

# 哈爾濱工業大學

# 实验报告

## 实 验（一）

题 目 Linux 下 C 工具应用

专 业 计算机科学与技术

学 号 1160300901

班 级 1603009

学 生 孙月晴

指 导 教 师 吴锐

实 验 地 点 G712

实 验 日 期 2017 年 10 月 10 日

# 计算机科学与技术学院

## 目 录

<b>第 1 章 实验基本信息</b>	<b>- 4 -</b>
1.1 实验目的	4 -
1.2.1 硬件环境	4 -
1.2.2 软件环境	4 -
1.2.3 开发工具	4 -
1.3 实验预习	4 -
<b>第 2 章 实验环境建立</b>	<b>- 9 -</b>
2.1 VMWARE 下中文 UBUNTU 安装 (5 分)	9 -
2.2 UBUNTU 与 WINDOWS 目录共享 (5 分)	9 -
<b>第 3 章 WINDOWS 软硬件系统观察分析</b>	<b>- 12 -</b>
3.1 查看计算机基本信息 (2 分)	12 -
3.2 设备管理器查看 (2 分)	14 -
3 隐藏分区与虚拟内存之分页文件查看 (2 分)	14 -
3.4 任务管理与资源监视 (2 分)	15 -
3.5 计算机硬件详细信息 (2 分)	15 -
4.1 计算机硬件详细信息 (3 分)	16 -
4.2 任务管理与资源监视 (2 分)	17 -
4.3 共享目录的文件系统信息 (3 分)	17 -
4.4 LINUX 下网络系统信息 (2 分)	17 -
<b>第 5 章 以 16 进制形式查看程序 HELLO.C</b>	<b>- 19 -</b>
5.1 请查看 HELLOWIN.C 与 HELLOLINUX.C 的编码 (3 分)	19 -
5.2 请查看 HELLOWIN.C 与 HELLOLINUX.C 的回车 (3 分)	20 -
<b>第 6 章 程序的生成 CPP、GCC、AS、LD</b>	<b>- 21 -</b>
6.1 请提交每步生成的文件 (4 分)	21 -
<b>第 7 章 计算机系统的基本信息获取编程</b>	<b>- 24 -</b>
7.1 请提交源程序文件 (10 分)	24 -
<b>第 8 章 计算机数据类型的本质</b>	<b>- 25 -</b>
8.1 请提交源程序文件 DATATYPE.C (10 分)	25 -
<b>第 9 章 程序运行分析</b>	<b>- 26 -</b>

9.1 SUM 的分析（20 分）	- 26 -
9.2 FLOAT 的分析（20 分）	- 26 -
<b>第 10 章 总结</b>	<b>- 29 -</b>
10.1 请总结本次实验的收获	- 29 -
10.2 请给出对本次实验内容的建议	- 29 -
<b>参考文献</b>	<b>- 30 -</b>

## 第 1 章 实验基本信息

### 1.1 实验目的

- 1.运用现代工具进行计算机软硬件系统的观察与分析
- 2.运用现代工具进行 Linux 下 C 语言的编程调试
- 3.初步掌握计算机系统的基本知识与各种类型的数据表示

### 1.2 实验环境与工具

#### 1.2.1 硬件环境

X64 CPU; 2GHz; 2G RAM; 256GHD Disk

#### 1.2.2 软件环境

Windows10 64 位; VirtualBox/Vmware 12 以上; Ubuntu 16.04 LTS 64 位;

#### 1.2.3 开发工具

Visual Studio 2010 64 位; CodeBlocks; vi/vim/gpedit+gcc

### 1.3 实验预习

- 1.实验课前,认真预习实验指导书
- 2.了解实验的目的、实验环境与软硬件工具、实验操作步骤,复习与实验有关的理论知识。
- 3.在 Windows 下编写 hellowin.c,显示“Hello 1160300199 学霸”  
(可用记事本、VS、CB等,换成自己的信息)

```
#include <stdio.h>
```

```
int main()  
{
```

```

    printf("Hello 1160300901 孙月晴\n");
    return 0;
}

```

4.在 Linux 下编写 hellolinux.c, 显示 “Hello 1160300199 学霸”  
 ( 可用 VI 、 VIM 、 EMACS 、 GEDIT 等, 换成自己的信息 )

```
#include <stdio.h>
```

```

int main()
{
    printf("Hello 1160300901 孙月晴\n");
    return 0;
}

```

5.编写 showbyte.c 以 16 进制显示文件 hello.c 等的内容: 每行 16 个字符, 上一行为字符, 下一行为其对应的 16 进制形式。  
 代码如下:

```

#include <stdio.h>
#include <mem.h>

typedef unsigned char *byte_pointer;

void show_bytes(byte_pointer start, size_t len);

int main() {
    char *s = "#include <stdio.h>\n"
              "\n"
              "int main() \n"
              "{\n"
              "    printf(\"Hello, world\\n\");\n"
              "    return 0;\n"
              "}";
    show_bytes((byte_pointer) s, strlen(s));
    return 0;
}

void show_bytes(byte_pointer start, size_t len) {
    size_t i, j;
    int flag = 0;

```

```

    for (i = 0; i < len; i++) {
        printf(" %.2c", start[i]);
        if ((i + 1) % 16 == 0) {
            printf("\n");
            for (j = flag; j < len; j++) {
                printf(" %.2x", start[j]);
                if ((j + 1) % 16 == 0) {

                    break;
                }
            }
            flag = j + 1;
            printf("\n");
        }
    }
    printf("\n");
    for (j = flag; j < len; j++) {
        printf(" %.2x", start[j]);
    }
}

```

6. 编写 `datatype.c`，定义 C 所有类型的全局变量，并赋初值。如整数可以是学号( 数字部分)，字符串可以是你的姓名，浮点数可以是身份证号的数字部分。主程序打印每个变量的变量名、变量值、变量地址、变量对应 16 进制的内存各字节。

代码如下：

```

#include <stdio.h>

int *p;
int s[3] = {1, 2, 3};
int num = 1160300901;
short snum = 901;
long lnum = 1160300901;
float fId = 1997;
double dId = 1997;
char cName= 's';

struct myself {
    int number;
    char name;
}

```

```
    float score;
} x = {1160300901, 's', 89};

union test {
    int num;
    char mark;
    float score;
};

enum weekday {
    sun, mon, tue
} ;

int main() {
    int a=sun;
    int b=mon;
    int var = 10;
    p = &var;
    printf("p = %d ,address = %d ,%X\n", *p, p, p);

    printf("s[0] = %d ,address = %d ,%X\n", s[0], &s[0], &s[0]);
    printf("s[1] = %d ,address = %d ,%X\n", s[1], &s[1], &s[1]);
    printf("s[2] = %d ,address = %d ,%X\n", s[2], &s[2], &s[2]);

    printf("int num = %d ,address = %d ,%X\n", num, &num, &num);

    printf("short num = %d ,address = %d ,%X\n", snum, &snum, &snum);

    printf("long num = %d ,address = %d ,%X\n", lnum, &lnum, &lnum);

    printf("float Id = %f ,address = %d ,%X\n", fId, &fId, &fId);

    printf("double Id = %f ,address = %d ,%X\n", dId, &dId, &dId);

    printf("char cName = %c ,address = %d ,%X\n", cName, &cName, &cName);

    printf("x.number = %d ,address = %d ,%X\n", x.number, &x.number,
&x.number);
    printf("x.name = %c ,address = %d ,%X\n", x.name, &x.name, &x.name);
    printf("x.score = %f ,address = %d ,%X\n", x.score, &x.score, &x.score);
```

```
printf("a = %d ,address = %d ,%X\n", a, &a, &a);
printf("b = %d ,address = %d ,%X\n", b, &b, &b);

printf("main 的地址: %p\n", main);
printf("printf 的地址: %p\n", printf);
union test Union;
Union.num=1603009;
Union.mark='s';
printf("b.num = %d ,address = %d ,%X\n",
Union.num,&Union.num,&Union.num);
printf("b.mark = %c ,address = %d ,%X\n",
Union.mark,&Union.mark,&Union.mark);
return 0;
}
```



