Loading Linux 3.19-lfs-7.7-systemd ...'

linux /boot/vmlinuz-3.19-lfs-7.7-systemd root=/dev/sdb1 ro

}

注意其中的ID号和您的不会是一样的。然后把这部分内容增加到宿主机文件/boot/grub2/grub.cfg的对应位置，如这条语句的后面：

### BEGIN /etc/grub.d/10\_linux ###

本实验手册是通过 sed 命令自动完成以上操作的（但是您应该了解如何利用 vim 等编辑工具来手动配置）。当然，您可更改这条菜单项的标题GNU/Linux，使之更加个性化一点。

## 进入新的系统

设置完 GRUB，现在终于到了进入新系统（即 LFS 目标系统）的时候了！

### Who is who

您应该知道，$LFS（即/mnt/lfs/）、sdb1、chrooted 环境都是同一片地方，只是我们这一次要以 LFS 目标系统进入了。

### How to get in

重启系统，会出现LFS系统的选择项，移动键盘的上下键选中它，按回车键进入，

Text

Description automatically generated

以root用户登录，输入在 6.25.3. Setting the root password 节设置的密码（脚本中已经将之放到 part-3.1.sh 的最后进行了）。



以上是进入新系统的情形。

### 脚本 part-3.6.sh

进入新系统后，可以运行该脚本里的一些命令以研究您的新系统。如：

uname -m

uname -r

uname -a

cat /etc/os-release

cat /etc/lfs-release

cat /etc/lsb-release

hostname

# 作业的提交

本课程要求学生进行课程总结、撰写博客并发表至主流技术论坛，还要通过PR（Pull Request）提交作业至 openEuler 开源社区仓库，掌握开源社区开发流程（若有兴趣，您还可以将课程总结以博客形式发表在 gitee 等社区博客）。

请大家通过我们的视频学习 PR 的提交，此处仅根据视频内容做一总结。

## 提交 PR 前的准备

Pull Request，在开源社区也常常被简称为PR，是自己修改源代码后，请求上游仓库采纳该修改时采取的一种行为。在提交 PR 之前，您应该已经完成以下操作：

在 [Gitee](https://gitee.com/) 平台（https://gitee.com/）上注册账号

注册账号时建议使用自己常用的私人邮箱注册。

签署个人 CLA

签署 CLA 的网址可以前往 openEuler 官方网站查看： <https://www.openeuler.org/zh/community/contribution/> ；

正确配置 git 账号

git config --global user.name "your-user-name"

git config --global user.email "your-email-address-on-gitee"

注意要配置成签署 CLA 的用户名和邮箱。

将您要提交 PR 的上游仓库 fork 到您自己的 gitee 账号下

比如，以 <https://gitee.com/openeuler-practice-courses/lanzhou_university_2021> 为例：



点击仓库右上角的“fork”按钮即可 fork。

本例假设 fork 到了一个叫 *Glibc* 的仓库中（您应该 fork 到自己仓库中）：

Graphical user interface, text, application, email

Description automatically generated

如图中所示，*Glibc* 为组织，而您的账号一般为个人，两者提交 PR 的流程基本是相同的。

完成后该仓库会出现在自己的主页下：

Graphical user interface, text, application

Description automatically generated

注意：如果之前已经fork该代码仓又想要删除后重新fork，则可以先进入自己的代码仓，然后进入“Settings（设置）”页面，点击左侧导航栏的“Delete（删除）”链接，在确认操作后，要求对用户进行密码校验确认。校验密码后即可删除仓库。

在gitee页面上fork完毕后，下面的步骤在git命令窗口运行git命令。

将自己账号下这个 fork 而来的仓库 git clone 到本地

git clone https://gitee.com/glibc/lanzhou\_university\_2021.git

cd lanzhou\_university\_2021/

git status # Be at master branch

创建自己的新分支

git git checkout -b *101010-zhaoxiaohu*

git status # Be in *101010-zhaoxiaohu* branch

注意命令中“*101010-zhaoxiaohu*”只是举例，同学们可以改成自己的“学号-姓名拼英”。

创建自己文件夹

mkdir *101010-zhaoxiaohu*

cd *101010-zhaoxiaohu*

在自己文件夹加入自己的实验截图等

此处以修改 README.md 为例进行说明：

touch README.md

vim README.md # Edit it, for instance.

进行 commit

git add ../*101010-zhaoxiaohu*/

git commit -s -a

和在 vim 中进行编辑相似，编辑 commit title 和 commit message。

第一次 push

git push --set-upstream origin *101010-zhaoxiaohu*

再次进行修改

此处仍以修改 README.md 文件为例：

vim README.md # Update it again, for instance.

