1. 参考网页：（理解英文字符与ASCII码）

<https://www.runoob.com/cprogramming/c-function-printf.html>

<https://www.runoob.com/cprogramming/c-examples-ascii-value-character.html>

编写C程序asciilist.c, 按以下格式在屏幕上显示从0到127的ascii的编码及字符。

ASCII码十进制，十六进制，字符

0 0

1 0x01

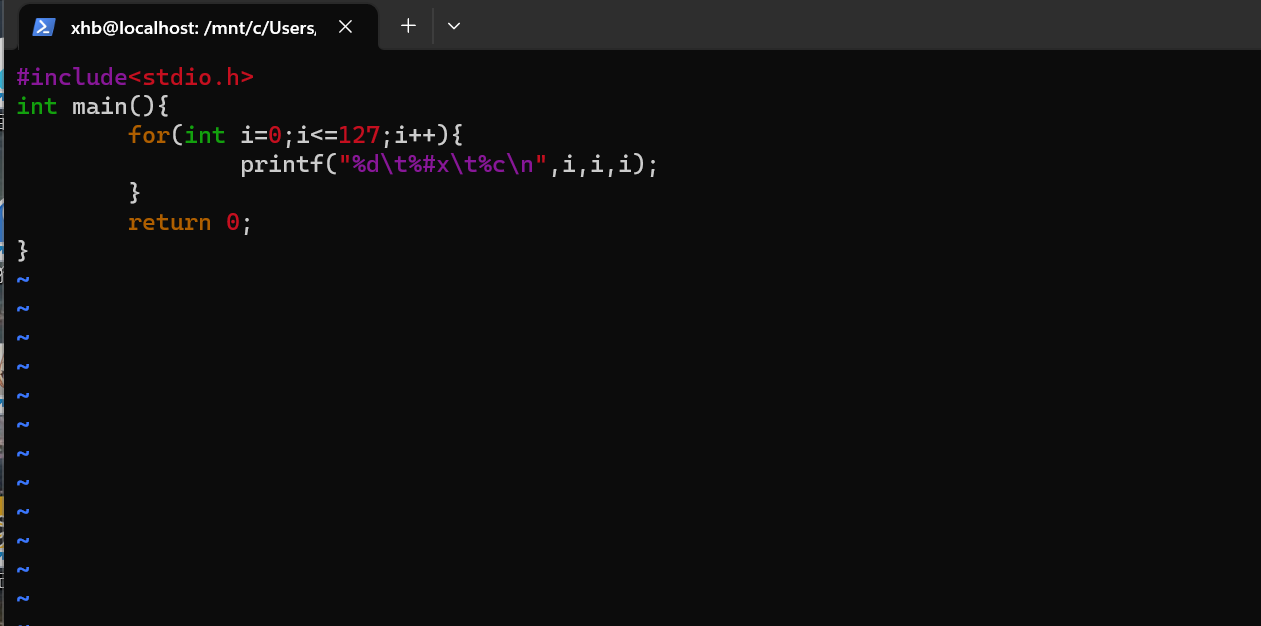
。

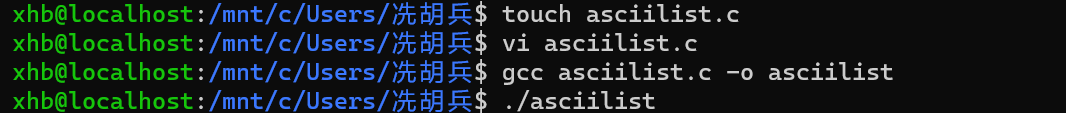
。

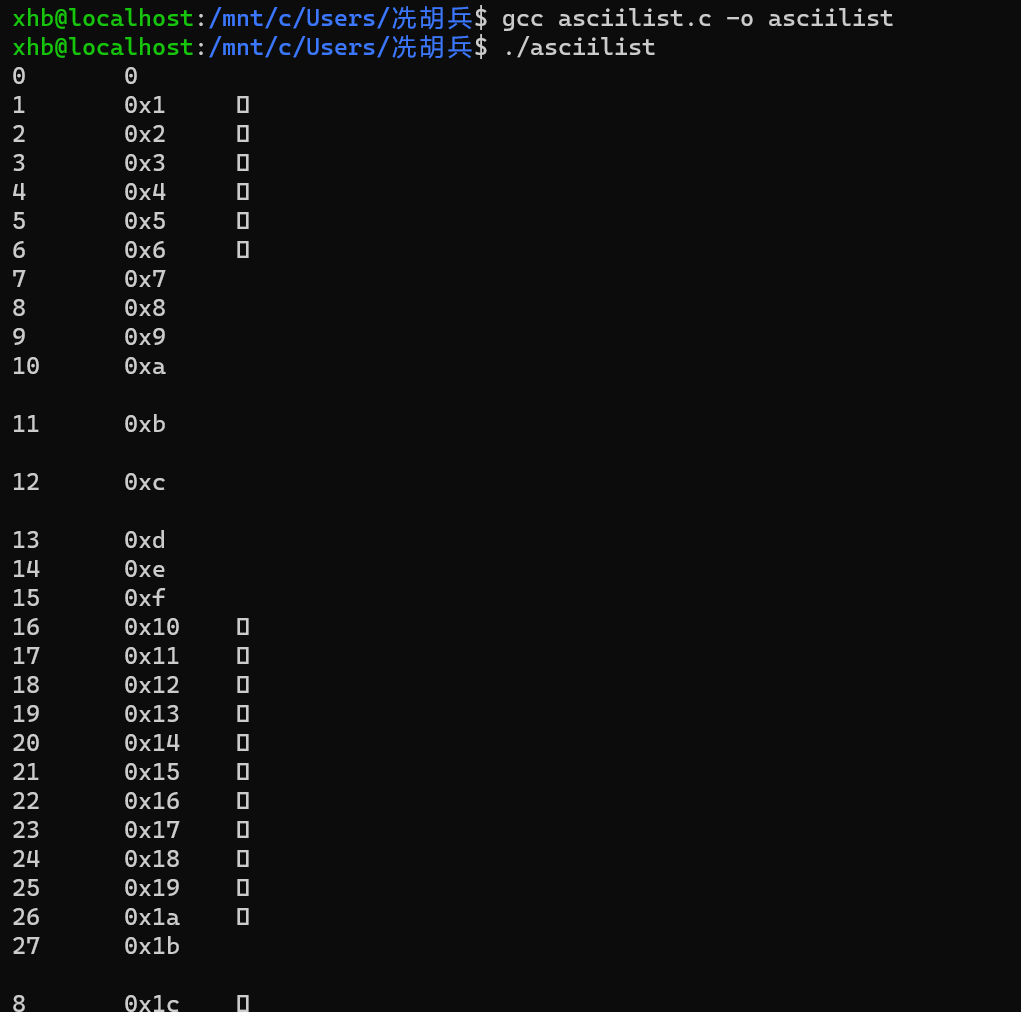
。

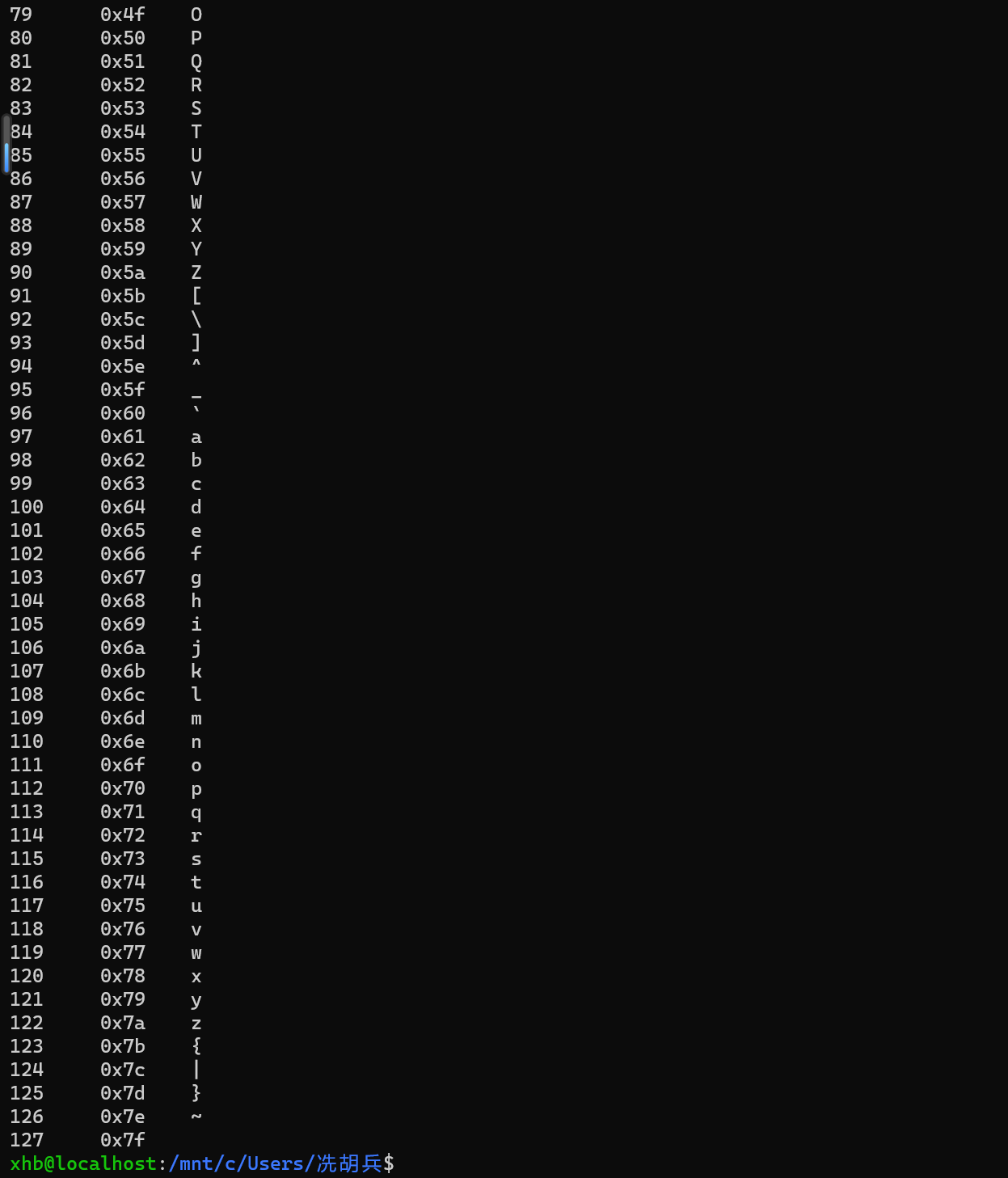
126 0x7e ~

127 0x7f









2. 以下为在屏幕显示Hello World! 的hello1.c程序。

（理解程序，进程，指令，数据，设备文件，RAM中的信息，保存到硬盘中）

#include <stdio.h>

#include <unistd.h>

#include  <string.h>

void main()

{

