

《操作系统原理》实验报告

姓名	熊睿	学号	U202011107	专业班级	网安 2004	时间	2022.12.25
----	----	----	------------	------	---------	----	------------

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 (推荐银河麒麟或优麒麟, 建议)

二、实验内容

- 1) 用 NASM 编写 MBR 引导程序, 在 BOCHS 虚拟机中测试。
- 2) 在 Linux (建议 Ubuntu 或银河麒麟或优麒麟) 下载剪和编译 Linux 内核, 并启用新内核。(其他发行版本也可以)
- 3) 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。(其他发行版本也可以)
- 4) 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 test-2022-11-21-20-42。

三、实验过程

3.1 linux 内核编译

1) 获取内核源码

在官网下载 linux-5.10.155 内核源码压缩包, 拖入虚拟机中, 使用 tar 命令解压进/usr/src 目录。

2) 配置内核特性

进入解压后生成的/usr/src/linux-5.10.155 文件夹使用 make menuconfig 命令配置内核特性, 不用做修改。

3) 编译内核

使用 `make -j 8` 命令编译内核，8 为分配的物理内核数量两倍，可以提高编译速度。

```

certs COPYING fs init lib missing-syscalls.d modules-only.symvers
root@boots-VMware-Virtual-Platform:/usr/src/linux-5.10.155# make -j 8
SYSHDR arch/x86/include/generated/asm/unistd_64_x32.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_x32.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_64.h
SYSTBL arch/x86/include/generated/asm/syscalls_64.h
warning: Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel
CALL scripts/atomic/check-atomics.sh
CC arch/x86/kernel/asm-offsets.s
CALL scripts/checksyscalls.sh
CC arch/x86/platform/pvh/enlighten.o
AS arch/x86/entry/entry_64.o
CC arch/x86/events/core.o
CC init/main.o
CC arch/x86/hyperv/hv_init.o
CC certs/system_keyring.o
CC arch/x86/xen/enlighten.o
CC kernel/fork.o
AR arch/x86/platform/pvh/built-in.a
CC kernel/exec_domain.o
CC arch/x86/entry/syscall_64.o
CC certs/blacklist.o
CC arch/x86/xen/mmu.o
CC arch/x86/entry/common.o
CC kernel/panic.o
AR certs/built-in.a
CC kernel/cpu.o
CC arch/x86/hyperv/mmu.o
CHK include/generated/compile.h
CC init/do_mounts.o
CC arch/x86/xen/time.o
CC arch/x86/events/amd/core.o

```

图 1 编译内核

[illegible]

图 2 编译完成

4) 安装内核、安装内核模块

使用 `make modules install` 和 `make install` 命令安装内核、安装内核模块。

```
root@Boots-Vmware-Virtual-Platform:/usr/src/linux-5.10.155/lib# cd ..
root@Boots-Vmware-Virtual-Platform:/usr/src/linux-5.10.155# ls boot
config-5.10.0-5-generic  config-5.10.155.old  initrd.img-5.10.155  initrd.img-old  System.map-5.10.0-5-generic  System.map-5.10.155.old  vmlinuz-5.10.0-5-generic  vmlinuz-5.10.155.old
config-5.10.155        grub                initrd.img-5.10.0-5-generic  initrd.img-5.4.0-74-generic  System.map-5.10.155  vmlinuz                vmlinuz-5.10.0-5-generic  vmlinuz-5.10.155
```

图 3 模块编译完成

5) 重启后选择新内核

重启后选择新内核，并用 `uname -mrs` 命令验证新内核是否已经启用。

```
certs  COPYING      fs      init      lib      .m:
root@boots-VMware-Virtual-Platform:/usr/src/linux-5.10.155# uname -mrs
Linux 5.10.155 x86_64
root@boots-VMware-Virtual-Platform:/usr/src/linux-5.10.155# S
```

图 4 新内核版本



图 5 开机时可以选择启动哪个内核



图 6 原来版本内核内核

3.2 添加 linux 系统调用

1) 编写函数调用

在 `/usr/src/linux-5.10.155/kernel` 中的 `sys.c` 中增加函数调用。

```

        return -EFAULT;
    }
    return 0;
}
SYSCALL_DEFINE2(SSD_Add,int,x,int,y){
    printk("%d",x+y);
    return 0;
}
SYSCALL_DEFINE3(SSD_Max,int,a,int,b,int,c){
    if(a>b)b=a;
    if(b>c)c=b;
    printk("%d",c);
    return 0;
}
#endif /* CONFIG_COMPAT */