再次 commit

git commit -s -a

再次 push

git push origin *101010-zhaoxiaohu* # the second, third... push

注意这时不需要像第一次那么 push 了。

查看 log

git log # Review

如此反复直到修改满意为止。

## 提交 PR

在前面的步骤中 git push 之后，自己仓库远端的代码上就会出现自己新建的分支（本例是 *101010-zhaoxiaohu*）：

Graphical user interface, text, application

Description automatically generated

在这个分支中检查自己提交的作业是否正常后就可以提交 PR 了。

新建 Pull Requests

点击“Pull Requests”按钮进入该页面：

Graphical user interface, text, application

Description automatically generated

点击“New Pull Request（新建 Pull Request）”按钮：

Graphical user interface, text, application

Description automatically generated

选择自己创建分支提交 PR

Graphical user interface, text, application, email

Description automatically generated

选中自己仓库的分支为刚才提交的分支，并填写标题文字和注释（注释请尽量详细，上图中的注释只是示例），然后点击右下角的“Create（创建）”按钮。

提交成功后原始远程仓（https://gitee.com/openeuler-practice-courses/lanzhou\_university\_2021）将会显示此次提交：

Graphical user interface, text, application, email

Description automatically generated

这时作为社区的一个 contributor，您的作业已经提交完毕。如果您提交的是代码，系统一般会自动进行 test。接下来要等待社区的 committer 进行 review，如果合乎要求，他/她会进行 /lgtm 动作。之后就需要社区的 maintainer 进行最后确认，如果一切顺利，他/她会进行 /approve 动作，这之后，您的作业（或代码）就会合到上游远程仓库里去。关于这些流程的具体命令及其含义请参考后文列出的相关链接。

## 参考链接

<https://clasign.osinfra.cn/sign/Z2l0ZWUlMkZvcGVuZXVsZXI=>

<https://gitee.com/openeuler/community/blob/master/zh/contributors/Gitee-workflow.md>

<https://gitee.com/openeuler/community/blob/master/en/sig-infrastructure/command.md>

<https://gitee.com/openeuler-practice-courses/lfs-course>

# 进阶任务

了解 QEMU 虚拟机技术以及 openEuler 20.09 及以上版本的 StratoVirt 虚拟化技术，尝试完成以下任务：

在 openEuler 开发环境，将刚构建出的 LFS 目标系统镜像作为虚拟机镜像，使用 QEMU 虚拟机启动它。

以 StratoVirt 虚拟机替代 QEMU 虚拟机，启动 LFS 目标系统镜像。

或者，

了解 openEuler 21.03 所用的5.10内核并以此构建 LFS 目标系统。

再者，

将此 LFS 实验移植到鲲鹏平台（基于 aarch64 架构）上。

相关资料可以参考以下链接：

<https://gitee.com/openeuler/kernel>

<https://www.qemu.org/>

<https://gitee.com/openeuler/stratovirt>

<https://repo.openeuler.org/openEuler-21.03/>

<https://www.hikunpeng.com/>

# 附录

## 如何将资料从 PC 复制到虚拟机

下面以在Windows 10中为例讲解如何将资料从 PC复制到虚拟机。我们假设将要拷贝的资料在本地 PC Windows 10 系统的 Downloads 文件夹下（其文件夹路径中不包含空格或中文名）。

在虚拟机中查看IP地址

ip a | grep 192.168

inet *192.168.56.102*/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8

可以看出该虚拟机本地IP地址为*192.168.56.102*。

在 PC 中打开 cmd 终端

在Windows的File Explorer中打开Downloads文件夹：

Graphical user interface, application, Word

Description automatically generated

然后在File Explorer的地址栏中输入“cmd”并按回车。这样会打开一个命令窗，并且路径就定位在Downloads文件夹下。

用scp命令进行拷贝

假设 lfs-packages-7.7-systemd.tar 文件已经下载到本地 PC 的 Downloads 文件夹。您可以用下列命令下载：

curl -O <https://zhuanyejianshe.obs.cn-north-4.myhuaweicloud.com/chuangxinshijianke/lfs-packages-7.7-systemd.tar>

然后用 scp 命令将其拷贝到虚拟机的 root 目录下：

scp ./lfs-packages-7.7-systemd.tar root@192.168.56.102:~/

这里只是拷贝一个文件，带“-r”选项还可以拷贝整个目录，请用“scp --help”查看其用法。

# 缩略语表

|  |  |  |
| --- | --- | --- |
| 缩略语 | 英文全称 | 中文释义 |
| ABI | Application Binary Interface | 应用程序二进制接口 |
| LFS | Linux From Scratch | 白手起家构建 Linux |
| LSB | Linux Standard Base | Linux 标准规范 |
| POSIX | Portable Operating System Interface for UNIX | UNIX 可移植操作系统接口 |
| VFS | Virtual File System | 虚拟文件系统 |