实验一 编译Linux内核

班级： 07112201 学号： 1120220715 姓名： 刘秉致

一、实验目的

一）理解操作系统内核的架构和功能

二）了解操作系统内核的编译过程

二、实验内容

在Fedora环境下替换和编译Linux内核，并将系统内核替换为新的内核。

三、实验步骤

1. **部署实验环境**

本次实验采用的环境如下：

|  |  |
| --- | --- |
| 虚拟机框架 | VMware Workstation 17.6.0 |
| 虚拟机系统 | Fedora Linux 40 (x86\_64) |
| 原始内核版本 | Linux 6.8.5 |
| 目标内核版本 | Linux 6.10.2 |

1. **依赖软件安装及目标内核下载**

实验依赖如下软件：gcc,make,openssl,flex,bison,openssl-devel,elfutils-libelf-devel,ncurses-devel,dwarves

使用命令进行安装

1. sudo yum install gcc make openssl flex bison openssl-devel elfutils-libelf-devel ncurses-devel dwarves

接下来下载并解压源码

1. mkdir linux\_kernel
2. cd linux\_kernel
3. wget https://mirrors.edge.kernel.org/pub/linux/kernel/v6.x/linux-6.10.12.tar.gz
4. tar xvf linux-6.10.12.tar.gz
5. **配置config文件**

Linux内核的构建过程会查找.config文件，也即内核的配置文件，用于指定Linux内核所有可能的配置选项。

当前系统中的内核配置文件为/boot/config-6.8.5-301.fc40.x86\_64，其中记录了当前内核在构建时的选项，可以使用命令进行查看

1. cat /boot/config-6.8.5-301.fc40.x86\_64

配置文件内部按功能分为不同区域，每个区域内包含了对多个功能的配置选项:

* Y表示将该功能编译进内核
* N表示不编译该功能
* M表示将该功能编译成可以在需要时动态加载到内核中的模块

由于编译选项过于繁杂，为了简化配置过程，这里使用当前内核的编译配置文件作为.config文件并使用make menuconfig来生成编译配置文件。

1. cd linux-6.10.12
2. cp /boot/config-6.8.5-301.fc40.x86\_64 .
3. make mrproper
4. mv config-6.8.5-301.fc40.x86\_64 .config
5. make menuconfig
6. **生成编译文件并安装**

