

华中科技大学

实 验 报 告

课 程： 操作系统原理

实验序号： 第 1 次实验

实验名称： 增加系统调用

院 系： 软件学院

专业班级：

学 号：

姓 名：

2023 年 03 月 14 日

一、实验目的

1. 理解并应用操作系统生成的概念和过程;
2. 理解并应用操作界面中的系统调用概念, 能在内核中增加系统调用;
3. 掌握和推广国产操作系统 (推荐银河麒麟或优麒麟)。

这次实验的目的是通过在 Linux 内核 (银河麒麟或优麒麟) 中增加系统调用, 来深入理解操作系统的工作原理和系统调用的实现方式。同时, 我们也希望通过这次实验, 能够进一步推广和应用国产操作系统。

二、实验内容

1. 在 Linux 内核 (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。系统调用的功能: `int Add(int, int)` 和 `int Max(int, int, int)`。

这次实验的主要内容是在 Linux 内核中增加两个系统调用, 并编写应用程序进行测试。

三、实验环境

操作系统: 麒麟 Kylin(V10)

内核版本: Linux 5.10.184

编译工具: gcc/g++

编辑工具: vi/vim/vscode/notepad++

主要依赖包: make, gcc, gdb, bison, flex, libncurses5-dev, libssl-dev, libidn11

四、程序的设计思路

4.1 程序结构 (或程序流程、或程序原理, 任选一个或多个组合)

增加系统调用需要在内核代码中改动三个文件：

- ✧ kernel/sys.c——系统调用函数实现
- ✧ arch/x86/include/asm/syscalls.h——系统调用表
- ✧ arch/x86/entry/syscalls/syscall_64.tbl——系统调用原型声明

修改内核后，需要重新编译、安装内核，即可使用新增的系统调用。另外需要额外编写用户态程序用来测试系统调用。

4.2 关键函数或参数或机制

(1) SYSCALL_DEFINE(name, ...)

SYSCALL_DEFINE()用于系统调用实现函数的定义，其中 n 为参数个数（0~5）。系统调用实现函数不能像常规函数一样直接定义，需要借助 SYSCALL_DEFINE()的宏展开。宏参数中 name 为函数名，可变参数为函数原型的参数列表。

例如：SYSCALL_DEFINE2(add, int, arg1, int, arg2)

(2) long int syscall(long int sysno, ...)

syscall()函数定义在 unistd.h 头文件中，用于直接根据系统调用号来使用系统调用。参数 sysno 是系统调用号，可变参数为需要向系统调用传递的 0~5 个参数。

例如：

```
ret = syscall(442, 3, 6, 9);
```

五、关键代码分析

5.1 程序关键片段一：添加系统调用号

在/usr/src/linux-5.10.184/arch/x86/entry/syscalls 目录，使用 gedit syscall_64.tbl 命令在 547 后面添加系统调用号 548 和 549，分别为 add 函数和 max 函数的调用

548	common add	sys_add
549	common max	sys_max

5.2 程序关键片段二：添加函数声明

在/usr/src/linux-5.10.184/arch/x86/include/asm 目录，使用 gedit syscalls.h 命令添加两个函数声明。

<pre>long _sys_add(int x, int y); long _sys_max(int x, int y, int z);</pre>
--

5.3 程序关键片段三：添加函数定义

在/use/src/linux-5.10.184/kernel 目录，使用 gedit sys.c 命令添加两个函数的定义：

<pre>SYSCALL_DEFINE2(add, int, arg1, int, arg2) { return (arg1 + arg2); } SYSCALL_DEFINE3(max, int, arg1, int, arg2, int, arg3) { if (arg1 > arg2) return (arg1 > arg3 ? arg1 : arg3); else return (arg2 > arg3 ? arg2 : arg3); }</pre>

5.4 程序关键片段四：编写测试程序

在测试目录编写测试程序：

```
#include <stdio.h>

#include <unistd.h>

#include <linux/kernel.h>

#include <sys/syscall.h>


int main()

{

    int nRet;

    nRet = syscall(548, 20, 18);

    printf("%d\n", nRet);

    nRet = syscall(549, 20, 18, 4);

    printf("%d\n", nRet);

    return 0;

}
```

六、程序运行结果和分析

使用下列命令对内核的安装包进行解压：



```
dekart@dekart-pc: /usr/src
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
dekart@dekart-pc:/usr/src$ sudo tar -xvJf linux-5.10.184.tar.xz
```

