

哈尔滨工业大学

实验报告

实 验（一）

题 目 计算机系统漫游

学 号 120L022115

班 级 2003007

学 生 王炳轩

指 导 教 师 吴锐

实 验 地 点 G709

实 验 日 期 2022-03-18

哈尔滨工业大学计算学部

目 录

第 1 章 实验基本信息	4 -
1.1 实验目的	4 -
1.2 实验环境与工具	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 -
第 2 章 实验环境建立	5 -
2.1 WINDOWS 下 HELLO 程序的编辑与运行 (5 分)	5 -
2.2 LINUX 下 HELLO 程序的编辑与运行 (5 分)	6 -
第 3 章 WINDOWS 软硬件系统观察分析	8 -
3.1 查看计算机基本信息 (2 分)	8 -
3.2 设备管理器查看 (2 分)	9 -
3 隐藏分区与虚拟内存之分页文件查看 (2 分)	11 -
3.4 任务管理与资源监视 (2 分)	12 -
3.5 CPUZ 下的计算机硬件详细信息 (2 分)	12 -
第 4 章 LINUX 软硬件系统观察分析	14 -
4.1 计算机硬件详细信息 (3 分)	14 -
4.2 任务管理与资源监视 (2 分)	14 -
4.3 磁盘任务管理与资源监视 (3 分)	15 -
4.4 LINUX 下网络系统信息 (2 分)	15 -
第 5 章 LINUX 下的 SHOWBYTE 程序	16 -
5.1 源程序提交 (8 分)	16 -
5.2 运行结果比较 (2 分)	16 -
第 6 章 程序的生成 CPP、GCC、AS、LD	17 -
6.1 请提交每步生成的文件 (10 分)	17 -
第 7 章 计算机数据类型的本质	18 -
7.1 运行 SIZEOF.C 填表 (5 分)	18 -
7.2 请提交源程序文件 SIZEOF.C (5 分)	18 -
第 8 章 程序运行分析	19 -
8.1 SUM 的分析 (10 分)	19 -
8.2 FLOAT 的分析 (10 分)	19 -

8.3 程序优化（20 分）	- 21 -
第 9 章 总结	- 23 -
9.1 请总结本次实验的收获.....	- 23 -
9.2 请给出对本次实验内容的建议.....	- 23 -
参考文献.....	- 24 -

第 1 章 实验基本信息

1.1 实验目的

运用现代工具进行计算机软硬件系统的观察与分析。

运用现代工具进行 Linux 下 C 语言的编程调试，掌握程序的生成步骤。

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

Surface Go 3: x64、Pentium G6500Y @ 1.1GHz、16GB RAM、128GB SSD。

1.2.2 软件环境

Windows 11、Windows Subsystem for Linux、Ubuntu 20.04

1.2.3 开发工具

Code::Blocks、gcc、vim

1.3 实验预习

上实验课前，必须认真预习实验 PPT。

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。

编写 `hello.c`，显示“Hello 120L022115 王炳轩”，并在不同的系统运行。

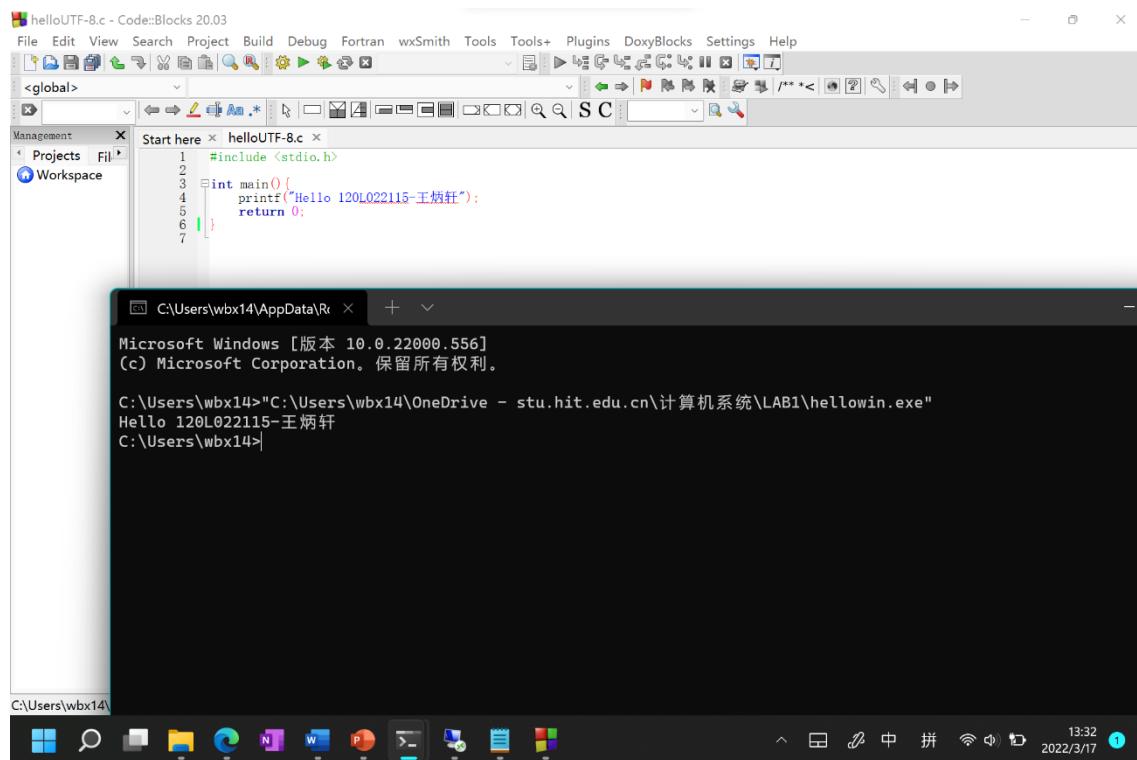
编写 `showbyte.c` 以 16 进制显示文件 `hello.c` 等的內容。

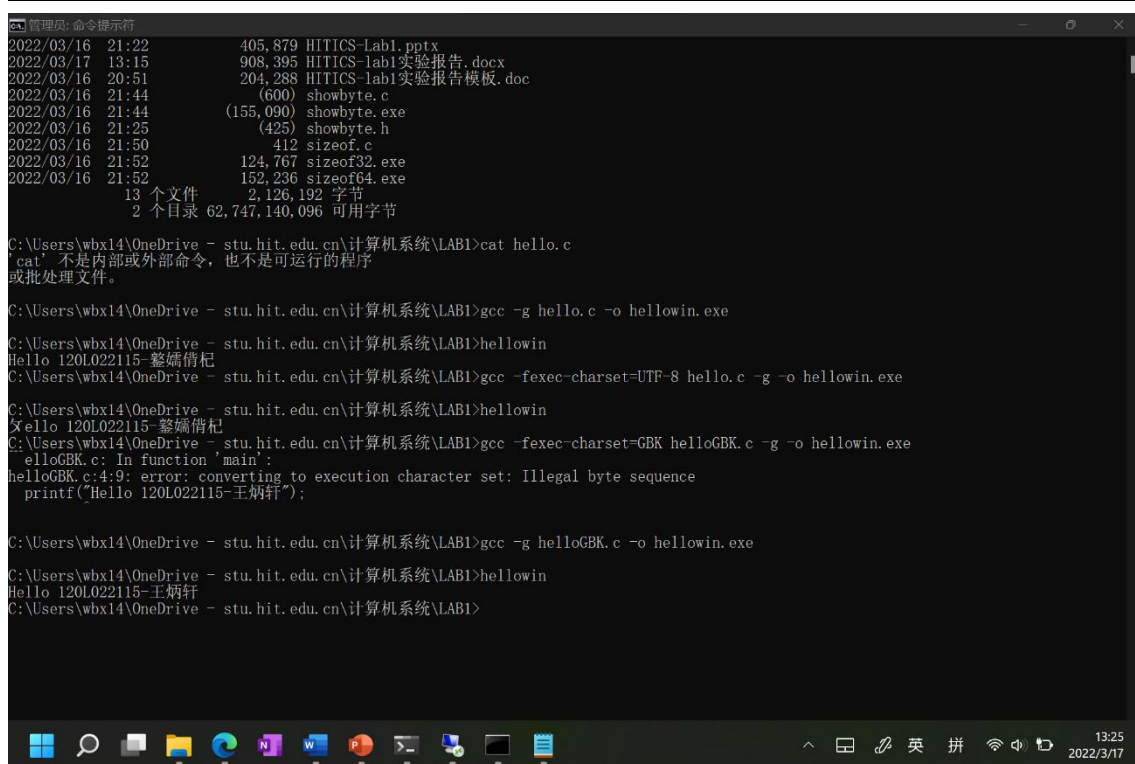
编写 `sizeof.c`，在不同的系统与架构运行，主程序打印每个类型变量的所占内存数。