```

图 7 新增函数调用

2) 系统调用函数声明

在/usr/src/linux-5.10.155/include/linux 中的 syscalls.h 中增加函数声明。

```

int __sys_getsockopt(int fd, int level, int optname, char __user *optval,
                    int __user *optlen);
int __sys_setsockopt(int fd, int level, int optname, char __user *optval,
                    int optlen);
asmlinkage long sys_SSD_Add(int x, int y);
asmlinkage long sys_SSD_Max(int a, int b, int c);

```

图 8 新增函数声明

3) 声明系统调用编号

在/usr/src/linux-5.10.155/arch/x86/entry/syscalls 的 syscall_64.tbl 中新增声明。

438	common	pidfd_getfd	sys_pidfd_getfd
439	common	faccessat2	sys_faccessat2
440	common	process_madvise	sys_process_madvise
498	64	SSD_Add	sys_SSD_Add
499	64	SSD_Max	sys_SSD_Max

```

#
# Due to a historical design error, certain syscalls are numbered differently
# in x32 as compared to native x86_64. These syscalls have numbers 512-547.
# Do not add new syscalls to this range. Numbers 548 and above are available
# for non-x32 use.

```

图 9 注册新调用

图 12 运行测试程序

3.3 编写批处理程序

1) 编写批处理程序

```
for each in $1/*
do
    #文件夹，不改名
    if [ -d $each ]; then
        echo "$each is a directory."
    #文件，改名
    elif [ -f $each ]; then
        echo "$each is a normal file."
        #通配符判断是否已经有日期后缀
        if [[ $each == *-20[0-9][0-9]-[012][0-9]-[0123][0-9]-[012][0-9]-
[0-5][0-9]-[0-5][0-9] ]]; then
            echo "$each >> ${each%%-*}-`date +%Y-%m-%d-%H-%M-%S`"

            mv $each ${each%%-*}-`date +%Y-%m-%d-%H-%M-%S`

        else
            echo "$each >> $each-`date +%Y-%m-%d-%H-%M-%S`"
            mv $each $each-`date +%Y-%m-%d-%H-%M-%S`

        fi
    fi
done
~
```

图 13 编写批处理程序

2) 运行批处理程序

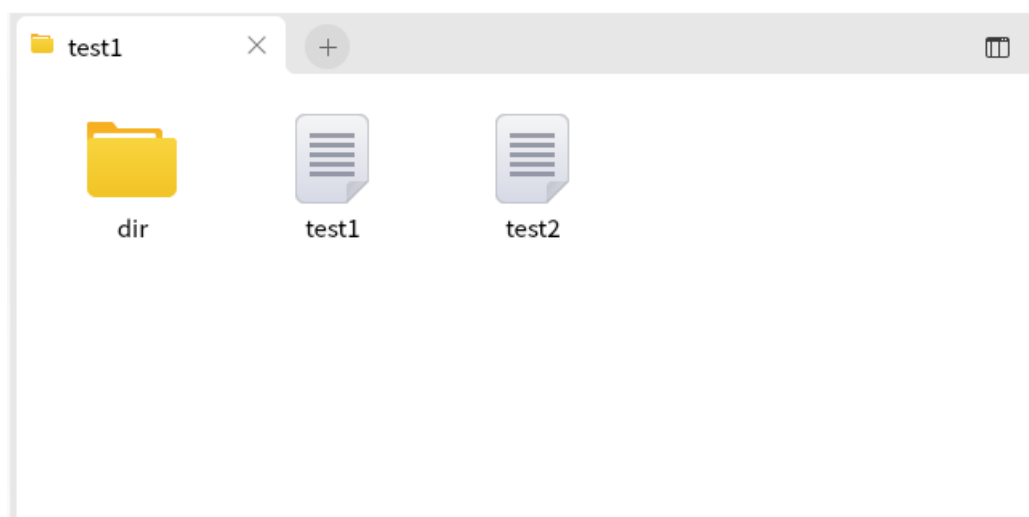


图 14 运行之前

```
oots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh ./test1
/test1/a-2022-11-23-14-53-34 is a normal file.
/test1/a-2022-11-23-14-53-34 >> ./test1/a-2022-12-26-13-51-15
/test1/b is a directory.
/test1/c-2022-11-23-14-53-34 is a normal file.
/test1/c-2022-11-23-14-53-34 >> ./test1/c-2022-12-26-13-51-15
oots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh ./test1
/test1/dir is a directory.
/test1/test1 is a normal file.
/test1/test1 >> ./test1/test1-2022-12-26-13-55-15
/test1/test2 is a normal file.
/test1/test2 >> ./test1/test2-2022-12-26-13-55-15
```

