

# 华中科技大学

## 网络安全安全学院

本科：《操作系统原理实验》  
实验报告

题目：用户界面实验

姓 名 \_\_\_\_\_

班 级 \_\_\_\_\_

学 号 \_\_\_\_\_

联系方式 \_\_\_\_\_

分 数 \_\_\_\_\_

评 分 人 \_\_\_\_\_

2024 年 11 月 26 日

## 报告要求

1. 报告不可以抄袭，发现雷同者记为 0 分。
2. 报告中不可以只粘贴大段代码，应是文字与图、表结合的，需要说明流程的时候，也应该用流程图或者伪代码来说明；如果发现有大量代码粘贴者，报告需重写。
3. 报告格式严格按照要求规范，并作为评分标准。

课程目标评价标准

课程目标	评价 环节	评价标准		
		高于预期（优良）	达到预期（及格）	低于预期(不及格)
目标 1. 能够依据实际工程问题中软硬件的约束，应用操作系统的基础理论、抽象和分层设计思想，对特定操作系统进行评价、分析，并能进行合理的优化和裁减。具备设计、实现、开发小型或简化的操作系统的能力，并能体现一定程度的创新性。能熟练应用操作系统的开发和调试技术和工具。	当堂验收	能很合理地设计程序结构、算法和主要数据结构；完成 80% 以上的预定功能。	能比较合理地设计程序结构、算法和关键的数据结构；完成 80% 以上的预定功能。	程序结构、算法和主要数据结构设计不很合理；仅完成不到 50% 的预定功能。
	实验报告	预定功能全部完成；截止日期前提交；源代码完整，且可编译；源代码注释清晰可读。	预定功能 80% 以上完成；截止日期前提交；源代码完整，且可编译；源代码注释比较清晰可读。	预定功能仅完成不到 50% 以上完成；截止日期以后提交；内容单薄；图文排版杂乱无章；源代码缺失或无法编译；源代码无注释或注释不清晰。
目标 2. 具备操作系统国产化和自主创新意识，掌握国产操作系统，例如麒麟操作系统，鸿蒙操作系统的应用和开发环境，支持和推广国产操作系统，积极建设国产操作系统的技术生态。	当堂验收	完全使用国产操作系统；代码规范；能正确回答老师的绝大部分提问。	完全使用国产操作系统；代码比较规范；能正确回答老师的半数以上提问。	仅部分功能使用国产操作系统；代码不够规范，注释缺乏；仅能正确回答老师的极少数提问。
	实验报告	技术框架非常合理高效；符合用户硬件环境和约束参数。报告内容充实；图文排版规范；使用国产操作系统完成全部实验。	技术框架基本合理高效；符合用户硬件环境和约束参数。报告内容基本充实；图文排版基本规范；大部分实验过程使用国产操作系统。	技术框架不合理，代码臃肿；不完全符合用户硬件环境和约束参数。报告内容单薄；图文排版杂乱无章；仅小部分实验过程使用国产操作系统，或完全没用。

报告评分表

评分项目		满分	得分	评分标准
用户界面实验	总体设计（目标 1）	15		15-11：能够给出明确需求，系统功能完整、正确和适当。 10-6：能够给出需求，但不够完整，能够阐述系统的设计，但不够完整、恰当和准确。 5-0：需求不够明确，系统设计不够完整、正确和恰当。
	详细设计（目标 1）	15		15-11：函数和数据结构描述完整，关系清晰，流程设计正确规范。 10-6：函数和数据结构描述基本完整，流程设计基本正确。 5-0：函数和数据结构描述不完整，流程设计有错误。
	代码实现（目标 1，目标 2）	10		10-8：代码能够实现设计的功能要求，考虑错误处理和边界条件。有充分的注释，代码格式规范。 7-5：代码能够实现基本的功能要求，但可能缺少错误处理和边界条件的考虑。关键部分有简单注释，代码格式较为规范。 4-0：代码未能完全实现功能要求，缺少错误处理和边界条件的考虑。注释不足，代码格式不够规范。
	测试及结果分析（目标 1，目标 2）	20		20-14：测试方法科学、完整，结果分析准确完备。 13-8：测试方法描述基本正确、完整，结果分析准确完备。 7-0：测试仅针对数据集的通过性进行描述。
	问题描述及解决方案（目标 1）	10		10-8：遇到的问题及解决方案真实具体 7-5：遇到描述不够详细，解决方案不够具体 4-0：没有写什么内容。
感想（含思政）（目标 2）		10		10-8：感想真实具体。 7-5：感想比较空洞。 4-0：没有写什么感想。
遇到的问题和解决方案，及意见和建议（目标 1）		10		10-8：意见和建议有的放矢。 7-5：意见和建议不够明确。 4-0：没有写什么内容。
文档格式（段落、行间距、缩进、图表、编号等）（目标 1）		10		基本要求：目录、标题、行间距、缩进、正文字体字号按照模板要求执行，图、表清晰且有标号。 10-8：格式规范美观，满足要求。 7-5：基本满足要求。 4-0：格式较为混乱。
总分		100		
教师签名			日期	

## 目 录

<b>1</b>	<b>实验概述.....</b>	<b>1</b>
1.1	用户界面实验 .....	1
1.2	实验目的.....	1
1.3	实验环境.....	1
1.4	实验内容.....	1
1.5	实验要求.....	1
<b>2</b>	<b>实验过程.....</b>	<b>2</b>
2.1	总体设计 .....	2
2.2	详细设计.....	2
2.3	代码实现.....	8
<b>3</b>	<b>测试与分析 .....</b>	<b>10</b>
3.1	系统测试及结果说明.....	10
3.2	遇到的问题及解决方法.....	11
3.3	设计方案存在的不足.....	11
<b>4</b>	<b>实验总结 .....</b>	<b>12</b>
4.1	实验感想.....	12
4.2	意见和建议.....	13

