

《计算机系统基础实验》

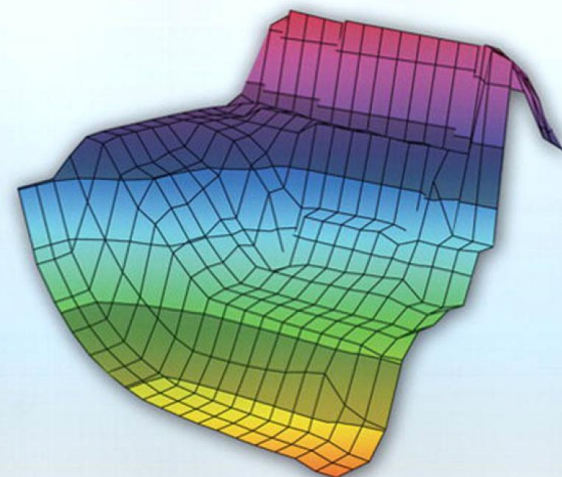
Introduction to Computer Systems

2022 春季

华中科技大学

张宇



2022年5月



- Lab1 数据表示
- Lab2 拆弹实验
- Lab3 缓冲区溢出攻击实验
- 注意：
 - 尽量自己弄懂自己做，考试也会考实验

■ CMU 深入理解计算机系统 原版实验

- 实验环境：linux 32位
- 实验技能：linux基本命令，编程环境
- 实验原理：定点数补码表示，浮点数IEEE 754表示
- 开机进入Linux 操作系统环境
- 配置IA-32兼容库（ia32-libs），注意高版本的Linux系统，该开发包已经被其他包替代了，大家要根据自己OS的版本通过百度或Google去查资料安装。

- 下载 lab1-handout.tar, 存放在下载目录
- 点击左侧dock图标 
 - 将刚下载的文件右键点击移动到主目录 (home)
- 点击左侧dock图标, 键入term, 打开终端
 - cd ~ 进入主目录  键入 ls 查看是否有下载文件
 - tar vxf lab1-handout.tar 解压代码框架
 - cd lab1-handout
 - ls 显示当前目录文件
 - make 编译生成可执行文件, 看看多了几个文件
 - 试试运行 ./btest ./dlc (失败的话可能是IA32库没安装)
 - vi bits.c 或 gedit bits.c

- 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 详细信息
- 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』
- cp mkdir rm mv chmod cat vi
- Linux常用命令

<http://blog.csdn.net/xiaoguaihai/article/details/8705992>

- **README** 实验细节的说明文件，请仔细阅读
- **bits.c** 工作文件，包含一组用于完成指定功能的函数的代码框架，按要求补充完成其函数体代码并“作为实验结果提交”。函数实现要求详细见注释。
- **btest.c** 实验结果正确性测试工具
- **dlc** 判断作为实验结果函数实现是否满足要求。
- **Makefile** 生成btest、fshow、ishow的Make文件。
- **ishow.c** 整型数据表示查看工具 `./ishow`
- **fshow.c** 浮点数据表示查看工具 `./fshow`

- Bits.c函数分类
 - 位操作函数
 - 补码运算函数
 - 浮点数表示函数
- 函数难度分级
 - 1,2,3,4

- 除浮点数函数实现外，只能使用顺序程序结构，禁用if, do, while, for, switch等。
- 有限操作类型，！ ~ & ^ | + << >> 各函数不一样
- 禁用（！ =、==、&&、|| 等组合操作符）
- 常量值范围 0~255
- 禁用强制类型转换
- 禁用整型外的任何其它数据类型
- 禁用定义和宏
- 不得使用函数
- 具体要求可参看bits.c各函数框架的注释


```
int Funct(arg1, arg2, ...) {  
    int var1 = Expr1;    //变量声明必须放在前面  
    ...  
    int varM = ExprM;  
  
    varJ = ExprJ;        //只允许用顺序结构  
    ...  
    varN = ExprN;  
    return ExprR;  
}
```

```
/*  sign - return
```

```
    1 if positive    0 if zero -1 if negative
```

```
Examples:  sign(130) = 1    sign(-23) = -1
```

```
Legal ops:  ! ~ & ^ | + << >>
```

```
Max ops: 10
```

```
Rating: 2 */
```

```
int sign(int x)
{
    return (x>>31) | (!!x);
}
```

- 可以使用循环和条件控制；
- 可以使用整型和无符号整型常量及变量（取值不受[0,255]限制）；
- 不使用任何浮点数据类型、操作及常量。
- 可以使用int和unsigned两种整型数据
- 禁用浮点数据类型、struct、union或数组结构。
- 浮点数函数均使用unsigned型数据表示浮点数据。
- float_abs等函数必须能处理全范围的变量值，包括(NaN)和infinity。

- 1、修改bits.c
- 2、语法检查（否则无法评分！！！！）
 - `$./dlc bits.c` #简单语法检查
 - `$./dlc -e bits.c` #检查操作运算符是否符合需求
 - dlc使用的是开源编译器，能通过gcc编译不一定能通过dlc检查
- 3、编译生成可执行文件
 - `$.make`
 - 修改bits.c必须make，make完成编译，链接，执行文件生成
- 4、正确性检查
 - `$./btest` #检查bits所有函数功能，失败给出测试用
 - `$./btest -f byteNot` #检查单个函数，失败给出测试用例
 - `$./btest -f byteNot -1 0xf -2 1` #规定测试用例检查

- 及时备份bits.c
- 最终提交文件必须能通过dlc, btest检查
 - 未通过检查可能是零分
 - 抄袭零分
- 最终提交文件名
 - CS1201_U201214795_姓名.c
 - 信安 IS 物联网 IT 计算机 CS 卓越班 ZY ACM班 ACM
 - 三个实验做完后，一次性提交给助教