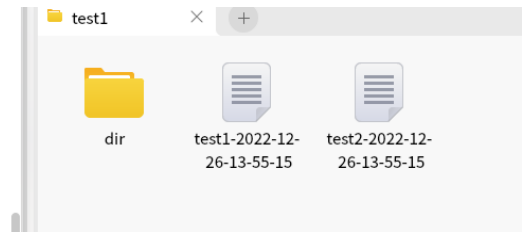


图 15 运行之后

四、实验结果

4.1 Linux 内核编译

```
certs COPYING fs init lib .m:
root@boots-VMware-Virtual-Platform:/usr/src/linux-5.10.155# uname -mrs
Linux 5.10.155 x86_64
root@boots-VMware-Virtual-Platform:/usr/src/linux-5.10.155# s
```

图 16 新内核编译成功版本

4.2 添加 Linux 系统调用

```
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
[ 3784.141800] usb 2-2.1: new full-speed USB device number 5 us
[ 3784.361976] usb 2-2.1: New USB device found, idVendor=0e0f,
cdDevice= 1.00
[ 3784.361980] usb 2-2.1: New USB device strings: Mfr=1, Produc
=3
[ 3784.362461] usb 2-2.1: Product: Virtual Bluetooth Adapter
[ 3784.362466] usb 2-2.1: Manufacturer: VMware
[ 3784.362468] usb 2-2.1: SerialNumber: 000650268328
[ 3785.018668] EXT4-fs (sda6): mounted filesystem with ordered
(null)
[ 3785.027903] EXT4-fs (sda6): re-mounted. Opts: (null)
[ 3842.682787] usb 2-2.1: USB disconnect, device number 5
[ 3843.076611] usb 2-2.1: new full-speed USB device number 6 us
[ 3843.321670] usb 2-2.1: New USB device found, idVendor=0e0f,
cdDevice= 1.00
[ 3843.321676] usb 2-2.1: New USB device strings: Mfr=1, Produc
=3
[ 3843.321687] usb 2-2.1: Product: Virtual Bluetooth Adapter
[ 3843.321690] usb 2-2.1: Manufacturer: VMware
[ 3843.321692] usb 2-2.1: SerialNumber: 000650268328
[ 4128.393163] 30
[ 4128.393304] 9
[ 4134.682130] 30

文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
#include <unistd.h>
#include <sys/syscall.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
    long ret;
    ret = syscall(498,12,10); //Add, ret = 30
    printf("ret:%ld\n",ret);
    ret = syscall(499,7,8,9);
    printf("ret:%ld\n",ret);
}
```

图 17 添加 Linux 系统调用

4.3 编写批处理程序

```
oots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh ./test1
/test1/a-2022-11-23-14-53-34 is a normal file.
/test1/a-2022-11-23-14-53-34 >> ./test1/a-2022-12-26-13-51-15
/test1/b is a directory.
/test1/c-2022-11-23-14-53-34 is a normal file.
/test1/c-2022-11-23-14-53-34 >> ./test1/c-2022-12-26-13-51-15
oots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh ./test1
/test1/dir is a directory.
/test1/test1 is a normal file.
/test1/test1 >> ./test1/test1-2022-12-26-13-55-15
/test1/test2 is a normal file.
/test1/test2 >> ./test1/test2-2022-12-26-13-55-15
```

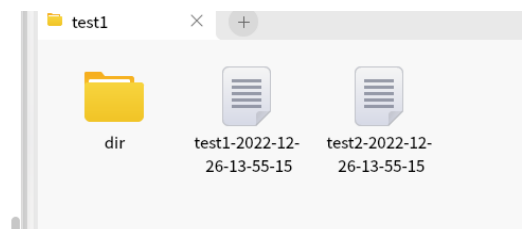


图 18 添批处理程序运行成功

五、实验错误排查和解决方法

5.1 Linux 内核编译

1) 内核选择错误

我的优麒麟的 Linux 版本是 Linux-5.10.0, 而我一开始选择的编译的内核是 Linux-5.9.6, 导致 `make modules_install` 时发生错误, 提示没有 `modules.order` 文件或是目录, 更换内核 5.10.155 后可以执行成功, 所以编译的时候尽量选择版本相近的内核.

```
sed: 无法读取 modules.order: 没有那个文件或目录
make: *** [Makefile:1416: modinst] 错误 2
```