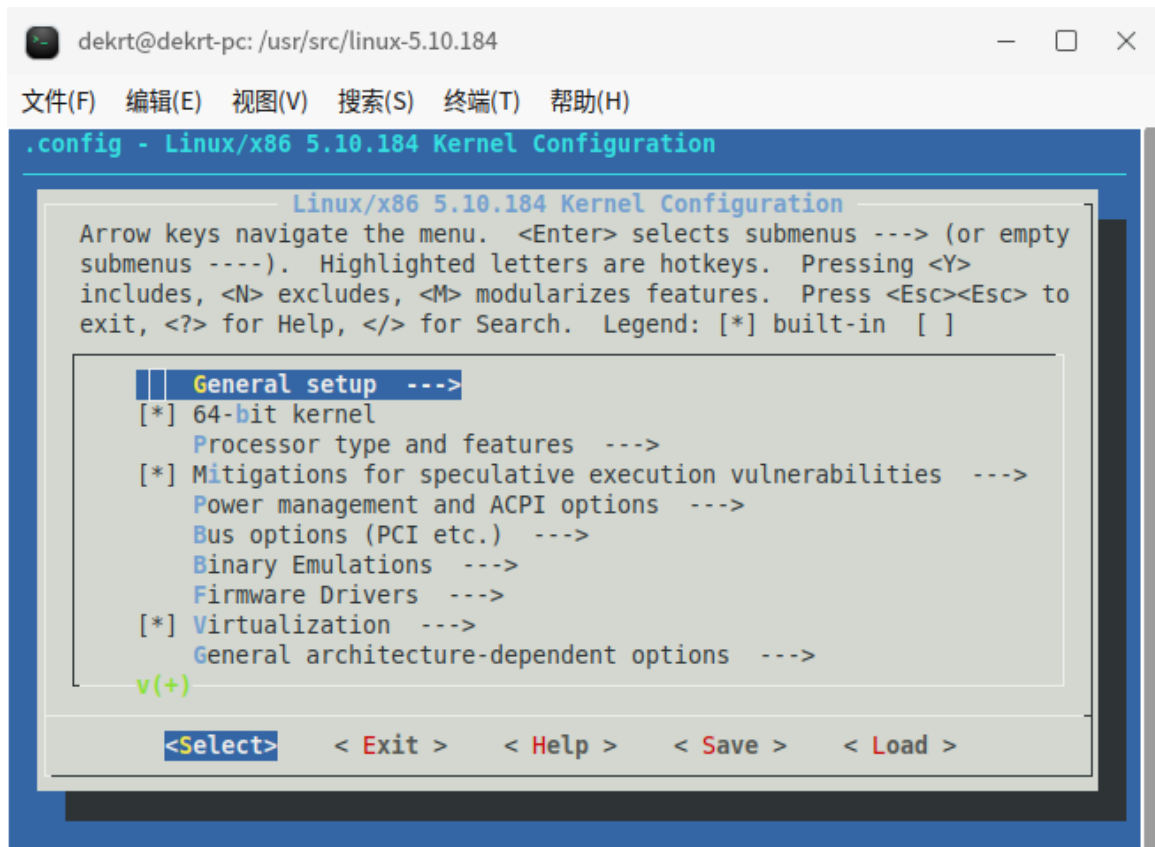
使用 `sudo apt install` 命令安装必须的依赖:

```
dekart@dekart-pc: /usr/src
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
linux-5.10.184/usr/include/
linux-5.10.184/usr/include/.gitignore
linux-5.10.184/usr/include/Makefile
linux-5.10.184/usr/initramfs_data.S
linux-5.10.184/virt/
linux-5.10.184/virt/Makefile
linux-5.10.184/virt/kvm/
linux-5.10.184/virt/kvm/Kconfig
linux-5.10.184/virt/kvm/async_pf.c
linux-5.10.184/virt/kvm/async_pf.h
linux-5.10.184/virt/kvm/coalesced_mmio.c
linux-5.10.184/virt/kvm/coalesced_mmio.h
linux-5.10.184/virt/kvm/eventfd.c
linux-5.10.184/virt/kvm/irqchip.c
linux-5.10.184/virt/kvm/kvm_main.c
linux-5.10.184/virt/kvm/vfio.c
linux-5.10.184/virt/kvm/vfio.h
linux-5.10.184/virt/lib/
linux-5.10.184/virt/lib/Kconfig
linux-5.10.184/virt/lib/Makefile
linux-5.10.184/virt/lib/irqbypass.c
dekart@dekart-pc: /usr/src$ sudo apt install gcc gdb bison flex libncurses5-dev lib
ssl-dev libidn11-dev libidn11 zlibc minizip build-essential openssl libelf-dev d
warves
```