char \*data="Hello World !\n";

printf("pid=%d, main address:%p,  data address:%p data:%s\n",getpid(),main,data,data);

write(2, data, strlen(data));

}

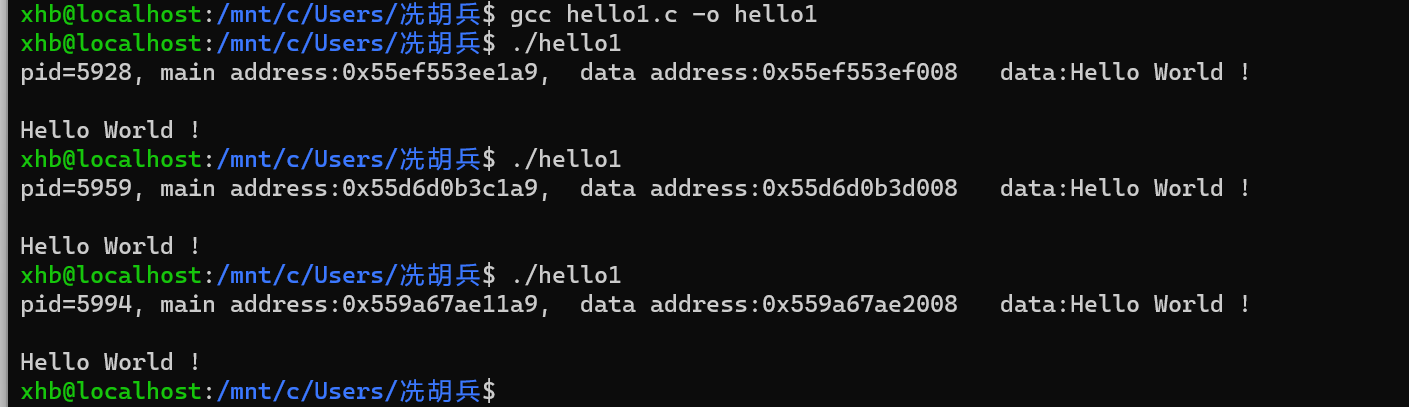
1）、将源程序编译生成可运行程序 hello1 , 运行三次，分析运行结果的差异。

gcc hello1.c -o hello1

./hello1

./hello1

./hello1



分析：  
运行三次后，发现printf函数输出程序的进程ID（getpid()）、mian函数的地址和data指针的地址都出现了差异。

进程ID出现差异：进程ID的分配是由操作系统内核进行的，可能是因为多线程的存在，那么获取进程的时刻可能不同，导致输出的进程ID不同

Main函数的差异和data指针地址出现差异的原因可能因为操作系统分配虚拟内存的方式不同而导致。

2）、参考网页：

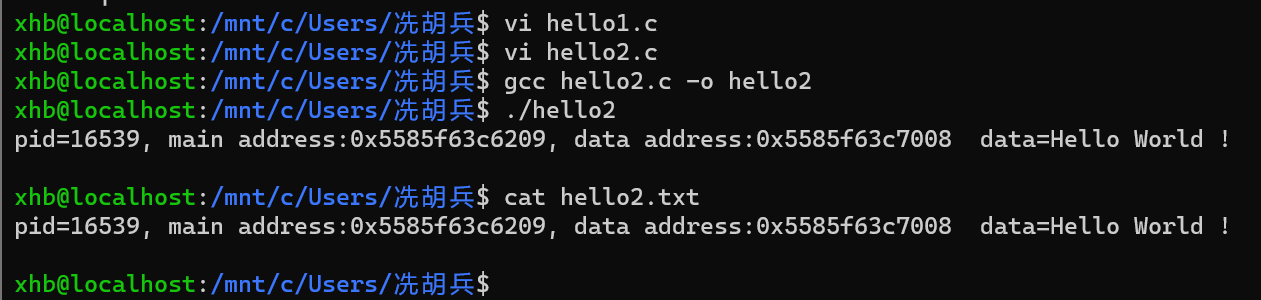
<https://linux.fasionchan.com/zh_CN/latest/system-programming/file-io/write.html>

