

**实验报告**

**实 验（二）**

题 目 DataLab 数据表示

专 业 计算机专业

学　　 号 1190200501

班　　 级 1903002

学 生 林燕燕

指 导 教 师 郑贵滨

实 验 地 点 G709

实 验 日 期 2021.04.02

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 4 -](#_Toc69989923)

[1.1 实验目的 - 4 -](#_Toc69989924)

[1.2 实验环境与工具 - 4 -](#_Toc69989925)

[1.2.1 硬件环境 - 4 -](#_Toc69989926)

[1.2.2 软件环境 - 4 -](#_Toc69989927)

[1.2.3 开发工具 - 4 -](#_Toc69989928)

[1.3 实验预习 - 4 -](#_Toc69989929)

[第2章 实验环境建立 - 5 -](#_Toc69989930)

[2.1 Ubuntu下CodeBlocks安装 - 5 -](#_Toc69989931)

[2.2 64位Ubuntu下32位运行环境建立 - 5 -](#_Toc69989932)

[第3章 C语言的数据类型与存储 - 6 -](#_Toc69989933)

[3.1 类型本质 - 6 -](#_Toc69989934)

[3.2 数据的位置-地址 - 6 -](#_Toc69989935)

[3.3 main的参数分析 - 8 -](#_Toc69989936)

[3.4 指针与字符串的区别 - 8 -](#_Toc69989937)

[第4章 深入分析UTF-8编码 - 9 -](#_Toc69989938)

[4.1 提交utf8len.c子程序 - 9 -](#_Toc69989939)

[4.2 C语言的strcmp函数分析 - 9 -](#_Toc69989940)

[4.3讨论：按照姓氏笔画排序的方法实现（选做，不做要求） - 9 -](#_Toc69989941)

[第5章 数据变换与输入输出 - 10 -](#_Toc69989942)

[5.1 提交**cs\_atoi.c** - 10 -](#_Toc69989943)

[5.2 提交**cs\_atof.c** - 10 -](#_Toc69989944)

[5.3 提交**cs\_itoa.c** - 10 -](#_Toc69989945)

[5.4 提交**cs\_ftoa.c** - 10 -](#_Toc69989946)

[第6章 整数表示与运算 - 11 -](#_Toc69989947)

[6.1 提交fib\_dg**.c** - 11 -](#_Toc69989948)

[6.2 提交**fib\_loop.c** - 11 -](#_Toc69989949)

[6.3 fib溢出验证 - 11 -](#_Toc69989950)

[6.4 除以0验证： - 11 -](#_Toc69989951)

[第7章 浮点数据的表示与运算 - 12 -](#_Toc69989952)

[7.1 正数表示范围 - 12 -](#_Toc69989953)

[7.2浮点数的编码计算 - 12 -](#_Toc69989954)

[7.3特殊浮点数值的编码 - 12 -](#_Toc69989955)

[7.4浮点数除0 - 13 -](#_Toc69989956)

[7.5 Float的微观与宏观世界 - 13 -](#_Toc69989957)

[7.6 讨论：任意两个浮点数的大小比较 - 13 -](#_Toc69989958)

[第8章 舍尾平衡的讨论 - 14 -](#_Toc69989959)

[8.1 描述可能出现的问题 - 14 -](#_Toc69989960)

[8.2 给出完美的解决方案 - 14 -](#_Toc69989961)

[第9章 总结 - 15 -](#_Toc69989962)

[9.1 请总结本次实验的收获 - 15 -](#_Toc69989963)

[9.2 请给出对本次实验内容的建议 - 15 -](#_Toc69989964)

[参考文献 - 16 -](#_Toc69989965)

