

# Lab1 报告

---

- 姓名：邵震哲
- 班级：1620204
- 学号：162020130
- 报告阶段：lab1
- 完成日期：2021.4.17
- 本次实验，我完成了所有内容。

## 目录

---

### Lab1 报告

#### 目录

1. Nuaa\_question1
2. Nuaa\_question2
3. Nuaa\_question3
4. Nuaa\_question4
5. Nuaa\_question5
6. Nuaa\_question6
7. 最终结果
8. 备注

## 1. Nuaa\_question1

---

- 思路  
先右移一位把最后一位舍弃，再左移一位，让最后一位补零
- 代码

```
int nuaa_question1(int x) {  
    return x>>1<<1;  
}
```

- 测试截图 (dlc btest)

```

shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 0 operators
dlc:bits.c:200:nuaa_question3: 0 operators
dlc:bits.c:211:nuaa_question4: 0 operators
dlc:bits.c:225:nuaa_question5: 0 operators
dlc:bits.c:240:nuaa_question6: 0 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question1
Score  Rating  Errors  Function
  1      1      0      nuaa_question1
Total points: 1/1
shaozhenzhe@Debian:~/lab1-handout$

```

## 2. Nuaa\_question2

- 思路

比较是否相同一般用异或，相同的话对应位置全部转化为0，再把比较的部分移到最低位，由于n是字节，所有右移的长度为  $8*n$ ，即  $n \ll 3$ 。右移完成后需要看最低位是否全为零，高位的也最好能置为0，因此和 0xff 进行与运算，如果低位全是0，那么结果还是0，如果低位不是0，结果也不是0，经过两次逻辑非可以转化成1。

- 代码

```

int nuaa_question2(int x, int y, int n) {
    return !!(((x^y)>>(n<<3))&0xff);
}

```

- 测试截图 (dlc btest)

```

shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 6 operators
dlc:bits.c:200:nuaa_question3: 0 operators
dlc:bits.c:211:nuaa_question4: 0 operators
dlc:bits.c:225:nuaa_question5: 0 operators
dlc:bits.c:240:nuaa_question6: 0 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question2
Score  Rating  Errors  Function
  2      2      0      nuaa_question2
Total points: 2/2
shaozhenzhe@Debian:~/lab1-handout$

```

## 3. Nuaa\_question3

- 思路

对两个数都进行两次逻辑非，转化为0或1，再进行与运算返回结果

- 代码

```
int nuaa_question3(int x, int y) {
    return (!!x & !!y);
}
```

- 测试截图 (dlc btest)

```
shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 6 operators
dlc:bits.c:200:nuaa_question3: 5 operators
dlc:bits.c:211:nuaa_question4: 0 operators
dlc:bits.c:225:nuaa_question5: 0 operators
dlc:bits.c:240:nuaa_question6: 0 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question3
Score  Rating  Errors  Function
   3       3       0      nuaa_question3
Total points: 3/3
shaozhenzhe@Debian:~/lab1-handout$
```

## 4. Nuaa\_question4

- 思路

把  $x$  的前  $n$  位移到末尾，也就是把  $x$  的前  $n$  位提取出来，把  $x$  左移  $n$  位，加上提取出来的部分就是结果。

提取前  $n$  位，需要把前  $n$  位移到低位去，也就是右移  $32-n$  位， $32-n$  可以表示为  $n$  取反加 1，最后保留低 5 位，高位清零（因为  $0 \leq n \leq 31$ ），因此和  $0x1f$  进行与运算。

右移时要注意补位的是符号位，因此要把右移多出来的  $32-n$  的符号位置 0。低  $n$  位保留，高  $32-n$  位置 0，很容易想到要和  $00\dots111$  进行与运算，只需要  $0xffffffff$  左移  $n$  位，再进行取反即可， $0xffffffff$  可以通过  $0x1 \ll 31 \gg 31$  得到。

最后  $x \ll n$  加上提取出来的前  $n$  位就是结果。

- 代码

```
int nuaa_question4(int x, int n) {
    int y = x >> ((~n + 1) & 0x1f);
    int p = 0x1 << 31 >> 31;
    p = ~(p << n);
    y = y & p;
    return (x << n) + y;
}
```

- 测试截图 (dlc btest)

```

shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 6 operators
dlc:bits.c:200:nuaa_question3: 5 operators
dlc:bits.c:215:nuaa_question4: 11 operators
dlc:bits.c:233:nuaa_question5: 2 operators
dlc:bits.c:248:nuaa_question6: 0 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question4
Score  Rating  Errors  Function
   3      3      0      nuaa_question4
Total points: 3/3
shaozhenzhe@Debian:~/lab1-handout$

