# 《操作系统原理》实验报告

姓名	杜宇晗	学号	U202112151	专业班级	信安 2104	时间	2023.11.21
----	-----	----	------------	------	---------	----	------------

## 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)下裁剪和编译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"test2023-11-21-20-42"

#### 三、实验环境和核心代码

#### 3.1 编写 MBR 引导程序并测试

实验环境: VMware Workstation Pro 17

UbuntuKylin 20.04

内核版本: 5.10.0-1023-oem

编辑工具: vim

安装 nasm

## 创建引导扇区

编写汇编代码,使用命令 vim boot. asm 创建 boot. asm 文件写入以下代码:

```
- □ ×
 kingqaquuu@ubuntu: ~/Desktop
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
org 07c00h
mov ax, cs
mov ds, ax
mov es, ax
call DispStr
jmp $
DispStr:
mov ax, BootMessage
mov bp, ax
mov cx, 16
mov ax, 01301h
mov bx, 000ch
mov dl, 0
int 10h
ret
BootMessage: db "Hello, OS world!"
times 510_- ($-$$) db 0
dw 0xaa55
```

输入:wq 保存并退出。再使用命令 nasm boot.asm -o boot.bin 生成 boot.bin 文件。

### 安装 Bochs

kingqaquuu@ubuntu:~/Desktop\$ sudo apt-get install vgabios bochs bochs-x bximage

# 使用 bximage 创建虚拟软盘

输入命令 bximage, 显示如下画面

```
kinggaguuu@ubuntu:~/Desktop$ bximage
                             bximage
 Disk Image Creation / Conversion / Resize and Commit Tool for Bochs
      $Id: bximage.cc 13481 2018-03-30 21:04:04Z vruppert $

    Create new floppy or hard disk image

Convert hard disk image to other format (mode)
Resize hard disk image

    Commit 'undoable' redolog to base image

Disk image info
0. Quit
Please choose one [0] 1
Create image
Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd
Choose the size of floppy disk image to create.
Please type 160k, 180k, 320k, 360k, 720k, 1.2M, 1.44M, 1.68M, 1.72M, or 2.88M.
 [1.44M]
What should be the name of the image?
[a.img]
Creating floppy image 'a.img' with 2880 sectors
The following line should appear in your bochsrc:
 floppya: image="a.img", status=inserted
```

选择生成 fd (软盘), 其他选择可以使用默认值,直接按回车。完成这一步后,当前目录就生成了软盘映像 a. img。

#### 将引导扇区写入软盘

使用命令 dd if=boot.bin of=a.img bs=512 count=1 conv=notrunc

#### 编写 Bochs 的配置文件

使用命令 vim bochsrc,将以下内容写入

# kingqaquuu@ubuntu: ~/Desktop

文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)

megs: 32

romimage: file=/usr/share/bochs/BIOS-bochs-latest vgaromimage: file=/usr/share/vgabios/vgabios.bin