# 第1章 实验基本信息

## 1.1 实验目的

熟练掌握计算机系统的数据表示与数据运算；

通过C程序深入理解计算机运算器的底层实现与优化；

掌握VS/CB/GCC等工具的使用技巧与注意事项；

## 1.2 实验环境与工具

### 1.2.1 硬件环境

X64 CPU；1.6GHz；8G RAM；256G SSD Disk；1T HDD Disk

### 1.2.2 软件环境

Windows10 64位；Vmware 14pro；Ubuntu 20.04.2 LTS 64位

### 1.2.3 开发工具

Visual Studio Code 64位；vim/gpedit+gcc

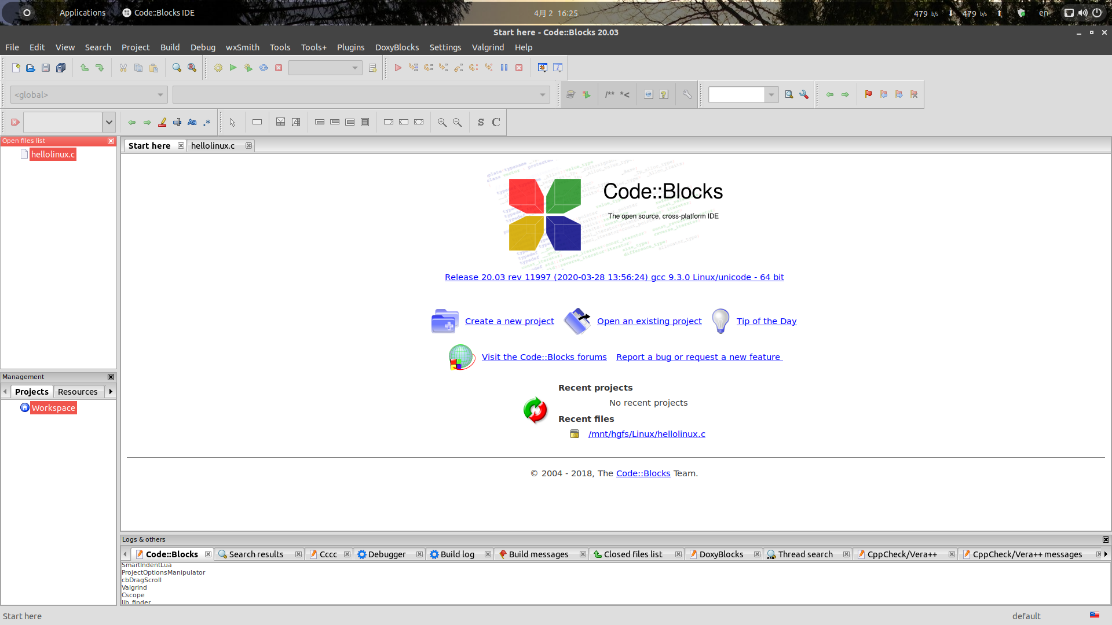
## 1.3 实验预习

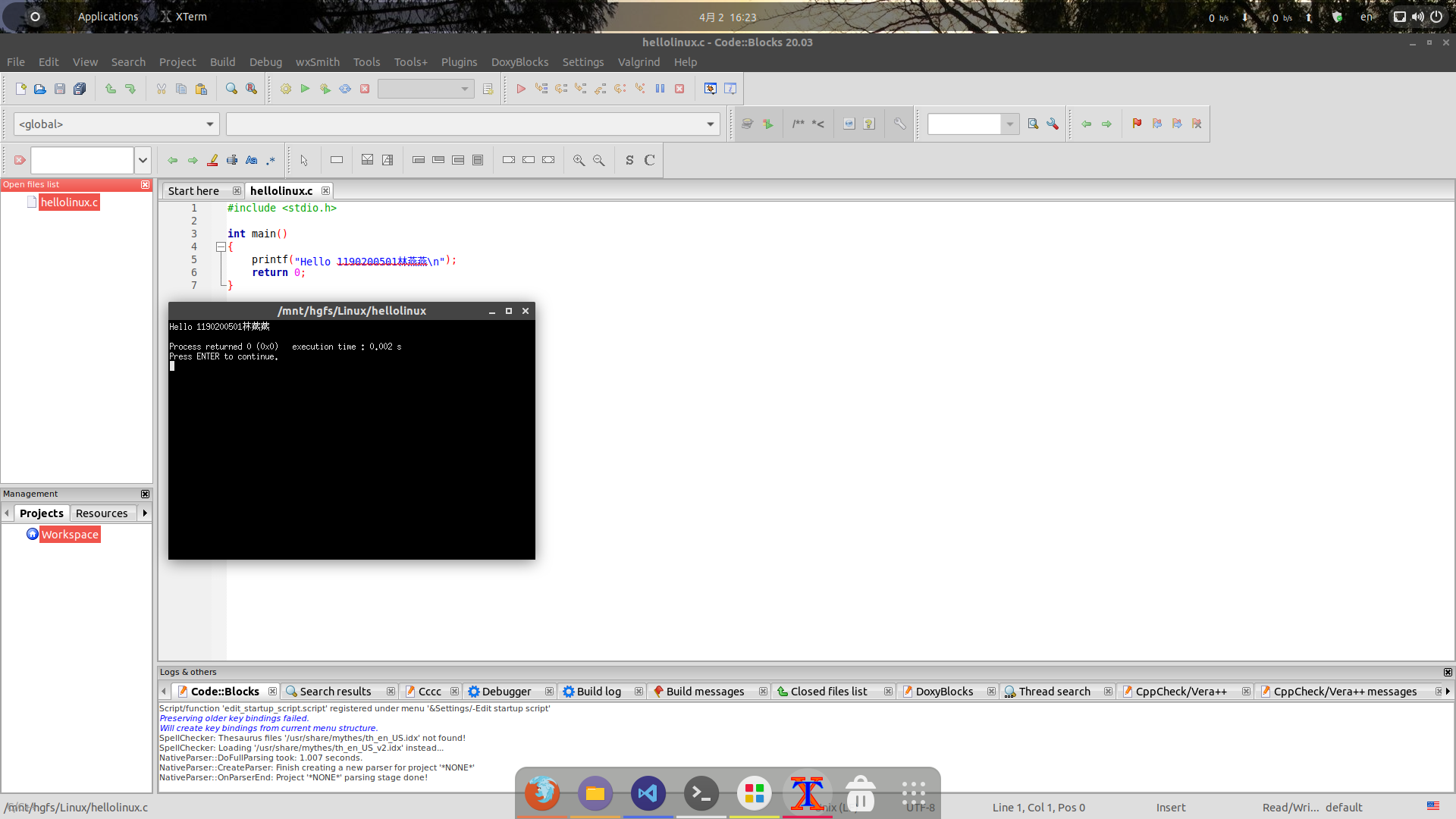
* 上实验课前，必须认真预习实验指导书（PPT或PDF）
* 了解实验的目的、实验环境与软硬件工具、实验操作步骤，复习与实验有关的理论知识。
* 采用sizeof在Windows的VS/CB以及Linux的CB/GCC下获得C语言每一类型在32/64位模式下的空间大小
* Char /short int/int/long/float/double/long long/long double/指针
* 编写C程序，计算斐波那契数列在int/long/unsigned int/unsigned long类型时，n为多少时会出错
* 先用递归程序实现，会出现什么问题？
* 再用循环方式实现。
* 写出float/double类型最小的正数、最大的正数（非无穷）
* 按步骤写出float数-1.1在内存从低到高地址的字节值-16进制
* 按照阶码区域写出float的最大密度区域范围及其密度，最小密度区域及其密度（区域长度/表示的浮点个数）