# 1 实验概述

## 1.1 用户界面实验

实验一：第 3 章 用户界面

## 1.2 实验目的

- (1) 理解并应用操作系统生成的概念和过程；
- (2) 理解并应用操作系统操作界面，系统调概念。

## 1.3 实验环境

操作平台：VMware Workstation Pro 16

操作系统：优麒麟 24.04，Linux 6.6.22，Window 11

编辑工具：gedit 46.2

## 1.4 实验内容

- (1) 在 Linux（建议麒麟版本）下载剪和编译 Linux 内核，并启用新内核（其他发行版本也可以）；
- (2) 在 Linux（建议麒麟版本）内核中增加 3 个新的系统调用，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）；
- (3) 在 Windows 下，编写“算命大师.bat”批处理程序，输入出生年月日，输出属相和星座；
- (4) 在 Linux 下，编写“算命大师”脚本程序，输入出生年月日，输出属相和星座。

## 1.5 实验要求

任务 1 是基础，任务 2 必做，任务 3 和任务 4 任做一个。

现场检查：任意完成一个即可。

由于任务 3 和 4 原理相似，我将四个任务均已完成，下面我将更加具体展开阐述完成过程和结果。

## 2 实验过程

### 2.1 总体设计

#### 2.1.1 任务 1 编译并启用新内核

该任务，在麒麟版 Linux 下，先查看当前内核版本，下载不高于此版本的内核源码解压至/usr/src，安装编译工具及虚拟机必备工具。进入源码文件夹，打开配置内核的图形窗口并修改配置文件，之后编译内核，安装内核模块并更新 grub 引导程序，重启后在 grub 引导界面选择新内核进入，完成新内核的启用。

#### 2.1.2 任务 2 为内核新增系统调用

该任务需修改系统调用表及声明系统调用服务例程在 arch/x86/entry/syscalls/syscall\_64.tbl 文件末尾添加系统调用声明，在 include/linux/syscalls.h 文件中添加函数声明，于 kernel/sys.c 文件中编写函数定义，最后重新编译内核并编写测试代码以验证功能。

#### 2.1.3 任务 3&4 编写“算命大师”脚本程序

该任务我针对 Linux 和 Windows 上分别设计了可执行的脚本程序。Linux 版脚本以 bash 为解释器，Windows 版用 chcp 65001 设字符集为 UTF-8。程序定义了存储生肖、星座及星座划分日期临界值的数组，通过 while true 循环获取用户输入的生日，检查格式和合法性后，通过恰当的算法进行计算并输出对应的属相和星座。

### 2.2 详细设计

#### 2.2.1 任务 1 编译并启用新内核

(1) \$ dpkg --list | grep linux-image 查看当前 linux 安装了哪些内核，\$ uname -r 查看正在使用的内核版本为 6.8.0-50-generic，后面下载的内核版本不能高于目前版本。

```
hyt@hyt915-VMware:~/Desktop$ dpkg --list | grep linux-image
ii linux-image-6.8.0-49-generic 6.8.0-49.49
    amd64 Signed kernel image generic
ii linux-image-6.8.0-50-generic 6.8.0-50.51
    amd64 Signed kernel image generic
ii linux-image-generic 6.8.0-50.51
    amd64 Generic Linux kernel image
hyt@hyt915-VMware:~/Desktop$ uname -r
6.8.0-50-generic
```

图 2 - 1 当前内核版本截图

(2) 将下载的内核源码解压到/usr/src 目录下，从网上下载不高于目前内核版本的 kernel 源代码，此处选择下载 linux-6.6.22.tar.xz，通过命令 `$ sudo mv linux-6.6.22.tar.xz /usr/src`，并进入资源文件夹/usr/src 下通过命令 `$ sudo tar -xf linux.6.6.22.tar.xz` 解压。

```
hyt@hyt915-VMware:/usr/src$ ls
linux-6.6.22                linux-headers-6.8.0-50
linux-headers-6.8.0-49     linux-headers-6.8.0-50-generic
linux-headers-6.8.0-49-generic
```

图 2 - 2 解压结果截图

下载软件包，使用如下图 2-3 所示命令安装编译的必要工具 gcc、gdb、bison、flex、libncurses-dev、libssl-dev 等以及虚拟机的必备工具。

```
hyt@hyt915-VMware:~$ sudo apt install flex bison build-essential libncurses-dev libssl-dev libelf-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
flex 已经是最新版 (2.6.4-8.2build1)。
bison 已经是最新版 (2:3.8.2+dfsg-1build2)。
build-essential 已经是最新版 (12.10ubuntu1)。
libncurses-dev 已经是最新版 (6.4+20240113-1ubuntu2)。
libssl-dev 已经是最新版 (3.0.13-0ubuntu3.4)。
libelf-dev 已经是最新版 (0.190-1.1build4)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 227 个软件包未被升级。
```