```

## 5. Nuaa\_question5

- 思路

得到绝对值只需要把符号位置0即可，其余部分不变，因此和 0x7fffffff 进行与运算即可。NaN 是阶码全为1，尾数部分不为0的情况。那么拿绝对值和 0x7f800000 进行比较，如果比它大，说明是 NaN，返回自身即可。

- 代码

```

unsigned nuaa_question5(unsigned uf) {
    int x = uf & 0x7fffffff;
    if (x > 0x7f800000)
        return uf;
    else
        return x;
}

```

- 测试截图 (dlc btest)

```

shaozhenzhe@Debian: ~/lab1-handout
shaozhenzhe@Debian:~/lab1-handout$ make clean
rm -f *.o btest fshow ishow *~
shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 6 operators
dlc:bits.c:200:nuaa_question3: 5 operators
dlc:bits.c:215:nuaa_question4: 11 operators
dlc:bits.c:233:nuaa_question5: 2 operators
dlc:bits.c:248:nuaa_question6: 0 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question5
Score  Rating  Errors  Function
   2      2      0      nuaa_question5
Total points: 2/2
shaozhenzhe@Debian:~/lab1-handout$

```

## 6. Nuaa\_question6

- 思路

取出阶码和尾数，取尾数时需要把省略的1加上（加在第24位上），取的操作是移动到低位，把其他位置0。接下来分情况讨论，阶码大于等于31时，超出了int的表示范围，返回 0x80000000，如果阶码小于0，说明是个小数，转化为int就变成0。

其他都是正常转化的数据，根据符号位分类，如果阶码大于23，说明尾数部分不够表示，需要左移  $x-23$  位，其他情况正常右移  $23-x$  即可。如果是负数，那么乘-1返回即可。

- 代码

```
int nuaa_question6(unsigned uf) {
    int x=((uf>>23)&0xff)-127;//取阶码
    int y=(uf&0x007fffff)+0x00800000;
    int p=y<<(x-23);
    int q=y>>(23-x);
    if(x>=31)
        return 0x80000000;
    if(x<0)
        return 0;
    if((uf&0x80000000)==0){
        if(x>23)
            return p;
        else
            return q;
    }
    else{
        if(x>23)
            return (p*(-1));
        else
            return (q*(-1));
    }
}
```

- 测试截图 (dlc btest)

```
shaozhenzhe@Debian:~/lab1-handout$ ./dlc bits.c
shaozhenzhe@Debian:~/lab1-handout$ ./dlc -e bits.c
dlc:bits.c:178:nuaa_question1: 2 operators
dlc:bits.c:191:nuaa_question2: 6 operators
dlc:bits.c:200:nuaa_question3: 5 operators
dlc:bits.c:215:nuaa_question4: 11 operators
dlc:bits.c:233:nuaa_question5: 2 operators
dlc:bits.c:267:nuaa_question6: 19 operators
shaozhenzhe@Debian:~/lab1-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
shaozhenzhe@Debian:~/lab1-handout$ ./btest -f nuaa_question6
Score  Rating  Errors  Function
4      4       0      nuaa_question6
Total points: 4/4
shaozhenzhe@Debian:~/lab1-handout$
```

## 7. 最终结果

- ./driver.pl 截图

```
shaozhenzhe@Debian: ~/lab1-handout
3. Running './dlc -Z' to identify operator count violations.

4. Compiling and running './btest -g -r 2' to determine performance score.
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
    if (arg_test_range[1] < 1)
        ~~~~~^~

5. Running './dlc -e' to get operator count of each function.

Correctness Results      Perf Results
Points  Rating  Errors  Points  Ops    Puzzle
1       1       0       2       2     nuaa_question1
2       2       0       2       6     nuaa_question2
3       3       0       2       5     nuaa_question3
3       3       0       2      11     nuaa_question4
2       2       0       2       2     nuaa_question5
4       4       0       2      19     nuaa_question6

Score = 27/27 [15/15 Corr + 12/12 Perf] (45 total operators)
shaozhenzhe@Debian:~/lab1-handout$
```

- 挑战教授截图（必须是你的学号）

2	9	5	11	2	16		8	162020225
2	7	4	11	2	19		8	162030302
2	6	5	11	2	19		8	162020130
2	7	5	10	2	20		7	xiangwandamowang
2	8	5	11	2	18		7	162030213

## 8. 备注

无