# 第2章 实验环境建立

## 2.1 Ubuntu下CodeBlocks安装

CodeBlocks运行界面截图：编译、运行hellolinux.c

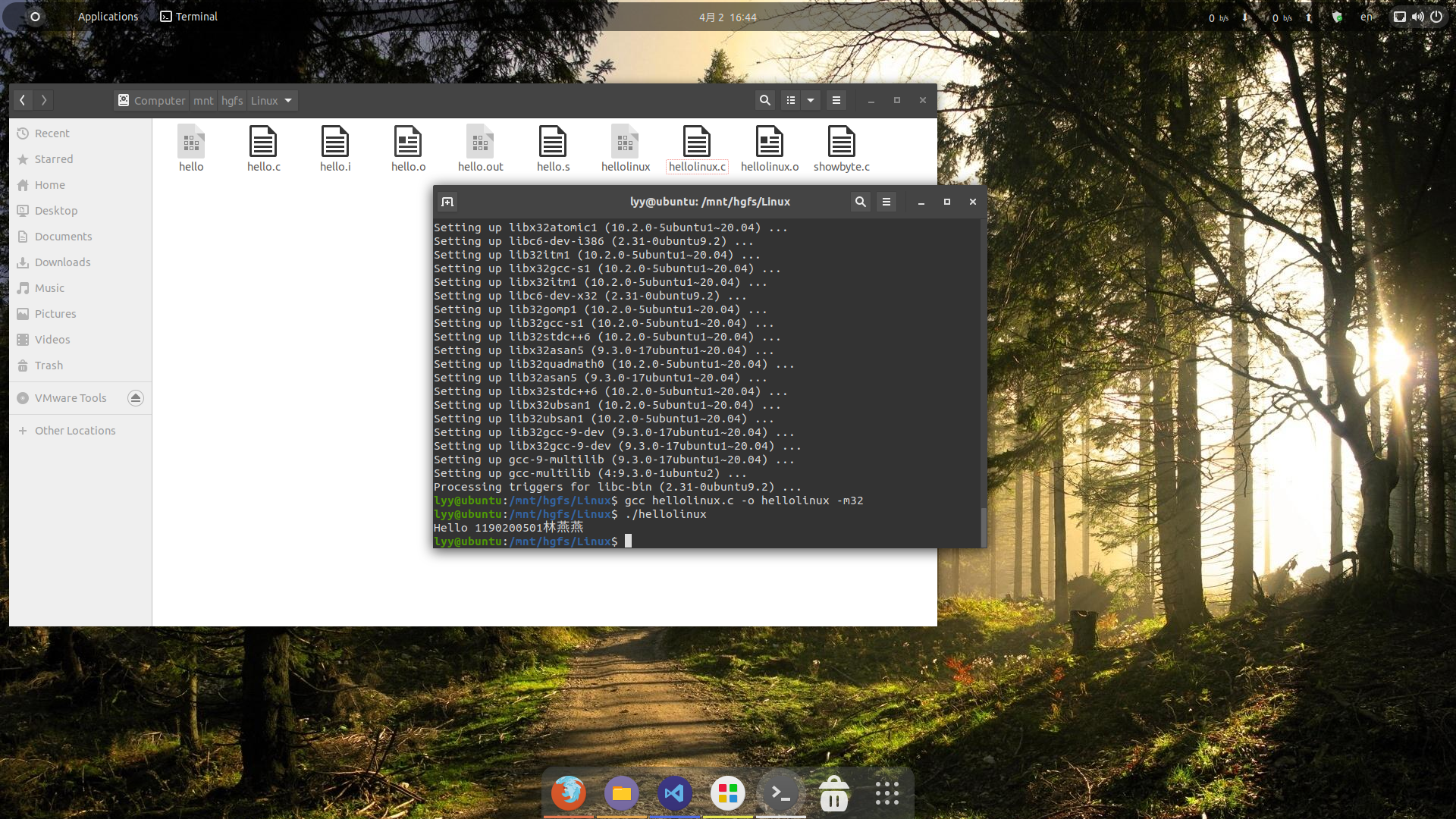




## 2.2 64位Ubuntu下32位运行环境建立

在终端下，用gcc的32位模式编译生成hellolinux.c。执行此文件。

Linux及终端的截图。



# 第3章 C语言的数据类型与存储

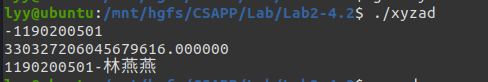
## 3.1 类型本质

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Win/VS/x86** | **Win/VS/x64** | **Win/CB/32** | **Win/CB/64** | **Linux/CB/32** | **Linux/CB/64** |
| char | 1 | 1 | 1 | 1 | 1 | 1 |
| short | 2 | 2 | 2 | 2 | 2 | 2 |
| int | 4 | 4 | 4 | 4 | 4 | 4 |
| long | 4 | 4 | 4 | 4 | 4 | 8 |
| long long | 8 | 8 | 8 | 8 | 8 | 8 |
| float | 4 | 4 | 4 | 4 | 4 | 4 |