安装完成后, 使用 `sudo apt-get update` 命令进行更新

```
dekart@dekart-pc: /usr/src
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
update-alternatives: 使用 /usr/bin/g++ 来在自动模式中提供 /usr/bin/c++ (c++)
正在设置 libncurses5-dev:amd64 (6.2-0kylin2) ...
正在设置 build-essential (12.8kylin1) ...
正在设置 libelf-dev:amd64 (0.176-1.1build1) ...
正在处理用于 man-db (2.9.1-1kylin0k1) 的触发器 ...
正在处理用于 libc-bin (2.31-0kylin9.2) 的触发器 ...
dekart@dekart-pc: /usr/src$ sudo apt update
命中:1 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1 InRelease
命中:2 http://archive2.kylinos.cn/deb/kylin/production/PART-V10-SP1/custom/partn
er/V10-SP1 default InRelease
命中:3 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1-2303-updates InRelease
命中:4 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1-2303-hwe-updates InRelease
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
有 2 个软件包可以升级。请执行 'apt list --upgradable' 来查看它们。
dekart@dekart-pc: /usr/src$ sudo apt-get update
命中:1 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1 InRelease
命中:2 http://archive2.kylinos.cn/deb/kylin/production/PART-V10-SP1/custom/partn
er/V10-SP1 default InRelease
命中:3 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1-2303-updates InRelease
命中:4 http://archive.kylinos.cn/kylin/KYLIN-ALL 10.1-2303-hwe-updates InRelease
正在读取软件包列表... 完成
dekart@dekart-pc: /usr/src$ make menuconfig
```

进入解压的目录下, 参考第五节对代码进行修改, 修改后使用 `make menuconfig` 命令进行配置。



使用 `make -j16 2>err.log` 命令进行编译，将错误信息重定向至当前目录下的 `err.log` 文件。编译内核成功后，会提示 `kernel is ready`,随后进行模块的安装。



使用 make install 命令进行新内核的安装。

```
root@dekart-pc: /usr/src/linux-5.10.184
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
DEPMOD 5.10.184
root@dekart-pc:/usr/src/linux-5.10.184# make install
sh ./arch/x86/boot/install.sh 5.10.184 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.10.184 /boot/vmlinuz-5.10.184
run-parts: executing /etc/kernel/postinst.d/dkms 5.10.184 /boot/vmlinuz-5.10.184
* dkms: running auto installation service for kernel 5.10.184
Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
make -j8 KERNELRELEASE=5.10.184 -C /lib/modules/5.10.184/build M=/var/lib/dkms/optilauncher/1.0.1.0/build...(bad exit status: 2)
Error! Bad return status for module build on kernel: 5.10.184 (x86_64)
Consult /var/lib/dkms/optilauncher/1.0.1.0/build/make.log for more information.

Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
make -j8 KERNELRELEASE=5.10.184 -C /lib/modules/5.10.184/build M=/var/lib/dkms/8168/8.048.00/build.....(bad exit status: 2)
Error! Bad return status for module build on kernel: 5.10.184 (x86_64)
```

重新启动后，选择新安装好的内核，使用 `uname -a` 和 `uname -r` 查看当前内核版本。

```
dekart@dekart-pc: ~/桌面
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
dekart@dekart-pc:~/桌面$ uname -r
5.10.184
dekart@dekart-pc:~/桌面$ uname -a
Linux dekart-pc 5.10.184 #1 SMP Thu Jun 15 10:17:51 CST 2023 x86_64 x86_64 x86_64 GNU/Linux
dekart@dekart-pc:~/桌面$
```


