《操作系统原理》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 蔡国庆 | 学号 | U202112087 | 专业班级 | 网安2103 | 时间 | 2024.1.5 |

**一、实验目的**

1. 理解操作系统引导程序/BIOS/MBR的概念和作用
2. 理解并应用操作系统生成的概念和过程
3. 理解并应用操作系统操作界面，系统调用概念
4. 掌握和推广国产操作系统
5. **实验内容**
6. 用NASM编写MBR引导程序，在BOCHS虚拟机中测试。
7. 在Linux（建议Ubuntu或银河麒麟或优麒麟）下裁剪和编译Linux内核，并启用新内核。
8. 为Linux内核（建议Ubuntu或银河麒麟或优麒麟）增加2个系统调用，并启用新的内核，并编写应用程序测试。
9. 在Linux （建议Ubuntu或银河麒麟或优麒麟） 或Windows下，编写脚本或批处理脚本。参数1个：指定目录。脚本的作用是把指定目录中的全部文件的文件名 加后缀，后缀是执行脚本时的日期和时分。例如：文件名“test”变成“test2023-11-21-20-42”.

**三、实验环境和核心代码**

表 1 实验环境说明

|  |  |
| --- | --- |
| **项目** | **版本** |
| *虚拟机软件* | *VMware Workstation Pro 17* |
| *操作系统* | *Ubuntu 16.04* |
| *Linux内核包* | *Linux-4.14* |
| *内核编译工具* | *make* |
| *代码编译工具* | *gcc* |
| *代码解释器* | *Python 3.12* |
| *文本编辑工具* | *Vim* |

**3.1 实验内容3）：Linux内核添加系统调用**

我下载的Linux内核包版本为Linux-4.14。

首先需要在Linux内核的系统调用表中添加两个新添加的系统调用表项，系统调用表文件的路径为/usr/src/linux-4.14.330/arch/x86/entry/syscalls，由于所使用的计算机架构为64位，所以在对应的syscall\_64.tbl进行系统调用添加。使用vim编辑工具打开文件，在调用表的末尾添加两个系统调用表项，编号548对应Add函数，编号549对应Max函数。



图 1 系统调用表中添加新系统调用表项

接下来，打开/usr/src/linux-4.14.330/include/linux/syscalls.h进行修改。在该文件中，需要将刚刚添加的两个系统调用对应的系统调用处理函数Add与Max进行声明。