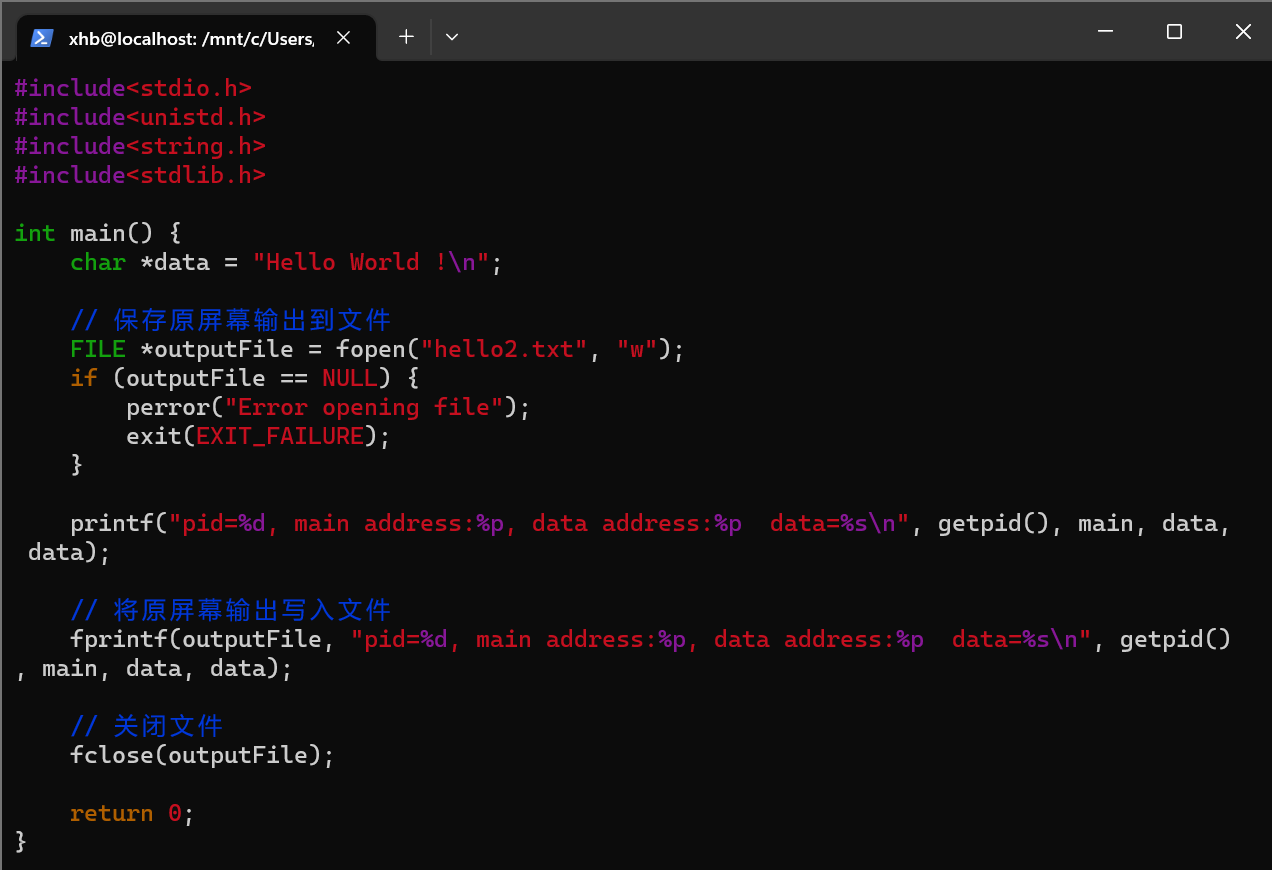
修改以上程序为hello2.c，将原屏幕输出保存到文件hello1.txt

. gcc hello2.c -o hello2 # 编译程序

./hello2 # 运行程序

cat hello2.txt # 显示 生成的 文件内容





3）、

<https://blog.51cto.com/yongxi/2365705>

参考以上网页，理解设备文件，及重定向命令。

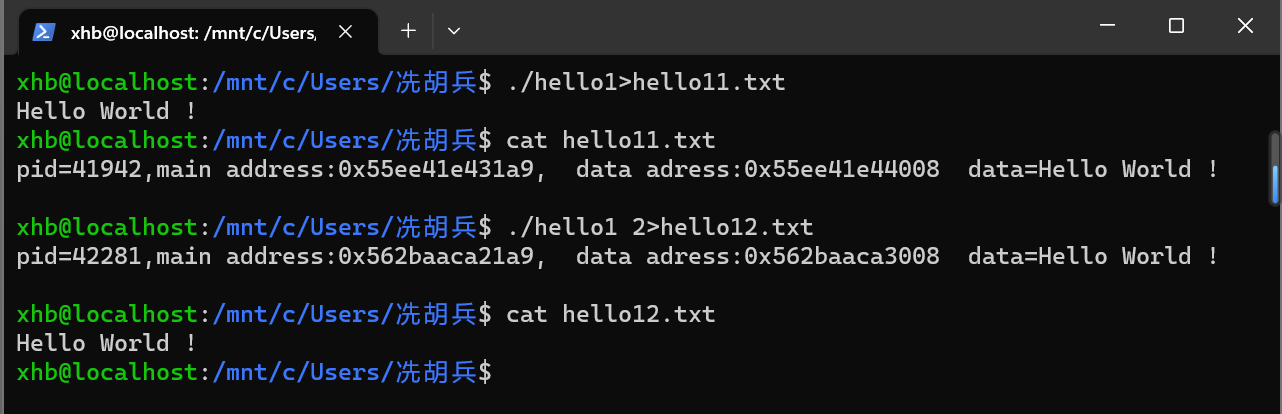
运行以下命令，解释命令的运行结果

./hello1 > hello11.txt

cat hello11.txt

./hello1 2> hello12.txt

cat hello12.txt



解释：

（1）./hello1>hello11.txt的运行结果是将hello1程序的标准输出重定向到‘’hello11.txt‘’文件中去。

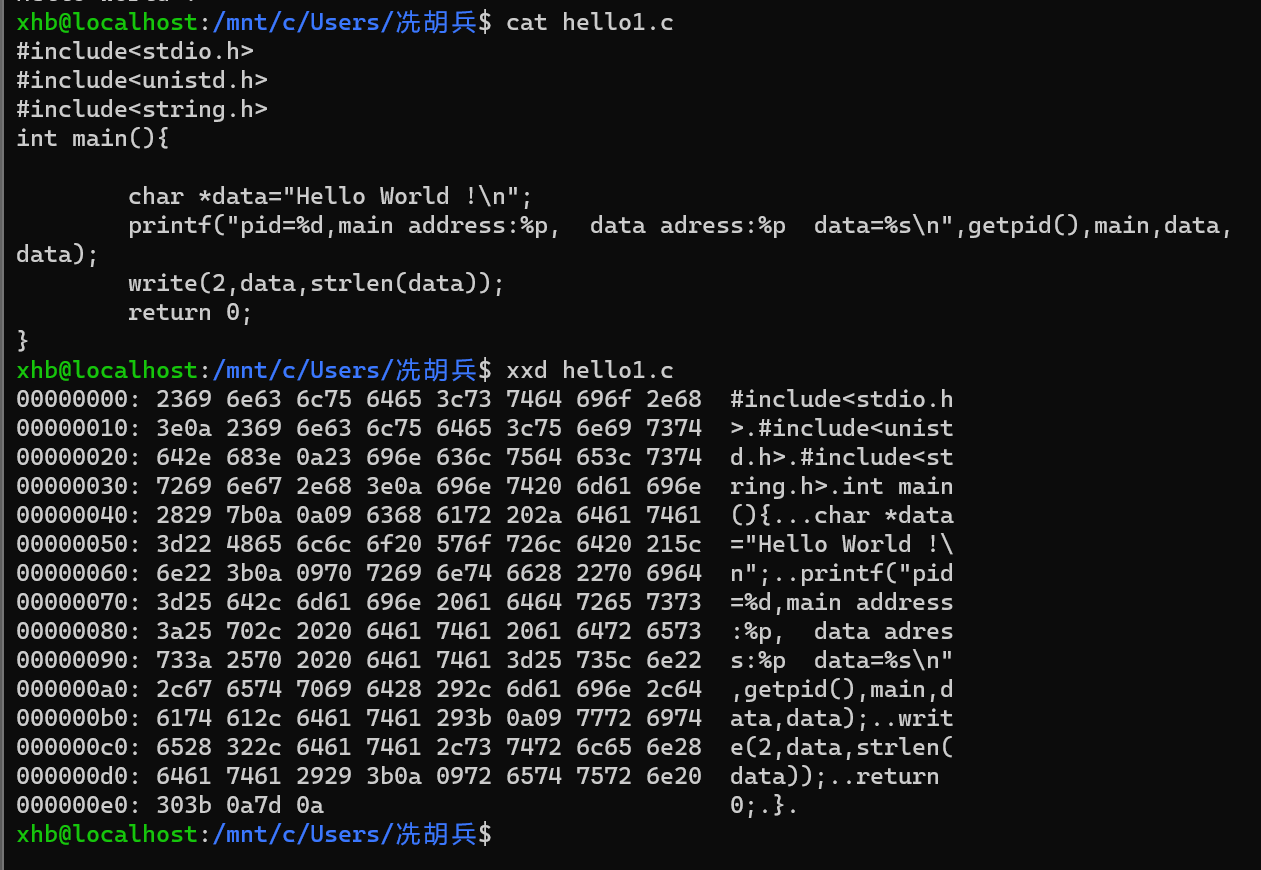
（2）cat hello11.txt是查看“hello11.txt”的文件内容

（3）./hello1 2>hello12.txt是将程序的标准错误输出重定向到“hello12.txt”中去

4）、运行以下命令，解释命令的运行结果

cat hello1.c

xxd hello1.c

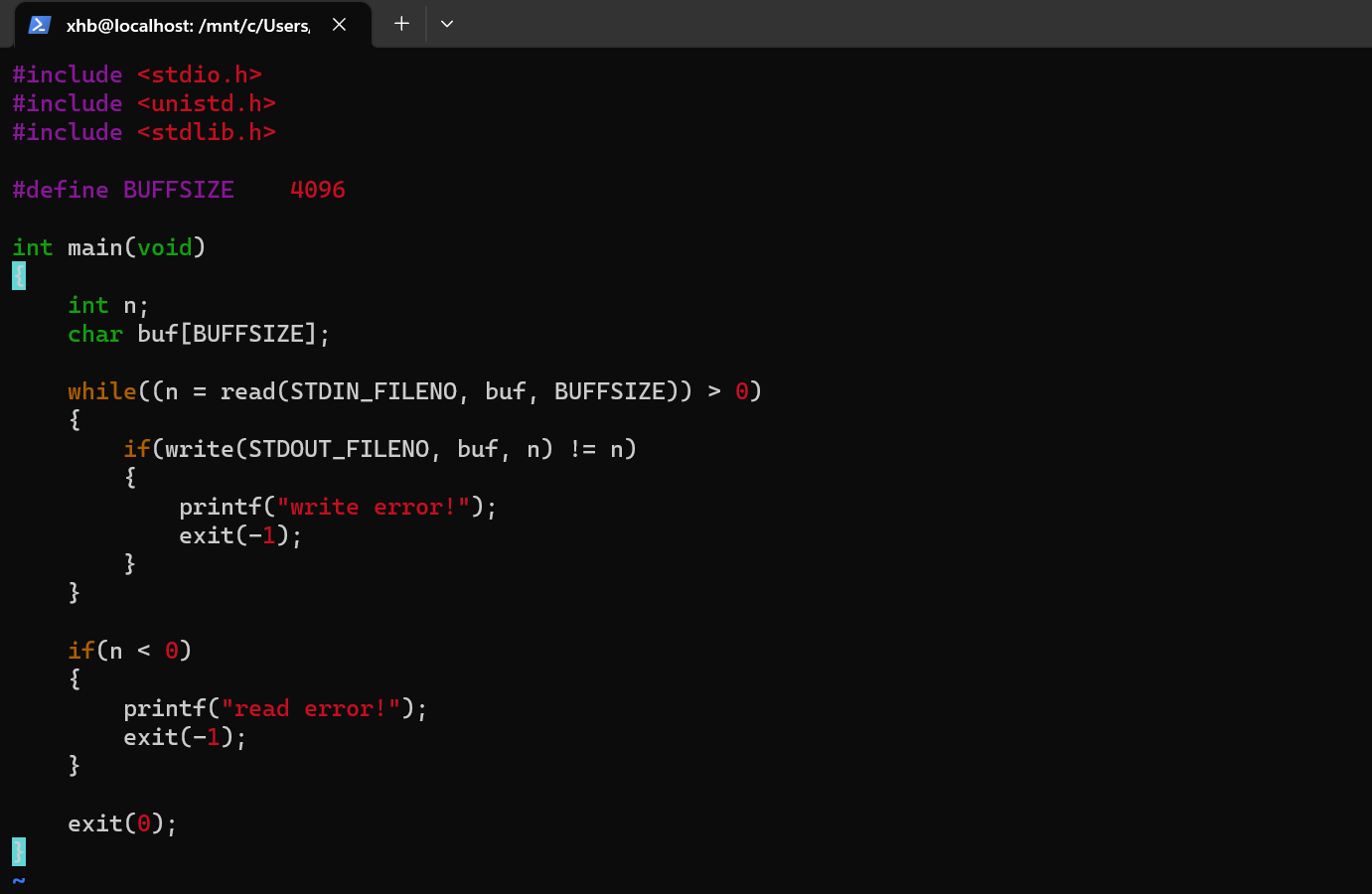


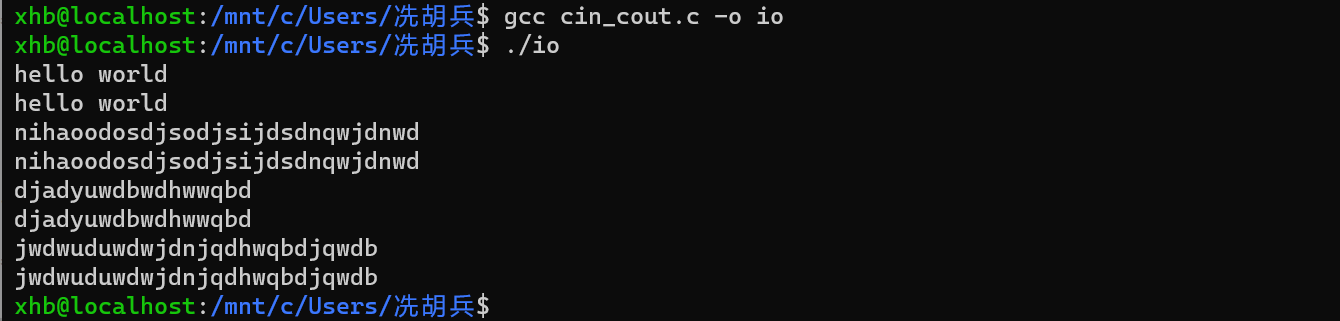
**解释：**

1. **cat 命令用于显示文件的内容。在这里，“hello1.c”是一个源代码文件，该命令将其内容打印到终端。执行命令后，将输出“hello.c”文件的文本内容，即C语言源代码。**
2. **“xxd”命令用于生成我问你间的十六进制和ASCII码表示，它将二进制文件转换成十六进制格式，并且在右侧显示ASCII字符。执行命令后，将输出“hello.c”文件的十六进制表示。每行包含十六个字节。左侧显示十六进制值，右侧是对应的ASCII字符表示。**
3. 写一个将标准输入复制到标准输出的c程序。（理解标准输入与标准输出）

<https://zhuanlan.zhihu.com/p/64609966>