floppya: 1 44=a.img, status=inserted

boot: floppy

log: bochsout.txt

mouse: enabled=<mark>0</mark>

# 启动 Bochs 虚拟机

输入命令 bochs -f bochsrc 进入后,返回终端,输入c按回车,则会直接显示出 Hello, OS world。

# 3.2 编译新内核并启用

实验环境: VMware Workstation Pro 17

UbuntuKylin 20.04

内核版本: 5.10.0-1023-oem

编辑工具: vim

uname -r 查看自己的 linux 内核为 5.10.0-1023-oem

kingqaquuu@ubuntu:~/Desktop\$ uname -r
5.10.0-1023-oem

从官网下载 linux-6.1.63 内核源码到/usr/src

安装需要的包

apt-get install lib32ncurses5-dev

apt-get install gcc gdb bison flex

apt-get install libncurses5-dev libssl-dev

apt-get install libidn11-dev libidn11

apt-get install zlibc minizip

```
apt-get install build-essential openssl
apt-get install libelf-dev
依次执行
    root@ubuntu:/usr/src/linux-6.1.63# make mrproper
    root@ubuntu:/usr/src/linux-6.1.63# make clean
    root@ubuntu:/usr/src/linux-6.1.63# make menuconfig
                                                                     - 🗆 X
 root@ubuntu: /usr/src/linux-6.1.63
文件(F) 编辑(E) 视图(V)
                        搜索(S) 终端(T) 帮助(H)
 .config - Linux/x86 6.1.63 Kernel Configuration
                    Linux/x86 6.1.63 Kernel Configuration
    Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
    submenus ----). Highlighted letters are hotkeys. Pressing <Y>
    includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
    exit, <?> for Help, </> for Search. Legend: [*] built-in []
            General setup --->
        [*] 64-bit kernel
            Processor type and features --->
        [*] Mitigations for speculative execution vulnerabilities (NEW)
            Power management and ACPI options --->
            Bus options (PCI etc.)
            Binary Emulations --->
        [*] Virtualization
            General architecture-dependent options
        [*] Enable loadable module support
        v(+)
          <Select>
                      < Exit >
                                 < Help >
                                             < Save >
                                                         < Load >
```

```
root@upuntu: /usr/src/iinux-6.1.63
 文件(F) 编辑(E)
                  视图(V)
                           搜索(S) 终端(T) 帮助(H)
# using defaults found in /boot/config-5.10.0-1023-oem
.config:421:warning: symbol value 'm' invalid for I8K
.config:2396:warning: symbol value 'm' invalid for MTD NAND ECC SW HAMMING
.config:2581:warning: symbol value 'm' invalid for PVPANIC
.config:8549:warning: symbol value 'm' invalid for VIDEO ZORAN DC30
.config:8550:warning: symbol value 'm' invalid for VIDEO ZORAN ZR36060
.config:8551:warning: symbol value 'm' invalid for VIDEO ZORAN BUZ
.config:8552:warning: symbol value 'm' invalid for VIDEO ZORAN DC10
.config:8553:warning: symbol value 'm' invalid for VIDEO ZORAN LML33
.config:8554:warning: symbol value 'm' invalid for VIDEO ZORAN LML33R10
.config:8555:warning: symbol value 'm' invalid for VIDEO ZORAN AVS6EYES
.config:9581:warning: symbol value 'm' invalid for ANDROID BINDER IPC
.config:9582:warning: symbol value 'm' invalid for ANDROID BINDERFS
.config:10323:warning: symbol value 'm' invalid for CRYPTO BLAKE2S X86
.config:10419:warning: symbol value 'm' invalid for CRYPTO ARCH HAVE LIB BLAKE2S
.config:10420:warning: symbol value 'm' invalid for CRYPTO LIB BLAKE2S GENERIC
configuration written to .config
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
lroot@ubuntu:/usr/src/linux-6.1.63#
```

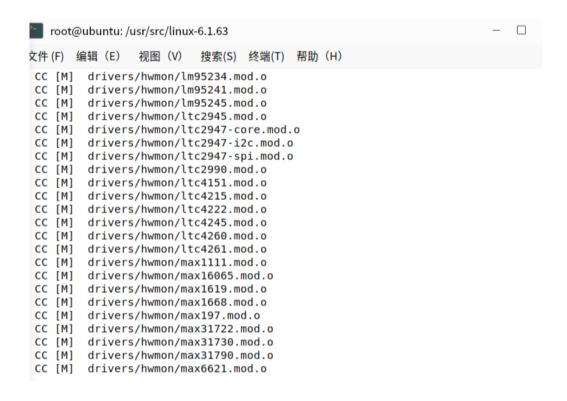
在. config 里

把两个 CONFIG\_SYSTEM 设置改为空字符串

```
root@ubuntu: /usr/src/linux-6.1.63
                                                                          文件(F) 编辑(E) 视图(V)
                           搜索(S) 终端(T) 帮助(H)
CONFIG PKCS7 TEST KEY=m
CONFIG SIGNED PE FILE VERIFICATION=y
# CONFIG FIPS SIGNATURE SELFTEST is not set
# Certificates for signature checking
CONFIG MODULE SIG KEY="certs/signing key.pem"
CONFIG MODULE SIG KEY TYPE RSA=v
# CONFIG MODULE SIG KEY TYPE ECDSA is not set
CONFIG SYSTEM TRUSTED KEYRING=V
CONFIG SYSTEM TRUSTED KEYS=""
CONFIG SYSTEM EXTRA CERTIFICATE=y
CONFIG SYSTEM EXTRA CERTIFICATE SIZE=4096
CONFIG SECONDARY TRUSTED KEYRING=y
CONFIG SYSTEM BLACKLIST KEYRING=y
CONFIG SYSTEM BLACKLIST HASH LIST=""
# CONFIG SYSTEM REVOCATION LIST is not set
# CONFIG SYSTEM BLACKLIST AUTH UPDATE is not set
# end of Certificates for signature checking
CONFIG_BINARY PRINTF=y
|-- 插入 --
                                                              11017,29
                                                                            94%
```