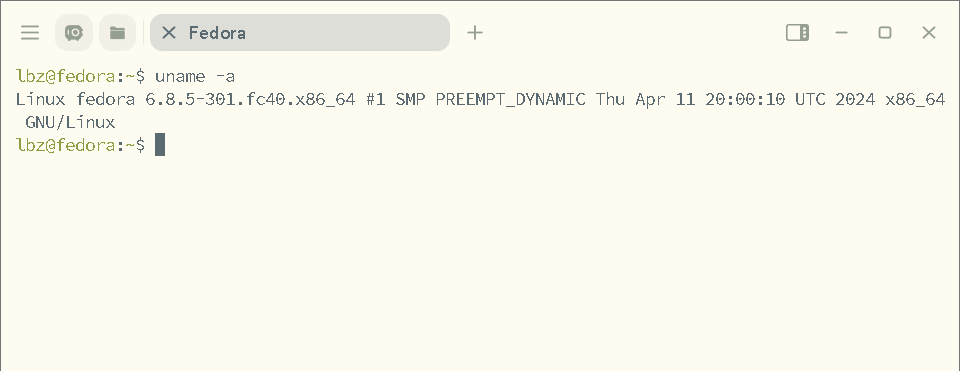
执行命令来进行多线程Make操作。

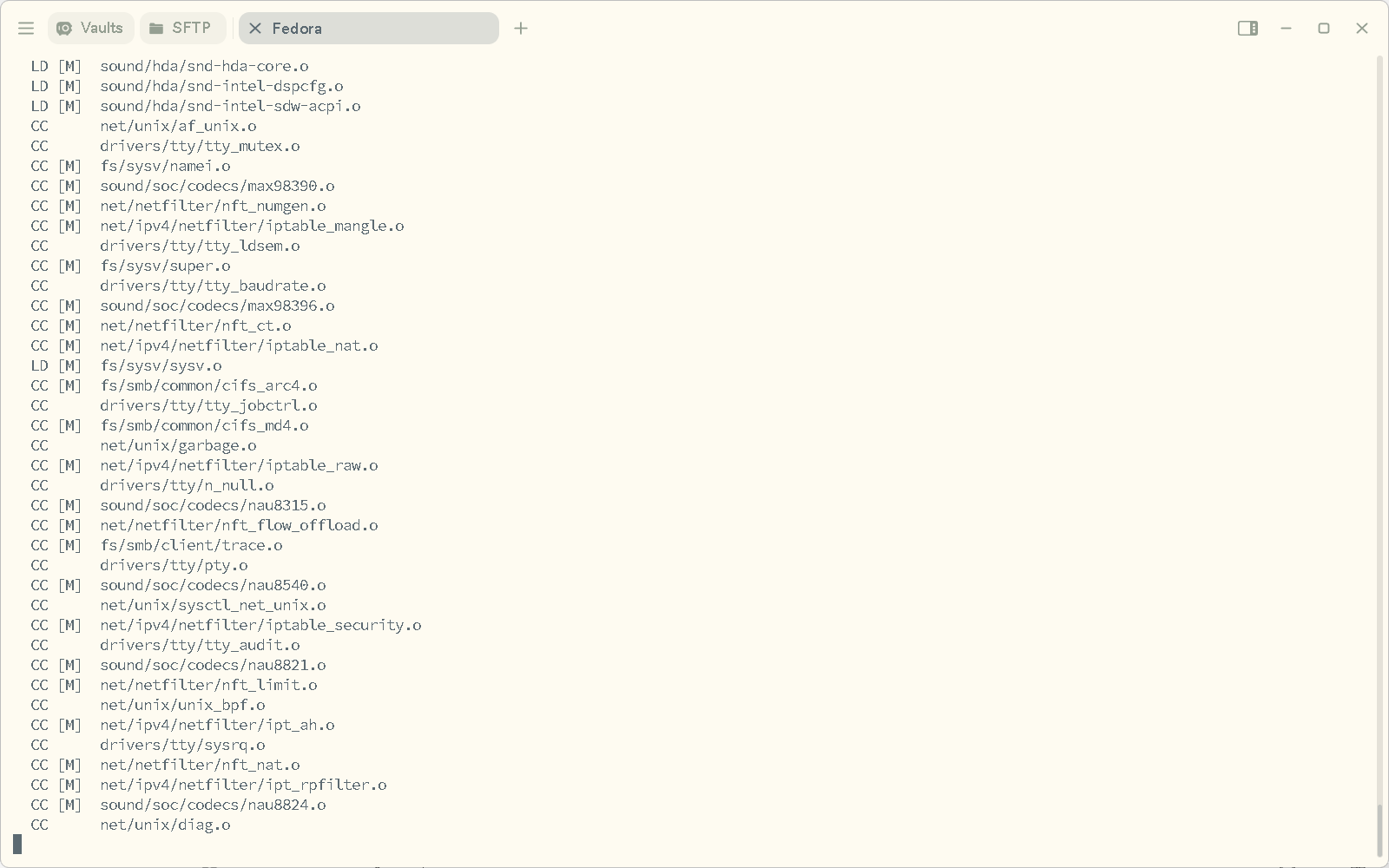
1. make -j$(nproc)

等待内核编译完毕后，运行命令安装编译后的内核并重新配置grub文件即可

1. make modules\_install
2. make install
3. reboot
4. sudo grub2-mkconfig -o /boot/grub2/grub.cfg

四、实验结果及分析

实验开始时的内核版本

实验过程中编译生成内核模块的可执行文件的过程

实验最终时的内核版本

五、实验收获与体会

这次实验中，我了解到了Linux内核的编译过程以及Linux内核的组织形式，并且了解到了一些功能模块的作用，如在编译过程中，由于我漏装openssl，导致部分功能模块的BTF不能正常生成，这引导我去查阅资料，认识和了解了BTF和CO-RE在Linux内核中的作用。

同时，编译的过程是十分漫长的，我成功编译内核文件大约花费了2小时左右，而之前由于各种依赖工具的缺失导致的编译错误所浪费的时间更是远不止这些。这让我对内核的认识有了不同的认识，正是因为内核的编译和安装花费的时间成本很高，内核才更加需要安全性和稳定性，避免频繁的编译、更新和修改。

附录：程序清单及说明（黑体四号字）

宋体小四号字，行距22磅

（列出文件名及说明即可，不需要在此处复制代码，代码直接以源文件形式提供，但源文件中对代码要有必要的注释和说明）