<https://segmentfault.com/a/1190000018650023>





4． 运行以下命令，解释每个命令的功能，给出在你的linux环境下的运行结果 ：（shell创建文本文件？）

$ cd ~

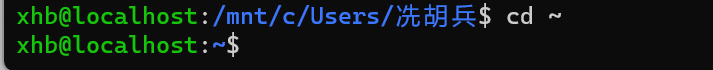
$ pwd

$ echo "hello cat " > cat.txt

$ ls -l cat.txt

$ cat cat.txt

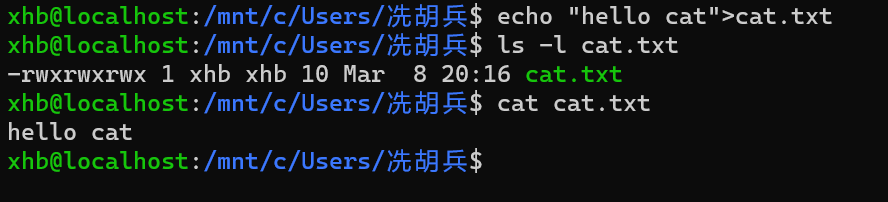
解释：  
（1）“cd”命令的功能是切换当前的工作目录，而“~”则是表示用户的家目录。执行这个命令后，当前工作目录将切换到用户的家目录



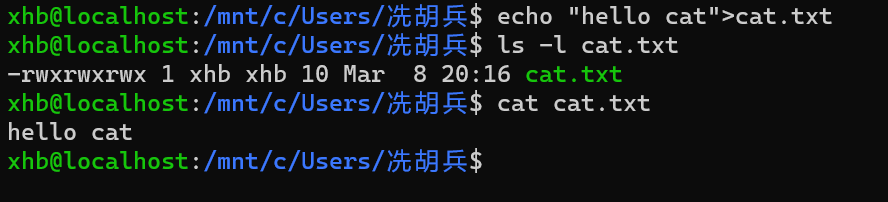
（2）“pwd”用于显示当前工作目录的路径



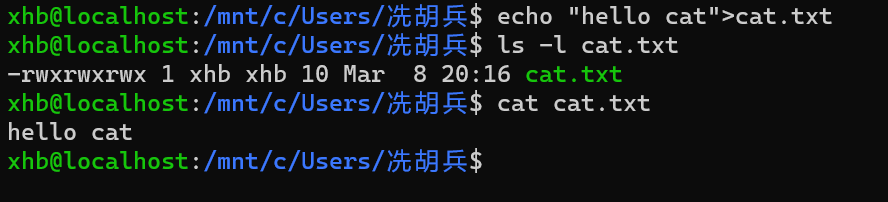
（3）“echo”命令用于向标准输出打印文本，“>”符号用于重定向，这里将“echo”命令的输出重定向到名为“cat.txt”的文件中。执行这个命令后，将在用户的家目录下创建一个名为“cat.txt”的文本文件，并将“hello cat”写入该文件。



（4）“ls”命令用于列出目录的内容，“-l”选项表示以长格式显示文件信息，包括文件权限、所有者、文件大小、创建时间等。



（5）“cat cat.txt”用于连接文件并显示文件的内容



5． 参考网页（理解命令行参数，普通文件的打开及读入）

<https://blog.csdn.net/xiaocainiaoshangxiao/article/details/13996459>

学习C语言命令行参数的用法。

编写C程序mycat.c，实现cat命令的最基本功能，显示普通文件的内容（文本或者十六进制）（用open,或者fopen）。并实现以下内容：

1）$ gcc mycat.c –o mycat

$ ./mycat   cat.txt

以文本（ascii码）显示文件cat.txt内容。 （cat.txt 为命令行参数，代表文件名）

2）$ ./mycat  -h  cat.txt

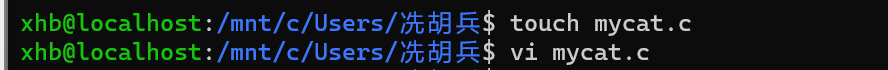
以十六进制显示文件cat.txt的内容。（命令行参数 -h 表示按十六进制显示文件内容）