图 2 - 3 下载软件包截图

(3)然后进入源码文件夹，配置内核，`$sudo make menuconfig` 打开配置内核的图形窗口。

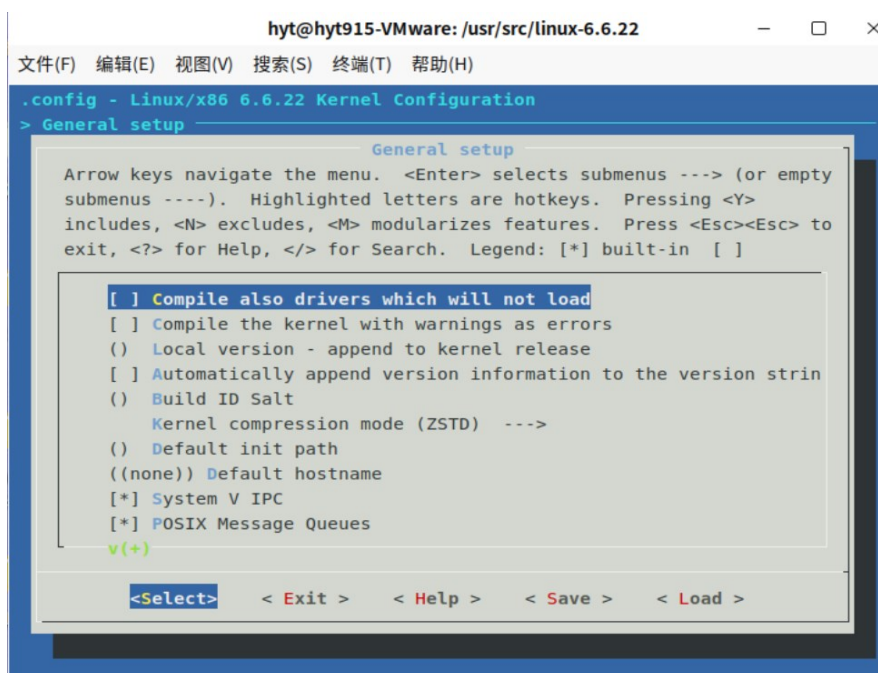


图 2 - 4 配置内核图形窗口



打开.config 文件，将配置文件中 CONFIG\_SYSTEM\_TRUSTED\_KEYS 的值改为空串。

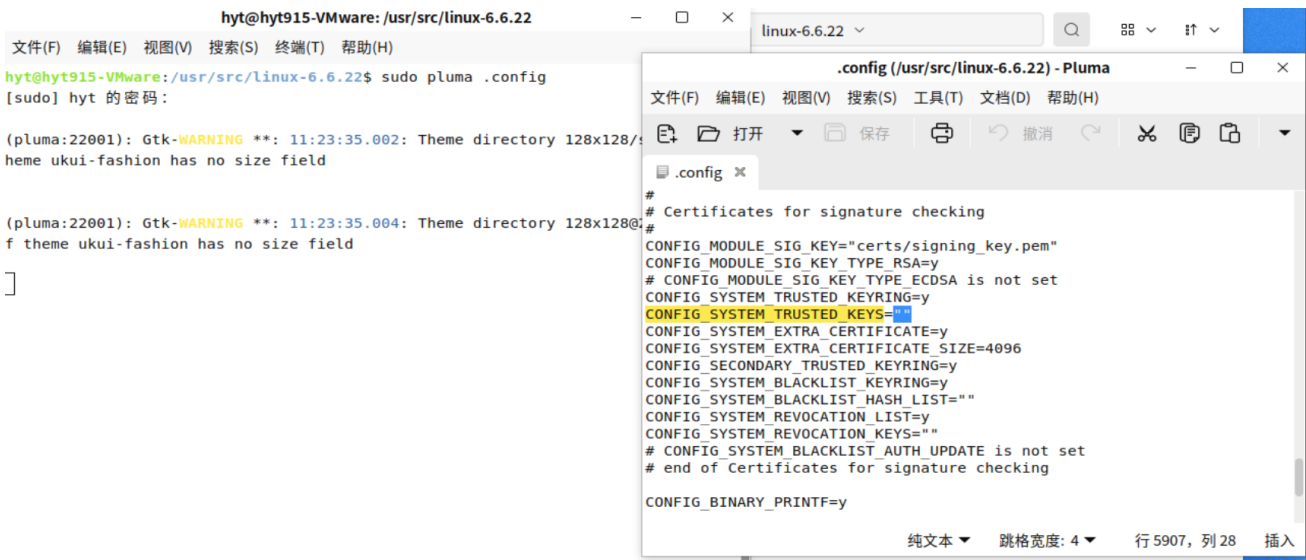


图 2 - 5 .config 文件截图

(4) 在源码目录下，通过 make 命令来编译内核。为加快编译速度，可结合 -jn 参数使用，例如执行 `$ sudo make -j6` 进行编译。其中，n 表示生成的线程数量，通常建议每个处理器分配一到两个线程，以避免编译速度过慢或内存溢出问题，数字越大速度越快，但是不要超过虚拟机分配的数量，可以通过查看虚拟机的硬件配置了解处理器分配的核心总数。

