

《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 “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

```
kingqaquuu@ubuntu: ~/Desktop$ sudo apt install 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

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

1. Create new floppy or hard disk image
2. Convert hard disk image to other format (mode)
3. Resize hard disk image
4. Commit 'undoable' redolog to base image
5. 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:
    floppy0: 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=0
~
```

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
```

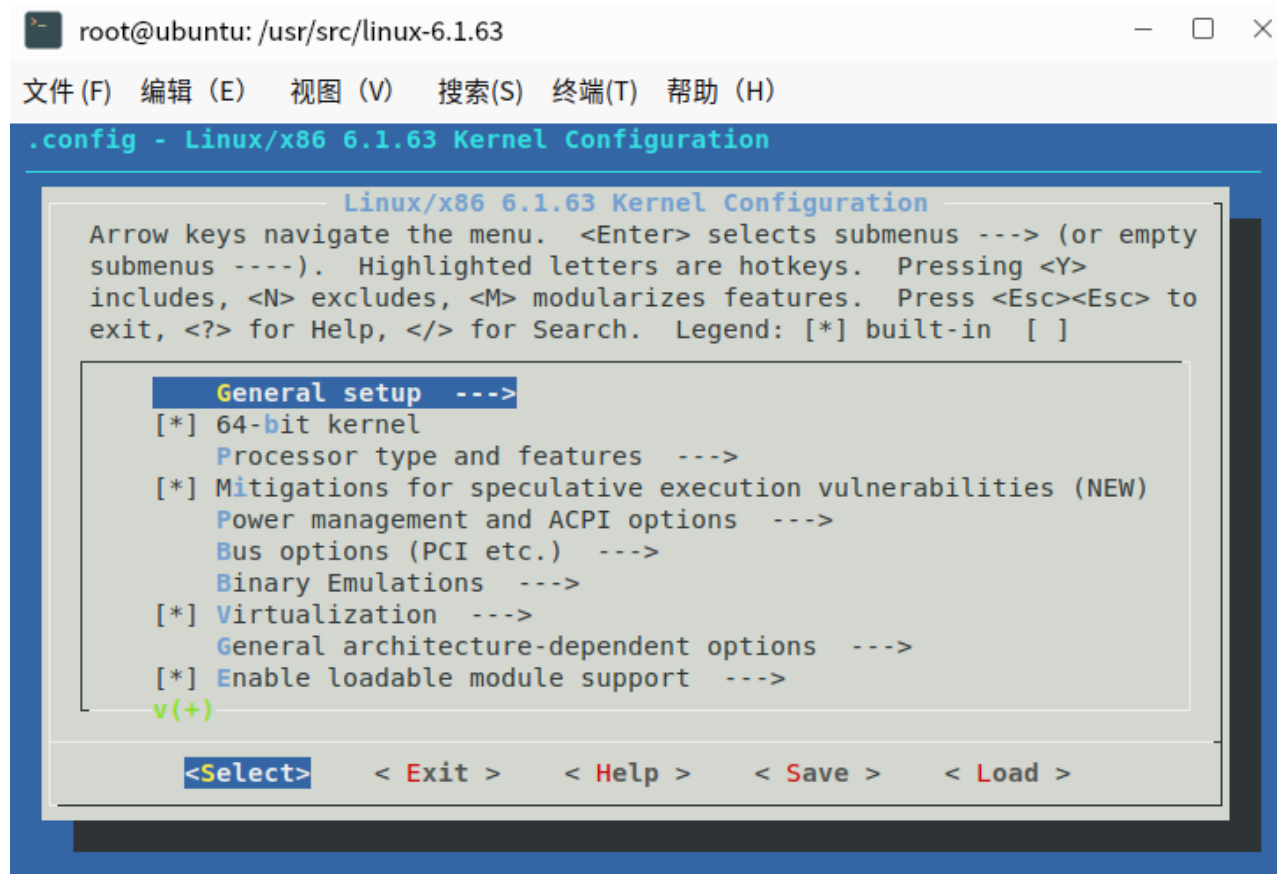
```
apt-get install zlibc minizip
```

```
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
```



```
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)
#
# 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'.

