

# lab1

2024/3/22

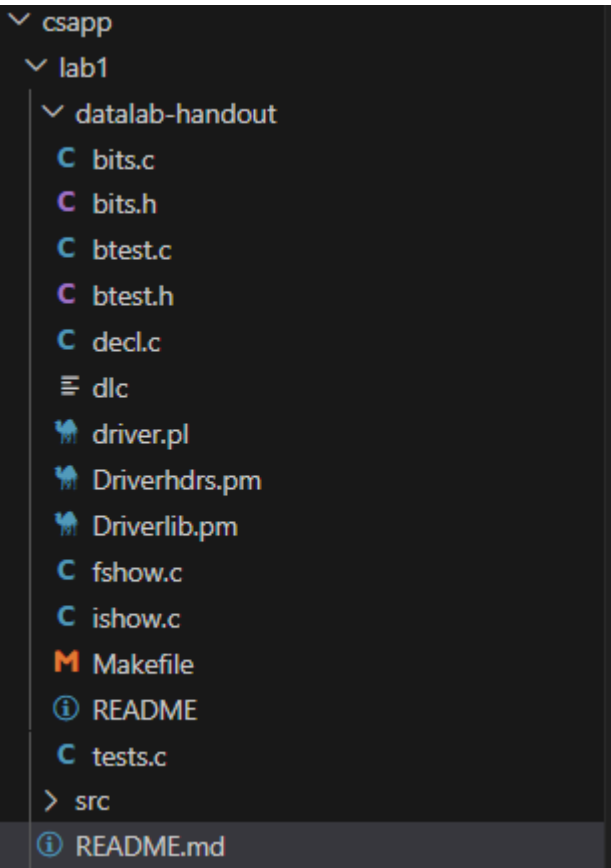
PB22111702 李岱峰

## 一.实验准备

linux环境准备就绪，配置如下：

```
ubuntu@VM8378-fengli-ics:~/csapp/lab1$ uname -a
Linux VM8378-fengli-ics 6.5.11-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-8 (2024-01-30T12:27Z) x86_64 x86_64 x86_64 GNU/Linux
ubuntu@VM8378-fengli-ics:~/csapp/lab1$
```

从网站上下载得到代码包，解压在文件夹里，文件组织如下：



以上，准备完成。

## 二.任务分析

Name	Description	Rating	Max ops
bitXor(x,y)	$x \oplus y$ using only $\&$ and $\sim$ .	1	14
tmin()	Smallest two's complement integer	1	4
isTmax(x)	True only if $x$ is largest two's comp. integer.	1	10
allOddBits(x)	True only if all odd-numbered bits in $x$ set to 1.	2	12
negate(x)	Return $-x$ with using $-$ operator.	2	5
isAsciiDigit(x)	True if $0x30 \leq x \leq 0x39$ .	3	15
conditional	Same as $x ? y : z$	3	16
isLessOrEqual(x, y)	True if $x \leq y$ , false otherwise	3	24
logicalNeg(x)	Compute $\neg x$ without using $\neg$ operator.	4	12
howManyBits(x)	Min. no. of bits to represent $x$ in two's comp.	4	90
floatScale2(uf)	Return bit-level equiv. of $2 \cdot f$ for f.p. arg. $f$ .	4	30
floatFloat2Int(uf)	Return bit-level equiv. of $(int) f$ for f.p. arg. $f$ .	4	30
floatPower2(x)	Return bit-level equiv. of $2 \cdot 0^x$ for integer $x$ .	4	30

- 1. 我需要使用bit操作，仅能使用"! ~ & ^ | + << >>
- 2. 我只能使用顺序语言，除特殊外不能跳转语句(loop)，条件判句(if)
- 3. 不使用stdio中的函数。
- 4. 完成所有操作

三.实验过程

1. 完成实验代码bits.c，实验原码请参考文件夹，这里只展示我认为很有收获的操作：

- 1. 三目运算符  $x ? y : z$ ，第一次见到这样的bits操作，使用减法将该减去的结果(y,z)减去即可。

```
int conditional(int x, int y, int z) { //x==0?z:y    x!=0    -> 0xffffffff|y
x==0 -> 0x00000000 ->~x|z
    int a=!!(x^0); //a=0 if x=0 else a =1
    int b=~a+1;     //b=0xffffffff if x=0 else b=0x00000000
    int c=~(y&~b)+1;
    int d=~(z&b)+1;
    return y+z+c+d;
}
```

