**操作系统课程设计实验报告**

实验题目： Linux 内核编译及添加系统调用

姓 名： 孙恺泽

学 号： 22270318

组 号： 无

专 业： 网络工程

班 级： 22272411

老师姓名： 曾英佩

日 期： 2024 年 04 月 20 日

目 录

[一 题目介绍 1](#_Toc9066)

[二 实验思路 2](#_Toc22209)

[三 遇到问题及解决方法 2](#_Toc10340)

[四 核心代码及实验结果展示 3](#_Toc6201)

[五 个人实验改进与总结 6](#_Toc25457)

[5.1 个人实验改进 6](#_Toc5843)

[5.2 个人实验总结 6](#_Toc19967)

[六 参考文献 6](#_Toc17199)

# 一 题目介绍

**1、实验介绍**

本实验通过修改Linux内核源码，添加新的Linux系统调用，替换编译后内核，并测试结果，了解Linux内核源码的编译方法和内核的安装方法，系统调用的概念、编写步骤和调用方法。

**2、任务描述**

* 掌握Linux 系统调用基本概念
* Linux 内核源码的编译和安装
* 添加Linux 的系统调用
* Linux 的系统调用的测试方法

**3、实验目的**

* 学习掌握Linux内核的编译和启用
* 学习掌握如何添加Linux 的系统调用
* 学习掌握测试Linux系统调用

# 二 实验思路

# 三 遇到问题及解决方法

1. 系统架构问题

刚做实验时没有使用华为云的弹性云服务器 ESC，而是使用的另外一台闲置的腾讯云轻量应用服务器，在编译 openEuler 内核时没有成功，因为这台腾讯云的服务器是 x86 架构。

因为x86架构和arm架构在指令集和硬件体系结构上存在显著差异。x86架构是复杂指令集计算机（CISC），而arm架构是精简指令集计算机（RISC）。x86架构的指令集更为复杂，可以使用内存或寄存器操作数进行ALU指令，而arm架构的指令仅在寄存器上操作，有少量指令用于从/向内存加载和存储数据。这种差异会影响到编译器生成的代码。

最终该实验还是使用的华为云弹性云服务器ESC完成。

1. 系统调用链接错误

在编译内核后的最终链接阶段出现，报错提示为“未定义的引用或符号”。查阅了相关资料，这种问题的产生原因通常是由于在添加新系统调用时，未正确配置系统调用表或相关文件，导致链接器无法找到相应的符号引用。后来发现在添加新系统调用时，需要在相应的头文件中声明系统调用函数的原型，并在宏定义中指定系统调用号。如果头文件或宏定义有误，链接器也会无法正确识别新系统调用。该问题最终得到解决。

# 四 核心代码及实验结果展示

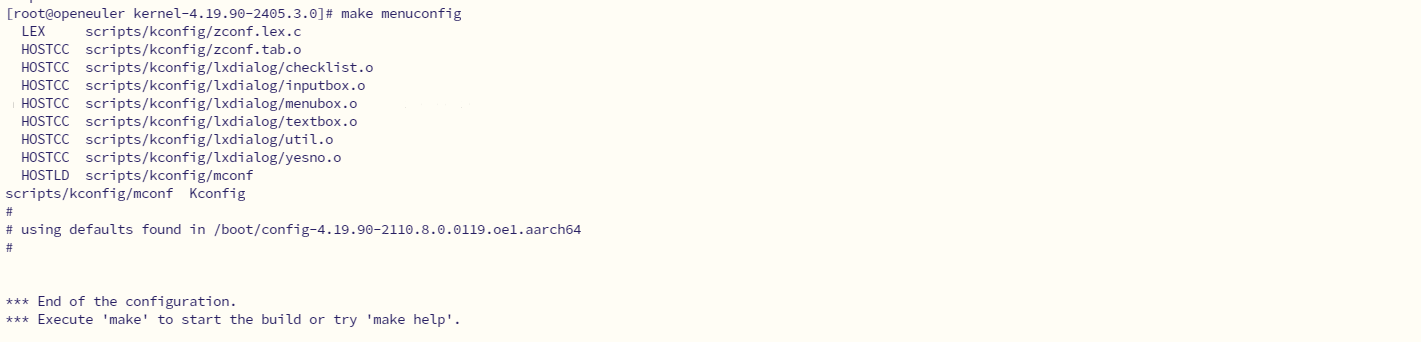
1. 从 GitHub 代码托管平台获取 openEuler 内核源码，并解压缩：

wget <https://gitee.com/openeuler/kernel/repository/archive/4.19.90-2405.3.0.zip>