| double | 8 | 8 | 8 | 8 | 8 | 8 |
| long double | 8 | 8 | 12 | 16 | 12 | 16 |
| 指针 | 4 | 8 | 4 | 8 | 4 | 8 |

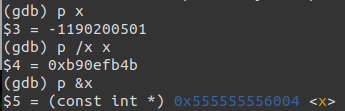
C编译器对sizeof的实现方式：sizeof是一个操作符，由编译器计算，在编译阶段计算出结果，在运行时是个常量

## 3.2 数据的位置-地址

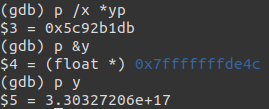
打印x、y、z输出的值：



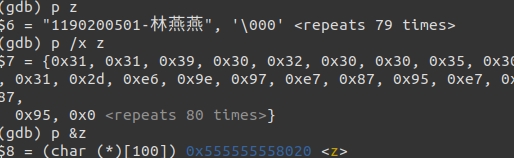
反汇编查看x、y、z的地址，每字节的内容：



x的地址0x555555556004, 每字节的内容0xb9 0e fb 4b



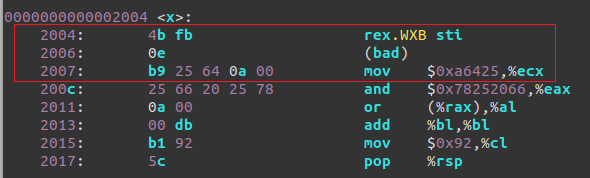
y的地址0x7fffffffde4c, 每字节的内容0x5c 92 b1 db



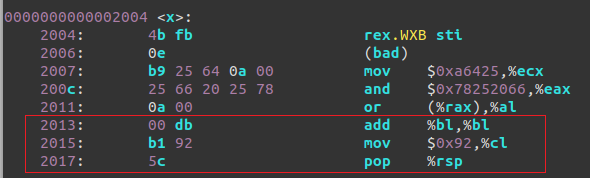
z的地址0x555555558020, 每字节的内容{0x31, 0x31, 0x39, 0x30, 0x32, 0x30, 0x30, 0x35, 0x30, 0x31, 0x2d, 0xe6, 0x9e, 0x97, 0xe7, 0x87, 0x95, 0xe7, 0x87,