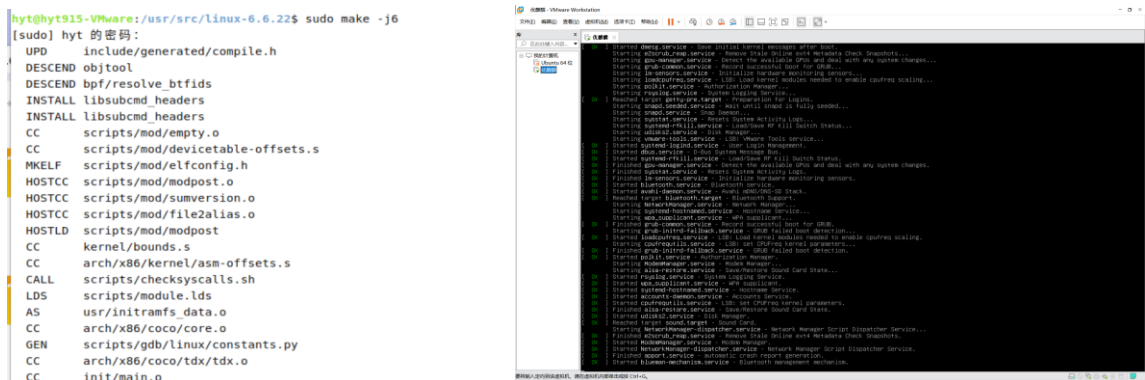


图 2 - 6 编译过程截图

此步时间较长，作者用了 30 分钟左右，编译完成后，可以使用 `$ sudo echo $?` 命令查看是否成功，输出 0 则编译成功，其他数字不成功。

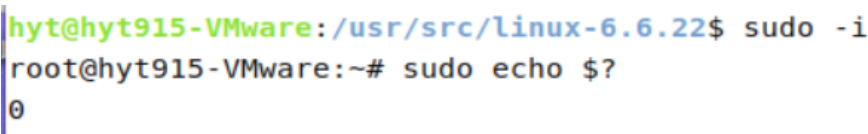


图 2 - 7 检查编译成功

(5) 安装内核模块，make modules\_install、make install。

```
hyt@hyt915-VMware:/usr/src/linux-6.6.22$ sudo make modules_install
INSTALL /lib/modules/6.6.22/modules.order
INSTALL /lib/modules/6.6.22/modules.builtin
INSTALL /lib/modules/6.6.22/modules.builtin.modinfo
SYMLINK /lib/modules/6.6.22/build
INSTALL /lib/modules/6.6.22/kernel/arch/x86/events/rapl.ko
SIGN /lib/modules/6.6.22/kernel/arch/x86/events/rapl.ko.zst
ZSTD /lib/modules/6.6.22/kernel/arch/x86/events/rapl.ko.zst
INSTALL /lib/modules/6.6.22/kernel/arch/x86/kernel/msr.ko
SIGN /lib/modules/6.6.22/kernel/arch/x86/kernel/msr.ko
ZSTD /lib/modules/6.6.22/kernel/arch/x86/kernel/msr.ko.zst
INSTALL /lib/modules/6.6.22/kernel/arch/x86/crypto/aesni-intel.ko
SIGN /lib/modules/6.6.22/kernel/arch/x86/crypto/aesni-intel.ko
ZSTD /lib/modules/6.6.22/kernel/arch/x86/crypto/aesni-intel.ko.zst
INSTALL /lib/modules/6.6.22/kernel/arch/x86/crypto/shal-ssse3.ko
SIGN /lib/modules/6.6.22/kernel/arch/x86/crypto/shal-ssse3.ko
ZSTD /lib/modules/6.6.22/kernel/arch/x86/crypto/shal-ssse3.ko.zst
INSTALL /lib/modules/6.6.22/kernel/arch/x86/crypto/sha256-ssse3.ko
SIGN /lib/modules/6.6.22/kernel/arch/x86/crypto/sha256-ssse3.ko

hyt@hyt915-VMware:/usr/src/linux-6.6.22$ sudo make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.6.22 /boot/vmlinuz-6.6.22
update-initramfs: Generating /boot/initrd.img-6.6.22
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 6.6.22 /boot/vmlinuz-6.6.22
run-parts: executing /etc/kernel/postinst.d/update-notifier 6.6.22 /boot/vmlinuz-6.6.22
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 6.6.22 /boot/vmlinuz-6.6.22
I: /boot/initrd.img.old is now a symlink to initrd.img-6.8.0-50-generic
I: /boot/initrd.img is now a symlink to initrd.img-6.6.22
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.6.22 /boot/vmlinuz-6.6.22
Sourcing file '/etc/default/grub'
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/UKUI/theme.txt
Found linux image: /boot/vmlinuz-6.8.0-50-generic
Found initrd image: /boot/initrd.img-6.8.0-50-generic
Found linux image: /boot/vmlinuz-6.8.0-49-generic
Found initrd image: /boot/initrd.img-6.8.0-49-generic
Found linux image: /boot/vmlinuz-6.6.22
Found initrd image: /boot/initrd.img-6.6.22
Found linux image: /boot/vmlinuz-6.6.22.old
Found initrd image: /boot/initrd.img-6.6.22
Found memtest86+x64 image: /boot/memtest86+x64.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

图 2 - 8 安装内核模块部分截图

并将新安装的内核设置为引导，更新 grub 引导程序。

```
hyt@hyt915-VMware:/usr/src/linux-6.6.22$ sudo update-initramfs -c -k 6.6.22
update-initramfs: Generating /boot/initrd.img-6.6.22
hyt@hyt915-VMware:/usr/src/linux-6.6.22$ sudo update-grub2
Sourcing file '/etc/default/grub'
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/UKUI/theme.txt
Found linux image: /boot/vmlinuz-6.8.0-50-generic
Found initrd image: /boot/initrd.img-6.8.0-50-generic
Found linux image: /boot/vmlinuz-6.8.0-49-generic
Found initrd image: /boot/initrd.img-6.8.0-49-generic
Found linux image: /boot/vmlinuz-6.6.22
Found initrd image: /boot/initrd.img-6.6.22
Found linux image: /boot/vmlinuz-6.6.22.old
Found initrd image: /boot/initrd.img-6.6.22
Found memtest86+x64 image: /boot/memtest86+x64.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

图 2 - 9 安装内核模块部分截图

(6) 重启进入 grub 引导界面，如果没有直接显示可以在开机时长按 shift，然后选择高级可以发现安装新内核后，原来的内核仍然存在，选择目标内核进入，这里由于名字比较长无法直接看出版本，注意选择界面是可以翻页的，可能并不只有四个内核，多次尝试即可。



图 2 - 10 内核选择截图

2.2.2 任务 2 为内核新增系统调用

(1) 修改系统调用表。通过命令\$ sudo gedit arch/x86/entry/syscalls/syscall\_64.tbl，进入系统调用号的配置，在末尾新增 int Max(int , int, int ); int GetPID( ); char \* GetCMD( );三个系统调