## 第 2 章 实验环境建立

### 2.1 Vmware 下中文 Ubuntu 安装 (5 分)

安装 Ubuntu，安装中文输入法（搜狗输入法），用户名为学号！

打开终端 term，输入 Hello 1160300199 学霸（用真实学号姓名代替）。

截图：要求有 Windows 状态行，Vmware 窗口，Ubuntu 窗口，终端 term 窗口，输入的“Hello 1160300199 学霸”信息

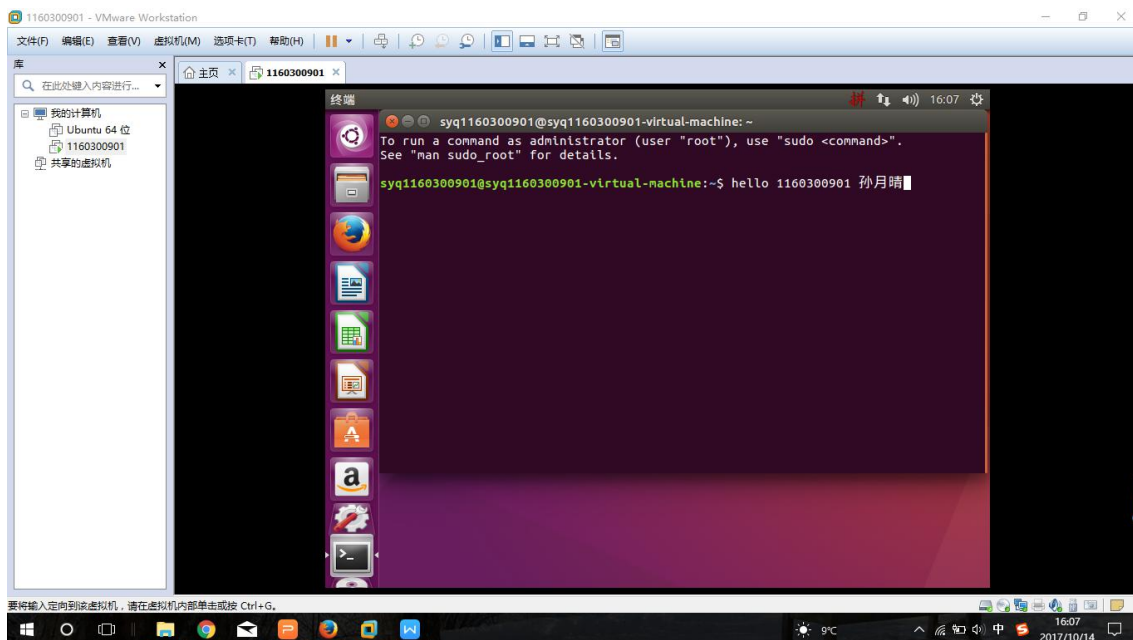


图 2-1 Vmware 下中文 Ubuntu 安装效果截图

### 2.2 Ubuntu 与 Windows 目录共享 (5 分)

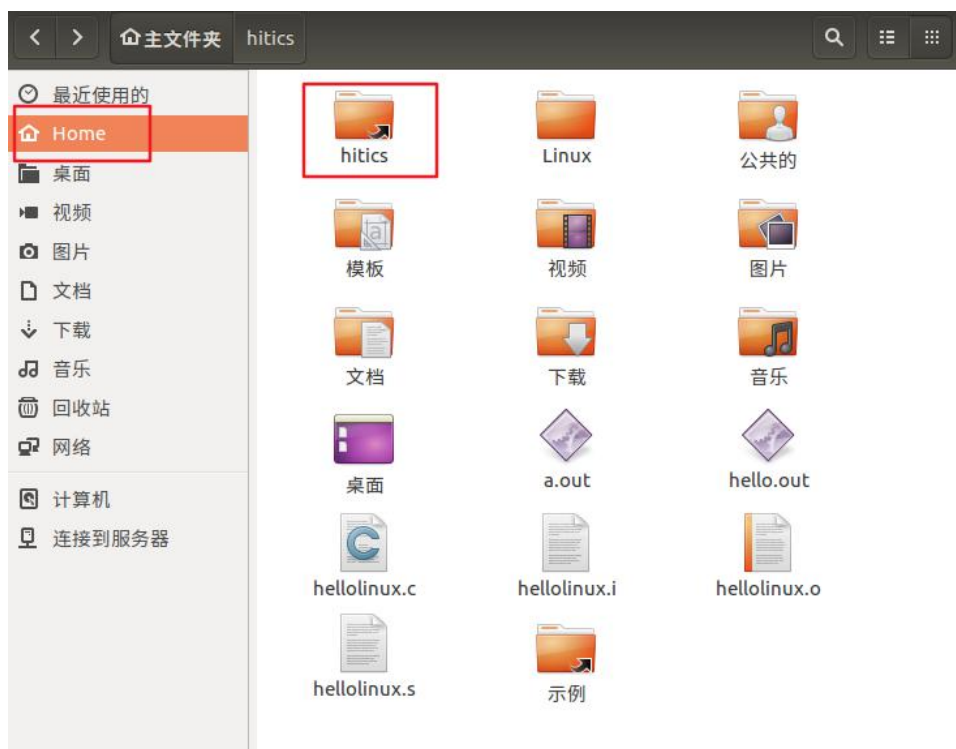
在 Windows 下建立一目录，将 helloworld.c 拷贝到此目录。在 vmware 下设置 Ubuntu 共享 hitics。

