

# ICS-LAB2 DataLab数据表示

哈尔滨工业大学  
计算机科学与技术学院

2019年9月

# 一、实验基本信息

## ■ 实验类型：综合型实验

## ■ 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

## ■ 实验指导教师

- 任课教师：郑贵滨
- 实验室教师：王晴、王宇
- TA：王庆尧、杨宇奇

## ■ 实验班级、人数与分组

- 1、5、6
- 一人一组

- **实验学时：3，18:30-21:00**
- **实验学分：5，本次实验按100分计算，折合成总成绩的5分。**
- **实验地点：G712、G709**
- **实验环境与工具：**
  - X64 CPU；2GHz；2G RAM；256GHD Disk 以上
  - Windows7 64位以上；VirtualBox/Vmware 11以上；Ubuntu 16.04 LTS 64位/优麒麟 64位；
  - Visual Studio 2010 64位以上；CodeBlocks；vi/vim/gpedit+gcc
- **学生实验准备：禁止准备不合格的学生做实验**
  - 个人笔记本电脑
  - 实验环境与工具所列明软件
  - 参考手册: Linux环境下的命令；GCC手册；GDB手册
  - <http://docs.huihoo.com/c/linux-c-programming/> C汇编Linux手册
  - <http://csapp.cs.cmu.edu/3e/labs.html> CMU的实验参考
  - <http://www.linuxidc.com/> <http://cn.ubuntu.com/>  
<http://forum.ubuntu.org.cn/>

## 二、实验要求

- 学生应穿鞋套进入实验室
- 进入实验室后在签到簿中签字
- 实验安全与注意事项
  - 禁止使用笔记本电脑以外的设备
  - 学行生不得自行开关空调、投影仪
  - 学生不得自打开窗户
  - 不得使用实验室内的其他实验箱、示波器、导线、工具、遥控器等
  - 认真阅读消防安全撤离路线
  - 突发事件处理：第一时间告知教师，同时关闭电源插排开关。
- 遵守学生实验守则，爱护实验设备，遵守操作规程，精心操作，注意安全，严禁乱拆乱动。
- 实验结束后要及时关掉电源，对所用实验设备进行整理，设备摆放和状态恢复到原始状态。
- 桌面整洁、椅子归位，经实验指导教师允许后方可离开

# 三、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）

# 四、实验内容与步骤

## ■ 1.环境建立

- Windows下Visual Studio 2019 64位
- Windows下CodeBlocks
- Ubuntu下安装CodeBlocks（为以后编程调试准备—可省略）
- 64位Ubuntu下32位编译与运行环境（为32位程序运行与反编译准备）

## ■ 2.Windows 环境下开发工具使用

- 解决方案->工程->源程序->头文件/资源文件等的软件工程理念
- Visual Studio 2019下的程序开发与调试：x86/x64切换下看看
  - scanf 问题与解决、各类开关的含义
  - 调试查看各类寄存器、变量/参数、汇编与机器代码、堆栈等等
  - 性能分析            源程序的编码切换！！执行对吗？
- CodeBlocks下的程序开发与调试：m32/m64的切换
  - 源程序是ansi还是utf-8编码。验证下。
  - 能变更源程序编码吗？                    采用类Linux编译调试环境。

# VS的使用

## ■ 建解决方案+工程：连接选择子系统平台console

- 添加源程序.c或者.cpp
- 选择64/32位，选择Debug/Release

## ■ 编译链接

- 菜单->生成，可选择编译、生成，重新生成等

## ■ 调试：调试菜单

- F10 单步调试，不进入子程序
- F11 单步调试，进入子程序
- F9 断点 F5 开始/继续 Shift+F5停止 不调试运行Ctrl+F5
- Ctrl+F10 运行到光标处
- Shift+F11 跳出子程序

## ■ 调试：查看菜单

- 看寄存器 Alt+5 看内存Alt+6 看汇编语言程序 Alt+8
- 鼠标右键可以增加内容，更改显示格式等等
- 增加内存窗口：Ctrl+Alt+M+2 3 4