用，如图 2-11 所示。

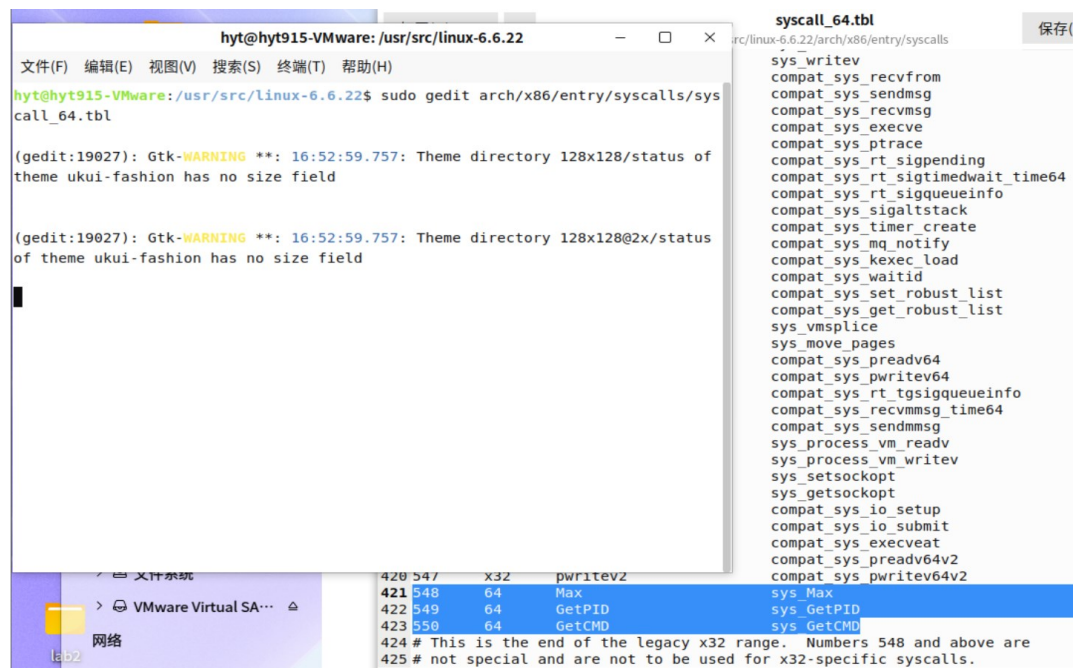


图 2 - 11 系统调用表修改截图

(2) 声明系统调用服务例程。通过命令 `$ sudo gedit include/linux/syscalls.h` 进入函数声明配置，添加三条新的函数声明，`asmlinkage long sys_Max(int a,int b,int c);asmlinkage long sys_GetPID(void);asmlinkage long sys_GetCMD(void);`如图 2-12 所示。

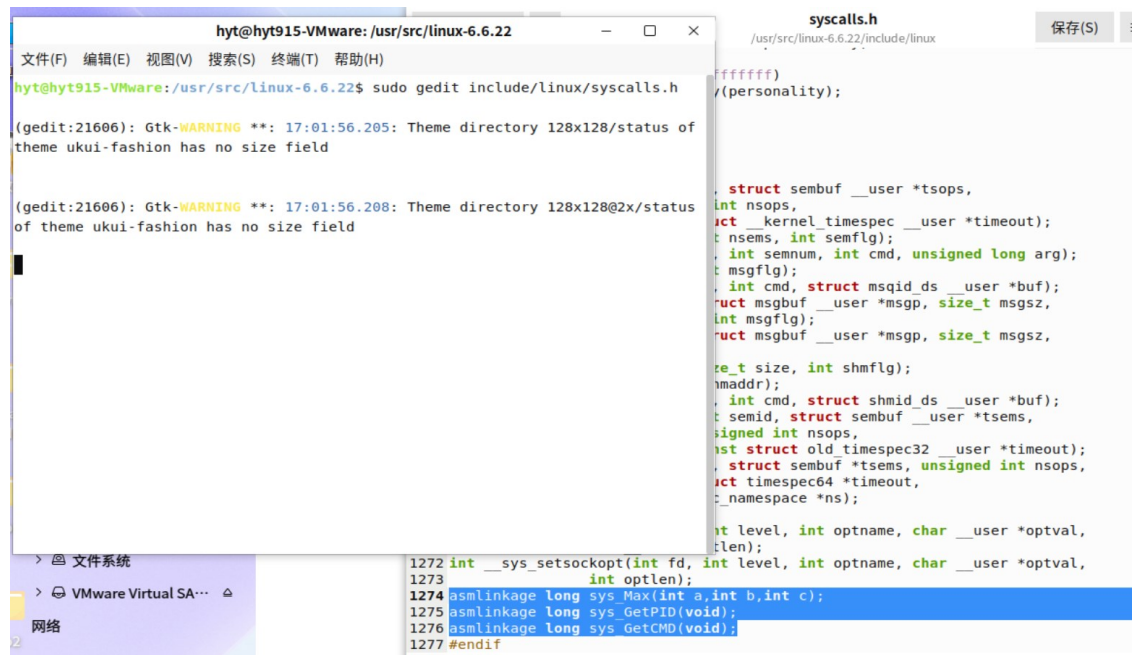
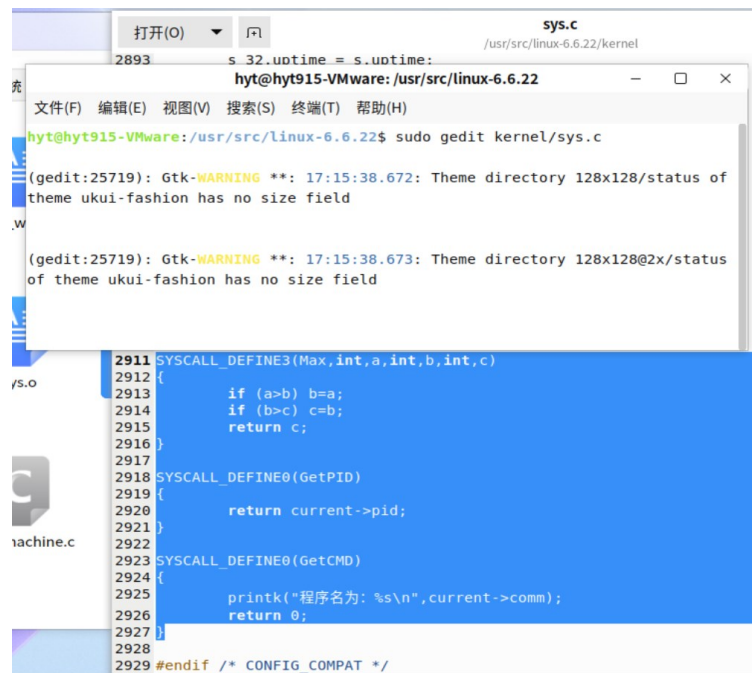