第 2 章 实验环境建立

2.1 Windows 下 hello 程序的编辑与运行 (5 分)

截图：要求有 Windows 状态行，Visual Studio 界面，源程序界面，运行结果界面。





```
管理员: 命令提示符
2022/03/16 21:22      405,879 HITICS-lab1.pptx
2022/03/17 13:15      908,395 HITICS-lab1实验报告.docx
2022/03/16 20:51      204,288 HITICS-lab1实验报告模板.doc
2022/03/16 21:44          (600) showbyte.c
2022/03/16 21:44      (155,090) showbyte.exe
2022/03/16 21:25          (425) showbyte.h
2022/03/16 21:50          412 sizeof.c
2022/03/16 21:52      124,767 sizeof32.exe
2022/03/16 21:52      152,236 sizeof64.exe
13 个文件      2,126,192 字节
2 个目录 62,747,140,096 可用字节

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>cat hello.c
'cat' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>gcc -g hello.c -o helloworld.exe

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>helloworld.exe
Hello 120L022115-王炳轩

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>gcc -fexec-charset=UTF-8 hello.c -g -o helloworld.exe

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>helloworld.exe
Hello 120L022115-王炳轩

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>gcc -fexec-charset=GBK helloGBK.c -g -o helloworld.exe
helloGBK.c:4:9: error: converting to execution character set: Illegal byte sequence
printf("Hello 120L022115-王炳轩");
^~~~~~

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>gcc -g helloGBK.c -o helloworld.exe

C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>helloworld.exe
Hello 120L022115-王炳轩
C:\Users\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1>
```

图 2-1 Windows 下 hello 运行截图

2.2 Linux 下 hello 程序的编辑与运行 (5 分)

截图：要求有 Ubuntu 的 OS 窗口，Codeblocks 界面，源程序界面，运行结果界面。

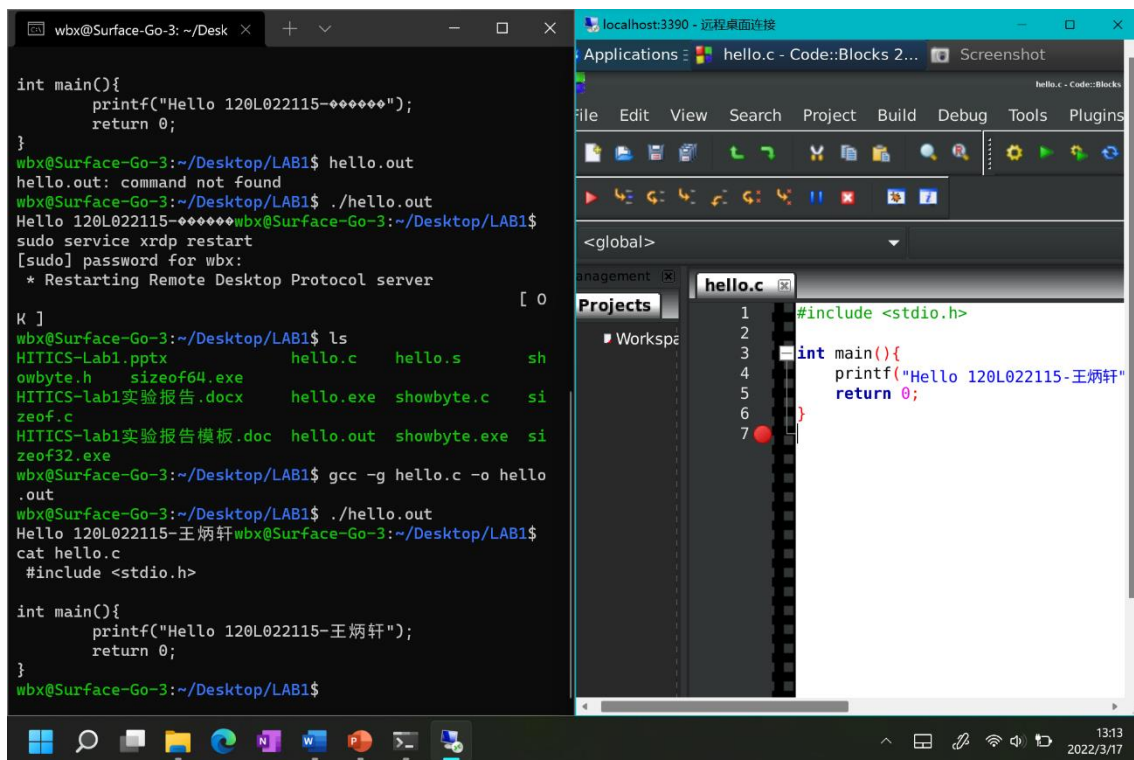
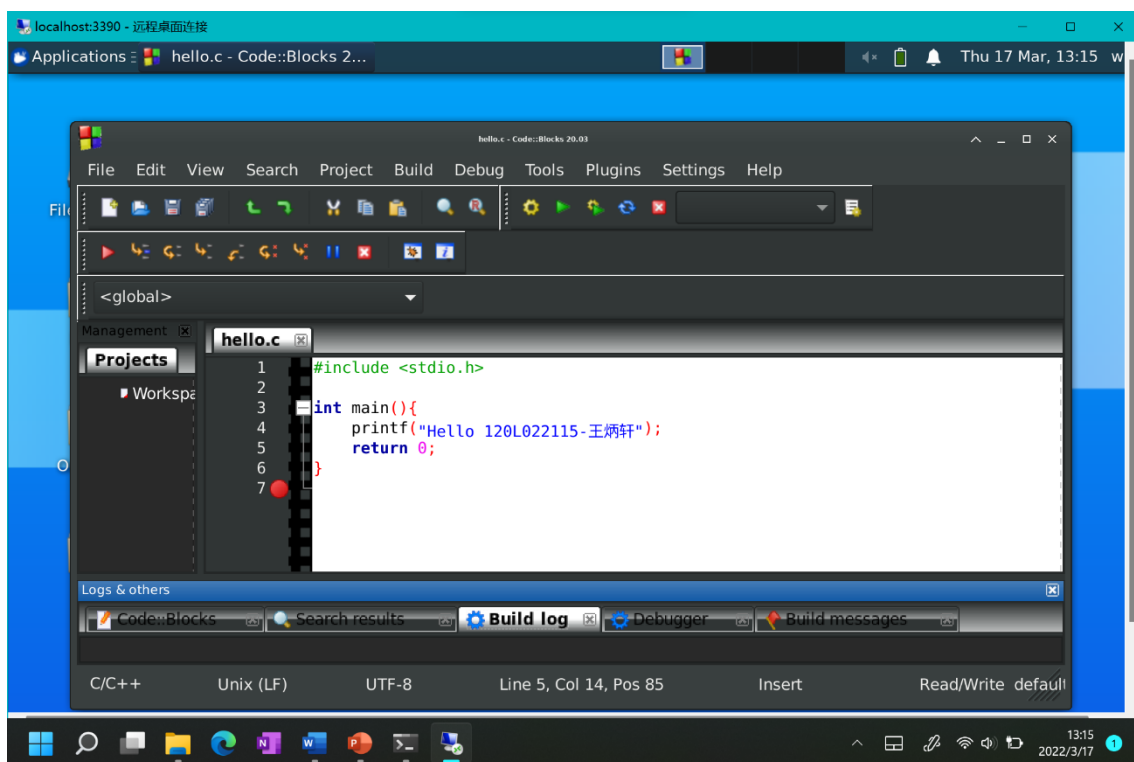
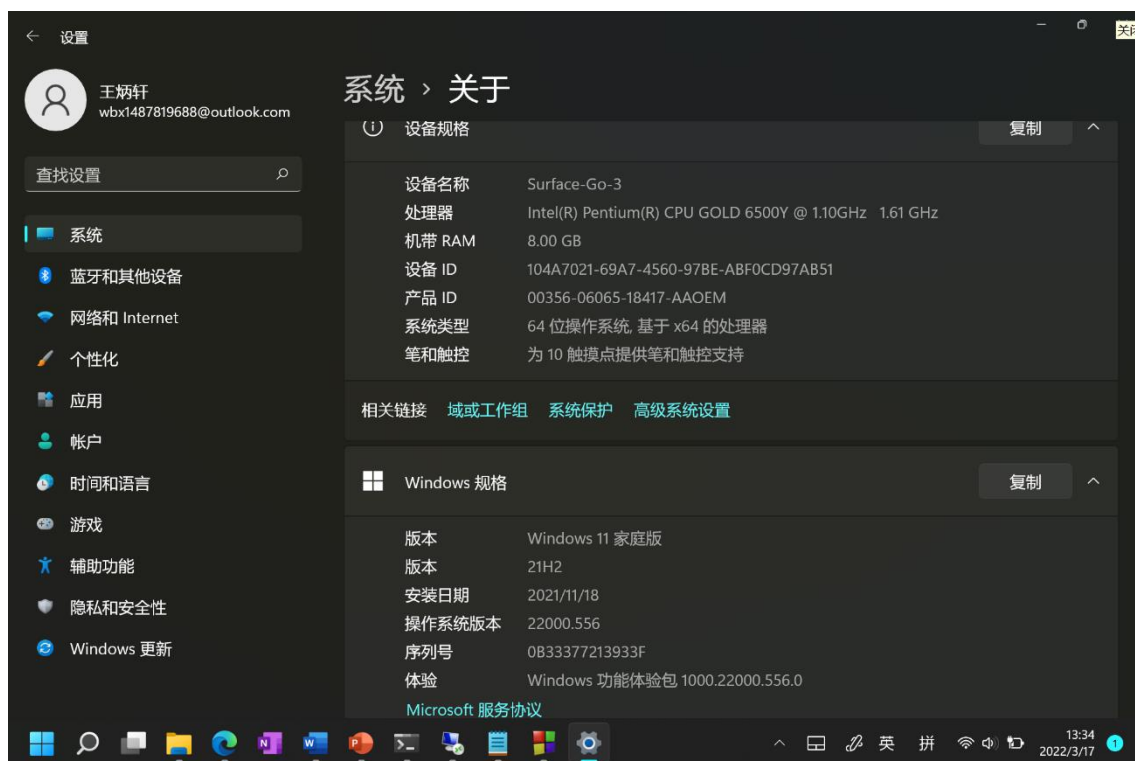