# CB的使用

## ■ 建立WorkPlace+Project工程： console/empty

- 添加源程序.c或者.cpp
- 选择64/32位，选择Debug/Release 项目属性GCC编译里 -m32/-m64等

## ■ 编译链接

- 菜单->生成，可选择编译^Shift+F9、生成^F9，重新生成^F11等
- F2 查看log编译等输出的窗口

## ■ 调试： 调试菜单： GDB同Linux， 可以代替Linux环境的学习

- 开始/继续 F8(无断点全速) ， Shift+F7 进入编译连接调试并到main。
- F7下一行 Shift+F7单步入 ^F7单步出
- Alt+F7下一指令 ALT+Shift+F7 下一指令入
- F5 断点 F4 运行到光标处 Shift+F8停止调试
- Shift+F11 跳出子程序 settings/debugger可改成Intel汇编

## ■ 调试： 查看菜单 Debugging Windows

- 寄存器 内存 汇编语言程序 监控变量窗口
- 汇编语言是Linux的AT&T汇编



# 3.Linux常用命令复习

- <http://blog.csdn.net/xiaoguaihai/article/details/8705992>
- **cd 切换目录   pwd 显示当前目录**
  - cd .. 上级目录   cd ~ 返回主目录   cd - 返回刚才的目录
- **ls 显示当前目录文件**
  - ls -l -a -h 目录 文件\*?                      注：可以单独用，可以组合
- **重要热键**
  - [Tab]        ---具有『命令补全』不『档案补齐』的功能
  - 光标上下键 ---上一个命令，下一个命令
  - [Ctrl]-c     ---让当前的程序『停掉』
- **文件与目录操作:cp   rm mkdir rmdir mv   chmod**
- **显示打印操作: cat nano more type**
- **编辑操作: vi   vim emacs   建议gedit**
  - 可用其他集成环境如visual code,Code Blocks
  - Gedit下更改C源程序（任意文件）的编码格式/换行方式。运行下看看！

# GDB的使用

- 可以先用 `readelf -a hello` 查看可执行文件的大致结构。第一条指令位置不是 `main`
- `(gdb)l (ist)`                      `(gdb) b main` 断点                      `(gdb) r` 运行 `c`继续
- `(gdb) disassemble main` /m c和asm一起排列 /r 看16进制代码
- `(gdb) x/15i main`    用 `x/i` 可以查看指令
- `(gdb) x/15 地址`    用 `x/` 可以查看数据    `s`字符串 `x`十六进制
- `(gdb) display/10i main` 命令自动显示当前正要执行汇编指令
- `(gdb) ni`; 单步执行下一条机器指令                      `n`单步执行一条C语句  
    `(gdb) si`; 单步执行入一条机器指令                      `s`单步执行一条C语句
- `(gdb) print $eax $RIP`    ➔反人类!!
- `(gdb) info reg`                      `info proc` all看本进程的所有信息
- `(gdb) layout prev | next | <layout_name>` 改变布局    `src/asm/split/regs`
- `(gdb) set disassembly-flavor intel` 改成Intel格式

## ■ 4. 类型的数据存储：类型本质是空间 sizeof

- 编写C程序在Windows的VS/CB下分别查看32/64位模式的各类型所占字节数:char/short int/int/long/float/double/ long long/long double/指针
- 在Linux下CB或GCC/GDB下32/64位模式下查看。 画一个表格!
- sizeof是函数吗? 通过汇编代码、反汇编可执行文件看sizeof 本质。

## ■ 5. 探索数据存储的位置：地址！（x86-32位）

- 常量：编写C子程序，全局变量x=整型常量=-学号（去掉学号中的字母）；
- 局部变量y=浮点常量=身份证号（身份证号中如有x替换为0）；
- 静态局部变量z=字符串常量=“学号-姓名”。
- 用printf分行打印x、y、z。反汇编查看这些常量出现在内存什么段？每个字节内容？代码中的形式？截图说明。
- 分析验证：x与y是什么阶段转换成补码与ieee754编码的？数值型常量与变量在存储空间上有什么区别！字符串型呢？常量表达式呢？