在 Ubuntu 下 Home 建立快捷链接 hitics 指向此共享目录,并在此目录建立 helloworld.c。

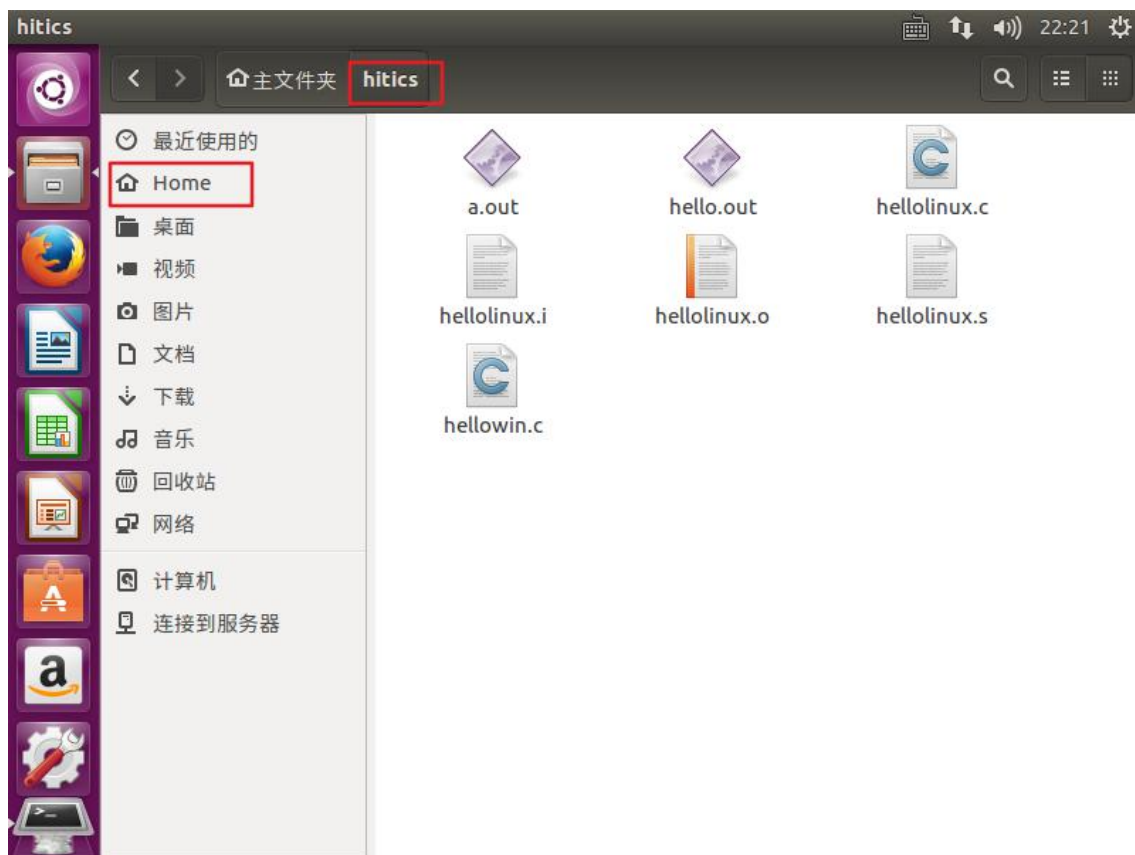
打开终端 term，进入此目录，输入 “ls -la” 指令。

截图：要求有 Ubuntu 的“文件”应用打开“Home”，能看到 hitics。term 窗口。

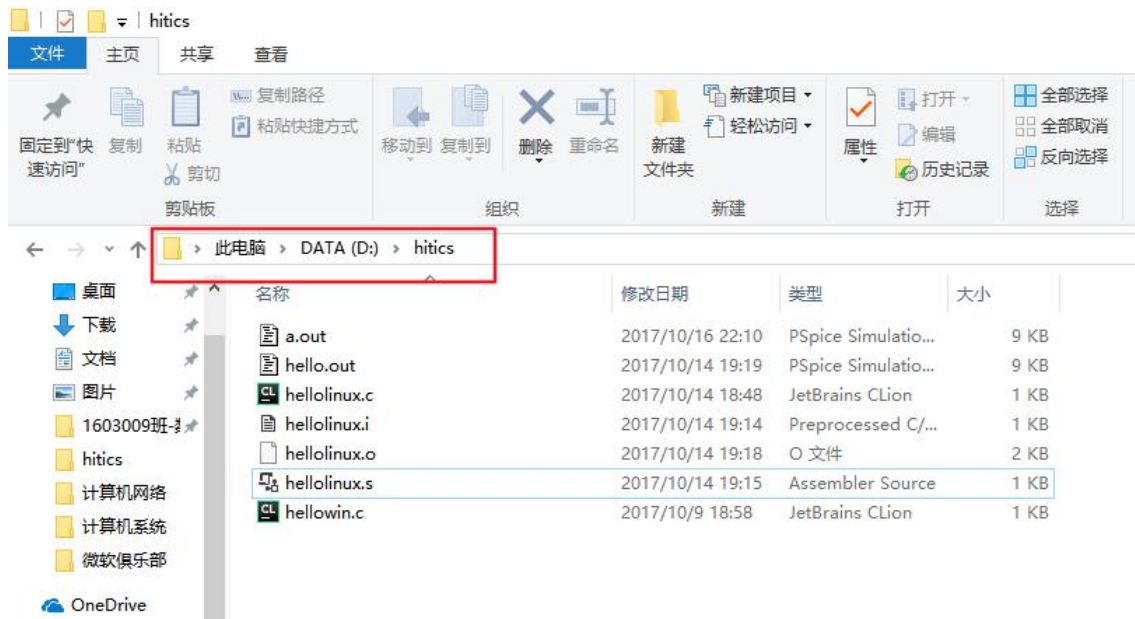
快捷链接 hitics 指向此共享目录:



Linux 中共享文件夹:



## Windows 中共享文件夹:



打开终端 term, 进入此目录, 输入 “ls -la” 指令:

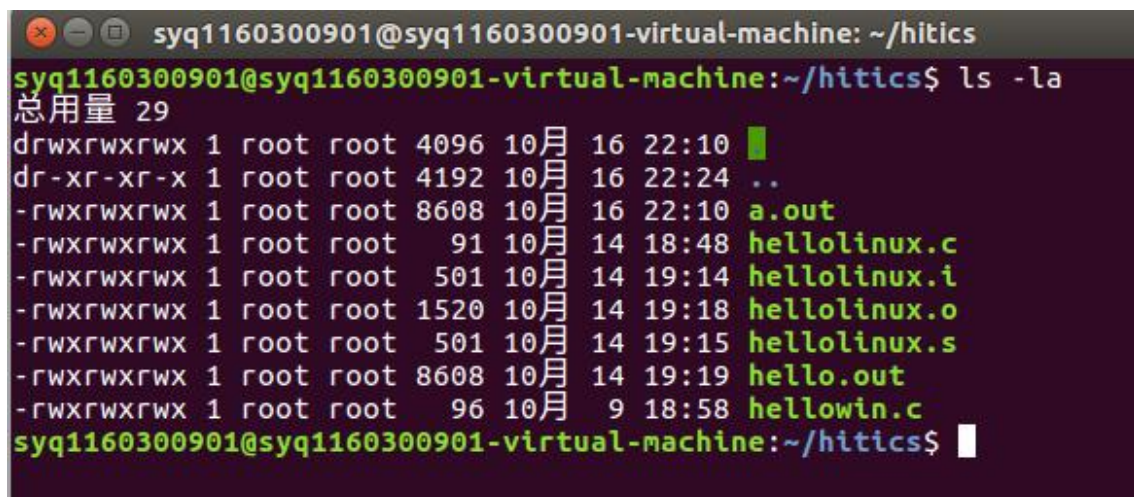


图 2-2 Ubuntu 与 Windows 共享目录截图

## 第 3 章 Windows 软硬件系统观察分析

### 3.1 查看计算机基本信息 (2 分)

截图：1.控制面板->系统

#### 查看有关计算机的基本信息

##### Windows 版本

Windows 10 家庭中文版

© 2017 Microsoft Corporation。保留所有权利。

##### 系统

制造商:	ASUSTek Computer Inc.
型号:	X555UQ
处理器:	Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz
已安装的内存(RAM):	8.00 GB (7.87 GB 可用)
系统类型:	64 位操作系统, 基于 x64 的处理器
笔和触控:	没有可用于此显示器的笔或触控输入

##### ASUSTek Computer Inc. 支持

网站: [联机支持](#)

##### 计算机名、域和工作组设置

计算机名:	DESKTOP-V0QK3MI
计算机全名:	DESKTOP-V0QK3MI
计算机描述:	
工作组:	WORKGROUP

##### Windows 激活

Windows 已激活 [阅读 Microsoft 软件许可条款](#)

产品 ID: 00342-33198-91875-AAOEM

2.命令行 systeminfo 执行结果(至少包含启动设备行)