图 2-2 Linux 下 hello 运行截图

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)

运行 Windows 管理工具中的“系统信息”程序，查看 CPU、物理内存、系统目录、启动设备、页面文件等信息，并截图



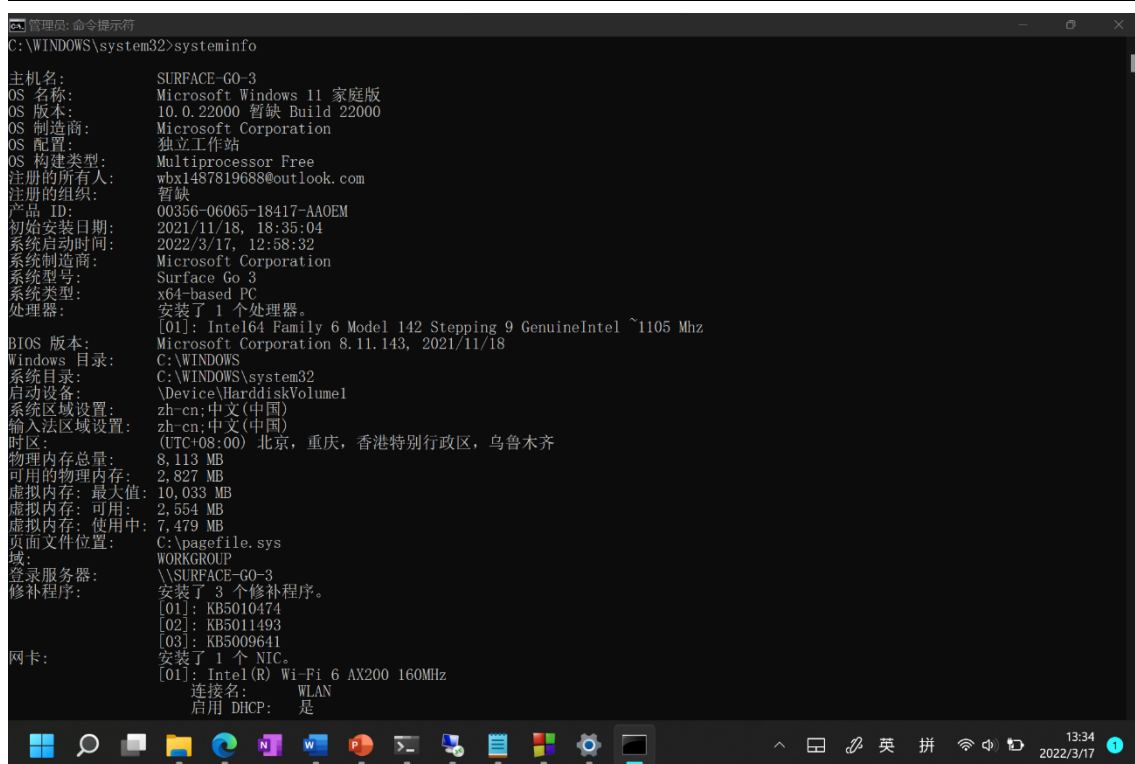
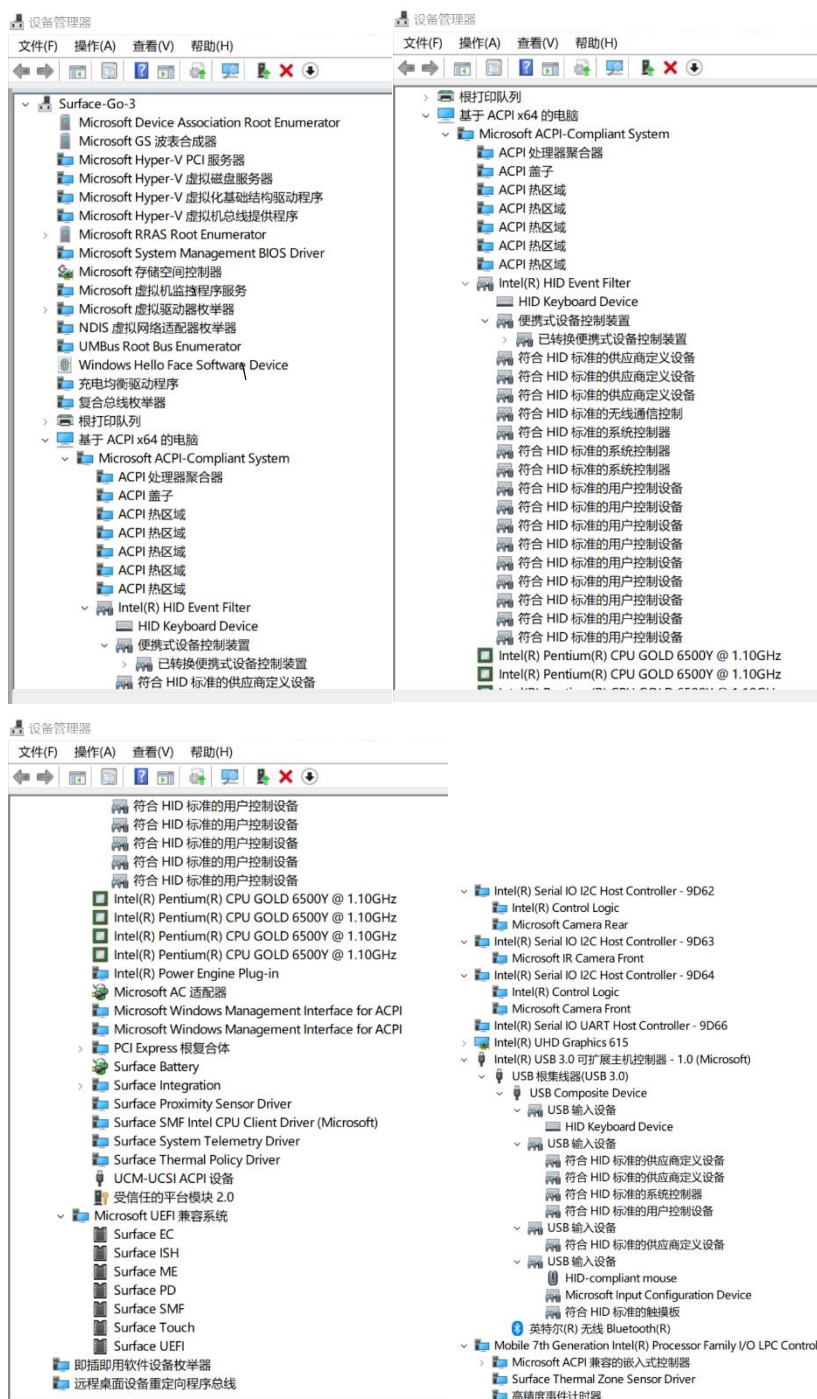