图 2 - 12 系统调用例程声明截图

(3) 编写增加的系统调用源码。通过命令 `$ sudo gedit kernel/sys.c` 进入对函数的定义，对

三条新函数进行定义，如图 2-13 所示。

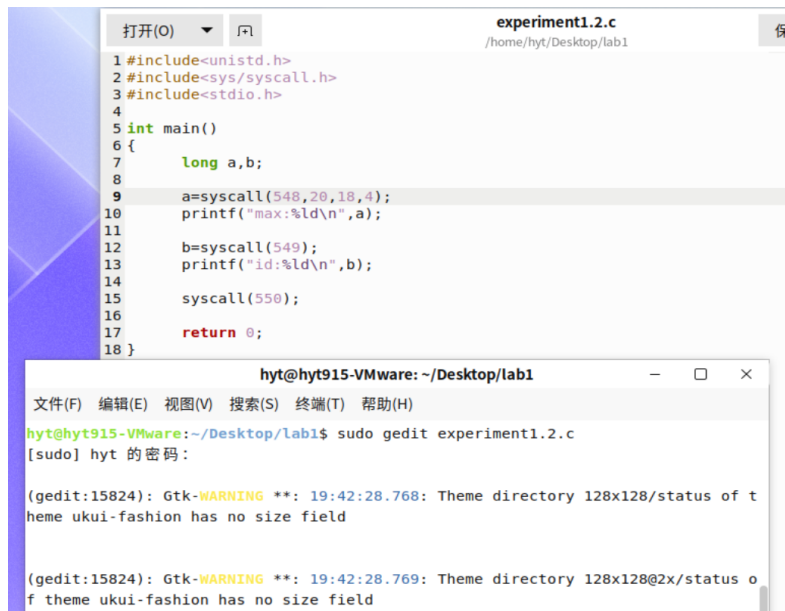


```
2893 s 32.uptime = s.uptime:
hyt@hyt915-VMware: /usr/src/linux-6.6.22
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
hyt@hyt915-VMware: /usr/src/linux-6.6.22$ sudo gedit kernel/sys.c
(gedit:25719): Gtk-WARNING **: 17:15:38.672: Theme directory 128x128/status of
theme ukui-fashion has no size field
(gedit:25719): Gtk-WARNING **: 17:15:38.673: Theme directory 128x128@2x/status
of theme ukui-fashion has no size field

2911 SYSCALL_DEFINE3(Max,int,a,int,b,int,c)
2912 {
2913     if (a>b) b=a;
2914     if (b>c) c=b;
2915     return c;
2916 }
2917 SYSCALL_DEFINE0(GetPID)
2918 {
2919     return current->pid;
2920 }
2921 SYSCALL_DEFINE0(GetCMD)
2922 {
2923     printk("程序名为: %s\n",current->comm);
2924     return 0;
2925 }
2926
2927
2928
2929 #endif /* CONFIG_COMPAT */
```

图 2 - 13 系统调用源码截图

(4) 重新编译内核，然后编写测试代码。通过命令\$ sudo gedit kernel/sys.c 进入对函数的定义，对三条新函数进行定义，如图 2-14 所示。



```
打开(O) 实验
experiment1.2.c
/home/hyt/Desktop/lab1
1 #include<unistd.h>
2 #include<sys/syscall.h>
3 #include<stdio.h>
4
5 int main()
6 {
7     long a,b;
8
9     a=syscall(548,20,18,4);
10    printf("max:%ld\n",a);
11
12    b=syscall(549);
13    printf("id:%ld\n",b);
14
15    syscall(550);
16
17    return 0;
18 }

hyt@hyt915-VMware: ~/Desktop/lab1
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
hyt@hyt915-VMware: ~/Desktop/lab1$ sudo gedit experiment1.2.c
[sudo] hyt 的密码:
(gedit:15824): Gtk-WARNING **: 19:42:28.768: Theme directory 128x128/status of t
heme ukui-fashion has no size field
(gedit:15824): Gtk-WARNING **: 19:42:28.769: Theme directory 128x128@2x/status o
f theme ukui-fashion has no size field
```

图 2 - 14 测试代码截图

2.2.3 任务 3&4 编写“算命大师”脚本程序

任务 3 和 4 编写脚本的核心部分算法是计算属相和星座。



zodiac[zodiac\_index]输出属相。属相是根据年份来计算的。在中国，属相是按 12 年循环的，从鼠年开始。由于公元 4 年是鼠年，所以 $(year-4) \% 12$  可以得到当前年份对应的生肖索引（0 为鼠，1 为牛，依此类推）。

```
47     zodiac_index=(year-4)%12 # 公元4年为鼠年
48     echo "属相: "${zodiac[zodiac_index]}
```

constellation[constellation\_index]输出星座。星座的计算是基于日期和月份的。首先检查输入日期的 day（日）是否大于该月的星座临界日期（mark[month]）。如果大于，则星座是该月的星座；否则是上一个月的星座。

```
51     if [ ${day} -gt ${mark[month]} ]
52     then
53         constellation_index=(month-1)%12
54     else
55         constellation_index=(month-2)%12
56     fi
57     echo "星座: "${constellation[constellation_index]}
```

## 2.3 代码实现

由于前两部分的命令和代码较为直接，同时作者 2.2 节行文针对特殊的部分已进行讲解，故该部分主要针对任务 3 和 4 的代码实现进行详细地解释和编写逻辑说明。

### 2.3.1 任务 3&4 编写“算命大师”脚本程序

任务 3 和 4 的脚本代码原理相同，这里主要以 linux 脚本程序为例对算法逻辑细节进行介绍，同时会对 Windows 版本的代码中需要注意的地方同步提及。

#### （1）脚本头部

```
1 #!/bin/bash
```

一行是“shebang”行，指定了脚本应该使用的解释器，这里使用的是 bash（Bash shell）。

```
@echo off
chcp 65001
title 算命大师
color 0b
```