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\孙月晴>systeminfo

主机名:                DESKTOP-VOQK3MI
OS 名称:               Microsoft Windows 10 家庭中文版
OS 版本:               10.0.15063 暂缺 Build 15063
OS 制造商:             Microsoft Corporation
OS 配置:               独立工作站
OS 构件类型:           Multiprocessor Free
注册的所有人:         孙月晴
注册的组织:            Microsoft
产品 ID:                00342-33198-91875-AAOEM
初始安装日期:          2017/10/8, 23:44:36
系统启动时间:          2017/10/8, 23:26:02
系统制造商:            ASUSTeK COMPUTER INC.
系统型号:              X555UQ
系统类型:              x64-based PC
处理器:                安装了 1 个处理器。
                        [01]: Intel64 Family 6 Model 78 Stepping 3 GenuineIntel ~2300 Mhz
BIOS 版本:              American Megatrends Inc. X555UQ.300, 2016/6/8
Windows 目录:           C:\WINDOWS
系统目录:               C:\WINDOWS\system32
启动设备:               \Device\HarddiskVolume1
系统区域设置:           zh-cn; 中文(中国)
输入法区域设置:         zh-cn; 中文(中国)
时区:                   (UTC+08:00) 北京, 重庆, 香港特别行政区, 乌鲁木齐
物理内存总量:           8,056 MB
可用的物理内存:         3,047 MB
虚拟内存: 最大值:       9,976 MB
虚拟内存: 可用:         4,088 MB
虚拟内存: 使用中:       5,888 MB
页面文件位置:           D:\pagefile.sys
域:                      WORKGROUP
登录服务器:              \\DESKTOP-VOQK3MI
修补程序:               安装了 5 个修补程序。
                        [01]: KB4021572
                        [02]: KB4022405
                        [03]: KB4025376
                        [04]: KB4038806
                        [05]: KB4040724
网卡:                   安装了 4 个 NIC。
                        [01]: VMware Virtual Ethernet Adapter for VMnet8
```



```

网卡: 安装了 4 个 NIC。
      [01]: VMware Virtual Ethernet Adapter for VMnet8
          连接名: VMware Network Adapter VMnet8
          启用 DHCP: 是
          DHCP 服务器: 192.168.233.254
          IP 地址
            [01]: 192.168.233.1
            [02]: fe80::d9ec:c97c:23a4:52dd
      [02]: VMware Virtual Ethernet Adapter for VMnet1
          连接名: VMware Network Adapter VMnet1
          启用 DHCP: 是
          DHCP 服务器: 192.168.253.254
          IP 地址
            [01]: 192.168.253.1
            [02]: fe80::d0ee:1c19:4c9c:2f32
      [03]: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC
          连接名: WLAN
          启用 DHCP: 是
          DHCP 服务器: 172.20.0.1
          IP 地址
            [01]: 172.20.15.187
            [02]: fe80::9cb6:5955:576c:a89b
            [03]: 2001:da8:ab:130:8d39:7e62:61a6:8a26
            [04]: 2001:da8:ab:130:9cb6:5955:576c:a89b
      [04]: Realtek PCIe GBE Family Controller
          连接名: 以太网
          状态: 媒体连接已中断
Hyper-V 要求: 虚拟机监视器模式扩展: 是
               固件中已启用虚拟化: 是
               二级地址转换: 是
               数据执行保护可用: 是

```

图 3-1 Windows 下计算机基本信息

### 3.2 设备管理器查看 (2 分)

按链接列出设备，找出所有的键盘鼠标设备。写出每一个设备的从根到叶节点的路径。

键盘: ACPI(\_SB\_)#ACPI(PCI0)#ACPI(LPCB)#ACPI(PS2K)

鼠标 1: HID\VID\_10C4&PID\_8105\6&36ECBE81&0&0000

鼠标 2: HID\ELAN1000&COL01\5&16D04072&0&0000

### 3 隐藏分区与虚拟内存之分页文件查看 (2 分)

写出计算机主硬盘的各隐藏分区的大小 (MB): 499MB

写出 pagefile.sys 的文件大小 (Byte): 1920MB=20132659920B

C 盘根目录下其他隐藏的系统文件名字为:

ProgramDate Intel \$WINDOWS.~BT

### 3.4 任务管理与资源监视 (2 分)

任务管理器

文件(F) 选项(O) 查看(V)

进程 性能 应用历史记录 启动 用户 详细信息 服务

名称	PID	状态	用户名	CPU	内存(专用...)	描述
系统中断	-	正在运行	SYSTEM	30	K	延迟过程调用和中断服务例程
系统空闲进程	0	正在运行	SYSTEM	32	8 K	处理器空闲时间百分比
System	4	正在运行	SYSTEM	05	20 K	NT Kernel & System

写出你的计算机的 PID 最小的两个任务的名称、描述。

1.系统空闲进程：处理器空闲时间百分比

2.System：NT Kernel&System

### 3.5 计算机硬件详细信息 (2 分)

CPU 个数： 1 个      物理核数： 2 个      逻辑处理器个数： 4 个

L1 Cache 大小： 128KB      L2 Cache 大小： 512KB      L3 Cache 大小： 3.0MB



图 3-2 Windows 下计算机硬件详细信息

## 第 4 章 Linux 软硬件系统观察分析

### 4.1 计算机硬件详细信息 (3 分)

CPU 个数: 1      物理核数: 1      逻辑处理器个数: 1

MEM    Total: 971MB      Used: 555MB

Swap    Total: 1021MB      Used: 227MB

```
syq1160300901@syq1160300901-virtual-machine: ~  
syq1160300901@syq1160300901-virtual-machine:~$ lscpu  
Architecture:          x86_64  
CPU 运行模式:          32-bit, 64-bit  
Byte Order:            Little Endian  
CPU(s):                1  
每个核的线程数:      1  
每个座的核数:        1  
Socket(s):             1  
NUMA 节点:            1  
厂商 ID:               GenuineIntel  
CPU 系列:              6  
型号:                 78  
Model name:            Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz  
步进:                 3  
CPU MHz:               2400.001  
BogoMIPS:              4800.00  
超管理器厂商:         VMware  
虚拟化类型:           完全  
L1d 缓存:              32K  
L1i 缓存:              32K  
L2 缓存:               256K  
L3 缓存:               3072K  
NUMA node0 CPU(s):    0  
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca  
cmov pat pse36 clflush dts mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm con  
stant_tsc arch_perfmon pebs bts nopl xtopology tsc_reliable nonstop_tsc aperfmpe  
rf pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadl  
ine_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch epb fsg  
sbased_tsc_adjust bmi1 avx2 smep bmi2 invpcid rdseed adx smap xsaveopt dtherm ida  
arat pln pts hwp hwp_notify hwp_act_window hwp_epp
```

```
syq1160300901@syq1160300901-virtual-machine:~$ free -m  
              total        used        free      shared  buff/cache   available  
Mem:           971         557          63           3          350         225  
Swap:          1021         227          794  
syq1160300901@syq1160300901-virtual-machine:~$ cat /proc/cpuinfo|grep "physical  
id"| sort| uniq| wc -l  
1  
syq1160300901@syq1160300901-virtual-machine:~$ cat /proc/cpuinfo|grep "cores"|un  
iq  
cpu cores      : 1  
syq1160300901@syq1160300901-virtual-machine:~$ cat /proc/cpuinfo|grep "processor  
"|wc -l  
1
```

图 4-1 Linux 下计算机硬件详细信息



## 4.2 任务管理与资源监视 (2 分)

输入 top 命令，按键 N 实现 PID 的排序：

```
1887 syq1160+ 20 0 941088 33284 10488 S 0.0 3.3 0:03.09 fcitx-qimp+
1885 syq1160+ 20 0 814508 22112 1888 S 0.0 2.2 0:00.64 evolution-+
```

写出 Linux 下的 PID 最小的两个任务的 PID、名称 (Command)。

1.PID: 1885 Command: fcitx-qimp+

2.PID: 1887 Command: evolution-+

## 4.3 共享目录的文件系统信息 (3 分)

写出 Linux 下的 hitics 共享目录对应的文件系统的基本信息：

名称: vmhgfs-fuse 容量: 420G 挂载点: /mnt/hgfs

