

# 시스템 프로그래밍

4주차: 실습 프로젝트 – Datalab (2)



T A

강범우, IT/BT 701호

[qjadr0630@hanyang.ac.kr](mailto:qjadr0630@hanyang.ac.kr)

**수업 관련 질문: Blackboard 토론 및 질의응답 게시판**



# Datalab 과제 - 2

- 과제 수행 환경
  - VirtualBox – Ubuntu18.04
  - 필수 자료: datalab2-handout.tar
    - 각자 본인의 gitlab 프로젝트 "Week4" 폴더
- 결과 제출 방법 및 기한
  - **bits.c** 소스 코드만 수정하여 gitlab 프로젝트에 commit
  - 10월 6일(화) 23:59까지 commit된 것만 인정  
-> 영상에서는 9월 30일로 잘못 공지



# 과제 수행 방법

- 필수 자료 확인
  - **datalab2-handout.tar**
    - \$ tar xvf datalab-handout.tar
    - 임의 프로그램 사용 가능
    - 과제 결과를 직접 테스트할 수 있는 소스와 프로그램 포함
- “**bits.c**” 파일 만 수정하여 과제 수행
  - 다른파일은 절대 수정하지 말 것.



## 과제 내용 (bits.c)

함수	설명	가능 연산자	점수	연산자 개수 제한
logicalShift(x, n)	변수 x를 n만큼 오른쪽으로 '논리' 시프트	! ~ & ^   + << >>	3	20
remainderPower2(x,n)	x를 $2^n$ 으로 나눈 나머지 ( $x\%(2^n)$ )	! ~ & ^   + << >>	1	20
specialBits()	0xFFCA3FFF을 반환	! ~ & ^   + << >>	1	3
floatAbsVal(uf)	uf의 절대값을 bit-level에서 반환	모든 연산자	2	10
floatScale4(uf)	부동소수점 uf에 대해 $4*uf$ 를 bit-level에서 반환	모든 연산자	4	30



## 과제 내용 (bits.c)

함수	설명	가능 연산자	점수	연산자 개수 제한
floatNegate(uf)	부동소수점 uf에 대해 -uf를 bit-level에서 반환	모든 연산자	2	10
satMul2(x)	x를 2만큼 곱하고, 만약 overflow가 발생할 경우 Tmin 혹은 Tmax를 반환	! ~ & ^   + << >>	3	20
thirdBits()	LSB부터 시작해서 3번째 비트 마다 1로 set 된 word를 반환 (Word는 32-bit)	! ~ & ^   + << >>	1	8
signMag2TwosComp(x)	주어진 x를 부호절대값 방식 으로 가정하고 해당 값을 2의 보수로 나타내어 반환	~ & ^   + << >>	4	15



## 과제 제출 주의 사항

- 컴파일 여부를 반드시 확인한 후에 제출 (make 명령 실행)
    - 제출한 bits.c 파일에서 컴파일 오류가 발생하는 경우에는 과제 점수가 부여되지 않음.
  - 과제 점수 확인 프로그램 실행 권장
    - btest, dlc, driver.pl 프로그램을 실행
- 본인 점수 미리 확인



# 과제 수행 - 따라 하기

- GitLab 본인 프로젝트에서 파일 확인 후 압축 풀기
  - “datalab-handout.tar”
  - \$ tar xvf datalab-handout.ar
- 과제 컴파일
  - \$ make

```
next@archi-01:~$ tar xvf datalab-handout.tar
next@archi-01:~$ cd datalab-handout/
next@archi-01:~/datalab-handout$ ls
bits.c  btest.c  decl.c  Driverhdrs.pm  driver.pl  ishow.c  README
bits.h  btest.h  dlc      Driverlib.pm   fshow.c   Makefile  tests.c
next@archi-01:~/datalab-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
next@archi-01:~/datalab-handout$
```





# 과제 수행 - 따라 하기

- 정답 확인 (btest, 오답인 경우)

```
next@archi-01:~/datalab-handout$ ./btest
Score  Rating  Errors  Function
ERROR: Test bitXor(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test allOddBits(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test byteSwap(-2147483648[0x80000000],0[0x0],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test replaceByte(-2147483648[0x80000000],0[0x0],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test rotateLeft(-2147483648[0x80000000],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test isTmin(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 1[0x1]
ERROR: Test fitsBits(-2147483648[0x80000000],1[0x1]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test sign(-2147483647[0x80000001]) failed...
...Gives 2[0x2]. Should be -1[0xffffffff]
ERROR: Test isPositive(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test addOK(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test float_f2i(0[0x0]) failed...
...Gives 2[0x2]. Should be 0[0x0]
Total points: 0/26
next@archi-01:~/datalab-handout$
```



# 과제 수행 - 따라 하기

- 정답 확인 (btest, 정답인 경우)

```
next@archi-01:~/datalab-handout$ ./btest
Score  Rating  Errors  Function
1      1       0      bitXor
2      2       0      allOddBits
2      2       0      byteSwap
3      3       0      replaceByte
3      3       0      rotateLeft
1      1       0      isTmin
2      2       0      fitsBits
2      2       0      sign
3      3       0      isPositive
3      3       0      addOK
4      4       0      float_f2i
Total points: 26/26
next@archi-01:~/datalab-handout$
```

- 프로그래밍 제약 확인 (dlc, 오류가 있는 경우)

```
next@archi-01:~/datalab-handout$ ./dlc bits.c
dlc:bits.c:176:bitXor: Illegal operator (==)
dlc:bits.c:177:bitXor: Illegal operator (^)
dlc:bits.c:178:bitXor: Illegal if
next@archi-01:~/datalab-handout$
```



# 과제 수행 - 따라 하기

- 총 점수 확인 (driver.pl) – 정확성과 성능

```
next@archi-01:~/datalab-handout$ ./driver.pl
1. Running './dlc -z' to identify coding rules violations.

2. Compiling and running './btest -g' to determine correctness score.
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c

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

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       7     bitXor
2       2       0       2       7     allOddBits
2       2       0       2      17     byteSwap
3       3       0       2       6     replaceByte
3       3       0       2      16     rotateLeft
1       1       0       2       5     isTmin
2       2       0       2       7     fitsBits
2       2       0       2       4     sign
3       3       0       2       4     isPositive
3       3       0       2       9     addOK
4       4       0       2      13     float_f2i

Score = 48/48 [26/26 Corr + 22/22 Perf] (95 total operators)
next@archi-01:~/datalab-handout$
```