0x95, 0x0 }

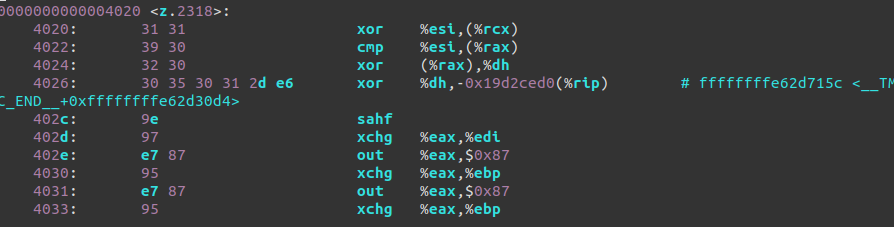
反汇编查看x、y、z在代码段的表示形式。



红框中为x在代码段的表示形式



红框中为y在代码段的表示形式



上图为z在代码段的表示形式

x与y在 编译 阶段转换成补码与ieee754编码。

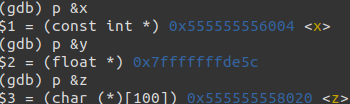
数值型常量与变量在存储空间上的区别是： 变量占据存储空间，常量不占据存储空间

字符串常量与变量在存储空间上的区别是： 字符串变量的名字（一维字符数组名）及其所需的存储空间是显式定义的，并通过名字来引用相应的字符串变量

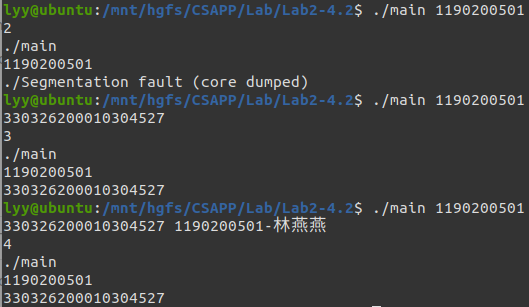
常量表达式在计算机中处理方法是： 常量表达式在编译时就已经被计算好了，直接储存到内存中

## 3.3 main的参数分析

反汇编查看x、y、z的地址截图4；

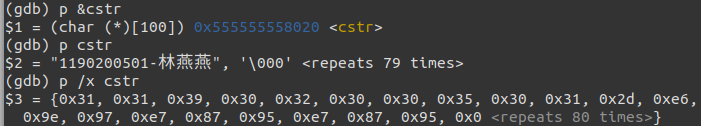


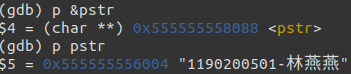
命令行传递参数，反汇编观察argc、argv的地址与内容，截图4。



## 3.4 指针与字符串的区别

cstr的地址与内容截图，pstr的内容与截图，截图5





pstr修改内容会出现什么问题 修改指针的时候会报错，因为修改后的内容超过了原来指针所指向内容的空间

# 第4章 深入分析UTF-8编码

## 4.1 提交utf8len.c子程序

## 4.2 C语言的strcmp函数分析

分析论述：strcmp到底按照什么顺序对汉字排序

每个汉字都有其对应的Unicode码， strcmp按照汉字对应的Unicode编码大小对汉字进行排序。所以用strcmp比较姓名的大小时，首先比较姓的Unicode编码大小，若一样，则继续比较下一位名的Unicode编码大小。

## 4.3讨论：按照姓氏笔画排序的方法实现（选做，不做要求）

分析论述：应该怎么实现呢？

用一个数据库或文件，储存每个汉字关于笔画数量的编码；

当每次排序时，调用此数据库或文件中汉字对应的编码按照笔画数量进行排序；

若姓氏笔画数量相同，则按照汉字对应的Unicode码进行排序，从而实现按照姓氏笔画排序的方法。

# 第5章 数据变换与输入输出

## 5.1 提交**cs\_atoi.c**

## 5.2 提交**cs\_atof.c**

## 5.3 提交**cs\_itoa.c**

## 5.4 提交**cs\_ftoa.c**

# 第6章 整数表示与运算

## 6.1 提交fib\_dg**.c**

## 6.2 提交**fib\_loop.c**

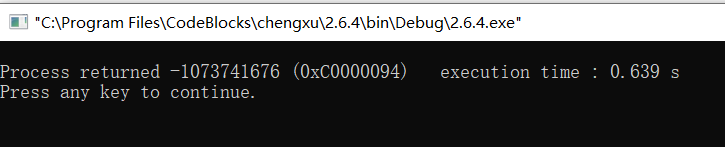
## 6.3 fib溢出验证

int 时从n= 47 时溢出，long时n= 47 时溢出。

unsigned int 时从n= 48 时溢出，unsigned long时n= 48 时溢出。

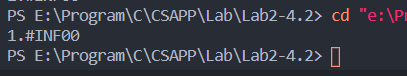
## 6.4 除以0验证：

除以0：



截图1

除以极小浮点数，截图：



# 第7章 浮点数据的表示与运算

## 7.1 正数表示范围

写出float/double类型最小的正数、最大的正数（非无穷）

float : 1.4E-45 3.4E38

double: 4.9E-324 1.8E308

## 7.2浮点数的编码计算

（1）按步骤写出float数-1.1的浮点编码计算过程，写出该编码在内存中从低地址字节到高地址字节的16进制数值

解：

* -1.1(10)=-1.00011001100110011001101(2)×2º

S=1；

* M=-1.00011001100110011001101(2)