要求：包括代码（含必要的注解），及在你的ubuntu上的运行结果的图片。

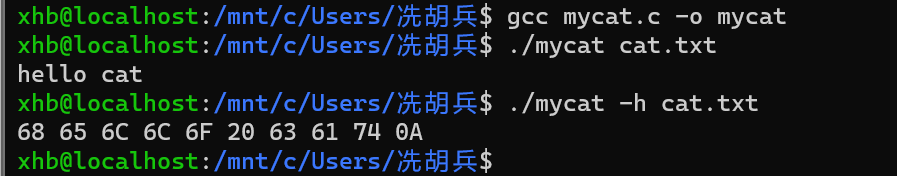
（如何用C实现打开一个文件，显示内容？）

如何打开文件及读入参考网页：

<https://book.itheima.net/course/223/1263669610003230722/1265878649361653762>







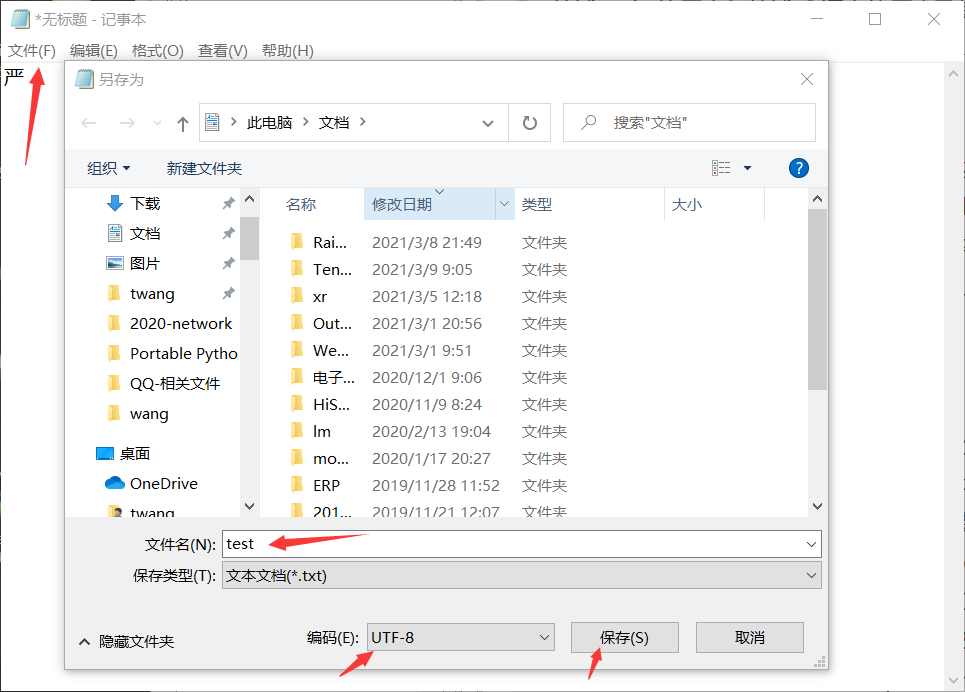
6．参考网页：

<http://www.ruanyifeng.com/blog/2007/10/ascii_unicode_and_utf-8.html>

http://www.unicode.org/charts/

理解字符编码ASCII，Unicode 和 UTF-8的关系。

打开"记事本"程序notepad.exe，新建一个文本文件（test.txt），内容就是一个”严”字，依次采用ANSI，Unicode，Unicode big endian和UTF-8编码方式保存。



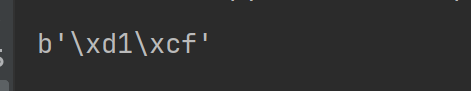
用前面安装的python，打开文本文件test.txt, 按二进制显示文件内容。解释输出结果。

f = open('test.txt', 'rb') #按二进制方式打开文件

print(f.read()) # 读出文件内容，并显示

f.close() # 关闭文件

1. ANSI



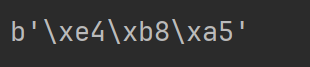
在这里，**\xd1** 和 **\xcf** 是十六进制表示的字节值。

解释：

* **\xd1** 对应 Unicode 编码的十六进制值，表示字符 "严"。
* **\xcf** 对应 Unicode 编码的十六进制值，表示字符 "汉"。

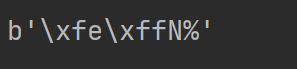
由于 ANSI 编码只能表示有限的字符集，而 "严" 和 "汉" 字符不在 ANSI 编码的范围内，因此它们被截断或映射为对应 ANSI 编码中的字符。在这里，**\xd1** 和 **\xcf** 分别对应 ANSI 编码中的一些特定字符。

1. UTF-8



**\xe4**、**\xb8** 和 **\xa5** 对应 Unicode 编码的十六进制值，表示字符 "严"

1. Unicode big endian



* **\xfe** 和 **\xff** 对应 Unicode Big Endian 编码的字节顺序标记，表示这是一个 Big Endian 编码的文件。
* **N** 和 **%** 对应 Unicode 编码的十六进制值，表示字符 "N" 和 "%"。

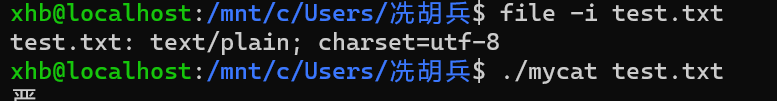
(4)unicode

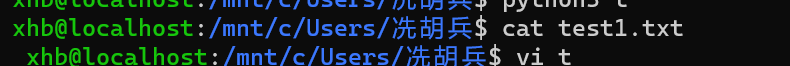


思考题：有兴趣同学，可以按字符方式显示文件内容。

再用上一题目编写的mycat , 以二进制方式打开显示文本文件test.txt，解释输出结果。

1. UTF-8



1. 