而在 Windows 上的批处理脚本中，我们呢可以使用 chcp 命令来设置字符集为 UTF-8，chcp 65001 是设置为 UTF-8 的命令，避免中文乱码的情况。

#### （2）定义数组

```
11 zodiac=("鼠" "牛" "虎" "兔" "龙" "蛇" "马" "羊" "猴" "鸡" "狗" "猪")
12 constellation=("水瓶座" "双鱼座" "白羊座" "金牛座" "双子座" "巨蟹座" "狮子座" "处女座" "天秤座" "天蝎座"
13 "射手座" "摩羯座")
13 mark=(20 19 21 20 21 22 23 23 23 24 23 22)
```

zodiac 数组存储了 12 生肖，顺序从鼠年开始。constellation 数组存储了 12 星座的名字，

顺序从水瓶座开始。mark 数组用于存储每个月的生日临界值（即星座划分的日期），例如 1 月的临界值是 20，表示 1 月 20 日后属于水瓶座，之前属于摩羯座。

### （3）主循环部分

```
15 while true
16 do
17     echo ""
18     read -p "请输入您的生日(格式: YYYY-MM-DD): " birthday
21     if [[ ${birthday} == "quit" || ${birthday} == "q" ]]
22     then
23         break
24     fi
```

这个 while true 循环会一直运行，直到用户输入 quit 或 q 来退出。read -p 用于提示用户输入生日，并将用户输入的内容存储在变量 birthday 中。

### （4）输入格式和日期合法性检查

```
27     if [[ ${#birthday} -ne 10 || ${birthday:4:1} != "-" || ${birthday:7:1} != "-" ]]
```

这里使用 if 语句检查输入的生日格式是否为 YYYY-MM-DD，即长度为 10，且第 5 位和第 8 位是"-"。如果格式不对，输出提示并继续要求用户输入。

```
33     if echo ${birthday} | grep -q -E "[0-9]{4}-[0-9]{2}-[0-9]{2}" && \
34     date -d $birthday +%Y%m%d > /dev/null 2>&1
```

使用 grep 检查日期格式是否符合 YYYY-MM-DD 正则表达式。然后尝试将输入的日期解析为有效日期，如果日期无效（如 2024-02-30），则会报错。如果输入的日期合法，则继续；否则，输出提示并要求重新输入。

```
42     year=${birthday:0:4}
43     month=${birthday:5:2}
44     day=${birthday:8:2}
```

合法则提取年月日信息，这一部分将用户输入的生日字符串分割为 year（年）、month（月）和 day（日），便于后续计算。然后根据 2.2.3 的算法计算属相和星座，并进行准确输出，为用户提供直观、准确的属相和星座查询结果。

### 3 测试与分析

#### 3.1 系统测试及结果说明

##### 3.1.1 任务 1 编译并启用新内核

可以在终端通过命令\$ `uname -r` 查看目前内核版本发现是新安装的 linux 版本 6.6.22，然后通过命令\$ `uname -a` 可以发现创建时间与编译开始时间相同，说明是新内核实验成功。

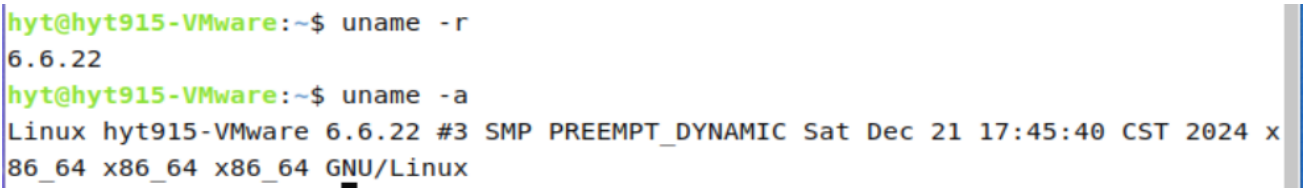


图 3 - 1 当前内核版本截图

##### 3.1.2 任务 2 为内核新增系统调用

编译运行效果如图 3-2 所示，达到预期完成任务。首先预期输出三个变量的最大值，输出符合预期（# `max(20,18,14)` ）；第二个系统调用直接输出系统调用号，符合预期;第三个系统需要通过 `dsmeg` 查看内核日志，输出的程序名和时间均可以准确对应，也是符合期望。



图 3 - 2 测试结果截图

##### 3.1.3 任务 3&4 编写“算命大师”脚本程序

由图 3-3 所示可以清晰地看出结果符合预期。①正确日期可以输出正确的属相和星座；②

均能够自动连续处理多个用户的年月日，中途不用退出程序；③然后对于用户输入错误时间和错误格式进行测试，均可以准确区分并正确处理异常；④具有良好的交互性，友好的人机界面。



```
算命大师
Active code page: 65001
+++++
欢迎使用算命大师程序
请输入您的出生日期
格式 YYYY-MM-DD
按q退出程序
+++++
请输入您的生日 (格式: YYYYMMDD) : 20050303
属相: 鸡
星座: 双鱼座

请输入您的生日 (格式: YYYYMMDD) : 88888888
输入有问题, 请重新输入
请输入您的生日 (格式: YYYYMMDD) : 20040115
属相: 猴
星座: 摩羯座

请输入您的生日 (格式: YYYYMMDD) : 111111
输入有问题, 请重新输入
请输入您的生日 (格式: YYYYMMDD) : q
谢谢使用, 程序结束!
Press any key to continue . . . |

hyt@hyt915-VMware:~/Desktop/lab1$ bash experiment1.4.sh
+++++
欢迎使用算命大师程序
请输入您的出生日期
格式 YYYY-MM-DD
需退出程序请输入quit或q
+++++

请输入您的生日 (格式: YYYY-MM-DD): 2004-11-23
属相: 猴
星座: 射手座

请输入您的生日 (格式: YYYY-MM-DD): 22031
输入格式错误哦!

请输入您的生日 (格式: YYYY-MM-DD): 2004-11-59
输入日期非法哦!

请输入您的生日 (格式: YYYY-MM-DD): q
hyt@hyt915-VMware:~/Desktop/lab1$
```