图 3-1 Windows 下计算机基本信息

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。



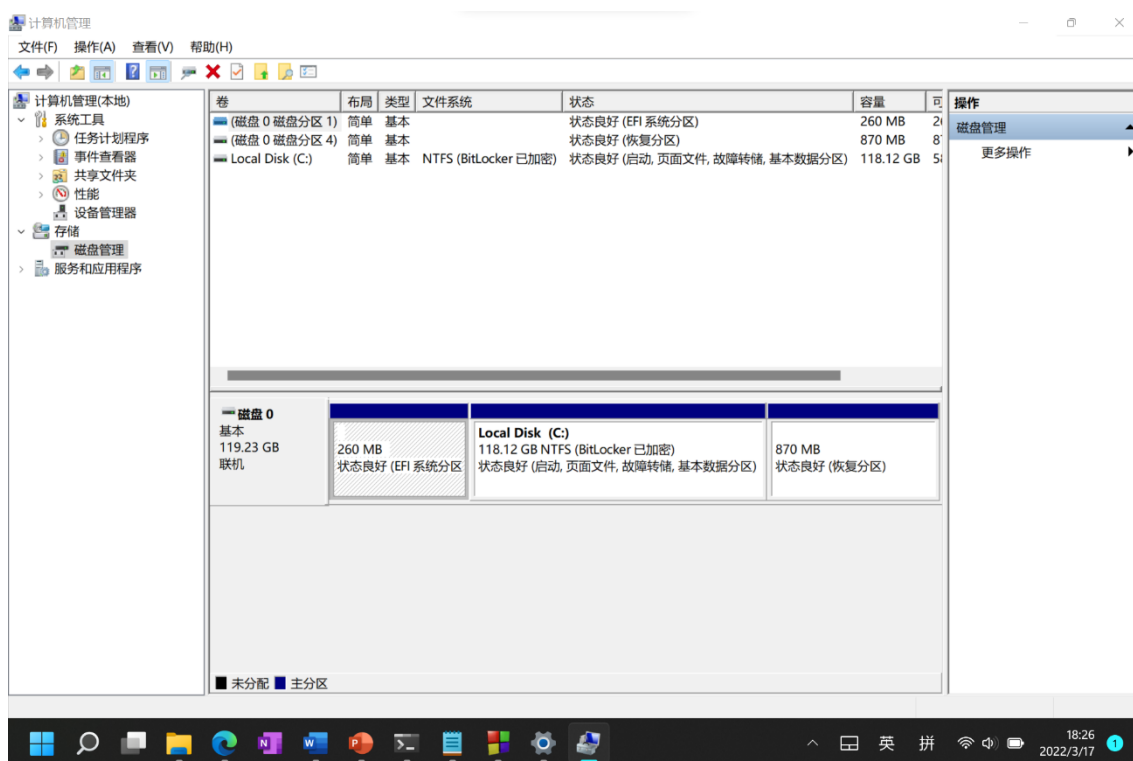
键盘: Surface-Go-3\基于 ACPI x64 的电脑\Microsoft ACPI-Compliant System\PCI Express 根复合体\Intel(R) USB 3.0 可扩展主机控制器 – 1.0 (Microsoft) \USB 根集线器(USB 3.0)\USB Composite Device\USB 输入设备\HID Keyboard Device

触摸屏: Surface-Go-3\基于 ACPI x64 的电脑\Microsoft ACPI-Compliant System\PCI Express 根复合体\Intel(R) USB 3.0 可扩展主机控制器 – 1.0 (Microsoft) \USB 根集线器(USB 3.0)\USB Composite Device\USB 输入设备\HID-compliant mouse

触摸板: Surface-Go-3\基于 ACPI x64 的电脑\Microsoft ACPI-Compliant System\PCI Express 根复合体\Intel(R) USB 3.0 可扩展主机控制器 – 1.0 (Microsoft) \USB 根集线器(USB 3.0)\USB Composite Device\USB 输入设备\符合 HID 标准的触摸板

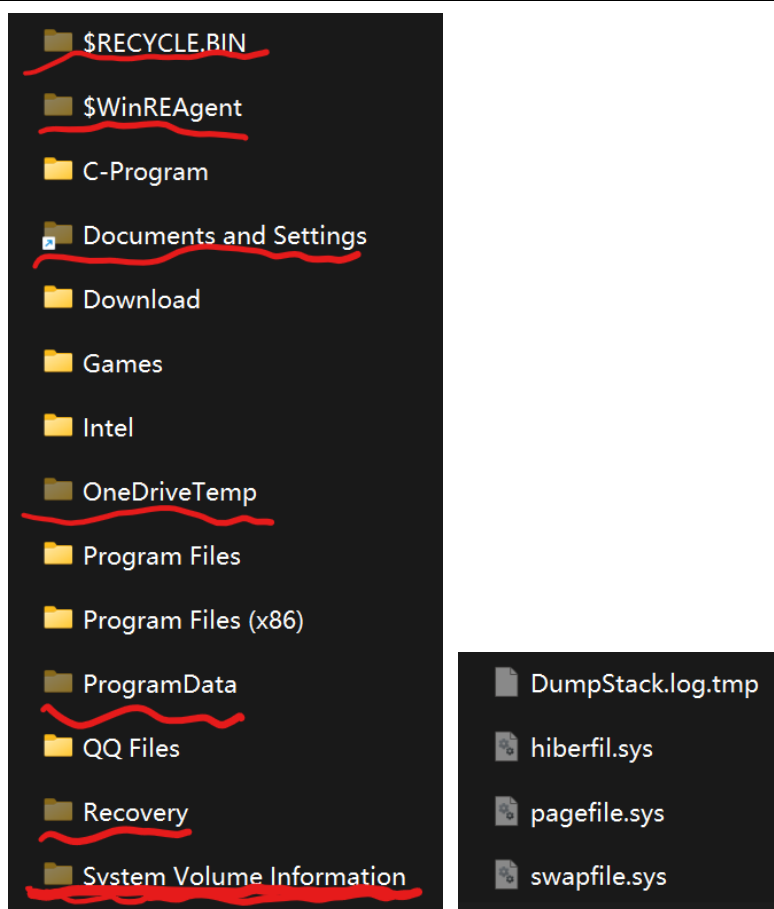
3 隐藏分区与虚拟内存之分页文件查看 (2 分)

写出计算机主硬盘的各隐藏分区的大小 (MB): 1130MB



写出 pagefile.sys 的文件大小 (Byte): 1.87GB

C 盘根目录下其他隐藏的系统文件名字为:



3.4 任务管理与资源监视 (2 分)

写出你的计算机的 PID 为“-”、最小与最大的 3 个任务的 PID、名称、描述。

名称	PID [^]	状态	用户名	CPU	内存(活动的...	体系结构	描述
系统中断	-	正在运行	SYSTEM	02	0 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	90	8 K		处理器空闲时间百分比
System	4	正在运行	SYSTEM	00	20 K		NT Kernel & System

名称	PID [^]	状态	用户名	CPU	内存(活动的...	体系结构	描述
IntelCpHeciSvc.exe	1768	正在运行	SYSTEM	00	416 K	x64	IntelCpHeciSvc Executable
conhost.exe	1760	正在运行	wbx14	00	412 K	x64	控制台窗口主进程
svchost.exe	1724	正在运行	LOCAL SE...	00	1,536 K	x64	Windows 服务主进程