结果输出了一个空格，表示无法显示“严”,它不在ASCII编码里面。

7．运行以下命令，给出在你的linux环境下的运行结果 ：（创建目录文件？显示目录文件内容？）

$ cd ~

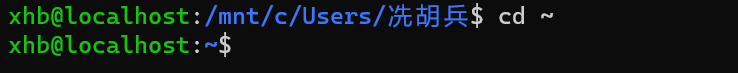
$ pwd

$ mkdir test-dir

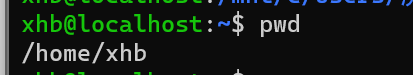
$ cp \*.\* ./test-dir

$ ls ./test-dir -l

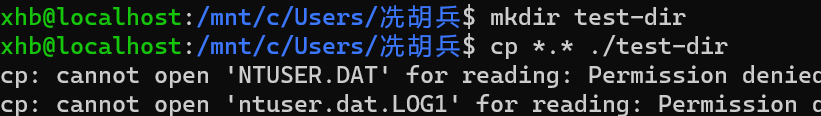
（1）



（2）



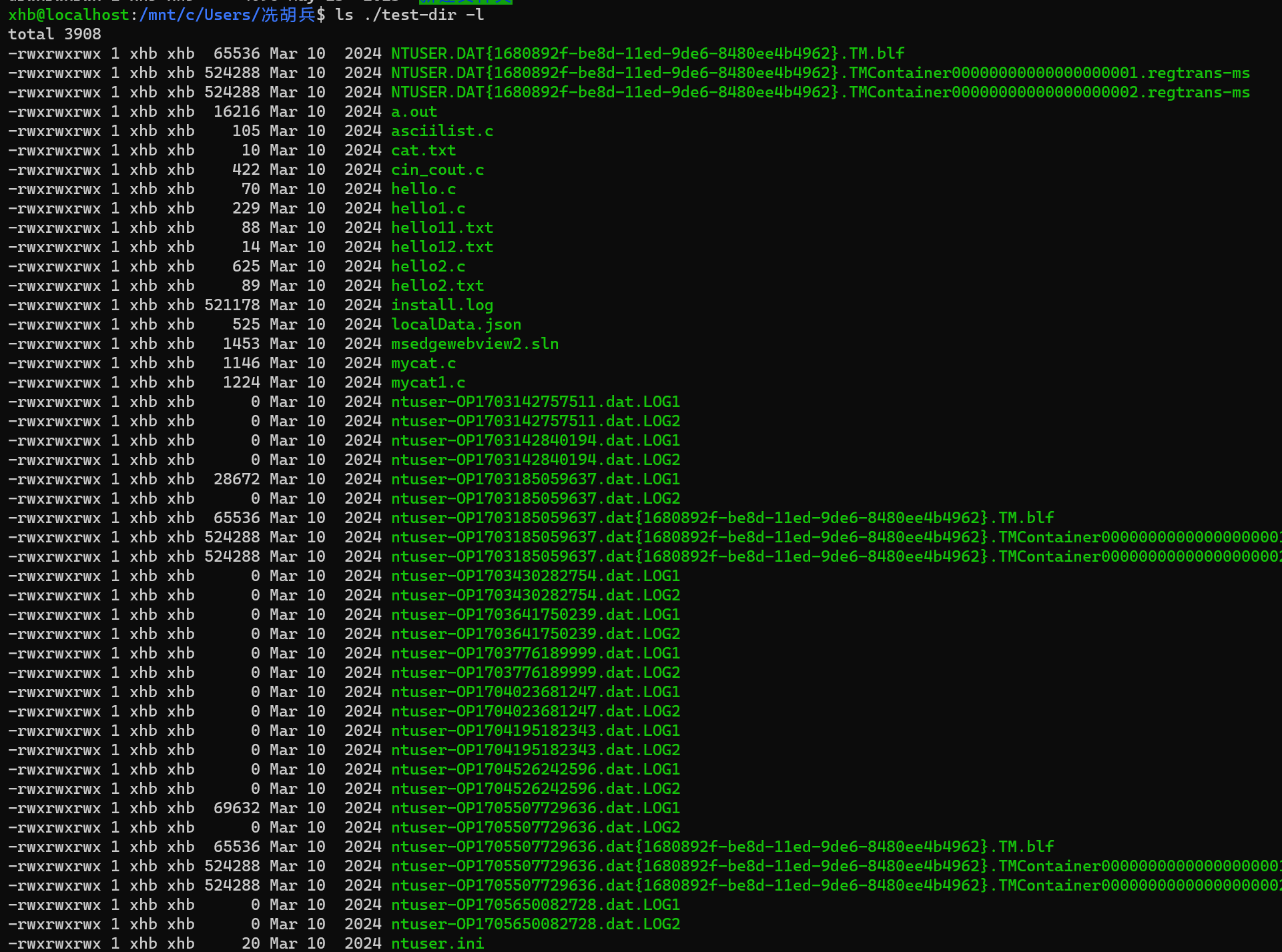
（3）



（4）



（5）



8．参考网页，

<https://blog.csdn.net/home198979/article/details/84340195>

编写C程序myls.c，实现ls命令的最基本功能，显示目录(目录文件)的内容。

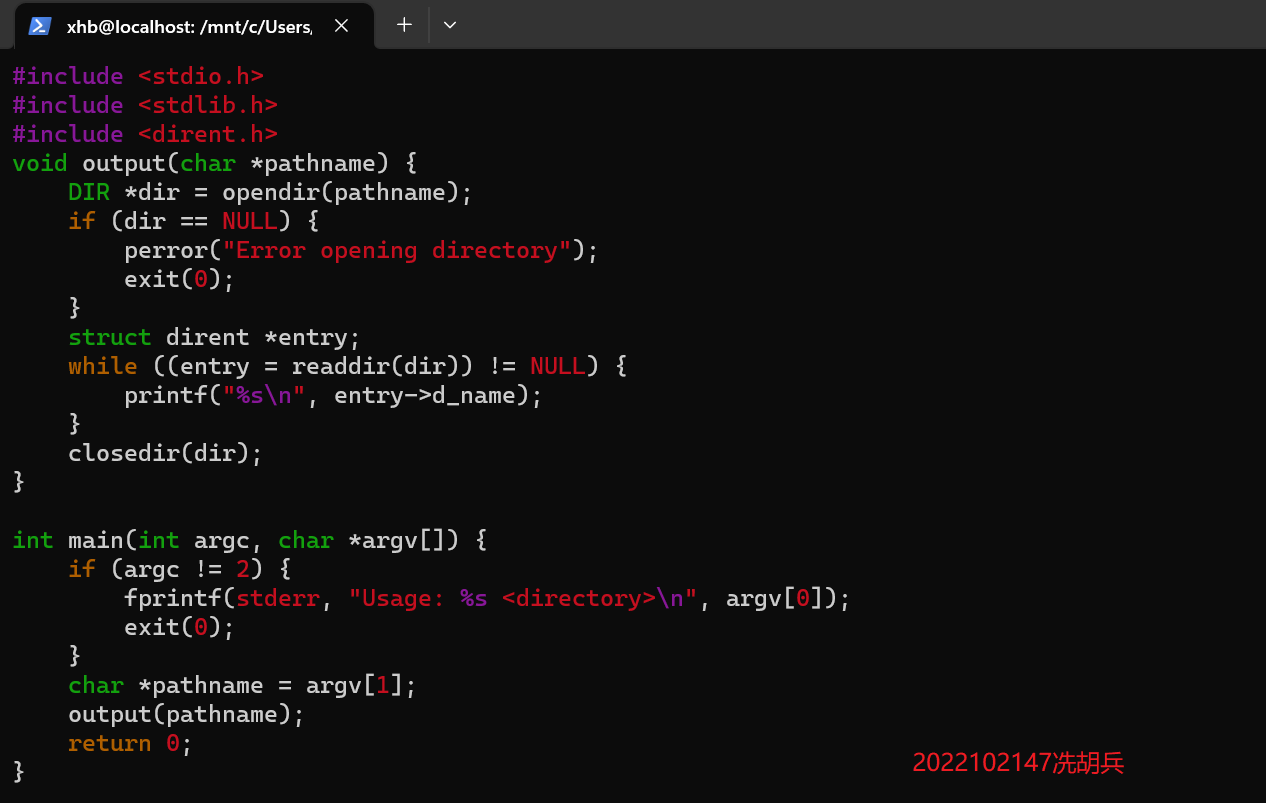
myls  .  显示当前目录下的文件。（用opendir）。

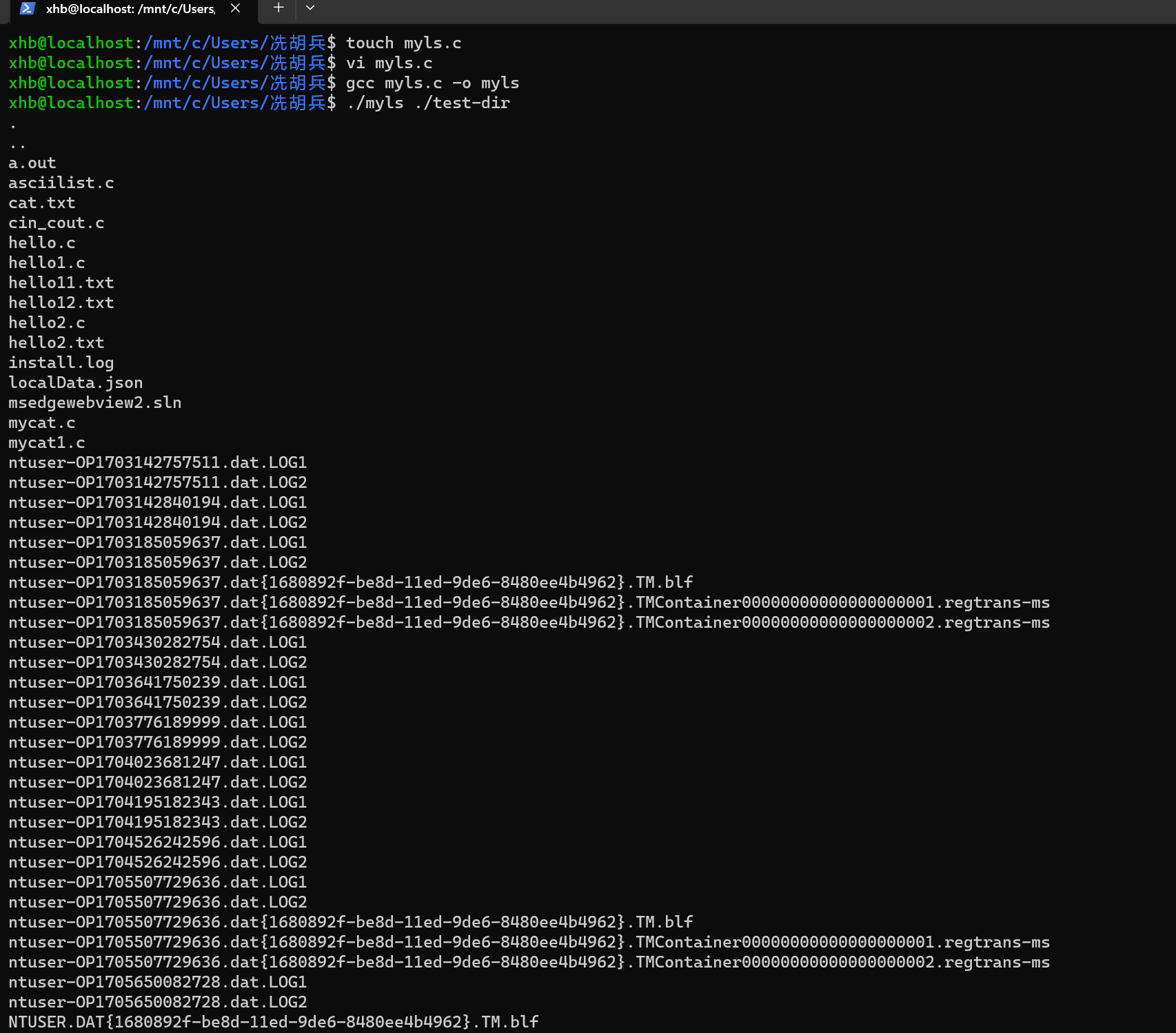
$ gcc myls.c –o myls

$ ./myls ./test-dir

包括代码（含必要的注解），及在你的ubuntu上的运行结果的图片。

（如何用C语言实现打开目录文件，显示目录文件中的内容？）





9． linux用户管理多用户的系统文件是哪个？有哪些属性？

**在Linux系统中，用户管理多用户的系统文件主要是 `/etc/passwd` 文件。`/etc/passwd` 文件包含了系统上所有用户的基本信息，每一行代表一个用户。**

**每行的格式如下：**

**username:password:UID:GID:GECOS:home\_directory:shell**

**其中：**

**username：用户登录名。**

**password：加密后的用户密码。通常是 'x'，表示密码存储在 `/etc/shadow` 文件中。