图 2 系统调用函数Add与Max声明

此外，还需要再打开/usr/src/linux-4.14.330 /kernel/sys.c文件，定义Add与Max系统调用函数。

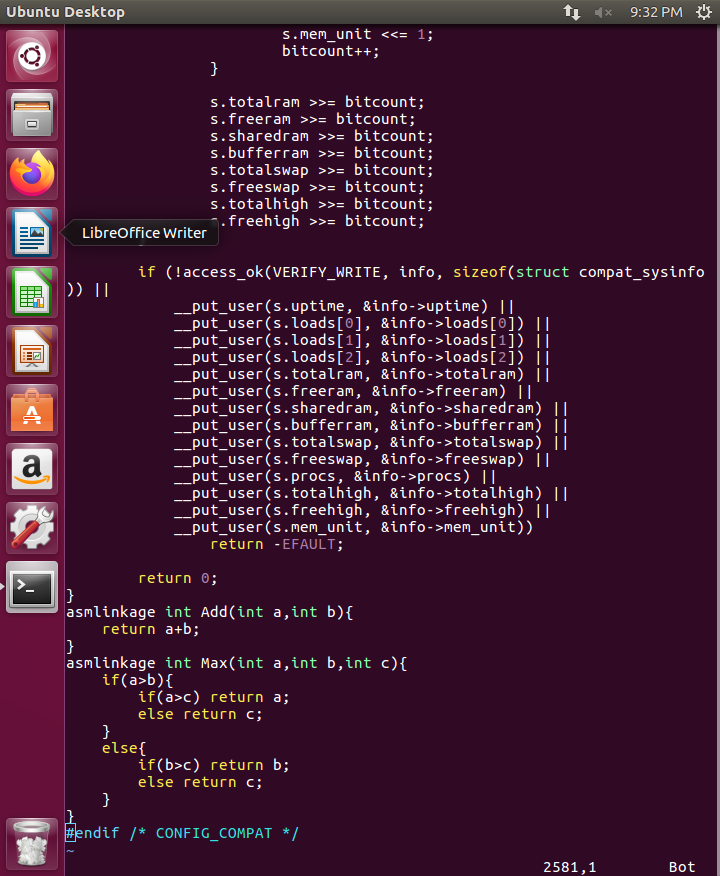


图 3 Add与Max函数实现代码

重新编译并安装添加了新系统调用的Linux内核包，随后重启虚拟机，进入修改过源码的系统，编写测试文件测试添加的系统调用，测试文件如下：

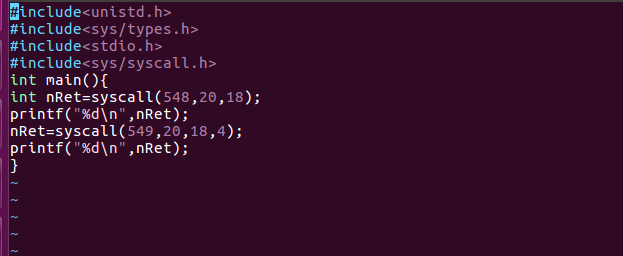
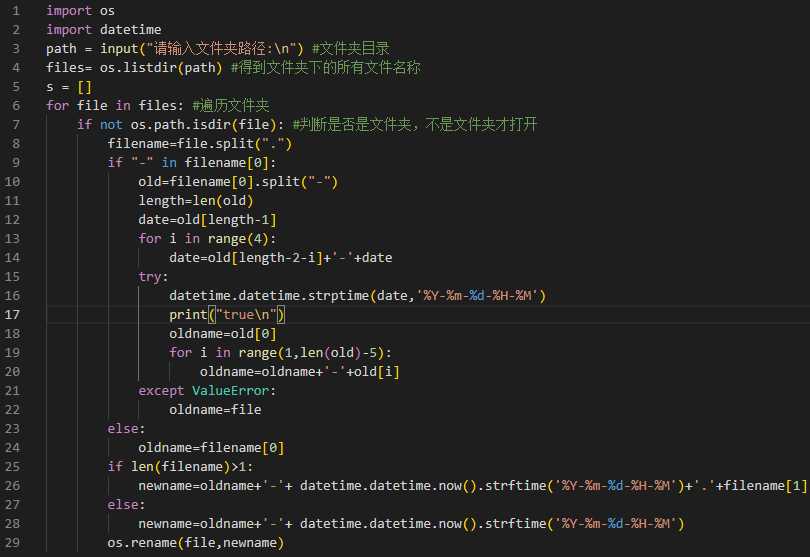


图 4 测试文件实现代码

**3.2 实验内容4）：编写时间后缀添加脚本**

由于对sh脚本语言不太熟练，所以使用python语言编写脚本实现，代码截图如下：

由于实验要求只为无时间后缀的问价添加时间后缀，为已有时间后缀的文件更新时间后缀，对于文件目录则不修改。所以，编写脚本时需要判断文件类型，调用os库中的os.path.isdir(file)函数进行判断。然后利用split函数分割文件名和文件后缀，接着根据“-”判断文件是否已有时间后缀，最后根据不同情况更新文件名

**四、实验结果**

**4.1 实验内容3）：Linux内核添加系统调用**

进入添加了系统调用的系统后，使用gcc编译器编译C语言测试程序并执行，输出结果分别为38与20，说明系统成功添加。且输出结果符合20+18=38，max{20，18，4}=20的计算结果，说明功能实现正常。

文本

描述已自动生成

图 5 系统调用测试结果示意图

**4.2 实验内容4）：编写时间后缀添加脚本**

使用mkdir命令创建一个用于测试的test文件目录：

表 2 test文件夹包含文件及预期修改结果

|  |  |  |
| --- | --- | --- |
| **文件名** | **类型** | **预期修改结果** |
| *test* | *文件* | *test-当前时间* |
| *test-2023-11-21-09-29.txt* | *文件* | *test-2当前时间.txt* |
| *testdir* | *文件目录* | *testdir* |

测试结果显示，test文件夹中文件都按照要求修改了后缀，而testdir文件目录名称未发生改变

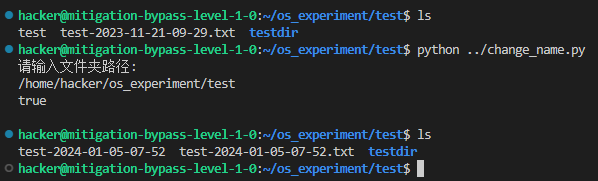


图 6 时间后缀添加脚本测试结果示意图

**五、实验错误排查和解决方法**

**5.1 实验内容3）：Linux内核添加系统调用**

完成该部分内容时，由于当时对Linux内核中实现系统不太了解，也不清楚系统调用涉及哪些文件，所以不确定具体需要更改哪些文件。于是上网查找试下系统调用的有关文件，发现这部分内容在理论课上讲到过，只是当时对具体实现细节并没有深入学习。通过翻看ppt与课本复习，确定需要在syscall\_table中加入调用表项，且在指定位置定义系统调用函数。由于实操和理论还是有一定不同，所以我又查找了相关教程，学习具体的操作步骤和涉及的内核文件。

在实现过程中，由于找的教程比较老，使用的是32位的系统架构，而我所安装的虚拟机使用的是64位，所以误修改了syscall\_32.tbl，导致编译内核时没有将添加的内容进行编译，安装后重新启动发现无法成功调用添加的系统调用，syscall始终返回-1。向成功完成的同学请教，明确了错误原因，改正后最终成功实现系统调用添加。

**5.2 实验内容4）：编写时间后缀添加脚本**

由于对shell脚本语言不太熟悉，所以使用了较为熟练的python语言进行编写。在完成实验任务的过程中，起初不知道如何判读文件和文件夹。查询后发现python的os库中有相关实现方法，所以直接调用。此外编写时没有遇到其他问题。为了学习shell脚本语言，我又找同学要了一份shell代码，阅读后发现shell中居然可以用正则式表达时间后缀，以此判断文件是否已具有时间后缀，和自己的代码比较后，感觉shell脚本更加简练高效。

**六、实验参考资料和网址**

1. [操作系统作业：给linux系统增加一个系统调用\_linux怎么向现有内核中加入一个新的系统调用-CSDN博客](https://blog.csdn.net/qq_41175905/article/details/80529311?ops_request_misc=&request_id=&biz_id=102&utm_term=)
2. <https://www.bing.com/>
3. **教学课件**