3.5 CPUZ 下的计算机硬件详细信息 (2 分)

CPU 个数: 1 物理核数: 2 逻辑处理器个数: 4

L3 Cache 大小: 4MB

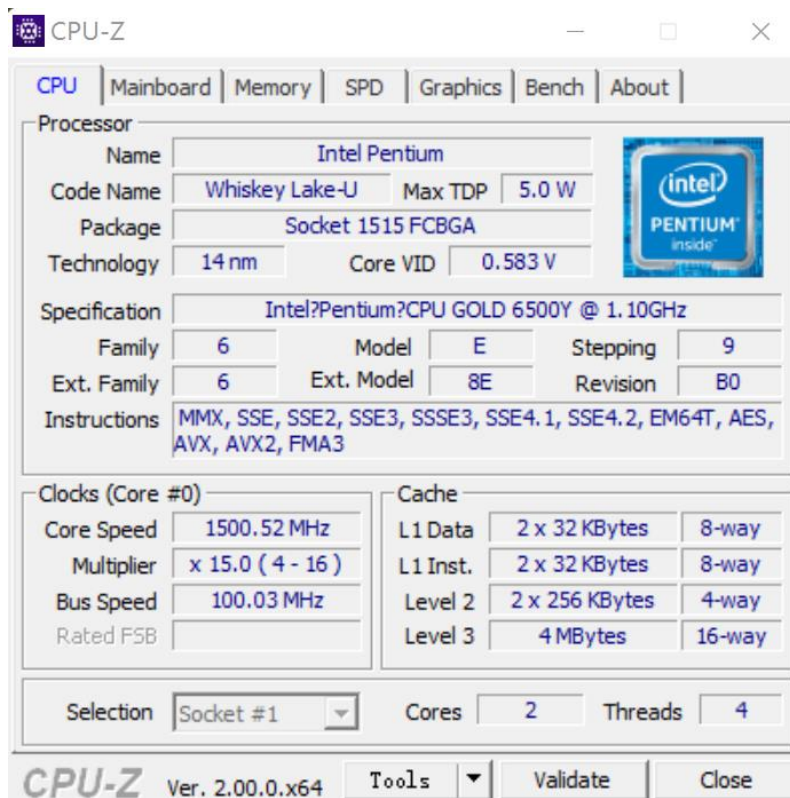
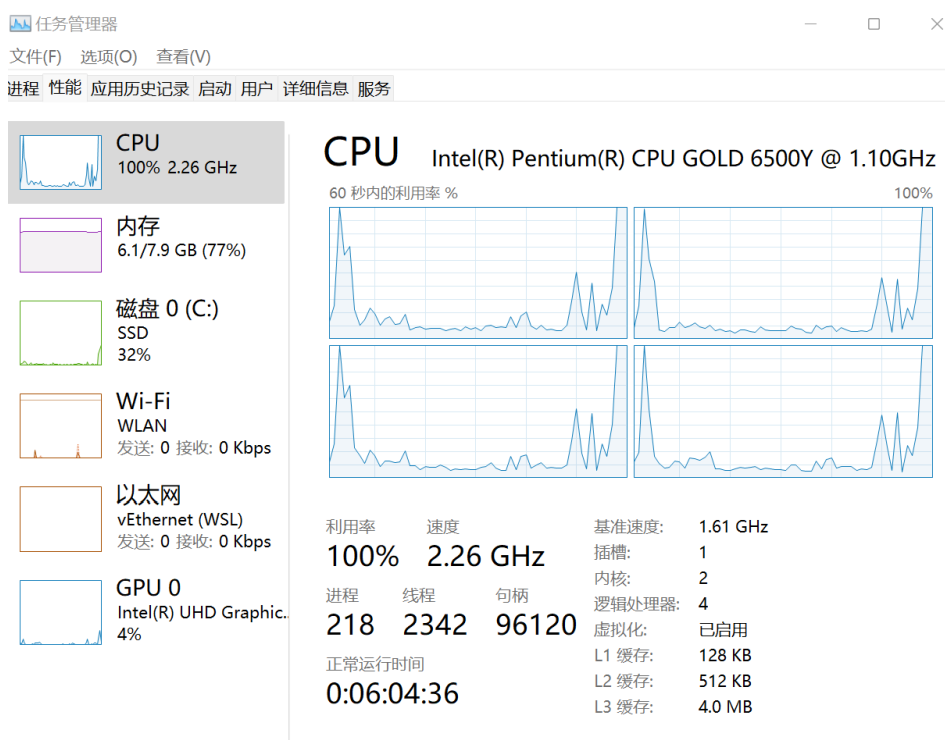


图 3-2 CPUZ 下 CPU 的基本信息

第 4 章 Linux 软硬件系统观察分析

(泰山服务器)

4.1 计算机硬件详细信息 (3 分)

CPU 个数: 96 物理核数: 96 逻辑处理器个数: 96

MEM Total: 197239480kB Used: 24669548kB Swap: 8388604kB

```
ex] stu_120L022115@node210: ~  
ex] stu_120L022115@node210: ~  
stu_120L022115@node210:~$ cat /proc/cpuinfo  
processor       : 0  
bogomips       : 200.00  
features       : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp  
p asimdfhm  
CPU implementer : 0x48  
CPU architecture: 8  
CPU variant    : 0x1  
CPU part       : 0xd01  
CPU revision   : 0  
  
processor       : 1  
bogomips       : 200.00  
features       : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp  
p asimdfhm  
CPU implementer : 0x48  
CPU architecture: 8  
CPU variant    : 0x1  
CPU part       : 0xd01  
CPU revision   : 0  
  
processor       : 2  
bogomips       : 200.00  
features       : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp  
p asimdfhm  
CPU implementer : 0x48  
CPU architecture: 8  
CPU variant    : 0x1  
CPU part       : 0xd01  
CPU revision   : 0  
  
stu_120L022115@node210:~$ cat /proc/meminfo  
MemTotal:        197239480 kB  
MemFree:         172569932 kB  
MemAvailable:    181370380 kB  
Buffers:         834744 kB  
Cached:          7468888 kB  
SwapCached:      764 kB  
Active:          11827436 kB  
Inactive:        6555936 kB  
Active(anon):    8770192 kB  
Inactive(anon):  1470460 kB  
Active(file):    3057244 kB  
Inactive(file):  5085476 kB  
Unevictable:     17140 kB  
Mlocked:         17140 kB  
SwapTotal:       8388604 kB  
SwapFree:        8359560 kB  
Dirty:           2332 kB  
Writeback:       0 kB  
AnonPages:       9696616 kB  
Mapped:          1714944 kB  
Shmem:           163016 kB  
Reclaimable:     1971444 kB  
Slab:            4137872 kB  
SReclaimable:    1971444 kB  
SUnreclaim:      2166428 kB  
KernelStack:    100464 kB  
PageTables:      92760 kB  
VFS_Unstable:    0 kB  
Source:          0 kB
```

图 4-1 Linux 下计算机硬件详细信息截图

4.2 任务管理与资源监视 (2 分)