#### 然后开始编译 输入下列命令

make -j16



make modules

make install

```
root@ubuntu: /usr/src/linux-6.1.63
                                                                              \square ×
                  视图 (V)
文件(F) 编辑(E)
                            搜索(S) 终端(T) 帮助(H)
z-6.1.63
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.1.63 /boot/vmlinu;
-6.1.63
update-initramfs: Generating /boot/initrd.img-6.1.63
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 6.1.63 /boot/vml
inuz-6.1.63
run-parts: executing /etc/kernel/postinst.d/update-notifier 6.1.63 /boot/vmlinu;
-6.1.63
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.1.63 /boot/vmlinuz-
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
正在生成 grub 配置文件 ..
找到主题: /usr/share/grub/themes/UKUI/theme.txt
找到 Linux 镜像: /boot/vmlinuz-6.1.63
找到 initrd 镜像: /boot/initrd.img-6.1.63
找到 Linux 镜像: /boot/vmlinuz-5.10.0-1023-oem
找到 initrd 镜像: /boot/initrd.img-5.10.0-1023-oem
找到 Linux 镜像: /boot/vmlinuz-5.4.0-73-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-73-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
root@ubuntu:/usr/src/linux-6.1.63# update-grub2
```

重启后选择内核 6.1.63



载入 Linux 6.1.63 ... 载入初始化内存盘... -

# 3.3 增加2个新调用并启用测试

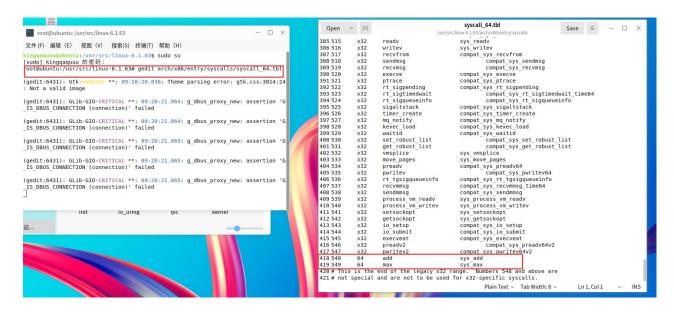
实验环境: VMware Workstation Pro 17

UbuntuKylin 20.04

内核版本: 5.10.0-1023-oem

编辑工具: gedit

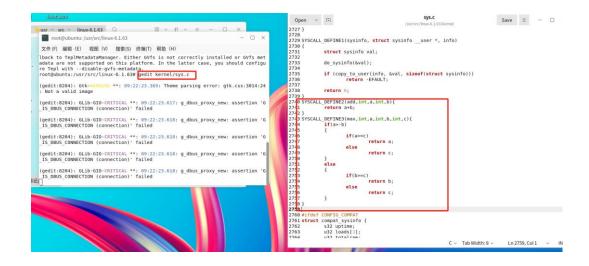
修改系统调用表



#### 声明系统调用服务例程



#### 实现自己的系统调用服务例程



## 利用任务二的过程 编译 然后切换内核并测试

### 3.4 批处理任务

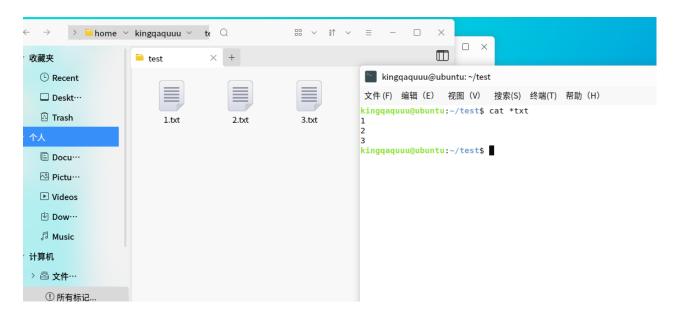
实验环境: VMware Workstation Pro 17

UbuntuKylin 20.04

内核版本: 5.10.0-1023-oem

编辑工具: vim

在/home/kingqaquuu/test 文件夹下,有三个txt 文件



### 在桌面编写脚本如下:

```
mysh.sh
  Open ∨ 🗐
                                                                                Save ≡ - □
 1#!/bin/bash
 2 post fix="txt"
3 echo "Please input the absolute path of the folder:"
 4 read Path
5 for file in $(ls ${Path})
 6 do
           if [ ${file#*.} == ${post_fix} ]
 8
                    count=$(wc -l ${Path}"/"${file} | awk '{print $1}' )
 9
10
                    if grep -q "$USER" ${Path}"/"${file}
11
                   sed -i "${count},${count}d" ${Path}"/"${file}
fi
13
14
15
16
                    cur_dateTime="`date "+%Y-%m-%d %H:%M:%S"`"
17
                    echo "$USER:${cur_dateTime}">> "${Path}/${file}"
           fi
18
19 done
```