在测试目录编写测试程序：

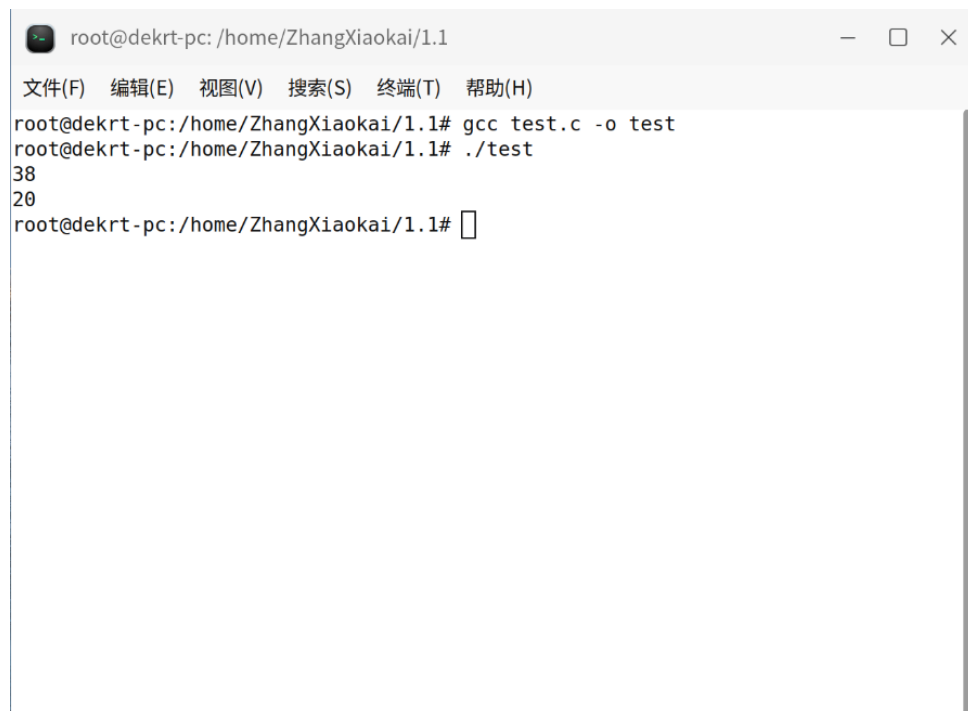


```
root@dekart-pc: /home/ZhangXiaokai/1.1
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)

#include <stdio.h>
#include <sys/syscall.h>
#include <unistd.h>
#include <linux/kernel.h>

int main()
{
    int nRet;
    nRet = syscall(548, 20, 18);
    printf("%d\n", nRet);
    nRet = syscall(549, 20, 18, 4);
    printf("%d\n", nRet);
    return 0;
}
~
~
~
~
~
~
~
-- 插入 --
```

运行测试程序，得到正确的输出结果。



```
root@dekart-pc: /home/ZhangXiaokai/1.1
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)

root@dekart-pc:/home/ZhangXiaokai/1.1# gcc test.c -o test
root@dekart-pc:/home/ZhangXiaokai/1.1# ./test
38
20
root@dekart-pc:/home/ZhangXiaokai/1.1#
```

七、实验错误排查和解决方法

7.1 环境配置问题/无法进行配置和安装

确认依赖程序是否安装

```
$ sudo apt install make gcc gdb bison flex libncurses5-dev libssl-dev libidn11 libelf-dev build-essential
```

7.2: 编译安装时提示“No space left on device”

空间分配不足：在 VMWare 中重新为 Kylin OS 分配存储，并在 disks 中将新分配的存储设置为可用。(No space left on device)

7.3: 编译时提示“没有规则可制作目标“debian/canonical-certs.pem”

修改"/usr/src/linux-5.11.11/.config"文件内容

CONFIG_SYSTEM_TRUSTED_KEYS="debian/canonical-certs.pem"修改为:

CONFIG_SYSTEM_TRUSTED_KEYS=""

保存后继续编译即可。

7.4: 编译时报错“tmp_vmlinux.btf: pahole (pahole) is not available. Failed to generate BTF for vmlinux“

修改 .config 文件中的 CONFIG_DEBUGINFO_BTF: 找到

CONFIG_DEBUGINFO_BTF=y, 改为 CONFIG_DEBUGINFO_BTF=n, 保存退出。

```
10640 #
10641 # Compile-time checks and compiler options
10642 #
10643 CONFIG_DEBUG_INFO=y
10644 # CONFIG_DEBUG_INFO_REDUCED is not set
10645 # CONFIG_DEBUG_INFO_COMPRESSED is not set
10646 # CONFIG_DEBUG_INFO_SPLIT is not set
10647 CONFIG_DEBUG_INFO_DWARF4=y
10648 CONFIG_DEBUG_INFO_BTF=y
10649 CONFIG_GDB_SCRIPTS=y
10650 CONFIG_FRAME_WARN=1024
```

7.5: 重启后无法进入高级设置

修改配置文件：打开/etc/default/grub，将 GRUB_TIMEOUT 设置为-1。运行命令
update-grub 完成修改。

八、实验参考资料和网址

- (1) 教学课件
- (2) 解决 Linux"没有规则可制作目标"debian/canonical-revoked-certs.pem"，由
"certs/x509_revocation_list" 需求。<https://www.xgboke.com/16586.html>
- (3) Ubuntu 官方网站：<https://ubuntu.com/>
- (4) Linux 内核官方网站：<https://www.kernel.org/>
- (5) 《深入理解 Linux 内核》第五版第三章
- (6) 《深入理解 Linux 内核》第五版第十二章