**

**UID：用户标识号（User ID），是一个唯一的数字用于标识用户。**

**GID：用户所属的主要组的组标识号（Group ID）。**

**GECOS：一些注释性描述，包括用户的真实姓名等。**

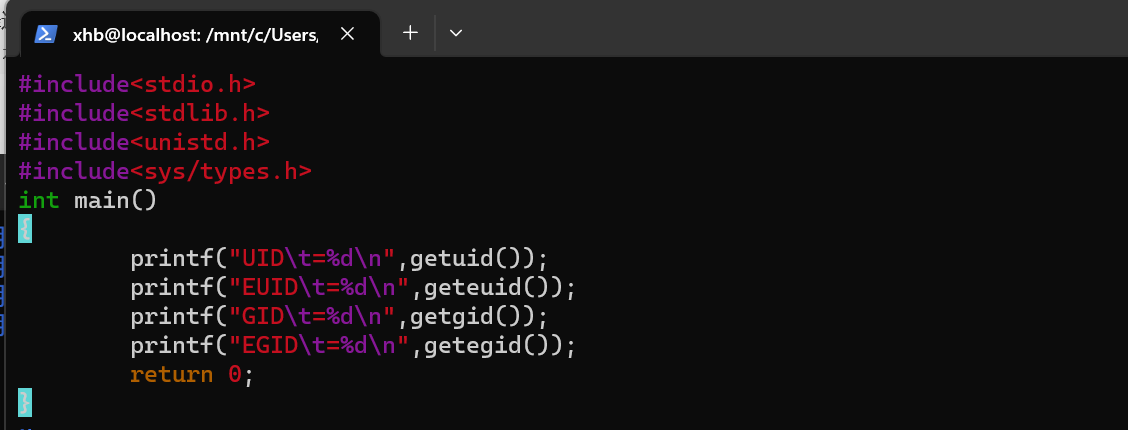
**home\_directory：用户的主目录，即用户登录后默认的工作目录。**

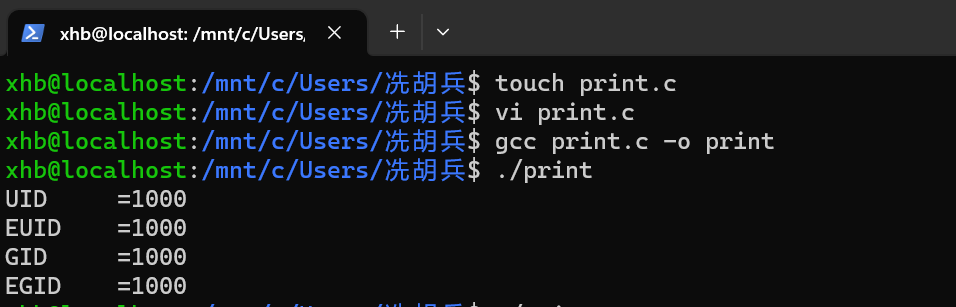
**shell：用户登录后默认使用的shell。**

10． https://www.cnblogs.com/itech/archive/2012/04/01/2429081.html。

参考以上网页内容，编写实例程序，测试并解释进程的用户ID（uid）和有效用户ID（euid），用户组ID（gid）和有效用户组ID（egid）从哪里获得？有什么用途？（请根据程序的结果来解释）

（理解UID相关的概念？）





1. 获取：

用户ID在登录系统时由系统分配，用于唯一标识一个用户。所有属于同一个用户的进程具有相同的用户ID。

有效用户ID用于权限检查，通常在执行育有特殊劝降的文件或操作时使用

用户组ID也是在登陆时由系统分配，用于将用户归类到一个或多个组中。

有效以用户组ID也用于权限检查，类似于有效用户ID

对于输出结果，发现四个输出的ID 均相同，这是系统分配的结果，在实际运用中，会出现不同的情况。

1. 用途：

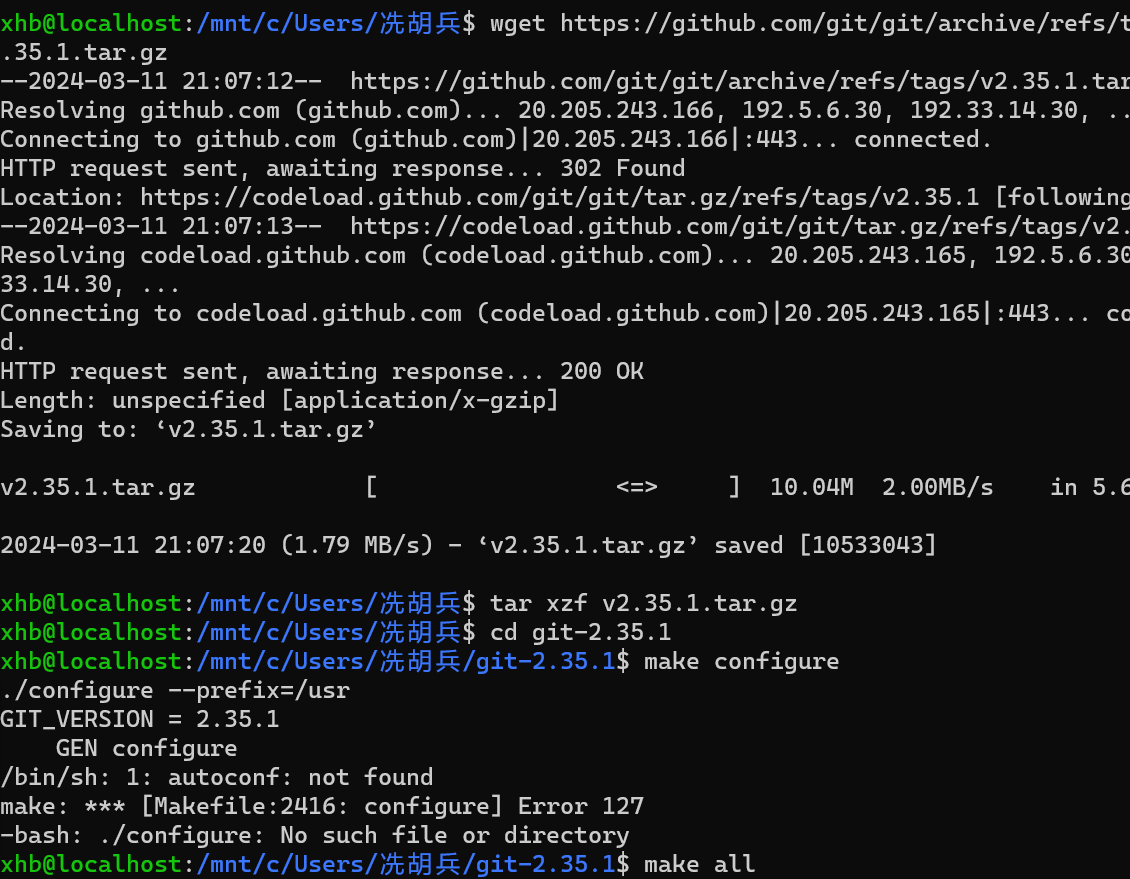
这些标识符是系统用于进行权限检查和控制访问的关键元素。它们确保了进程只能访问其具有权限的资源。

当进程以不同的用户身份运行时，有效用户ID和有效用户组ID变得特别重要。例如，一个普通用户可能需要执行一个只有管理员才有权限执行的程序，此时可以通过提升有效用户ID来获取相应的权限。

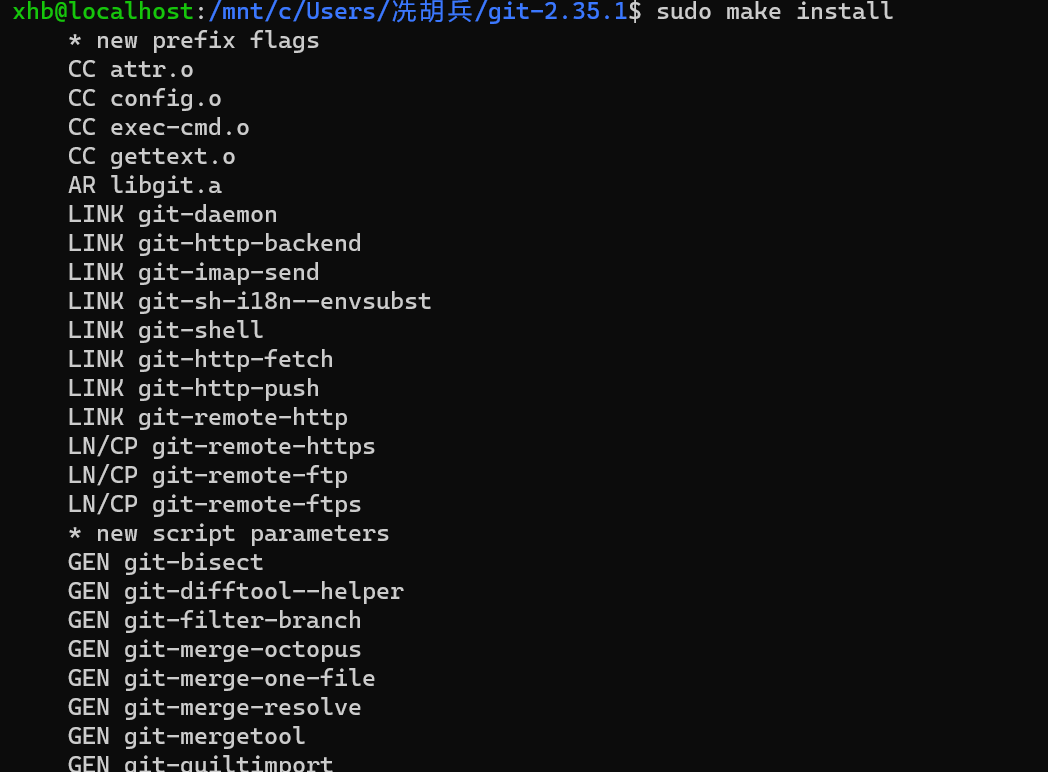
11．https://www.howtoing.com/how-to-install-git-on-ubuntu-18-04