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

PID:1,system

PID:2,kthreadd


```

stu_120L022115@node210: ~
top - 19:25:58 up 1 day, 8:55, 8 users, load average: 1.18, 1.61, 2.10
Tasks: 1845 total, 1 running, 1844 sleeping, 0 stopped, 0 zombie
%cpu(s): 1.5 us, 0.3 sy, 0.0 ni, 98.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 192616.7 total, 168474.6 free, 14035.3 used, 10106.8 buff/cache
MiB Swap: 8192.0 total, 8162.4 free, 29.6 used, 177141.5 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 7662 lijianc+  20   0 5877700 2.0g 174904 S  79.1  1.1 561:43.33 prometheus
 2491 root      20   0 1774400 865876 100608 S  30.1  0.4 554:43.00 k3s-server
 8509 root      20   0 11.9g  1.1g 110944 S  19.0  0.6 468:09.75 exe
1636885 nobody   20   0 729304 16880 4900 S  11.8  0.0 0:12.79 node_exporter
 6582 1000l     20   0 1935976 690796 34844 S  8.2  0.4 773:01.37 adapter
 8081 root      20   0 1106948 339532 79028 S  2.9  0.2 108:08.39 rancher
 9802 root      20   0 753504 41092 28184 S  2.9  0.0 9:12.29 coredns
 7073 lijianc+  20   0 716704 13788 6888 S  2.0  0.0 4:03.81 pushprox-proxy
1646738 stu_120+  20   0 12008 5456 2952 R  2.0  0.0 0:01.78 top
 3283 root      20   0 929036 155440 83856 S  1.3  0.1 143:05.67 containerd
 1392 root      19  -1 397112 153352 149736 S  0.7  0.1 3:07.35 systemd-journal
 8795 root      20   0 852056 129496 77432 S  0.7  0.1 15:41.55 containerd
12555 root      20   0 1031472 194512 69524 S  0.7  0.1 16:51.62 agent
 1 root      20   0 174040 16644 7244 S  0.3  0.0 5:53.30 systemd
 2 root      20   0 0 0 0 S  0.3  0.0 0:03.40 kthreadd
 11 root      20   0 0 0 0 I  0.3  0.0 5:27.23 rcu_sched
 1444 root      20   0 24124 7280 3340 S  0.3  0.0 5:45.54 systemd-udevd
 2536 root      20   0 8163052 82316 45040 S  0.3  0.0 6:34.02 dockerd
 7081 nobody   20   0 755756 52436 33368 S  0.3  0.0 7:03.63 metrics-server
10020 root      20   0 713888 10736 7368 S  0.3  0.0 1:09.58 containerd-shim
10526 root      20   0 713888 10112 7556 S  0.3  0.0 1:14.34 containerd-shim
10527 root      20   0 713888 10788 7876 S  0.3  0.0 1:11.93 containerd-shim
10687 root      20   0 919424 82480 43512 S  0.3  0.0 11:24.68 fleetagent
503070 stu_120+  20   0 6481628 393772 120212 S  0.3  0.2 2:10.12 gnome-shell
505372 stu_120+  20   0 2519096 188496 81028 S  0.3  0.1 3:00.97 Isolated Web Co
1382475 stu_120+  20   0 6485036 397024 123612 S  0.3  0.2 0:58.94 gnome-shell
1607366 root      20   0 0 0 0 I  0.3  0.0 0:00.12 kworker/40:2-events
1639564 root      20   0 0 0 0 I  0.3  0.0 0:00.11 kworker/23:1-events
1642541 root      20   0 0 0 0 I  0.3  0.0 0:00.04 kworker/93:1-events
1642569 root      20   0 0 0 0 I  0.3  0.0 0:00.06 kworker/73:1-events
1642683 root      20   0 0 0 0 I  0.3  0.0 0:00.06 kworker/74:0-events
1642801 root      20   0 0 0 0 I  0.3  0.0 0:00.07 kworker/64:2-egroup_pidlist_destroy
1642809 root      20   0 0 0 0 I  0.3  0.0 0:00.01 kworker/u194:2-events_power_efficient
1643803 root      20   0 0 0 0 I  0.3  0.0 0:00.06 kworker/4:1-hclge
1647731 root      20   0 0 0 0 I  0.3  0.0 0:00.01 kworker/0:1-hclge
1647934 root      20   0 24124 5412 1460 S  0.3  0.0 0:00.01 systemd-udevd
1647935 root      20   0 24124 5412 1460 S  0.3  0.0 0:00.01 systemd-udevd
 3 root      0 -20 0 0 0 I  0.0  0.0 0:00.00 rcu_gp
 4 root      0 -20 0 0 0 I  0.0  0.0 0:00.00 rcu_par_gp

```

4.3 磁盘任务管理与资源监视 (3分)

1. /dev/sda 设备的大小_20GiB_, 类型_devtmpfs__
2. Units_sectors of 1 * 512 = 512 bytes __Sector Size__ 512 bytes / 512 bytes

4.4 Linux 下网络系统信息 (2分)

写出机器正联网用的网卡 IPv4 地址: __192.168.11.210__

mac 地址: __6c:eb:b6:15:93:2b__

图 4-2 Linux 下网络系统信息

第 5 章 Linux 下的 showbyte 程序

(10 分)

5.1 源程序提交 (8 分)

showbyte.c 与实验报告放在一个压缩包里



showbyte.c

5.2 运行结果比较 (2 分)

运行 `od -Ax -tcx1 hello.c` 以及 `showbyte.c`, 结果截图。

```
wbx@Surface-Go-3:~/Desktop/LAB1$ od -Ax -tcx1 hello.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 > \r \n \r \n i n t m a i n ( )
      68 3e 0d 0a 0d 0a 69 6e 74 20 6d 61 69 6e 28 29
000020  { \r \n \t p r i n t f ( " H e l l
      7b 0d 0a 09 70 72 69 6e 74 66 28 22 48 65 6c 6c
000030  o 1 2 0 L 0 2 2 1 1 5 - 315 365 261
      6f 20 31 32 30 4c 30 32 32 31 31 35 2d cd f5 b1
000040  376 320 371 " ) ; \r \n \t r e t u r n
      fe d0 f9 22 29 3b 0d 0a 09 72 65 74 75 72 6e 20
000050  0 ; \r \n } \r \n
      30 3b 0d 0a 7d 20 0d 0a
000058
```

图 5-1 OD 的输出结果

```
wbx@Surface-Go-3:~/Desktop/LAB1$ ./showbyte.out
. . . # i n c l u d e < s t d
ef bb bf 23 69 6e 63 6c 75 64 65 20 3c 73 74 64
i o . h > \r \n \r \n i n t m a i
69 6f 2e 68 3e 0d 0a 0d 0a 69 6e 74 20 6d 61 69
n ( ) { \r \n \t p r i n t f ( " H
6e 28 29 7b 0d 0a 09 70 72 69 6e 74 66 28 22 48
e l l o 1 2 0 L 0 2 2 1 1 5 -
65 6c 6c 6f 20 31 32 30 4c 30 32 32 31 31 35 2d
. . . . . . . . . " ) ; \r \n \t r
e7 8e 8b e7 82 b3 e8 bd a9 22 29 3b 0d 0a 09 72
e t u r n 0 ; \r \n } \r \n
65 74 75 72 6e 20 30 3b 0d 0a 7d 0d 0a
```

图 5-2 showbyte 的输出结果

第 6 章 程序的生成 Cpp、Gcc、As、ld

6.1 请提交每步生成的文件 (10 分)

hello.i hello.s hello.o hello.out (附上 hello.c)



第 7 章 计算机数据类型的本质

7.1 运行 sizeof.c 填表 (5 分)

	Win/VS/x86	Win/VS/x64	Linux/M32	Linux/M64
char	1	1	1	1
short	2	2	2	2
int	4	4	4	4
long	4	4	4	8
long long	8	8	8	8
float	4	4	4	4
double	8	8	8	8
long double	12	16	12	16
指针	4	8	4	8

7.2 请提交源程序文件 sizeof.c (5 分)