```
syq1160300901@syq1160300901-virtual-machine:~$ df -h
文件系统 容量 已用 可用 已用% 挂载点
udev      465M  0    465M   0% /dev
tmpfs     98M   6.4M   91M    7% /run
/dev/sda1  19G   4.5G   14G   26% /
tmpfs     486M  252K   486M   1% /dev/shm
tmpfs     5.0M  4.0K   5.0M   1% /run/lock
tmpfs     486M  0     486M   0% /sys/fs/cgroup
tmpfs     98M   68K    98M    1% /run/user/1000
vmhgfs-fuse 420G  59G   361G   14% /mnt/hgfs
```

## 4.4 Linux 下网络系统信息 (2 分)

写出本虚拟机的 IPv4 地址: 192.168.199.202

mac 地址: 00:0c:29:c6:37:f4

```
syq1160300901@syq1160300901-virtual-machine:~$ ifconfig
ens33  Link encap:以太网 硬件地址 00:0c:29:c6:37:f4
       inet 地址:192.168.199.202 广播:192.168.199.255 掩码:255.255.255.0
       inet6 地址: 2001:da8:ab:130:9339:c6bb:251f:d004/64 Scope:Global
       inet6 地址: fe80::16e4:c450:a9b3:4975/64 Scope:Link
       inet6 地址: 2001:da8:ab:130:6d9d:5947:4439:f0c3/64 Scope:Global
       UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
       接收数据包:147462 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:4203 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:21621183 (21.6 MB)  发送字节:323020 (323.0 KB)

lo     Link encap:本地环回
       inet 地址:127.0.0.1 掩码:255.0.0.0
       inet6 地址: ::1/128 Scope:Host
       UP LOOPBACK RUNNING  MTU:65536  跃点数:1
       接收数据包:1507 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:1507 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:100097 (100.0 KB)  发送字节:100097 (100.0 KB)
```

图 4-1 Linux 下网络系统信息

## 第5章 以16进制形式查看程序 Hello.c

### 5.1 请查看 HelloWin.c 与 HelloLinux.c 的编码 (3分)

HelloWin.c 采用 ANSI 编码, HelloLinux.c 采用 UTF-8 编码, 你的姓名 孙月晴 分别编码为: CB EF D4 C2 C7 E7 与 e5 ad 99 e6 9c 88 e6 99 b4。

HelloWin.c 在 Linux 下用 gcc 缺省模式编译后运行结果为: 如图

```

syq1160300901@syq1160300901-virtual-machine: ~/hitics
syq1160300901@syq1160300901-virtual-machine:~/hitics$ gcc hellowin.c
syq1160300901@syq1160300901-virtual-machine:~/hitics$ ./a.out
Hello 1160300901

```

Windows 下用工具 winhex 查看 hellowin.c, 如图:

main.c																	ANSI ASCII
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	23	69	6E	63	6C	75	64	65	20	3C	73	74	64	69	6F	2E	#include <stdio.
00000010	68	3E	0D	0A	0D	0A	69	6E	74	20	6D	61	69	6E	28	29	h> int main()
00000020	0D	0A	7B	0D	0A	20	20	20	20	70	72	69	6E	74	66	28	{ printf(
00000030	22	48	65	6C	6C	6F	20	31	31	36	30	33	30	30	39	30	"Hello 116030090
00000040	31	CB	EF	D4	C2	C7	E7	5C	6E	22	29	3B	0D	0A	20	20	lËiÔÂÇç\n");
00000050	20	20	72	65	74	75	72	6E	20	30	3B	0D	0A	7D	0D	0A	return 0; }

Linux 下运行 `od -Ax -tcx1 hello.c` 看 hellolinux.c, 如图:

```

syq1160300901@syq1160300901-virtual-machine: ~
Hello 1160300901孙月晴
syq1160300901@syq1160300901-virtual-machine:~$ od -Ax -tcx1 hellolinux.c
000000 # i n c l u d e < s t d i o .
23 69 6e 63 6c 75 64 65 20 3c 73 74 64 69 6f 2e
000010 h > \n i n t m a i n ( ) { \n \t
68 3e 0a 69 6e 74 20 6d 61 69 6e 28 29 7b 0a 09
000020 p r i n t f ( " H e l l o
70 72 69 6e 74 66 28 22 48 65 6c 6c 6f 20 31 31
000030 6 0 3 0 0 9 0 1 345 255 231 346 234 210 346 231
36 30 33 30 30 39 30 31 e5 ad 99 e6 9c 88 e6 99
000040 264 \ n " ) ; \n
b4 5c 6e 22 29 3b 0a 20 20 20 20 20 20 20 20 72
000050 e t u r n 0 ; \n } \n
65 74 75 72 6e 20 30 3b 0a 7d 0a
00005b
syq1160300901@syq1160300901-virtual-machine:~$ hexdump -C hellolinux.c
00000000 23 69 6e 63 6c 75 64 65 20 3c 73 74 64 69 6f 2e |#include <stdio.|
00000010 68 3e 0a 69 6e 74 20 6d 61 69 6e 28 29 7b 0a 09 |h>.int main(){..|
00000020 70 72 69 6e 74 66 28 22 48 65 6c 6c 6f 20 31 31 |printf("Hello 11|
00000030 36 30 33 30 30 39 30 31 e5 ad 99 e6 9c 88 e6 99 |60300901.....|
00000040 b4 5c 6e 22 29 3b 0a 20 20 20 20 20 20 20 72 |.\n");. r|
00000050 65 74 75 72 6e 20 30 3b 0a 7d 0a |return 0;.|
0000005b

```


## 5.2 请查看 HelloWin.c 与 HelloLinux.c 的回车 (3 分)

Windows 下的回车编码为: 0D 0A , Linux 下的回车编码为: 0a。

交叉打开文件的效果是 用文本编辑器交换打开后效果相同。

如图:

在 windows 中打开 hellolinux.c



The screenshot shows the Notepad++ application window titled "D:\hitics\hellolinux.c - Notepad++". The menu bar includes 文件(F), 编辑(E), 搜索(S), 视图(V), 编码(N), 语言(L), 设置(T), 工具(O), 宏(M), 运行(R), and 帮助(H). The toolbar contains various icons for file operations and editing. The editor window shows the file "hellolinux.c" with the following C code:

```
1  #include <stdio.h>
2  int main() {
3      printf("Hello 1160300901孙月晴\n");
4      return 0;
5  }
```

在 linux 中打开 hellowin.c



The screenshot shows the gedit application window titled "hellowin.c (~/.hitics) - gedit". The window has a sidebar with icons for the Dash, Home, Firefox, and Files applications. The top bar includes a "打开(O)" button and a keyboard shortcut icon. The editor window shows the file "hellowin.c" with the following C code:

```
#include <stdio.h>

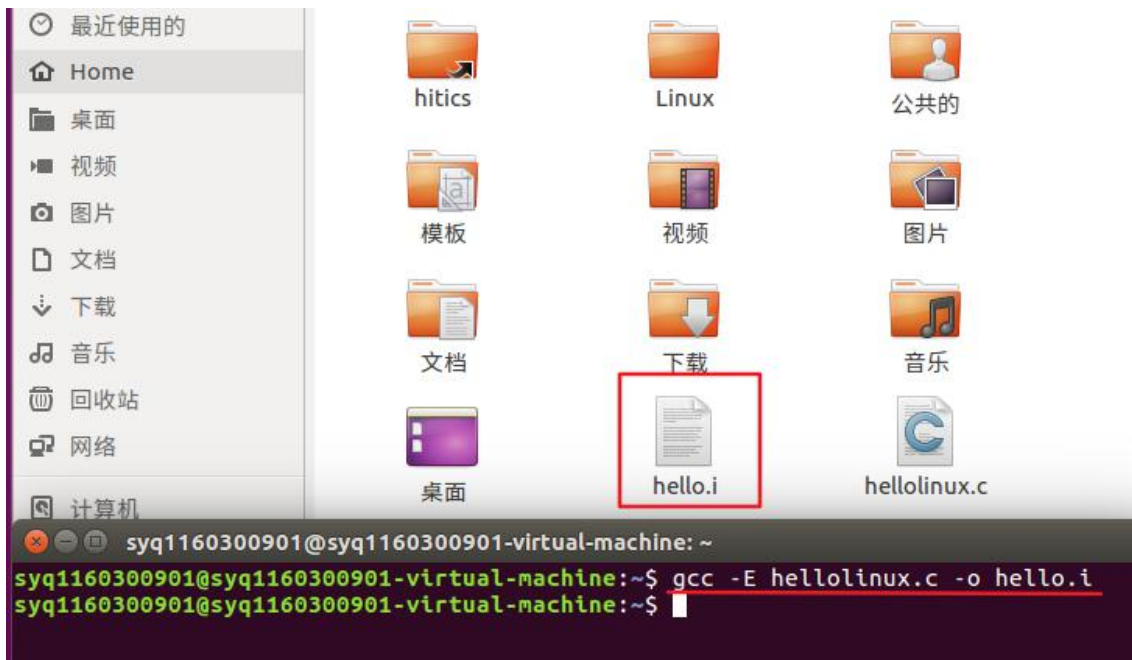
int main()
{
    printf("Hello 1160300901孙月晴\n");
    return 0;
}
```

## 第 6 章 程序的生成 Cpp、Gcc、As、ld

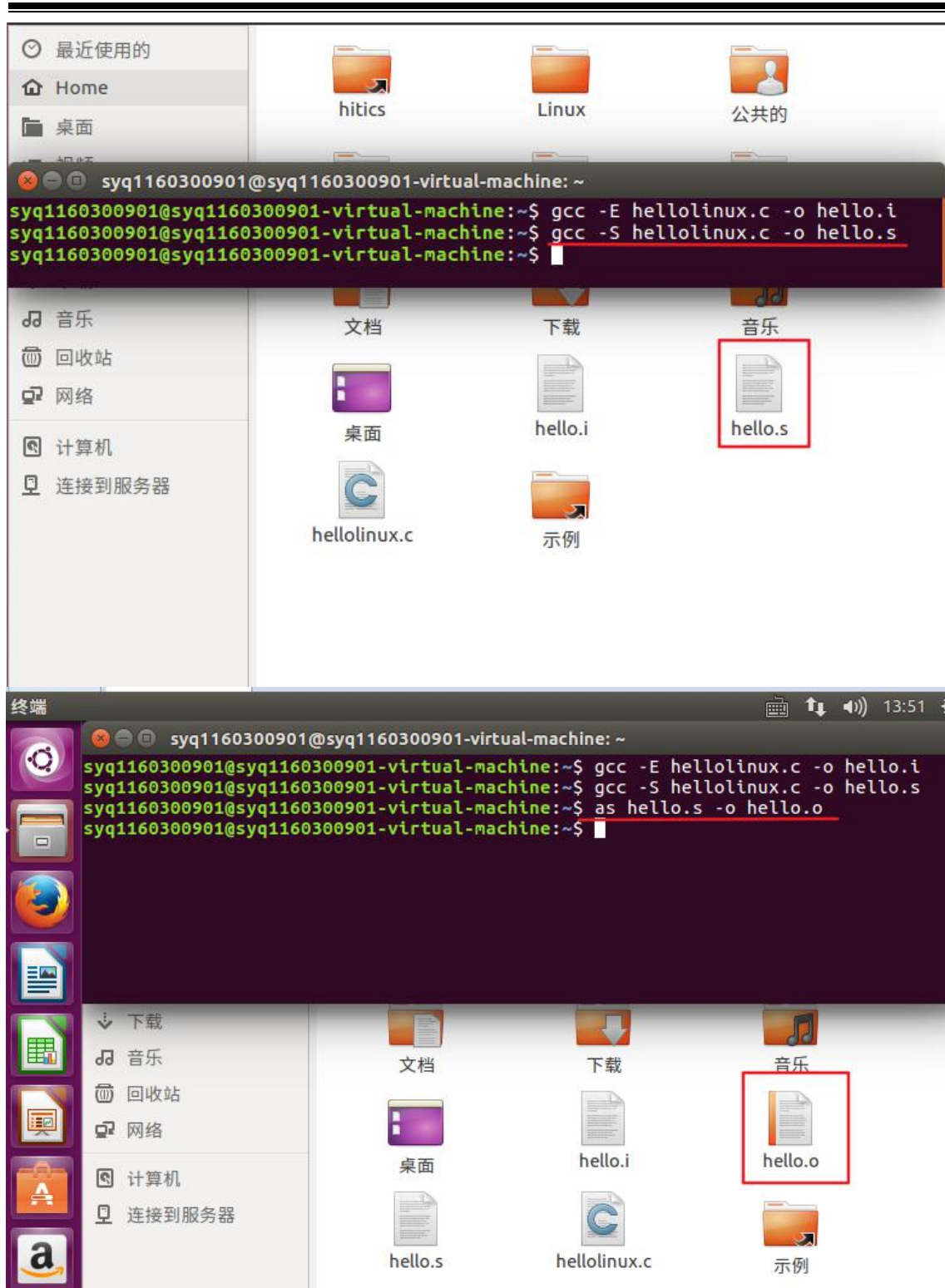
### 6.1 请提交每步生成的文件（4 分）

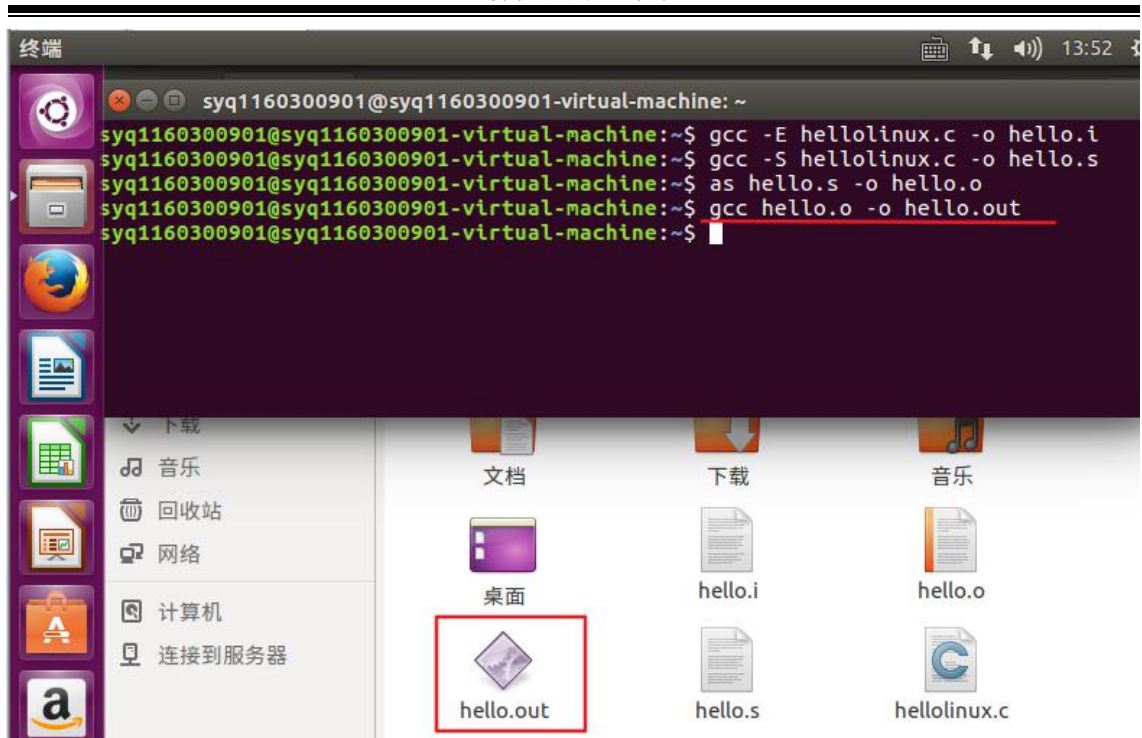
hello.i hello.s hello.o hello.out

文件生成过程：







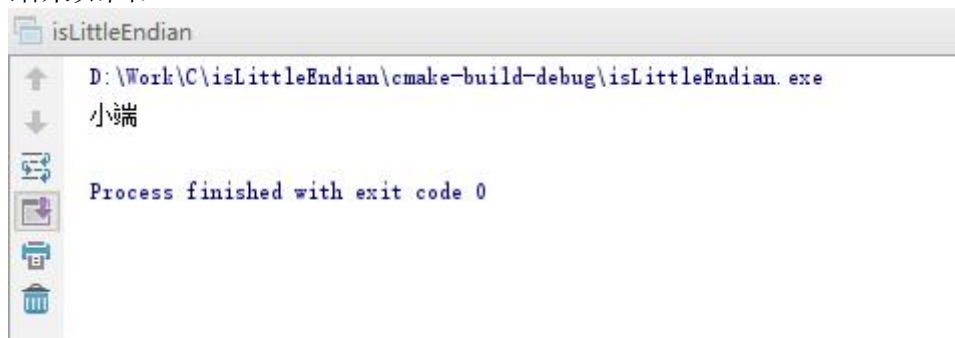


## 第 7 章 计算机系统的基本信息获取编程

### 7.1 请提交源程序文件 (10 分)

#### (1) isLittleEndian.c

运行结果如图：

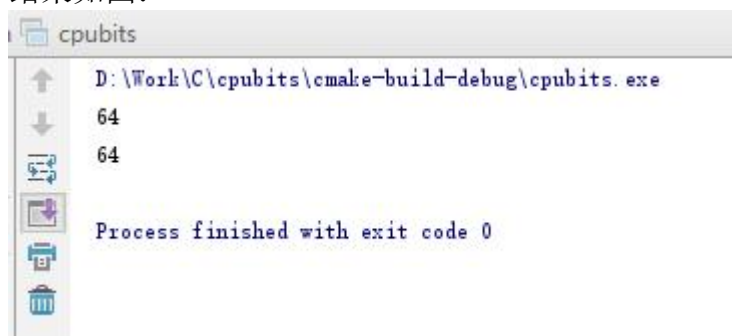


