

**实验报告**

**实 验（二）**

题 目 DataLab 数据表示

专 业 计算机类

学　　 号 1190200523

班　　 级 1903002

学 生 石翔宇

指 导 教 师 郑贵滨

实 验 地 点 G709

实 验 日 期 2021.4.2

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 4 -](#_Toc20402368)

[1.1 实验目的 - 4 -](#_Toc20402369)

[1.2 实验环境与工具 - 4 -](#_Toc20402370)

[1.2.1 硬件环境 - 4 -](#_Toc20402371)

[1.2.2 软件环境 - 4 -](#_Toc20402372)

[1.2.3 开发工具 - 4 -](#_Toc20402373)

[1.3 实验预习 - 4 -](#_Toc20402374)

[第2章 实验环境建立 - 5 -](#_Toc20402375)

[2.1 Ubuntu下CodeBlocks安装 - 5 -](#_Toc20402376)

[2.2 64位Ubuntu下32位运行环境建立 - 5 -](#_Toc20402377)

[第3章 C语言的数据类型与存储 - 6 -](#_Toc20402378)

[3.1 类型本质（1分） - 6 -](#_Toc20402379)

[3.2 数据的位置-地址（2分） - 6 -](#_Toc20402380)

[3.3 main的参数分析（2分） - 6 -](#_Toc20402381)

[3.4 指针与字符串的区别（2分） - 6 -](#_Toc20402382)

[第4章 深入分析UTF-8编码 - 8 -](#_Toc20402383)

[4.1 提交utf8len.c子程序 - 8 -](#_Toc20402384)

[4.2 C语言的strcmp函数分析 - 8 -](#_Toc20402385)

[4.3讨论：按照姓氏笔画排序的方法实现 - 8 -](#_Toc20402386)

[第5章 数据变换与输入输出 - 9 -](#_Toc20402387)

[5.1 提交**cs\_atoi.c** - 9 -](#_Toc20402388)

[5.2 提交**cs\_atof.c** - 9 -](#_Toc20402389)

[5.3 提交**cs\_itoa.c** - 9 -](#_Toc20402390)

[5.4 提交**cs\_ftoa.c** - 9 -](#_Toc20402391)

[5.5 讨论分析OS的函数对输入输出的数据有类型要求吗 - 9 -](#_Toc20402392)

[第6章 整数表示与运算 - 10 -](#_Toc20402393)

[6.1 提交fib\_dg**.c** - 10 -](#_Toc20402394)

[6.2 提交**fib\_loop.c** - 10 -](#_Toc20402395)

[6.3 fib溢出验证 - 10 -](#_Toc20402396)

[6.4 除以0验证： - 10 -](#_Toc20402397)

[第7章 浮点数据的表示与运算 - 11 -](#_Toc20402398)

[7.1 正数表示范围 - 11 -](#_Toc20402399)

[7.2浮点数的编码计算 - 11 -](#_Toc20402400)

[7.3特殊浮点数值的编码 - 11 -](#_Toc20402401)

[7.4浮点数除0 - 11 -](#_Toc20402402)

[7.5 Float的微观与宏观世界 - 11 -](#_Toc20402403)

[7.6 讨论：任意两个浮点数的大小比较 - 11 -](#_Toc20402404)

[第8章 舍尾平衡的讨论 - 12 -](#_Toc20402405)

[8.1 描述可能出现的问题 - 12 -](#_Toc20402406)

[8.2 给出完美的解决方案 - 12 -](#_Toc20402407)

[第9章 总结 - 13 -](#_Toc20402408)

[9.1 请总结本次实验的收获 - 13 -](#_Toc20402409)

[9.2 请给出对本次实验内容的建议 - 13 -](#_Toc20402410)

[参考文献 - 14 -](#_Toc20402411)

# 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

* Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
* 16GB RAM
* 1TB HDD + 512G SSD