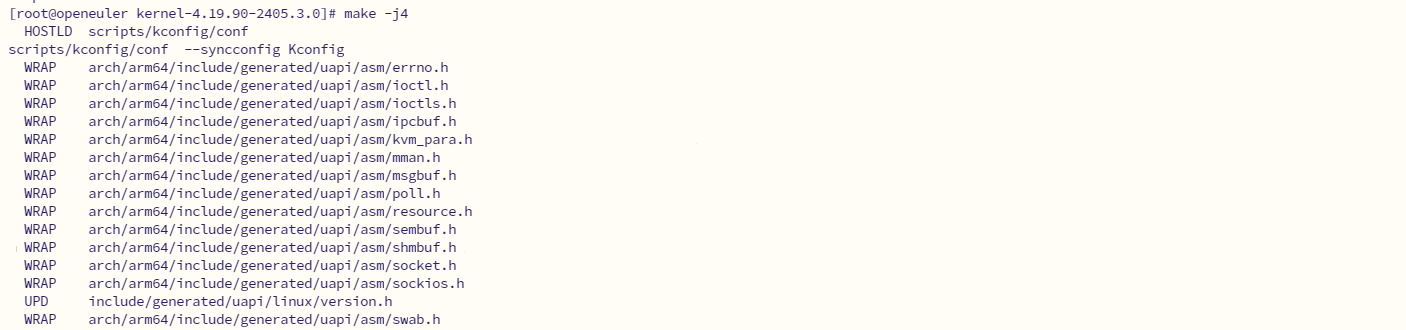


1. 编译内核

make openeuler\_defconfig

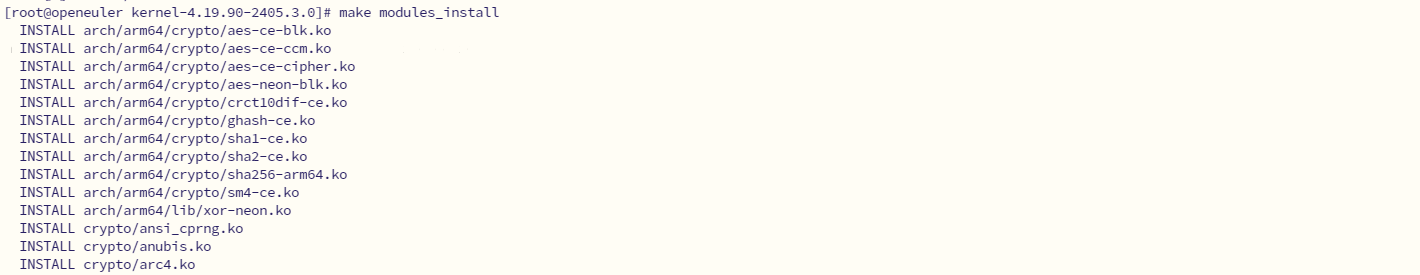


make -j4

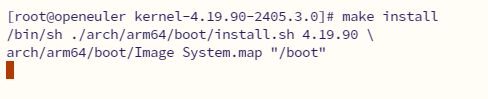


1. 安装内核

make modules\_install（安装内核模块）



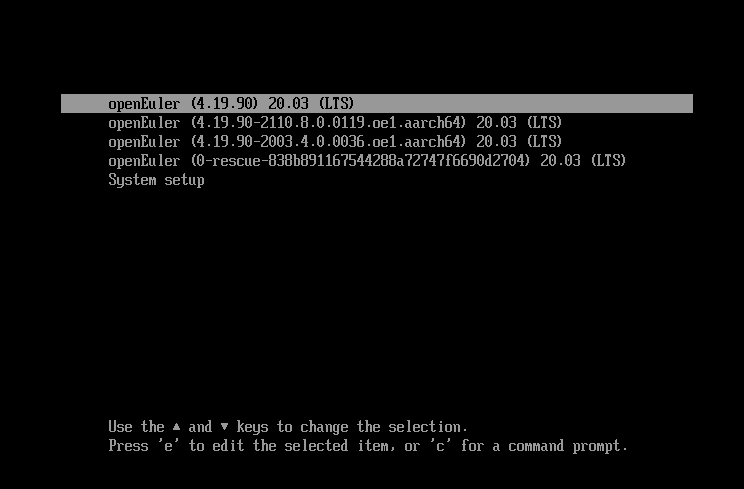
Make install（安装内核）

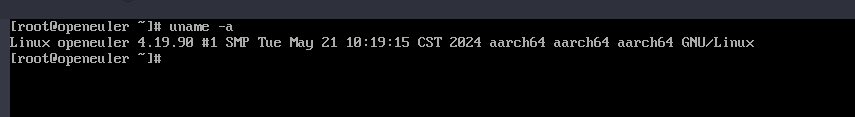


1. 更新系统引导，启动新编译的内核

grub2-mkconfig -o /boot/grub2/grub2.cfg

在华为云 VNC 远程登录页面进入新编译的系统：

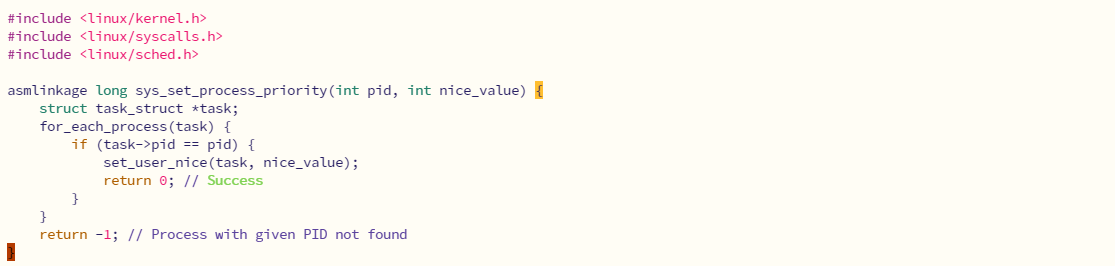




1. 系统调用

（1）修改或返回指定进程的优先级

在系统内核源码中创建一个目录，名为 my\_syscalls\_1，在该目录下创建一个 my\_syscalls\_1.c，在其中编写第一个系统调用：



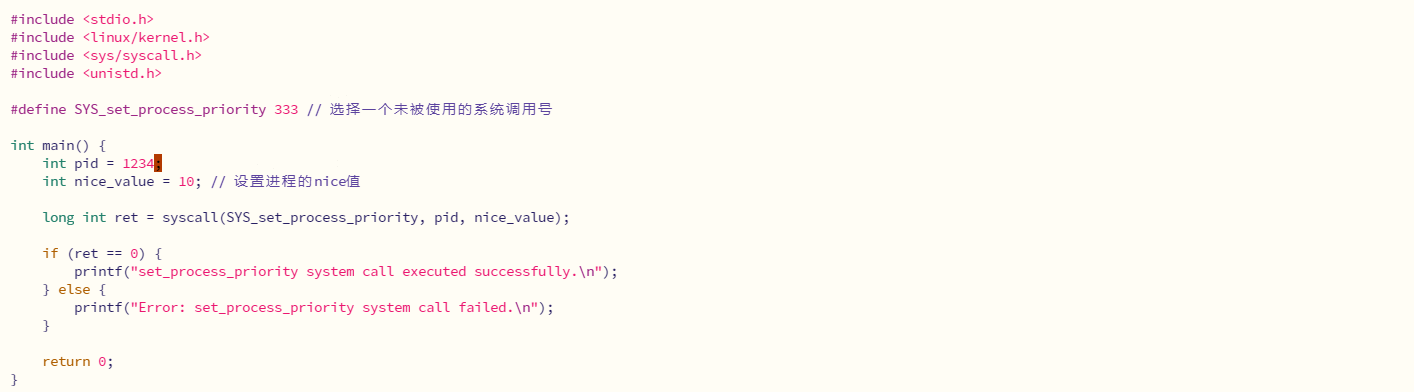
在该目录下再创建一个 Makefile 文件，写入：

