

课程实验报告

课 程 名 称： 计算机系统

实验项目名称： 课程实验二

专 业 班 级： 计科1907

姓 名： 杨杰

学 号： 201908010705

指 导 教 师： 黄丽达、赵欢

完 成 时 间： 2021 年 4 月 16 日

信息科学与工程学院

|  |  |
| --- | --- |
| **实验题目：**LAB2-datalab | |
| **实验目的：**修改bits.c的C语言代码，使其通过所有在不违反任何编码准则的情况下，在btest中进行测试，进一步熟悉整型及浮点数的位级表示，实现常用二进制运算的常用方法。 | |
| **实验环境：**个人电脑、linux发行版本、dlc编译环境 | |
| **实验内容及操作步骤：**  **4.1 实验内容**  (1)替换bits.c中各个函数中的return，实现相应功能，并通过btest测试，具体格式如下：  int Funct(arg1, arg2, …) {  /\* brief description of how your implementation works \*/  int var1 = Expr1;  int varM = ExprM;  varJ = ExprJ;  …  varN = ExprN;  return ExprR;  }  (2)补充函数要求如下：  每一个“Expr”只能使用如下规则：  ①　数字只能使用0到255（0xff），不能使用像0xffffffff这样大的数字  ②　函数参数和局部变量(没有全局变量)  ③　一元运算目：! ~  ④　二元运算目：& ^ | + << >>  下面的操作不被允许：  ①　使用任何控制结构，如if, do, while, for, switch等。  ②　定义或使用任何宏。  ③　在此文件中定义任何其他函数。  ④　调用任何库函数。  ⑤　使用任何其他的操作，如&&, ||, -, or ?:  ⑥　使用任何形式的casting  ⑦　使用除int以外的任何数据类型。这意味着你不能使用数组、结构等。  对于需要你执行浮点运算的问题，编码规则较不严格。允许使用循环和条件控制也可以同时使用int和unsigned。可以使用任意整数和无符号常量。  **4.2 操作步骤**  (1)首先将代码包datalab-handout复制到Ubuntu系统中（复制之前需要安装好vmware-tools）,接下来的实验都是在该文件目录下进行。  (2)补充bits.c中的所有函数，并遵循实验内容中提到的补充规则，另外需要注意编码过程中运算符的合法性和最大操作符数。  (3)实现代码如下：  **函数一：**    **代码解释**：实现x和y的按位与运算，由于有运算符数量的限制，这里可以采用德摩根律进行转化后再求解，即x & y=~~(x&y)=~((~x)|(~y))。  **函数二：**    **代码解释：**首先将所取字节移动到最右边，再和0xff相与使得前面3个字节清零。  **函数三：**    **代码解释：**逻辑右移是左端补0，算术右移是填充符号位，这里直接将x右移得到的是算术右移的结果，这里需要产生一个掩码mask来消除算术右移n的符号位，采用的办法是0相与消除符号位，其余部分和1相与不变。  **函数四：**    **代码解释：**输入一个整型数字，输出该数字二进制表示中有多少个1，可以通过二分法进行查找进行记录，先计算每两位中1的个数，并用对应的2位来进行存储，然后计算每四位中1的个数，用对应的4位进行存储，最后得到16位中1的个数，即x中1的个数。  **函数五：**    **代码解释**：不使用!而实现!功能。第一步~x+1求补码，与x本身进行相或后得到最高有效位，如果x非0那么最高有效位必为1，否则为0。将该数逻辑右移31位后得到要么全0(x为0)要么全1（x非0）的数，然后按位取反再与1相与就可以实现！的操作。  **函数六：**    **代码解释：**32位补码能表示的最小数是0x80000000，这里直接将1左移31位即可。  **函数七：**    **代码解释：**对于正数来说，从第31位到第n-1位必须全是0，因此左移（32-n）位再右移（32-n）位一定和原数相同，如果不相同，一定是因为第31位到第n-1位有至少一个1。对于负数，也是类似的情形，从31位到第n-1位必须全是1，这里定义一个移位数shift,将左移右移之后的结果和原数x通过异或比较，相同则可表示。其中32-n是通过32+(~n)+1计算的。  **函数八：**    **代码解释：**    至于为什么负数时偏移量为2^n-1，可参考《深入理解计算机系统》(原书第2版)P64-P66.  **函数九：**    **代码解释：**取x的相反数，用0-x也就是0+x的补码，所以直接返回~x+1即可。  **函数十：**    **代码解释：**如果x大于0则返回1，否则返回0，因此直接判断符号位，但需要特殊考虑0。  **函数十一：**    **代码解释：**首先对x,y进行符号位的判断，如果x,y异号的话，x>=0（即x符号位为0，y符号位为1）,则说明x>y,返回0，如果x,y同号直接进行相减判断符号即可。  这里为什么要用y-x而不是x-y呢？  又为什么取x的符号位与1相与呢？  返回值为什么是这个表达式呢？    **函数十二：**    **代码解释：**该函数实际上就是要找到最高位1的索引（0~31）。因此可以先将int型数据x右移16位，并进行两次取反操作，如果得到的值为1，则说明x的高16位中存在至少一个1，那么bitsNumber应加上16；如果得到的值为0，则说明高16位中不存在1。然后再将x右移(bitsNumber+8)位，同样进行两次取反操作，如果得到的值为1，则说明在(bitsNumber+8)和(bitsNumber+15)这8位中至少有一个1，那么bitsNumber应加上8；如果得到的值为0，则说明这8位中不存在1。依次类推，继续将x右移(bitsNumber+4)，(bitsNumber+2)，(bitsNumber+1)位，并进行同样的操作即可得到最终结果。  **函数十三：**    **代码解释：**首先判断浮点数是否为nan，如果为nan则返回本身，否则返回-f。判断浮点数是否是nan可以采用f和0x7f80000相与的结果来判断。阶码全1，尾数不全为0为nan；阶码全1，尾数全为0，符号位为0为正无穷；阶码全1，尾数全为0，符号位为1为负无穷。  如果不为nan的话，将符号位取反即可。这里通过异或操作做到，和1异或为取反，和0异或为不变。  **函数十四：**    **代码解释：**将整形转化为无符号浮点数，即求浮点数。先取的符号位，再将剩余部分全部取为正数形式，即absx，即可以得到无符号的数值。然后将有数字的部分直接移动到最高位，记录移动的位数，再将其右移9位（因尾数只要23位即可）。对于阶码部分，e=32-shiftleft，bias为127，exp=e+bias=159-shiftleft，if部分做舍入处理。    **函数十五：**    **代码解释：**将浮点数乘2，对于阶码全0的非规格数，对其尾数乘2即可，即直接左移一位，但因为左移会丢失符号位，所以要提前记录符号位。对于规格化数，直接对其阶码+1即可。对于阶码全1的nan和正无穷以及负无穷，直接返回本身即可（因为无穷乘以2还是无穷）。  **实验结果及分析：**  (1)使用dlc检测bits.c是否有错误    结果显示没有错误，说明代码编写规范。  注：此处有一个warning，上网查阅资料知这是正常的。    (2)使用dlc的-e选项检查各个操作数是否符合要求      经检查，均符合要求。  (3)使用btest检验函数实现代码的功能正确性，先使用make编译生成btest可执行文件（这里用make,对当前目录下所有程序进行编译）    (4)调用btest命令检查bits.c中所有函数功能的正确性，以便下一步查找错误原因，执行./btest bits.c后结果如下：    这是经过调试之后的测试结果，所有功能得到验证。 (5)测试结束后或者每次用btest测试时，需使用make clean删除生成的可执行文件。    (6)测试btest其它功能 ./btest –h：打印出相关提示信息    ./btest –f logicalShift:测试某特定的函数    ./ishow x; ./fshow y:显示整数和浮点数的位级表示    **收获与体会：**  (1)通过该实验进一步熟悉了整型及浮点数的位级表示，实现常用二进制运算的常用方法。  (2)对基本的二进制运算的总结  按位与（&）：参与运算的数字转换为二进制，而后逐位对应进行与运算。  按位或（|）：参与运算的数字转换为二进制，对应位相或。  按位异或（^）: 参与运算的数字转换为二进制，对应位相异或，规则为两位相同为0，不同为1。异或运算能够实现位翻转，高效交换两个变量的值等功能。  按位取反（~）：将一个数按位取反,即~0 = 1,~1 = 0。  逻辑非（！）：将真值结果取非，如!5=0,!0=1。  (3)算术移位和逻辑移位的区别  左移：x<<y表示将x左移y位，左边的位全部丢弃，在右边全部补0。  右移：x>>y表示将x右移y位，右边的位全部丢弃，对于逻辑移位，左边补0，对于算术移位，左边填充符号位。  (4)实验过程中遇到不会做的题目，通过上网查阅资料，回顾课本知识点，体会到了C语言中使用二进制运算也可以实现很多功能，如两个数的大小比较，不用循环统计二进制数据中1的个数，获取某一字节等等，对整数和浮点数的位级表示等相关知识也有一定的提升。 | |
| 实  验成绩 |  |

实验报告撰写说明

1．实验题目和目的

请从实验指导资料中获取。

2．实验步骤和内容

包括：

（1）本次实验的要求；

（2）源程序清单或者主要伪代码；

（3）预期结果；

（4）上机执行或调试结果：包括原始数据、相应的运行结果和必要的说明（截图）；

3．实验体会

调试中遇到的问题及解决办法；若最终未完成调试，要试着分析原因；调试程序的心得与体会；对课程及实验的建议等。