E=0

* frac=00011001100110011001101(2)

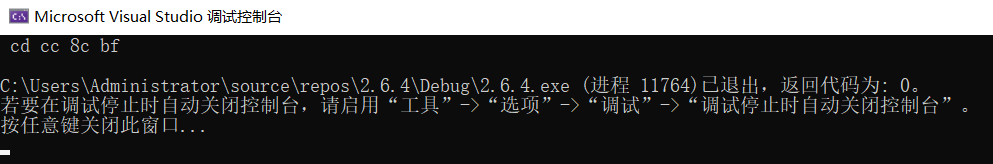
exp=E + Bias = 127 = 01111111

编码结果：10111111100011001100110011001101

十六进制：BF8CCCCD

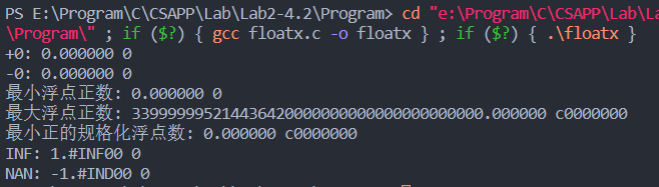
从低地址到高地址数值：CDCC8CBF

（2）验证：编写程序，输出值为-1.1的浮点变量其各内存单元的数值，截图。



## 7.3特殊浮点数值的编码

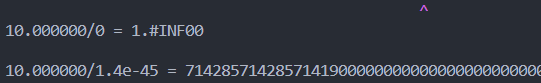
（1）构造多float变量，分别存储+0-0，最小浮点正数，最大浮点正数、最小正的规格化浮点数、正无穷大、Nan,并打印最可能的精确结果输出（十进制/16进制）。截图。



（2）提交子程序floatx.c

## 7.4浮点数除0

（1）编写C程序，验证C语言中float除以0/极小浮点数后果，截图



（2）提交子程序float0.c

## 7.5 Float的微观与宏观世界

按照阶码的数值区域，float编码最密集区域的阶码编码是： -1.1111(23个1)×2-126~1.1111(23个1)×2-126, 该区域中相邻浮点数编码的数值的间距是： 1.40e-45 ；

float编码最稀疏区域的阶码编码是： 1.0000(23个1)×2127~1.11111(23个1)×2127,该区域中相邻浮点数编码的数值的间距是：20282409603651670423947251286016。

最小正数变成十进制科学记数法，最可能能精确到多少1.401298464324817e-45

最大正数变成十进制科学记数法，最可能能精确到多少3.4028234663852886e37

## 7.6 讨论：任意两个浮点数的大小比较

论述比较方法以及原因。

方法：当符号相同时，将两个浮点数相减并对差取绝对值，取一极小值如1e-7与差的绝对值比较：如果差的绝对值小于该值，则可以认为相等，如果大于该值，判断差是否大于零，若大于零，则被减的浮点数大，若小于零，则减的浮点数大。

理由：计算机表示浮点数有精度限制，对于超出了精度限制的浮点数，计算机会将精度之外的小数部分截断。可能使本来不等的两个浮点数相等，因此不能直接对浮点数进行大小比较。

# 第8章 舍尾平衡的讨论

## 8.1 描述可能出现的问题

舍尾时采用四舍五入，会造成数据的误差，当数据合计时，误差累积导致与实际合计值不匹配。

## 8.2 给出完美的解决方案

求和后进行舍尾

# 第9章 总结

## 9.1 请总结本次实验的收获

本次实验我了解了各种变量在内存中的地址以及各种变量在不同操作系统和不同位数机器的大小，学会了怎么用反汇编的方式查看变量在内存中的地址和内容，知道了反汇编的操作步骤，学习了字符的utf-8编码方式。

## 9.2 请给出对本次实验内容的建议

希望ppt内容能描述详细一些

注：本章为酌情加分项。

# 参考文献

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[5] KANAMORI H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

[6] 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.