obj-y := my\_syscalls.o

返回内核源码根目录，修改 Makefile 文件内容：

core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ my\_syscalls/

重新编译内核，并重启，编写 test\_1.c 编译运行，验证新添加的系统调用生效：



（2）返回当前系统的名称和版本

在系统内核源码中添加第二个系统调用：



其余步骤同上。重新编译运行后编写 test\_2.c 确认其效果。

1. 完成试验

完成实验后，关闭系统：



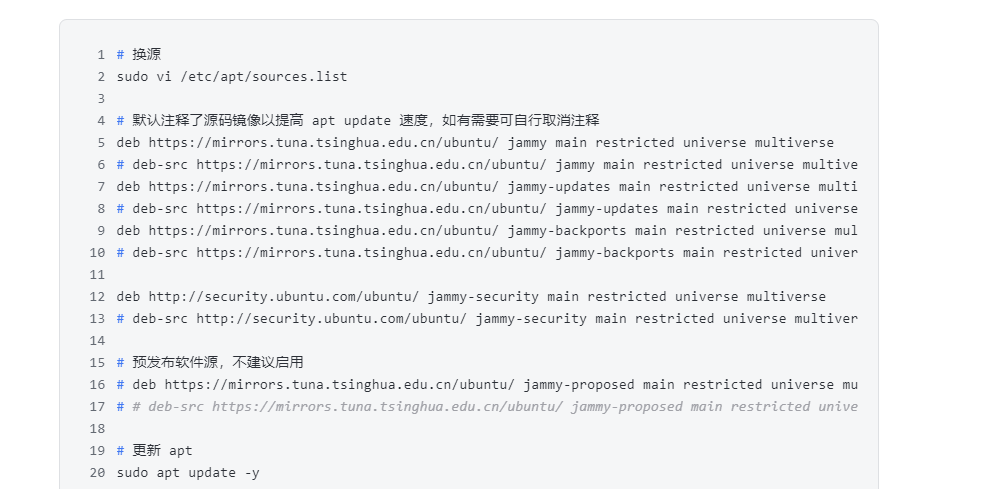
# 五 个人实验改进与总结

## 5.1 个人实验改进

有一说一，其实没有对实验做出很大的改进，因为在网上搜索的资料不够具体，而且很多文档没有提及具体的环境，或者说网上的文档普遍略过了问题的解决方法，以及问题产生的原理。

建议在实验过程中加入更详细的注释和文档说明。在编写新系统调用的过程中，应该养成良好的注释习惯，解释每个关键步骤的作用和原理。这样做不仅有助于他人理解代码，也有助于自己在日后回顾时快速理解。此外，编写一份详细的实验报告，记录整个实验的步骤、遇到的问题及解决方法，以及实验结果的展示。通过撰写实验报告，可以更好地总结实验经验，加深对Linux内核编译和系统调用的理解。