图 3-3 任务三（左）和 任务四（右）测试结果

综上所述，实验一的四个任务均可以准确完成测试要求。

## 3.2 遇到的问题及解决方法

（1）问题一：编译完内核重启后，无法进入引导程序，直接开机。

解决方法：修改 grub 配置文件。打开 grub 配置文件，发现 GRUB\_TIMEOUT 为 0，意味着系统不会在 grub 界面停留，于是我将其修改为 30，然后更新配置文件并重启，此时能够看到 grub 界面。另外，注意到 GRUB\_TIMEOUT\_STYLE 的值为 hidden，将它改为 countdown 后，grub 界面会出现一个倒计时器，时间结束时如果没有做出选择，则选择默认选项。

（2）问题二：编写内核新增系统调用程序，出现参数不对应或者逻辑问题。

解决方法：其实本身的代码并不难写，同时网上有可以进行参考的文档，困难在于需要操作者足够细心，确保一步步都是准确无误的，因为重新编译的时间较长，我一开始由于疏忽重新编译了几次，积极与同学们进行讨论，非常感谢大家，最终顺利完成实验。

（3）问题三：编写“算命大师”脚本程序时产生乱码问题。

解决方法：均需要设置中文环境确保中文输出的正确性。

## 3.3 设计方案存在的不足

对国产操作系统的熟练度需要提高。



## 4 实验总结

### 4.1 实验感想

#### 4.1.1 任务 1 编译并启用新内核实验感想

在优麒麟操作系统上进行 Linux 内核编译时，我深刻体会到操作系统内核强大的灵活性。从操作系统安全的角度来看，内核编译和裁剪不仅仅是为了定制操作系统以提高效率，更是为了能够根据安全需求定制安全策略。通过了解内核结构和各个组件的作用，我们能够更好地发现潜在的安全漏洞，并采取相应的防范措施。

在使用优麒麟操作系统的过程中，我感受到其在国内环境中的良好适配性。然而，内核编译过程中的依赖问题不容忽视，这些问题如果处理不当，可能会成为安全隐患。例如，错误的依赖库版本可能导致内核运行不稳定，进而可能被恶意攻击者利用。通过查阅相关文档和参与社区讨论解决这些问题的过程，使我更加明白在操作系统开发和维护过程中，保障每一个环节的正确性和安全性是至关重要的。成功启用新内核后，我认识到合理定制内核对于构建安全可靠的操作系统环境具有重要意义，它是防止外部攻击和保障系统稳定运行的基础。

#### 4.1.2 任务 2 为内核新增系统调用实验感想

新增系统调用的任务让我深入了解了优麒麟内核的定制性。从操作系统安全层面考虑，系统调用是用户空间与内核空间交互的重要接口，也是安全防护的关键区域。通过修改内核代码增加新的系统调用，我清晰地认识到系统调用在整个操作系统安全架构中的重要性。

在这个过程中，我不仅学习了如何扩展内核，还加深了对系统调用安全机制的理解。与其他 Linux 发行版相比，优麒麟提供了更优化的配置选项，这有助于我们更好地实施安全策略。在成功编写用户空间应用程序并测试新的系统调用后，我更加意识到正确地管理和控制系统调用对于防止非法访问和恶意操作至关重要。任何对系统调用的不当使用或者未授权的调用，都可能导致严重的安全问题，如权限提升、数据泄露等。因此，深入了解内核系统调用机制并确保其安全性，是保障操作系统安全运行的核心环节之一。

#### 4.1.3 任务 3&4 编写“算命大师”脚本程序实验感想

编写“算命大师”脚本程序是一个兼具趣味性和挑战性的任务。在优麒麟操作系统的终端编写 Bash 脚本时，我遇到了字符编码和脚本优化方面的挑战。从操作系统安全的角度来看，

脚本程序的编写和运行涉及到系统资源的使用和数据处理，如果处理不当，可能会导致安全漏洞。

例如，字符编码问题可能导致脚本解析错误，进而引发命令执行异常，这可能会被攻击者利用来执行恶意命令。优麒麟强大的终端工具和包管理器为解决这些问题提供了帮助，但同时也提醒我们在使用这些工具时，需要确保其本身的安全性。在 Windows 系统下实现批处理脚本，让我对命令行编程有了更深入的了解，并且通过在不同操作系统之间进行兼容性调试，我认识到跨平台脚本编程中的安全问题不容忽视。不同操作系统的安全机制和默认设置存在差异，在编写跨平台脚本时，必须充分考虑这些差异，以防止因兼容性问题导致的安全风险，例如脚本在某个平台上正常运行，但在另一个平台上却因安全配置不同而引发漏洞。

## 4.2 意见和建议

通过这次实验，我不仅加深了对操作系统底层机制的理解，也对国产操作系统有了更为深入的认识。优麒麟作为一款国内自主研发的操作系统，其稳定性和适配性为开发者提供了良好的使用体验，这也为我的操作系统学习之路提供了有益的参考和借鉴。同时此次实验为我对内核攻击和防御进一步研究奠定了基础，让我更加体会到系统安全与稳定性的重要性，极大提高了动手实践和问题解决能力，作为网安专业的学生也为未来职业发展奠定了基础。

## 原创性声明

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

已阅读并同意以下内容。

判定为不合格的一些情形：

- (1) 请人代做或冒名顶替者；
- (2) 替人做且不听劝告者；
- (3) 实验报告内容抄袭或雷同者；
- (4) 实验报告内容与实际实验内容不一致者；
- (5) 实验代码抄袭者。

作者签名：