图 19 编译失败错误

2) 读写权限错误

根据提示可以看到 `lib/raid6/mktables:Permission denied`, 需要给 `lib/raid6/mktables` 这个文件读写权限, 利用 `chmod 777` 指令进行操作, 或者直接用 `root` 账号进行实验.

```
CONTAINERS
CALL      scripts/checksyscalls.sh
CALL      scripts/atomic/check-atomics.sh
DESCEND   objtool
CHK       include/generated/compile.h
CHK       kernel/kheaders_data.tar.xz
TABLE     lib/raid6/tables.c
/bin/sh: 1: lib/raid6/mktables: Permission denied
make[2]: *** [lib/raid6/Makefile:90: lib/raid6/tables.c] 错误 126
make[2]: *** 正在删除文件“lib/raid6/tables.c”
make[1]: *** [scripts/Makefile.build:500: lib/raid6] 错误 2
make: *** [Makefile:1784: lib] 错误 2
```

图 20 读写权限错误

5.2 添加 Linux 系统调用

1) 不理解如何观察 Linux 系统调用

按照实验要求, 进行 `test.c` 的编写和编译生成 `exe` 文件, 执行 `exe` 文件后没有在终端中直接观察到变化和输出, 通过和同学交流讨论, 才知道, 可以用 `dmesg` 和 `dmesg -c` 来观察, 并且运行程序要用 `./` (指代当前目录) 来运行.

```
boots@boots-VMware-Virtual-Platform:~/桌面$ gcc test.c -o 111
boots@boots-VMware-Virtual-Platform:~/桌面$ 111
Could not find command-not-found database. Run 'sudo apt update' to populate it.
111: 未找到命令
boots@boots-VMware-Virtual-Platform:~/桌面$ ./111
ret:0
ret:0
boots@boots-VMware-Virtual-Platform:~/桌面$
```

图 21 无法观察输出

5.3 编写批处理程序

1) 脚本文件执行命令不熟悉

还是脚本命令执行不是很熟悉，在查找相关资料后了解，对于 sh 文件的执行，需要指定一个文件路径，例如./test1 表面当前路径下的 test1 文件夹，如果只执行./change.sh 默认对/路径进行执行显示如下图所示。

```
boots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh
/backup is a directory.
/bin is a directory.
/boot is a directory.
/cdrom is a directory.
/data is a directory.
/dev is a directory.
/etc is a directory.
/home is a directory.
/lib is a directory.
/lib32 is a directory.
/lib64 is a directory.
/libx32 is a directory.
/lost+found is a directory.
/media is a directory.
/mnt is a directory.
/opt is a directory.
/proc is a directory.
/root is a directory.
/run is a directory.
/sbin is a directory.
/srv is a directory.
/sys is a directory.
```

图 22 change.sh 执行输出

我们需要对自己创建的文件夹里的内容进行修改，文件夹不修改，文件修改成执行是日期名字，所以使用./change.sh ./test1 命令，观察到预期的结果。



```
boots@boots-VMware-Virtual-Platform: ~/桌面
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
boots@boots-VMware-Virtual-Platform:~/桌面$ ./change.sh ./test1
./test1/dir is a directory.
./test1/test1-2022-12-26-13-55-15 is a normal file.
./test1/test1-2022-12-26-13-55-15 >> ./test1/test1-2022-12-26-15-13-05
./test1/test2-2022-12-26-13-55-15 is a normal file.
./test1/test2-2022-12-26-13-55-15 >> ./test1/test2-2022-12-26-15-13-05
boots@boots-VMware-Virtual-Platform:~/桌面$
```

图 23 ./change.sh ./test1 执行输出

六、实验参考资料和网址

- (1) 教学课件
- (2) [Adding a New System Call — The Linux Kernel documentation](#)
- (3) https://blog.csdn.net/qq_46106285/article/details/121507087

(4) 《一个操作系统的实现》