我也是在进行试验操作时编写了详细地文档，包括各个步骤可能遇到的问题、问题对应的原理和解决方法，然后编写了用于自动化的Shell脚本，方便之后的同学更快地完成操作系统实验内容。以下展示部分截图：



## 5.2 个人实验总结

在完成了一系列操作系统实验后，我深刻体会到了Linux内核编译和系统调用的精妙之处。从最初的架构选择错误，到系统调用链接的困难，每一个挑战都是对我的知识和解决问题能力的考验。通过这次实验，我不仅学会了如何从GitHub获取openEuler内核源码，还掌握了编译、安装和更新系统引导的技巧。更重要的是，我学会了如何添加新的系统调用，并确保它们在系统中正确运行。

这次实验的过程虽然充满了挑战，但也同样充满了成就感。每当我解决一个问题，或是成功地让新的系统调用运行起来时，那种满足感是无法用言语表达的。我也意识到了文档和注释的重要性。在实验过程中，详细的注释和文档不仅帮助我理解了每个步骤的原理，也为未来可能遇到相同问题的人提供了宝贵的参考。

# 六 参考文献

1. [实验一 openEuler操作系统安装与内核编译](https://blog.csdn.net/qq_46744173/article/details/122198126)
2. [Linux 中型实验 OpenEuler下添加新的系统调用](https://blog.csdn.net/XYH_233/article/details/126600823)
3. [Devv\_](https://devv.ai)