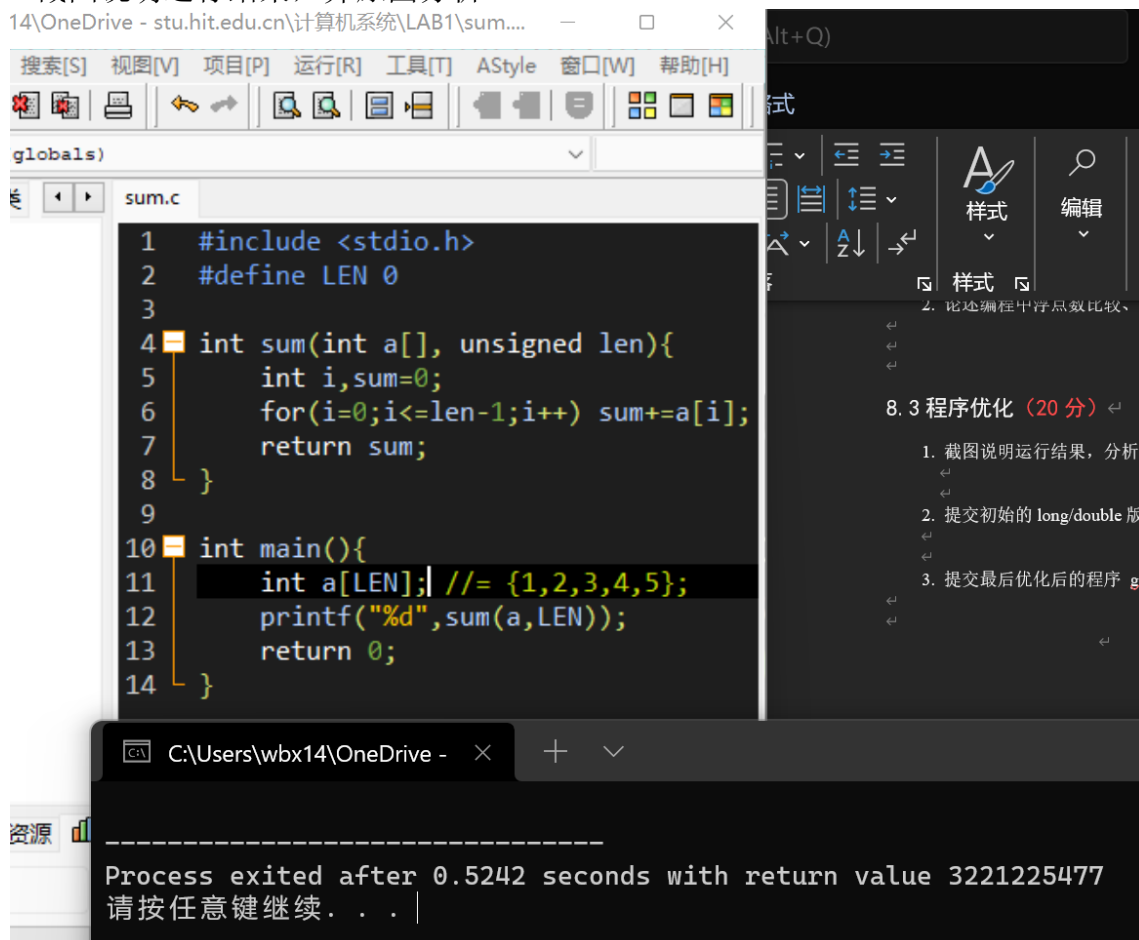
由于 sizeof 在 32 位、64 位系统返回值类型不同（分别为 unsigned int、long unsigned int），因此分为了 2 个文件。



第 8 章 程序运行分析

8.1 sum 的分析 (10 分)

1. 截图说明运行结果，并原因分析。



结果：程序异常退出。

分析原因：len-1 = UMAX，i 需要循环 UMAX 次，此时必然超过数组长度。

2. 论述改进方法

将函数参数 unsigned 改为 int 即可。

8.2 float 的分析 (10 分)

1. 运行结果截图，分析产生原因。

```
rs\wbx14\OneDrive - stu.hit.edu.cn\计算机系统\LAB1\float...
编辑[E] 搜索[S] 视图[V] 项目[P] 运行[R] 工具[T] AStyle 窗口[W] 帮助[H]
(globals)
查看类 float.c
1 include <stdio.h>
2
3 int main(){
4     float f;
5
6     for(;;){
7         printf("Input a float:");
8         scanf("%f",&f);
9         printf("This float is %f\n",f);
10        public int __cdecl printf (const char *
11    }
12
13    return 0;

C:\Users\wbx14\OneDrive -
Input a float:61.419997
This float is 61.419998
Input a float:61.419998
This float is 61.419998
Input a float:61.419999
This float is 61.419998
Input a float:61.420000
This float is 61.419998
Input a float:61.420001
This float is 61.420002
Input a float:0
This float is 0.000000

中止
compiler paths
列: 35
```

分析原因：浮点数有精度限制，也并不是连续的。输入一个实数，保存的不一定是该数，而是浮点数所能表示的最近的那个数。

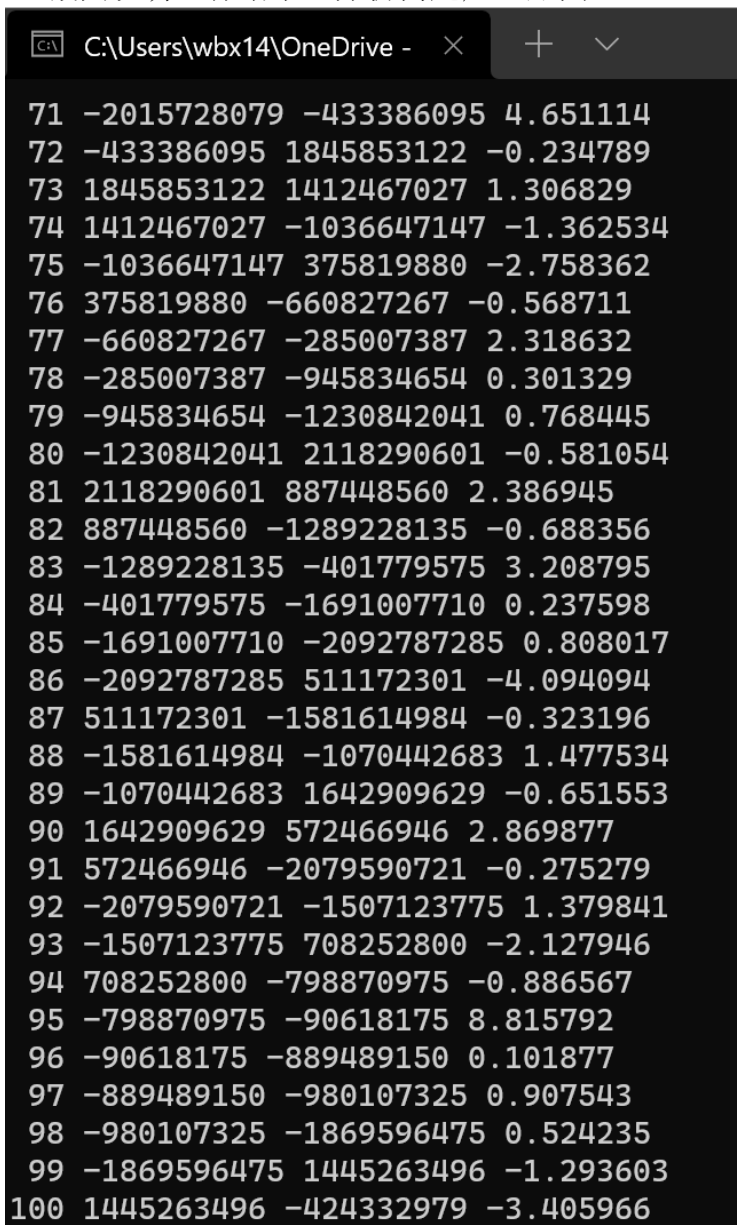
2. 论述编程中浮点数比较、汇总统计等应如何正确编程。

比较时，做差，当差小于一个很小的数时，默认他们相等。

汇总统计时应该遵循浮点数的有效位数和精度限制。

8.3 程序优化 (20 分)