- 2. 查位数的bits级操作，二分法从高位开始找

```
int howManyBits(int x) {
    int b16,b8,b4,b2,b1,b0;
    int flag=x>>31;
    x=(flag&x)|(~flag&x);    //x>=0不变 ,x<0按位取反
    b16=(!!(x>>16))<<4;    //如果高16位不为0,则我们让b16=16
    x>>=b16;                //如果高16位不为0 则右移动16位 来看高16位的情况
    b8=(!!(x>>8))<<3;
    x>>=b8;
    b4=(!!(x>>4))<<2;
```

```

    x >>= b4;
    b2=(!!(x>>2))<<1;
    x >>= b2;
    b1=!!(x>>1);
    x>>= b1;
    b0=x;
    return b0+b1+b2+b4+b8+b16+1;//加1是因为符号位,其他的是从低开始1的位数,从后向前
找,不重不漏
}

```

## 2. 进行编译检查, 包括符号是否正确和运行结果是否正确

### 1. 准备工作中的错误:

```

ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ sudo ./dlc bits.c
sudo: ./dlc: 找不到命令
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ ./dlc bits.c
bash: ./dlc: 权限不够

```

权限不足, 使用chmod指令赋予权限后成功

```

ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ make btest
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
bits.c: In function 'isAsciiDigit':
bits.c:214:10: warning: suggest parentheses around operand of '!' or change '&' to '&&' or '!' to '~' [-Wparentheses]
 214 |     return !((x+a)>>31)&((b+x)>>31); //x>a && x<=b -> !(x-a)>>31 && !(b-x)>>31
      |            ^~~~~~
In file included from btest.c:16:
/usr/include/stdio.h:27:10: fatal error: bits/libc-header-start.h: 没有那个文件或目录
 27 | #include <bits/libc-header-start.h>
      |

```

查明是缺少必要的库, 在linux中更新lib库解决

## 2. 符号使用:

### 1. 第一次编译出错

```

ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ chmod 777 dlc
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ ./dlc bits.c
dlc:bits.c:146:bitXor: Illegal operator (|)
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$

```

查明是在bitXor中使用了不该使用的操作符所致

## 2. 第二次编译正确

```
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ ./dlc bits.c
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$
```

## 3. 运行结果

### 1. 第一次运行

```
1      1      0      isTmax
2      2      0      allOddBits
2      2      0      negate
3      3      0      isAsciiDigit
3      3      0      conditional
3      3      0      isLessOrEqual
ERROR: Test logicalNeg(-2147483648[0x80000000]) failed...
...Gives 1[0x1]. Should be 0[0x0]
4      4      0      howManyBits
ERROR: Test floatScale2(-2147483648[0x80000000]) failed...
...Gives 0[0x0]. Should be -2147483648[0x80000000]
4      4      0      floatFloat2Int
ERROR: Test floatPower2 failed.
      Timed out after 10 secs (probably infinite loop)
Total points: 24/36
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$
```

三个函数错误，错误原因如图。其中第三个问题无法解决，就修改了运行时间，加参数-T 20

### 2. 修好后运行

```
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$ ./btest -T 20
Score  Rating  Errors  Function
1      1      0      bitXor
1      1      0      tmin
1      1      0      isTmax
2      2      0      allOddBits
2      2      0      negate
3      3      0      isAsciiDigit
3      3      0      conditional
3      3      0      isLessOrEqual
4      4      0      logicalNeg
4      4      0      howManyBits
4      4      0      floatScale2
4      4      0      floatFloat2Int
4      4      0      floatPower2
Total points: 36/36
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$
```

### 3. 最终测试

需要注意的是，我将测试最后一个函数的运行时间提高到了20，否则无法在响应时间内通过测试。

```
Correctness Results      Perf Results
Points  Rating  Errors  Points  Ops    Puzzle
1       1       0       2       8     bitXor
1       1       0       2       1     tmin
1       1       0       2       8     isTmax
2       2       0       2       9     allOddBits
2       2       0       2       2     negate
3       3       0       2      10     isAsciiDigit
3       3       0       2      15     conditional
3       3       0       2       5     isLessOrEqual
4       4       0       2       6     logicalNeg
4       4       0       2      36     howManyBits
4       4       0       2      17     floatScale2
4       4       0       2      18     floatFloat2Int
4       4       0       2      10     floatPower2

Score = 62/62 [36/36 Corr + 26/26 Perf] (145 total operators)
ubuntu@VM8378-fengli-ics:~/csapp/lab1/datalab-handout$
```

拿到了全部的分數。

### 四.总结

bits实验很有趣，可以获得一些前所未有的编程体验。同时也丰富了如makefile、linux chmod的使用经验。