- 5. 探索数据存储的位置：地址！（x86-32位）
  - main参数的位置：将前述x、y、z三个变量的数值作为命令行参数，传递给main函数并打印输出！分析main参数在内存中的位置以及存储形式（argc/argv）。截图！
  - 指针与字符串的区别：全局变量 `char cstr[100]="学号-姓名"`；`char *pstr="学号-姓名"`。编程，把字符串“身份证号”复制到cstr、pstr，会有什么问题出现？
- 6. 深入分析程序的编码
  - 源程序的编码：和OS、编辑器、编译器相关，Linux / windows/Mac 下的编码与回车处理不同，所以不同编码在不正确的使用环境下可能有编译以及错误输出。
  - 分析验证：VS/CB/GCC下不同源程序编码是怎么处理的？

## ■ 7. Unicode与UTF-8编码

- Unicode.org 学习！ Code chart CJK: U4E00.PDF 基本多文种平面
- <https://blog.csdn.net/oyji1992/article/details/80030366> 17个平面
- UTF-8是Unicode的一种编码方式，C有各种编码转换函数

//0000 0000 – 0000 007F:      0xxxxxxx      7位

//0000 0080 – 0000 07FF:      110xxxxx 10xxxxxx      11 位

//0000 0800 – 0000 FFFF:      1110xxxx 10xxxxxx 10xxxxxx      16位

//0001 0000 -- 001F FFFF:      11110xxx 10xxxxxx 10xxxxxx 10xxxxxx      21位

- //常用汉字cjk 4e00-9fbf 所以三个字节,2万多个中日韩汉字
- 编写子程序 int utf8len(char\*) 返回字符个数（一个utf8编码一个字符）
- 分析验证字符串比较函数，不同编码的串按姓名排序的正确性！
- 讨论：按照姓氏笔画排序怎么实现呢？ 输入法/显示打印软件中编码。

## 8.数据变换与输入输出

- **Int/float与char\*的转换情景：**
  - 编译器碰到的常数
  - Printf/scanf的实现
- **编写函数cs\_atoi/cs\_atof 字符串转正数/浮点数**
  - 不用C语言内置函数如sprintf/atoi等
  - 字符串限定为十进制，处理效果同scanf
- **编写函数cs\_itoa/cs\_ftoa 正数/浮点数转字符串**
  - 不用C语言内置函数itoa/sprintf等
  - 字符串限定为十进制，处理效果同printf
- **讨论：OS的函数对输入输出的数据有类型要求吗？**

## 9.整数表示与运算

- 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？ 0, 1, 1, 2, 3, 5, 8.....
  - 再用循环方式实现。
- 编写C程序，验证C语言中整数除以0/极小浮点数后果，截图

# 10. 千年虫/万年虫模拟

- **2000年问题：** 年份而已，只有两位，问题是什么？怎么解决
- **9999年问题：** 4位年份，问题是什么？怎么解决？
  - 请在BIOS与OS中验证你的机器到底会有什么问题？包括C中获取日期时间的函数。
- **2038虫问题：** 计算机的时间获取与long的计数问题
  - 请在BIOS与OS中验证你的机器到底会有什么问题？包括C中获取日期时间的函数。会出问题吗？



# 11.浮点数据的表示与运算

- 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，编写子程序在内存验证，截图
- 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）。
- 编写C程序，验证C语言中float除以0/极小浮点数后果，截图

# 12.Float的微观与宏观世界

- 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）
- 锱铢必究的微观世界：最可能精确到多少呢？变成十进制科学记数法呢？
- 心系宇宙的宏观世界：心有多粗呢？十进制科学记数法呢？
- 讨论：怎么判断两个浮点数的大小呢？

# 13. 舍尾平衡的讨论：当天税务局长

- 全国税收是从一个个单位/个人的每一笔消费/营收等计算出来的。
- 不同时间段、不同行业、不同县市省国的统计口径不同，最后的统计数据也不一样，经常出现分项数据的和与总数不一致的情况。
- 浮点数的二进制表示，导致有精度问题。浮点数据的密度不同，也导致数据的精度。
- 不要找借口、不要推脱！
- 设计一套完美的舍尾平衡算法吧！！！！！！！！
- .....

## 五、实验报告格式

- 按照实验报告模板所要求的格式与内容提交。
- 实验后3周内提交至课代表并打包给授课教师。
- 本次实验成绩按100分计
  - 按时上课，签到5分
  - 按时下课，不早退5分
  - 课堂表现：10分，不按操作规程、非法活动扣分。
  - 实验报告：80分。具体参见实验报告各环节的分值
- 学生提交1个文件包，课代表提交1个包