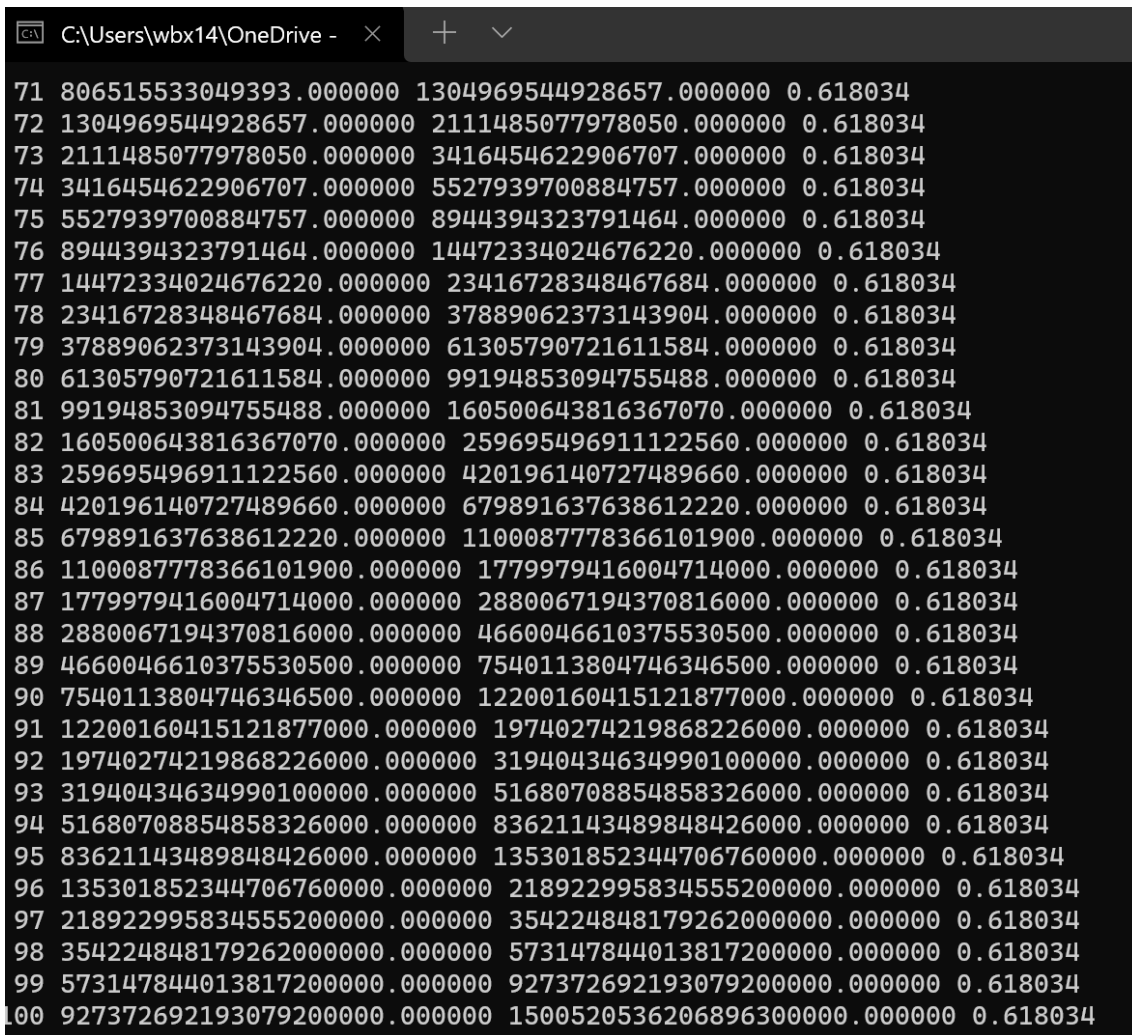
1. 截图说明运行结果，分析问题产生原因。



```
C:\Users\wbx14\OneDrive - ...  
71 -2015728079 -433386095 4.651114  
72 -433386095 1845853122 -0.234789  
73 1845853122 1412467027 1.306829  
74 1412467027 -1036647147 -1.362534  
75 -1036647147 375819880 -2.758362  
76 375819880 -660827267 -0.568711  
77 -660827267 -285007387 2.318632  
78 -285007387 -945834654 0.301329  
79 -945834654 -1230842041 0.768445  
80 -1230842041 2118290601 -0.581054  
81 2118290601 887448560 2.386945  
82 887448560 -1289228135 -0.688356  
83 -1289228135 -401779575 3.208795  
84 -401779575 -1691007710 0.237598  
85 -1691007710 -2092787285 0.808017  
86 -2092787285 511172301 -4.094094  
87 511172301 -1581614984 -0.323196  
88 -1581614984 -1070442683 1.477534  
89 -1070442683 1642909629 -0.651553  
90 1642909629 572466946 2.869877  
91 572466946 -2079590721 -0.275279  
92 -2079590721 -1507123775 1.379841  
93 -1507123775 708252800 -2.127946  
94 708252800 -798870975 -0.886567  
95 -798870975 -90618175 8.815792  
96 -90618175 -889489150 0.101877  
97 -889489150 -980107325 0.907543  
98 -980107325 -1869596475 0.524235  
99 -1869596475 1445263496 -1.293603  
100 1445263496 -424332979 -3.405966
```

分析原因：发生了整数溢出。

修改为 double 后的运行结果：



```
C:\Users\wbx14\OneDrive - ...  
71 806515533049393.000000 1304969544928657.000000 0.618034  
72 1304969544928657.000000 2111485077978050.000000 0.618034  
73 2111485077978050.000000 3416454622906707.000000 0.618034  
74 3416454622906707.000000 5527939700884757.000000 0.618034  
75 5527939700884757.000000 8944394323791464.000000 0.618034  
76 8944394323791464.000000 14472334024676220.000000 0.618034  
77 14472334024676220.000000 23416728348467684.000000 0.618034  
78 23416728348467684.000000 37889062373143904.000000 0.618034  
79 37889062373143904.000000 61305790721611584.000000 0.618034  
80 61305790721611584.000000 99194853094755488.000000 0.618034  
81 99194853094755488.000000 160500643816367070.000000 0.618034  
82 160500643816367070.000000 259695496911122560.000000 0.618034  
83 259695496911122560.000000 420196140727489660.000000 0.618034  
84 420196140727489660.000000 679891637638612220.000000 0.618034  
85 679891637638612220.000000 1100087778366101900.000000 0.618034  
86 1100087778366101900.000000 1779979416004714000.000000 0.618034  
87 1779979416004714000.000000 2880067194370816000.000000 0.618034  
88 2880067194370816000.000000 4660046610375530500.000000 0.618034  
89 4660046610375530500.000000 7540113804746346500.000000 0.618034  
90 7540113804746346500.000000 12200160415121877000.000000 0.618034  
91 12200160415121877000.000000 19740274219868226000.000000 0.618034  
92 19740274219868226000.000000 31940434634990100000.000000 0.618034  
93 31940434634990100000.000000 51680708854858326000.000000 0.618034  
94 51680708854858326000.000000 83621143489848426000.000000 0.618034  
95 83621143489848426000.000000 135301852344706760000.000000 0.618034  
96 135301852344706760000.000000 218922995834555200000.000000 0.618034  
97 218922995834555200000.000000 354224848179262000000.000000 0.618034  
98 354224848179262000000.000000 573147844013817200000.000000 0.618034  
99 573147844013817200000.000000 927372692193079200000.000000 0.618034  
100 927372692193079200000.000000 1500520536206896300000.000000 0.618034
```

2. 提交初始的 long/double 版本的 g1.c 与 g2.c。

3. 提交最后优化后的程序 g.c

第 9 章 总结

9.1 请总结本次实验的收获

学会了基本的一些 Linux 指令。
学会使用 GCC 生成程序及知道了其过程。
了解了一些计算机系统的知识。

9.2 请给出对本次实验内容的建议

1、建议对 sizeof.c 分为 32/64 位分别编写，因为在 Linux x64 环境下通过 GCC 编译，sizeof() 的返回值为 long unsigned，而 32 位则是 unsigned，若使用同一个源代码，在编译时会出错（报警）。

错误信息参考如下：

```
sizeof32.c:12:20: warning: format '%u' expects argument of type 'unsigned int', but argument 2 has type 'long unsigned int' [-Wformat=]
   12 | printf("ldouble\t%u\n",sizeof(long double));
      |                    ^~
      |                    |
      |                    long unsigned int
      |                    unsigned int
      |                    %lu
sizeof32.c:13:20: warning: format '%u' expects argument of type 'unsigned int', but argument 2 has type 'long unsigned int' [-Wformat=]
   13 | printf("pointer\t%u\n",sizeof(void*));
      |                    ^~
      |                    |
      |                    long unsigned int
      |                    unsigned int
      |                    %lu
```

2、PPT 内容和泰山服务器不匹配，需要更新。在泰山服务器中，学生账户没有 sudo 权限，需要更新 PPT 内容（包含查看隐藏磁盘分区等命令，没有 sudo 权限则无法使用）。

注：本章为酌情加分项。

参考文献

为完成本次实验你翻阅的书籍与网站等

- [1] [Ubuntu 编译出现 fatal error: bits/libc-header-start.h: No such file or directory 张志翔 的博客 -CSDN 博客](https://blog.csdn.net/qg_19734597/article/details/102943559)
https://blog.csdn.net/qg_19734597/article/details/102943559
- [2] <https://blog.csdn.net/ycacmzc/article/details/114951047>
- [3] [Linux 查看系统块大小 - 潇湘隐者 - 博客园 \(cnblogs.com\)](http://cnblogs.com)