root@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=y
# CONFIG_MODULE_SIG_KEY_TYPE_ECDSA is not set
CONFIG_SYSTEM_TRUSTED_KEYRING=y
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

```
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)
CC [M] drivers/hwmon/lm95234.mod.o
CC [M] drivers/hwmon/lm95241.mod.o
CC [M] drivers/hwmon/lm95245.mod.o
CC [M] drivers/hwmon/ltc2945.mod.o
CC [M] drivers/hwmon/ltc2947-core.mod.o
CC [M] drivers/hwmon/ltc2947-i2c.mod.o
CC [M] drivers/hwmon/ltc2947-spi.mod.o
CC [M] drivers/hwmon/ltc2990.mod.o
CC [M] drivers/hwmon/ltc4151.mod.o
CC [M] drivers/hwmon/ltc4215.mod.o
CC [M] drivers/hwmon/ltc4222.mod.o
CC [M] drivers/hwmon/ltc4245.mod.o
CC [M] drivers/hwmon/ltc4260.mod.o
CC [M] drivers/hwmon/ltc4261.mod.o
CC [M] drivers/hwmon/max1111.mod.o
CC [M] drivers/hwmon/max16065.mod.o
CC [M] drivers/hwmon/max1619.mod.o
CC [M] drivers/hwmon/max1668.mod.o
CC [M] drivers/hwmon/max197.mod.o
CC [M] drivers/hwmon/max31722.mod.o
CC [M] drivers/hwmon/max31730.mod.o
CC [M] drivers/hwmon/max31790.mod.o
CC [M] drivers/hwmon/max6621.mod.o
```

make modules

make install

```
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)
z-6.1.63
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.1.63 /boot/vmlinuz-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/vmlinuz-6.1.63
run-parts: executing /etc/kernel/postinst.d/update-notifier 6.1.63 /boot/vmlinuz-6.1.63
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.1.63 /boot/vmlinuz-6.1.63
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
```