  参考以上网站资料，下载源码版本的git应用软件，下载后编译，安装。给出你的安装过程，并解释每一个命令的含义（完成的功能）。个别网络慢的同学，可以直接用我放在QQ上的文件git.zip。（我们后面会介绍git软件）

（理解基于源码的软件安装？）







* **wget**：从网络下载文件。
* **tar**：解压缩文件。
* **cd**：更改当前工作目录。
* **make configure**：运行 make 来生成 configure 脚本。
* **./configure --prefix=/usr**：配置软件的编译选项，指定安装目录为 /usr。
* **make all**：编译源代码，生成可执行文件。
* **sudo make install**：以管理员权限安装 Git。

12．[python 读写 excel(xlsx) 文件](https://www.1024sou.com/article/39696.html)。（pythonr的仓库？从仓库中安装合适的wheel？）

默认仓库：<https://pypi.org/simple/>

1）写入xls文件代码

安装轮子（第三方库）openpyxl。

pip install openpyxl

from openpyxl import Workbook

# 创建一个Workbook对象pi

wb = Workbook()

# 获取当前活跃的sheet，默认是第一个sheet

ws = wb.active

ws['A1'] = '用户名'

ws['B1'] = '年龄 '

ws['C1'].value = '男/女'

row1 = ['小黄', "20", '女']

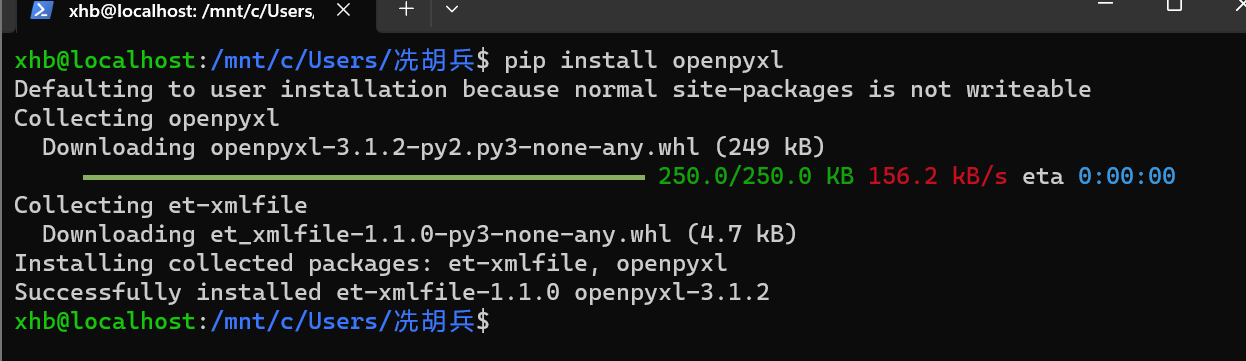
row2 = ['小绿', "13", '男']

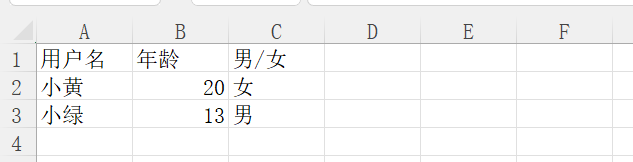
ws.append(row1)

ws.append(row2)

wb.save("a.xlsx")

## 用excel打开python当前目录下生成的文件 a.xlsx，给出其内容。





2）读取代码

from openpyxl import load\_workbook

wb = load\_workbook("a.xlsx")

# 获取所有表格(worksheet)的名字

sheets = wb.sheetnames

# 第一个表格的名称

sheet\_first = sheets[0]

# 获取特定的worksheet

ws = wb[sheet\_first]

# 获取表格所有行和列，两者都是可迭代的

rows = ws.rows

content = []

# 迭代所有的行

for row in rows:

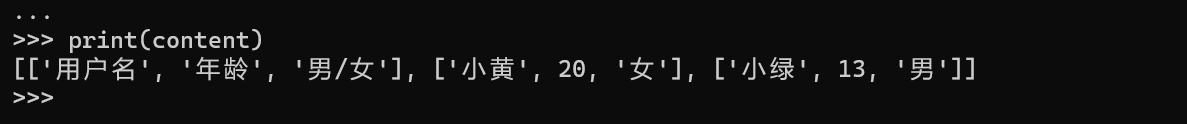
line = [col.value for col in row]

content.append(line)

print(content)

参考：<https://geek-docs.com/python/python-tutorial/python-openpyxl.html> Openpyxl 教程

pip install openpyxl安装的openpyxl包（python源码）通常被安装在[python](https://so.csdn.net/so/search?q=python&spm=1001.2101.3001.7020)安装目录下的lib\site-packages\openpyxl目录下。



13．[python](https://www.1024sou.com/article/39696.html) 控制鼠标移动与交互，屏幕快照

安装轮子（第三方库）pyautogui

pip install pyautogui

例子代码：

## 使用 pyautogui 模块相关函数，可以模拟鼠标及键盘操作， 完整说明文档见http://pyautogui.readthedocs.org/

# pip install pyautogui

# 要注意的是，模拟移动鼠标与击键可能太快，导致其他程序跟不上，并且程序可能失去控制，

# 需要掌握如何从问题中恢复，至少要能中止它。

# 防止或恢复GUI自动化问题

# 1) 使用pyautogui.PAUSE设置每个PyAutoGUI函数调用在执行动作后暂停的秒数

# 2) pyautogui自动防故障功能：将鼠标移到屏幕的左上角，来抛出failSafeException异常

import pyautogui

pyautogui.PAUSE = 1

pyautogui.FAILSAFE = True # 启用自动防故障功能

width,height = pyautogui.size() # 屏幕的宽度和高度

print(width,height)

pyautogui.position() # 鼠标当前位置

## 控制鼠标交互

# pyautogui.click() 封装了 pyautogui.mouseDown()和pyautogui.mouseUp(), 这两个函数也可以单独使用

pyautogui.click(30,30,button='left')

pyautogui.mouseDown(30,30,button='left')

# pyautogui.doubleClick() 双击左键， pyautogui.rightClick() 双击右键 pyautogui.doubleClick(30,30,button='left')

参考：<https://pyautogui.readthedocs.io/en/latest/quickstart.html> pyautogui文档

pip install pyautogui安装的pyautogui包（python源码）通常被安装在[python](https://so.csdn.net/so/search?q=python&spm=1001.2101.3001.7020)安装目录下的lib\site-packages\pyautogui目录下。

14．参考网页：

<https://blog.csdn.net/Nobitah/article/details/104764816>

参考程序 disp\_A.c, 显示字符A

#include <string.h>

#include <unistd.h>

#include <stdio.h>

char dot\_A[][8]={

"0000000",

"0001000",

"0010100",

"0100010",

"1000001",

"1000001",

"1111111",

"1000001",

"1000001"

};

void disp\_A(char dispc)

{

int i,j;

for(i=0;i<9;i++)

// printf("%s\n",dot\_A[i]);

{

for(j=0;j<7;j++)

if(dot\_A[i][j] == '1') printf("%c",dispc);

else printf(" ");

printf("\n");

}

}

int main(int argc, char \*argv[])

{

int i;

if (argc < 3)

{

printf("Usage: a.out char1 char2 \n");

return 0;

}

// if(argv[1][i] == 'A') disp\_A(argv[2][0]);

if(argv[1][i] == 65) disp\_A(argv[2][0]); // A 的ASCII码 是 65

}

将以上程序命名为 disp\_A.c,

wang@DESKTOP-PABFQ0S:~$ gcc disp\_A.c -o disp\_A #编译成程序 disp\_A

wang@DESKTOP-PABFQ0S:~$ ./disp\_A A @ # 运行程序 disp\_A， 后面是二个命令行参数

@

@ @

@ @

@ @

@ @

@@@@@@@

@ @

@ @

修改以上示例程序，增加数字字符‘0’~‘9’十个字符的显示点阵。修改完成后，运行以下程序测试修改的结果。

/disp\_A 0 @

/disp\_A 1 @

/disp\_A 2 @

/disp\_A 3 @

/disp\_A 4 @

/disp\_A 5 @

/disp\_A 6 @

/disp\_A 7 @

/disp\_A 8 @

/disp\_A 9 @

说明：提交编程作业时，需要附上源程序，并加上合适的注解。

源代码：