### 1.2.2 软件环境

* Windows 10 21H1
* Ubuntu 20.04 LTS

### 1.2.3 开发工具

* VSCode，CodeBlocks，gcc+gdb

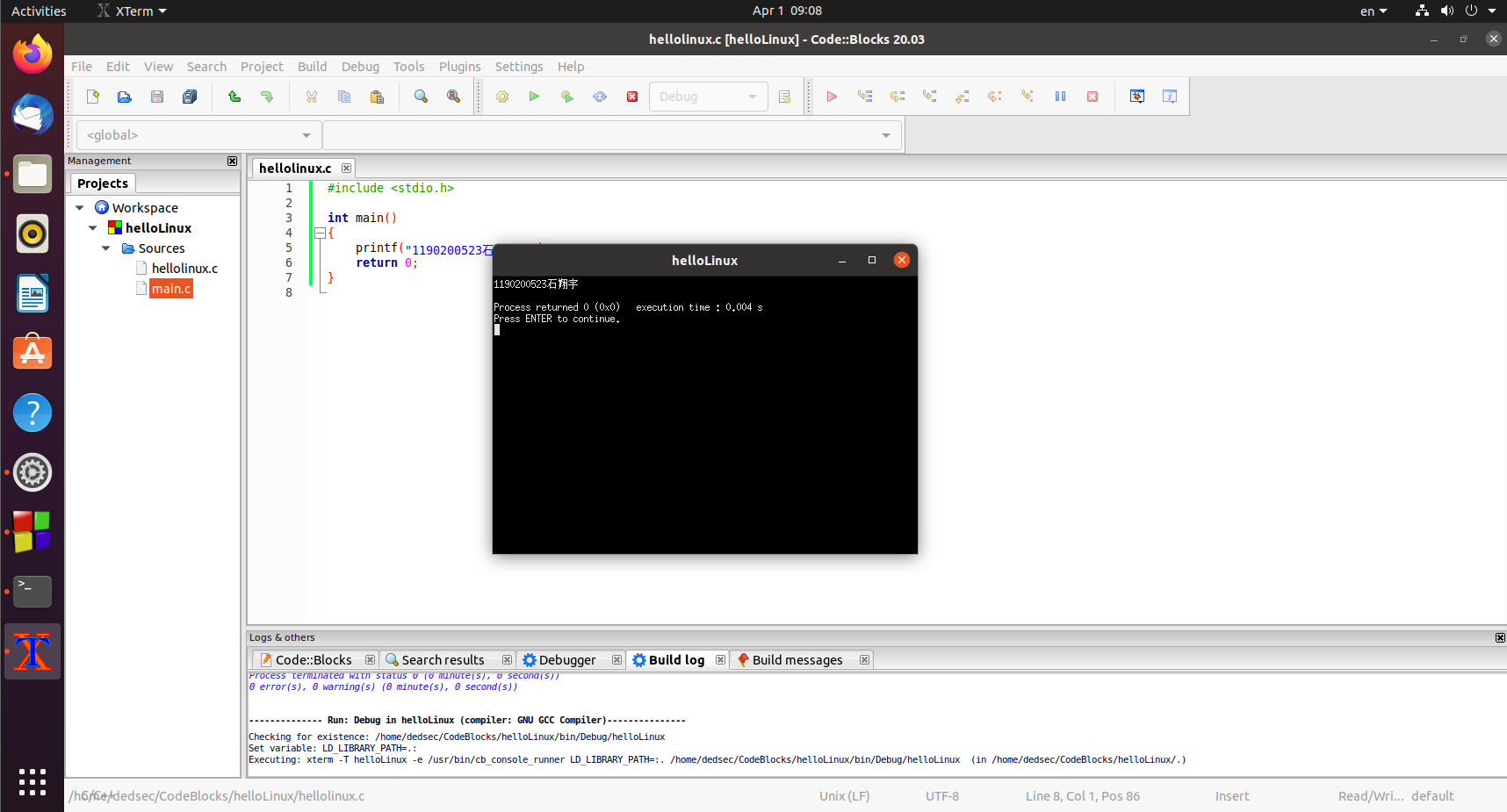
## 1.3 实验预习

* 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

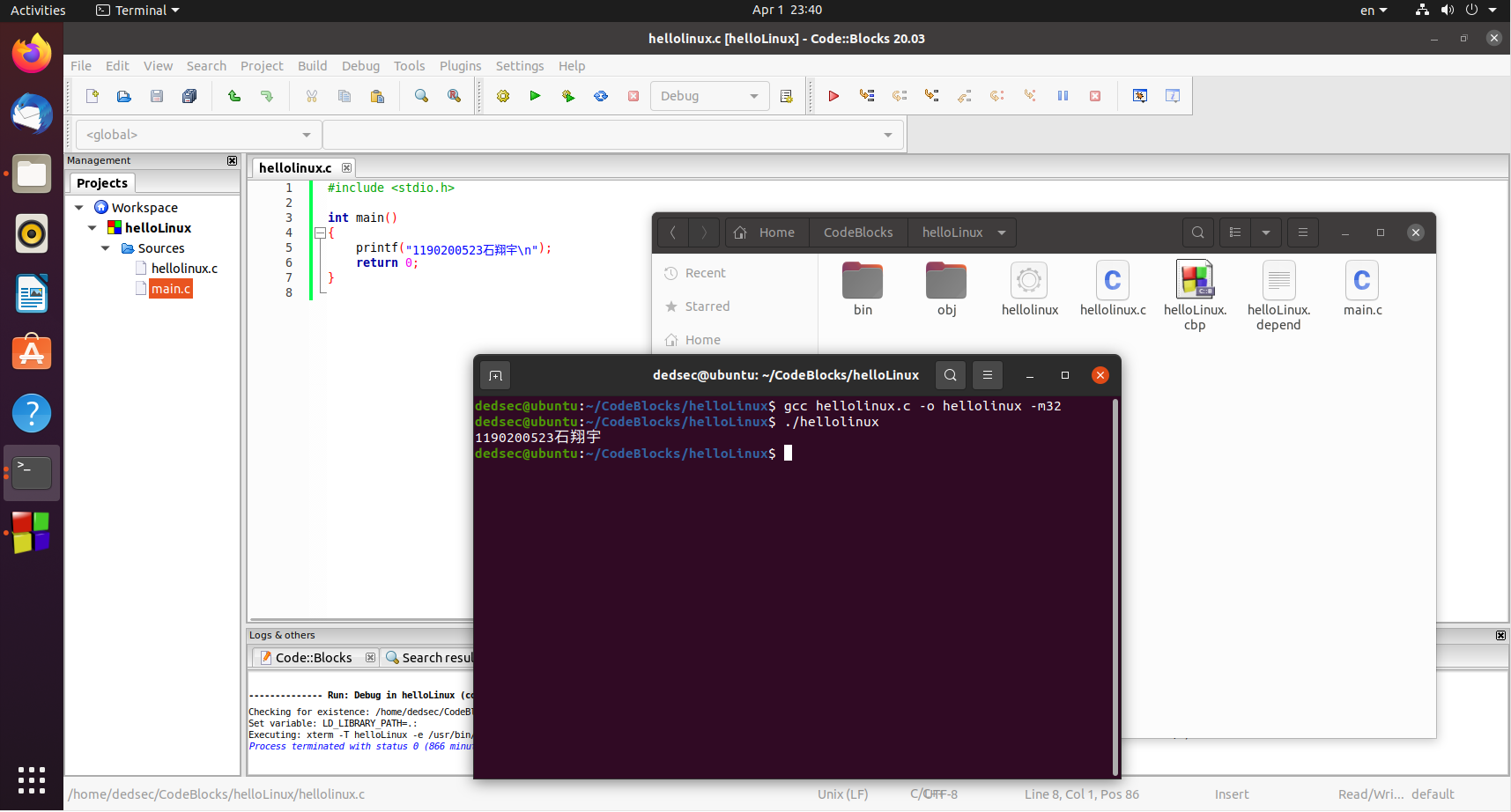
CodeBlocks运行界面截图：编译、运行hellolinux.c



## 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 |

**3.1提示：在linux和windows两种平台下，用两种编译器观察。不限定非得是Codeblock**

C编译器对sizeof的实现方式：sizeof是一个操作符，由编译器来计算，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：截图1

反汇编查看x、y、z的地址，每字节的内容：截图2，标注说明

反汇编查看x、y、z在代码段的表示形式。截图3，标注说明

x与y在\_\_\_\_\_\_\_\_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：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

字符串常量与变量在存储空间上的区别是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

常量表达式在计算机中处理方法是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.2提示：**

**①在linux下生成可执行程序，假设是a.out。然后用objdump -dx a.out > a-dump.s 生成反汇编文件a-dump.s，查看a-dump.s**

**②gdb ./a.out ↙ layout asm ↙ b main ↙ r ↙ disp argc ↙….**

## 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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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

## 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？

# 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**

## 5.5 讨论分析OS的函数对输入输出的数据有类型要求吗

论述如下：

# 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=\_\_\_\_\_\_\_\_\_时溢出，long时n=\_\_\_\_\_\_\_\_\_\_\_\_时溢出。

unsigned int 时从n=\_\_\_\_\_\_\_\_\_时溢出，unsigned long时n=\_\_\_\_\_\_\_\_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）

## 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

（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：\_\_\_\_\_\_\_\_\_\_, 该区域中相邻浮点数编码的数值的间距是：\_\_\_\_\_\_\_\_\_\_；float编码最稀疏区域的阶码编码是：\_\_\_\_\_\_\_\_\_\_, 该区域中相邻浮点数编码的数值的间距是：\_\_\_\_\_\_\_\_\_\_。

最小正数变成十进制科学记数法，最可能能精确到多少\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

## 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。

# 第8章 舍尾平衡的讨论

## 8.1 描述可能出现的问题

## 8.2 给出完美的解决方案

# 第9章 总结

## 9.1 请总结本次实验的收获

## 9.2 请给出对本次实验内容的建议

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] 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.