```
isLittleEndian
D:\Work\C\isLittleEndian\cmake-build-debug\isLittleEndian.exe
小端
Process finished with exit code 0
```

#### (2) cpuWordSize.c

1. 根据栈中两个连续存放指针的地址偏移量来计算字长
2. 编译原理中，long 型变量一般占 cpu 个字节数, 用 `8 * sizeof(long *)` 来判断

运行结果如图：



```
cpubits
D:\Work\C\cpubits\cmake-build-debug\cpubits.exe
64
64
Process finished with exit code 0
```

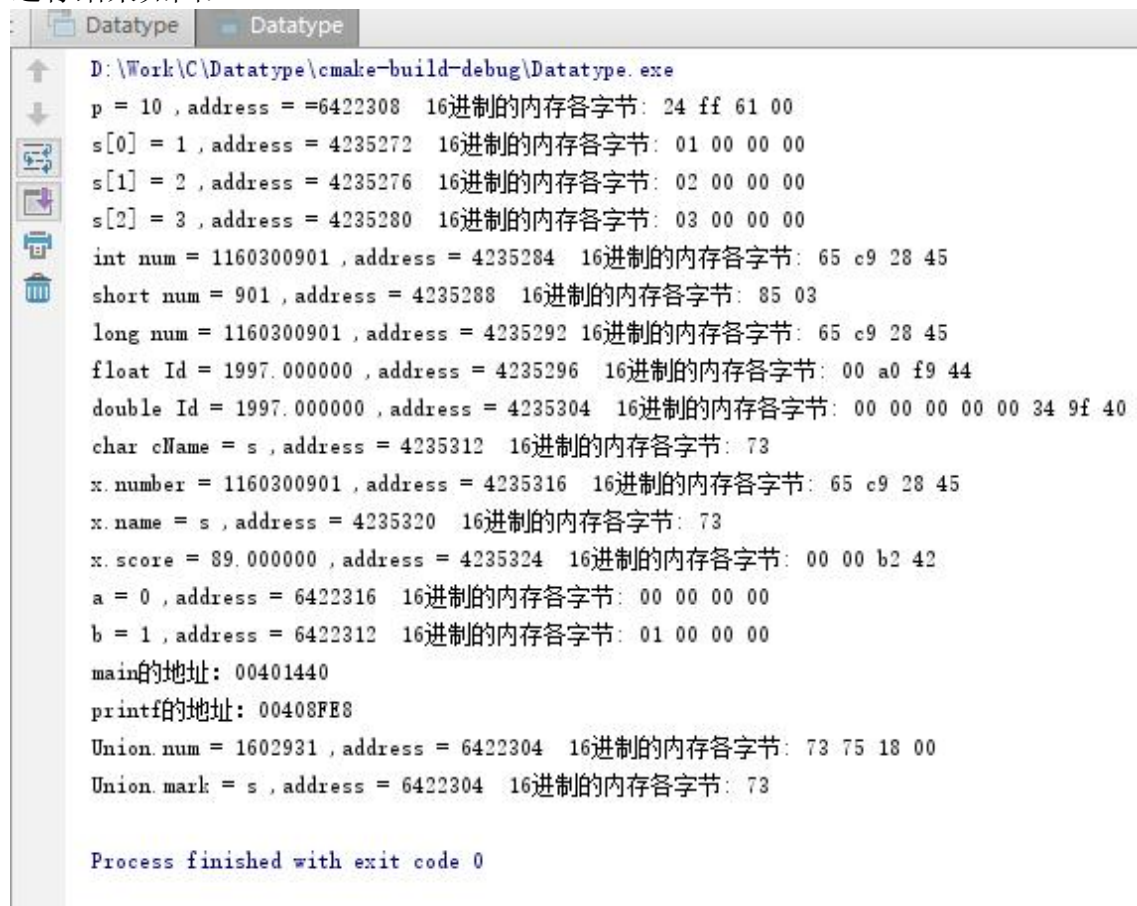


## 第 8 章 计算机数据类型的本质

### 8.1 请提交源程序文件 Datatype.c (10 分)

要求有 main 函数进行测试。

运行结果如图：