update-grub2

重启后选择内核 6.1.63



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

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

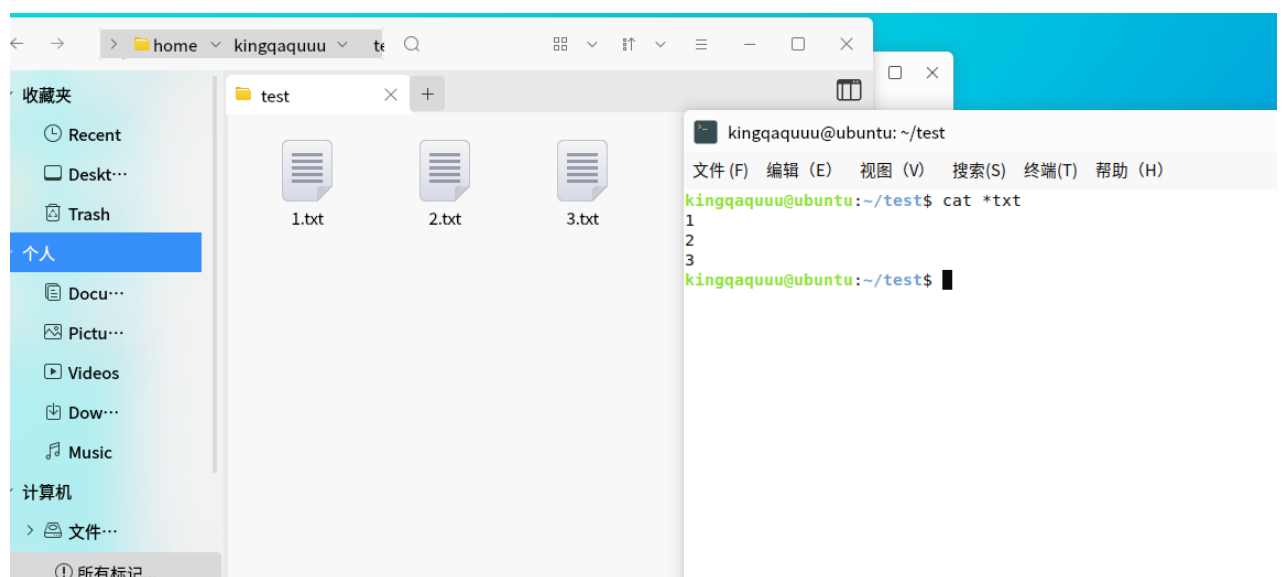
实验环境：VMware Workstation Pro 17

UbuntuKylin 20.04

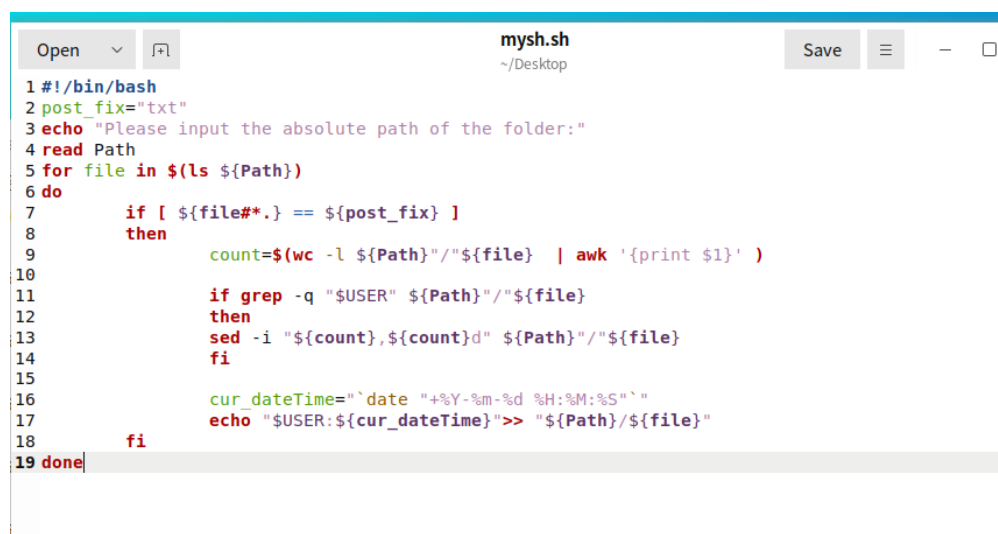
内核版本：5.10.0-1023-oem

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：

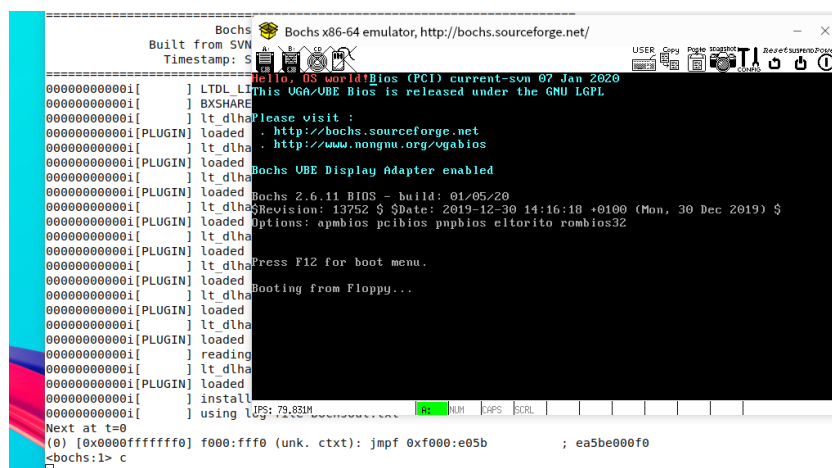


编写保存后运行

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。

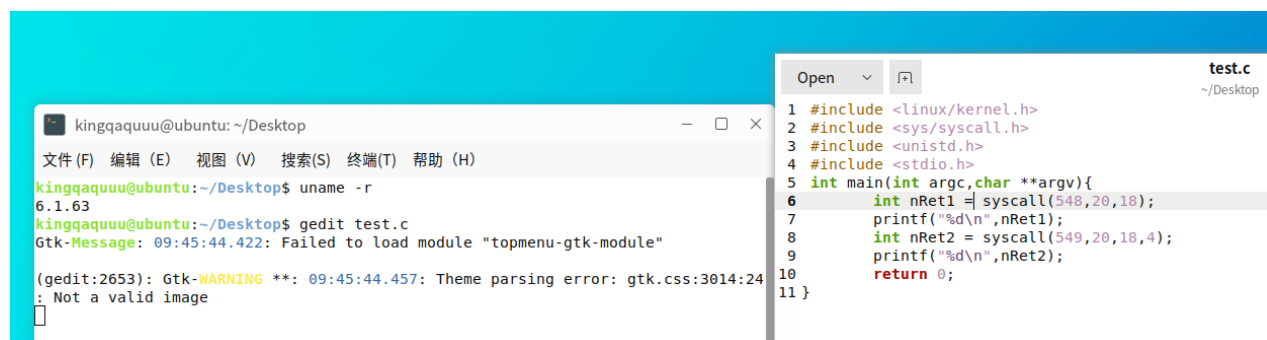


4.2 编译新内核并启用

输入 `uname -r` 查看当前内核发现是 6.1.63 成功

```
kingqaquuu@ubuntu:~/Desktop$ uname -r
6.1.63
```

4.3 增加 2 个新调用并启用测试



```
kingqaquuu@ubuntu: ~/Desktop$ gcc 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 instead.
```

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