编写保存后运行

### 四、实验结果

#### 4.1 编写 MBR 引导程序并测试

输入命令 bochs -f bochsrc 进入后,返回终端,输入c按回车,则会直接显示出Hello, OS world。

```
Bochs Sochs x86-64 emulator, http://bochs.sourceforge.net/
Built from SVN
Timestamp: S
                                                                                                                                                                                                                                                                                                                                                                                                                USER Copy Popte Ballint T Reset suprince Cope I
00000000001 | J
00000000001 | J
000000000001 | J
000000000001 | LUGIN |
000000000001 | PLUGIN |
000000000001 | PLUGIN |
000000000001 | PLUGIN |
000000000001 | PLUGIN |
000000000001 | DUGIN |
000000000001 | DUGIN |
000000000001 | DUGIN |
                                                                                                                                           dello, <mark>OS world!B</mark>ios (PCI) current-svn 07 Jan 2021
This UGA/UBE Bios is released under the GNU LGPL
                                                                                                     LTD. LITHIS VGA./UEE Bios is released under the GNU LGPL

BXSHARE

lt dlmovlease visit:
loaded . http://bochs.sourceforge.net
t dlma . http://bochs.sourceforge.net
t dlma . http://www.nongnu.org/vgabios
loaded pochs UBE Display Adapter enabled
loaded mochs 2.6.11 BIOS - bwild: 01/05/20
t dlma Revision: 13752 $ $ bate 2.019-12-30 14:16:18 *0100 (Mon. 30 Dec 2019) $
loaded options: apmbios pribios pnphios eltorito rombios32
t dlma
loaded
lt dlma reserved by the source of the 
 00000000000001[ ]
00000000000001[PLUGIN]
  000000000000i
                                                                                                        loaded
lt_dlha
loaded
  00000000000i[PLUGIN]
  0000000000001
  00000000000i[PLUGIN]
     reading
     11000000000000
  0000000000001[PLUGIN]
 00000000000i[
00000000000i[
                                                                                                        instal
                                                                                             ] install
] using llfs: 79.831M
    (0) [0x0000fffffff0] f000:fff0 (unk. ctxt): jmpf 0xf000:e05b
                                                                                                                                                                                                                                                                                                                                                        ; ea5be000f0
  <books:1> c
```

# 4.2 编译新内核并启用

输入 uname -r 查看当前内核发现是 6.1.63 成功

```
K<mark>ingqaquuu@ubuntu:~/Desktop$ uname -r</mark>
6.1.63
```

### 4.3 增加2个新调用并启用测试

```
| Copen | Cop
```

```
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)

kingqaquuu@ubuntu:~/Desktop$ uname -r
6.1.63
kingqaquuu@ubuntu:~/Desktop$ gedit test.c
Gtk-Message: 09:45:44.422: Failed to load module "topmenu-gtk-module"

(gedit:2653): Gtk-WARNING **: 09:45:44.457: Theme parsing error: gtk.css:3014:24
: Not a valid image
kingqaquuu@ubuntu:~/Desktop$ acc test.c -o test
kingqaquuu@ubuntu:~/Desktop$ ./test
38
20
kingqaquuu@ubuntu:~/Desktop$
```

可以看到成功调用了系统调用中的 add 和 max。

#### 4.4 批处理任务

运行后输入 test 的地址, 然后输出所有 txt 文件, 可以发现添加成功



# 五、实验错误排查和解决方法

#### 5.1 编写 MBR 引导程序并测试

遇到报错 keyboard\_mapping' is deprecated - use 'keyboard' option instead

Bochs is exiting with the following message:
[ ] bochsrc:25: 'keyboard\_mapping' is deprecated - use 'keyboard' option in stead.

将 keyboard mapping 配置注释掉后解决了报错问题

#### 5.2 编译新内核并启用

刚开始使用的是 ubuntu20.04 分配了 60g 的内存,但是在 make modules 这一步的时候

报错了,发现内存不足了,导致我后面都无法进行,后来询问同学才知道应该分配 100g 的内存,我也正好下载了优麒麟 22.04,但是优麒麟 22.04 是 jammy,下载一些包的时候需要修改它的下载源才能下下来,也有很多问题,很麻烦,我又找了镜像,下载了 20.04 版本的优麒麟进行实验,这个时候实验的过程就异常的顺利

# 5.3 增加2个新调用并启用测试

没有遇到什么问题

### 5.4 批处理任务

没有遇到什么问题

# 六、实验参考资料和网址

(1) 教学课件