```
D:\Work\C\Datatype\cmake-build-debug\Datatype.exe
p = 10 , address = 6422308 16进制的内存各字节: 24 ff 61 00
s[0] = 1 , address = 4235272 16进制的内存各字节: 01 00 00 00
s[1] = 2 , address = 4235276 16进制的内存各字节: 02 00 00 00
s[2] = 3 , address = 4235280 16进制的内存各字节: 03 00 00 00
int num = 1160300901 , address = 4235284 16进制的内存各字节: 65 c9 28 45
short num = 901 , address = 4235288 16进制的内存各字节: 85 03
long num = 1160300901 , address = 4235292 16进制的内存各字节: 65 c9 28 45
float Id = 1997.000000 , address = 4235296 16进制的内存各字节: 00 a0 f9 44
double Id = 1997.000000 , address = 4235304 16进制的内存各字节: 00 00 00 00 00 34 9f 40
char cName = s , address = 4235312 16进制的内存各字节: 73
x.number = 1160300901 , address = 4235316 16进制的内存各字节: 65 c9 28 45
x.name = s , address = 4235320 16进制的内存各字节: 73
x.score = 89.000000 , address = 4235324 16进制的内存各字节: 00 00 b2 42
a = 0 , address = 6422316 16进制的内存各字节: 00 00 00 00
b = 1 , address = 6422312 16进制的内存各字节: 01 00 00 00
main的地址: 00401440
printf的地址: 00408FE8
Union.num = 1602931 , address = 6422304 16进制的内存各字节: 73 75 18 00
Union.mark = s , address = 6422304 16进制的内存各字节: 73

Process finished with exit code 0
```

## 第 9 章 程序运行分析

### 9.1 sum 的分析 (20 分)

```
int sum(int a[], unsigned len)
{
    int i, sum=0;
    for(i=0; i<=len-1; i++)
        sum+=a[i];
    return sum;
}
```

(1) 为什么程序这样的运行结果？

从有符号数到无符号数的隐式强制类型很容易引起错误。

因为参数 len 是无符号的，计算  $0-1$  将使用无符号计算，这等价于模数加法。结果得到 UMax。≤ 比较同样使用无符号数比较，而因为任何数都是小于或者等于 UMax 的，所以这个比较总是为真！因此代码将试图访问数组 a 的非法元素。

(2) 怎么改进程序？

有两种方法可以改正这段代码，其一是将 len 声明为 int 类型，其二是将 for 循环的测试条件改为  $i < len$ 。

### 9.2 float 的分析 (20 分)

运行结果如图：

```

这个浮点数的值为: 61.419998
请输入一个浮点数: 61.419998
61.419998 D:\Work\C\float\cmake-build-debug\float.exe
这个浮点数的值为: 61.419998 请输入一个浮点数: 10.186810
请输入一个浮点数: 61.419999 10.186810
61.419999 这个浮点数的值为: 10.186810
这个浮点数的值为: 61.419998 请输入一个浮点数: 10.186811
请输入一个浮点数: 61.420000 10.186811
61.420000 这个浮点数的值为: 10.186811
这个浮点数的值为: 61.419998 请输入一个浮点数: 10.186812
请输入一个浮点数: 10.186812
这个浮点数的值为: 10.186812
61.419999 请输入一个浮点数: 10.186813
请输入一个浮点数: 61.419999 10.186813
这个浮点数的值为: 61.419998 这个浮点数的值为: 10.186813
请输入一个浮点数: 61.420000 请输入一个浮点数: 10.186814
61.420000 10.186814
这个浮点数的值为: 61.419998 这个浮点数的值为: 10.186814
请输入一个浮点数: 61.420001 请输入一个浮点数: 10.186815
61.420001 10.186815
这个浮点数的值为: 61.420002 这个浮点数的值为: 10.186815
请输入一个浮点数: 0 请输入一个浮点数: 0
0 0
这个浮点数的值为: 0.000000 这个浮点数的值为: 0.000000

```

各输入数的 IEEE 标准浮点数表示为:

第一组:

```

61.419997
01000010011101011010111000010011
61.419998
01000010011101011010111000010011
61.419999
01000010011101011010111000010100
61.420000
01000010011101011010111000010100
61.420001
01000010011101011010111000010100

```

第二组:

```

10.186810
01000001001000101111110100101100
10.186811
01000001001000101111110100101101
10.186812

```

01000001001000101111110100101110

10.186813

01000001001000101111110100101111

10.186814

01000001001000101111110100110000

10.186815

01000001001000101111110100110001

(1) 根据 IEEE 规则, float 在计算机中的存储默认就近舍入。Float 用的是科学计数法, 数越小, 小数部分越精确, 也就是说, 可表示的数并不是均匀分布的, 越靠近原点数越稠密, 表示的精度越高。

第一组数的阶码是 5, 十进制小数的精确的位数大约到  $2^{-23} \times 2^5 \approx 3.8 \times 10^{-6}$ , 也就是说, 能精确到小数点后第 5 位, 第 6 位是不精确的, 大约连续 4 个 6 位小数的标准浮点格式转化为十进制输出的结果是相同的, 应该是这几个数从大到小排列在中间的一个数;

第二组数的阶码是 3, 十进制小数的精确的位数大约到  $2^{-23} \times 2^3 \approx 9 \times 10^{-7}$ , 也就是说, 能精确到小数点后第 6 位, 第 7 位是不精确的, 那么每个 6 位小数都可以精确地输出。

(2) 使用浮点数是应注意对精度的要求, 越靠近原点, 数越稠密, 表示的精度越高。如果精确度要求很高, 数又比较大, 可以先除以 10 的  $i$  次方, 使其更接近原点, 计算完再乘以 10 的  $i$  次方, 这样可以稍微提高精确度。

## 第 10 章 总结

### 10.1 请总结本次实验的收获

1. 首先通过本次实验我熟练掌握了 linux 下 C 工具的应用,通过 Vmware 12、Ubuntu 16.04 LTS 64 位以及 Visual Studio 、vi/vim/gpedit+gcc 等实验环境与工具,独立解决了实验中的问题;
2. 运用现代工具任务管理器、资源监视器、性能监视器、winhex、鲁大师等进行计算机软硬件系统的观察与分析,熟悉了 windows 下和 Linux 下命令行的使用;
3. 初步掌握计算机系统的基本知识与各种类型的数据表示,更加熟悉了计算机的基本结构和 C 语言基本数据类型;
4. 实验的过程中遇到了很多棘手的问题,比如虚拟机不能联网、解析不了 `cn.archive.ubuntu.com` 的域名、软件不能下载和更新等,通过查阅资料和反复调试最后都一个个解决了,虽然用了很长时间解决这些问题,但是自己收获也很多,我准备把实验中遇到的问题的解决方法写篇博客,希望也能对别人问题的解决提供帮助。

### 10.2 请给出对本次实验内容的建议

1. 希望实验的 ppt 和 pdf 文件以及实验模板能统一一致,在实验过程中发现有些同学因为 pdf 文件误解了实验内容,目标不清晰;有些 ppt 中要求要做的实验模板上并没有要求,比如 64 位 Linux 下 32 位编译运行;
2. 我个人觉得本次实验的分值分布有些不合理,比如 2.2 Ubuntu 与 Windows 目录共享,中途遇到了很多问题,花费了很多时间才成功,但大约有 5 分,但是最后两题每道题 20 分。

注:本章为酌情加分项。

## 参考文献

- [1] 大卫 R.奥哈拉伦，兰德尔 E。布莱恩特. 深入理解计算机系统[M]. 机械工业出版社.2017.7
- [2] KANAMORI H. Shaking Without Quaking[J]. Science， 1998， 279（5359）： 2063-2064.
- [3] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science ， 1998 ， 281 ： 331-332[1998-09-23]. <http://www.sciencemag.org/cgi/collection/anatmorp>.