

# System Programming

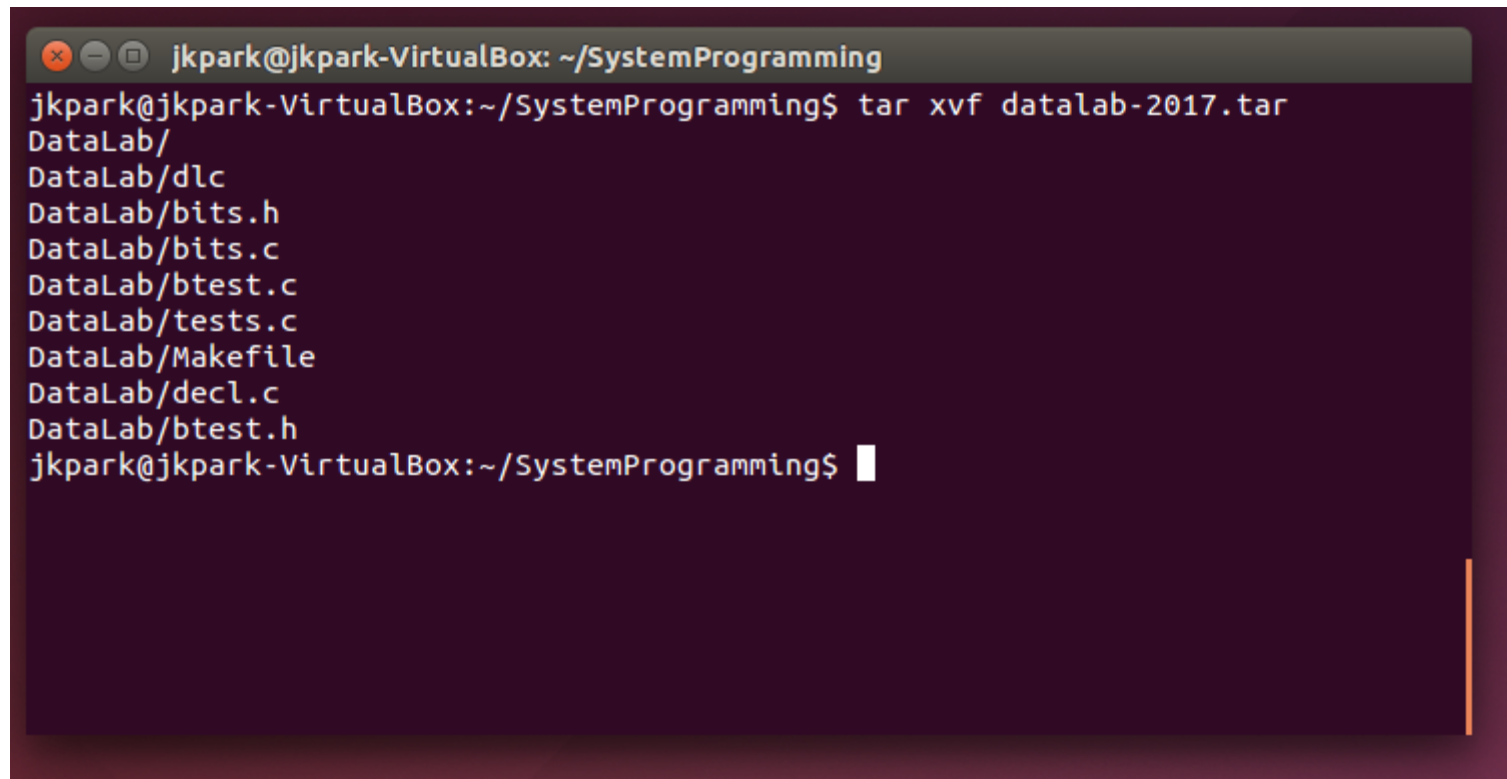
Hyun-Wook Jin  
System Software Laboratory  
Department of Computer Science & Engineering  
Konkuk University  
jinh@konkuk.ac.kr

# 과제 설명

- 목적
  - Bit Level 표현과 조작에 숙달
- 평가
  - 총 14개의 문제
  - 각각의 문제는 난이도별 점수
  - 총점 40점
- 기한
  - 2017년 10월 12일 23시 59분
  - 기한 꼭 지키기

# DATALAB

- Datalab-2017.tar 형태의 압축파일 제공
  - # tar -xvf datalab-2017.tar



```
jkpark@jkpark-VirtualBox: ~/SystemProgramming
jkpark@jkpark-VirtualBox:~/SystemProgramming$ tar xvf datalab-2017.tar
DataLab/
DataLab/dlc
DataLab/bits.h
DataLab/bits.c
DataLab/btest.c
DataLab/tests.c
DataLab/Makefile
DataLab/decl.c
DataLab/btest.h
jkpark@jkpark-VirtualBox:~/SystemProgramming$
```

# 과제를 위해 사용할 파일

- `tests.c`
  - 답을 비교하게 될 비교답안지
- `bits.c`
  - 문제를 풀게 될 문제지
- `dlc`
  - 제약조건을 지켰는지 확인하기 위한 실행 파일
- `btest`
  - 문제를 제대로 풀었는지 확인해줄 검사기



# 문제지(bits.c) 작성 요구사항

- 아래와 같이 본인의 학번, 이름 반드시 넣을 것
  - 제출 파일에 학번, 이름이 누락된 경우 채점 불가
  - 채점 불가 시, 해당 과제 0점 처리할 것!

```
*  
* STEP 1: Fill in the following struct with your identifying info.  
*/  
team_struct team =  
{  
    /* Team name: Replace with either:  
    Your login ID if working as a one person team  
    or, ID1+ID2 where ID1 is the login ID of the first team member  
    and ID2 is the login ID of the second team member */  
    "20XX12345",  
    /* Student name 1: Replace with the full name of first team member */  
    "Hong Gil-Dong",  
    /* Login ID 1: Replace with the login ID of first team member */  
    "Hong Gil-Dong",  
  
    /* The following should only be changed if there are two team members */  
    /* Student name 2: Full name of the second team member */  
    "",  
    /* Login ID 2: Login ID of the second team member */  
    ""  
};
```

# 문제지(bits.c) 문제 확인

```
* isZero - returns 1 if x == 0, and 0 otherwise
* Examples: isZero(5) = 0, isZero(0) = 1
* Legal ops: ! ~ & ^ | + << >>
* Max ops: 2
* Rating: 1
*/
int isZero(int x) {
    return 2;
}
```

- 함수에 대한 간단한 설명
- Examples : 몇 가지 예
- Legal ops : 사용할 수 있는 연산
- Max ops : 최대 연산 갯수
- Rating : 배당 점수

# 비교답안지(test.c) 추가 확인

```
#include <limits.h>
int test_isZero(int x) {
    return x == 0;
}
```

- bits.c 에서 확인할 수 있는 함수에 대해 좀 더 익숙한 표현으로 표현되어있는 식을 확인할 수 있다.
- Bits.c 에 비어있는 수식을 완성하여 주어진 연산들을 통해 tests.c에 해당하는 함수와 같은 결과를 얻어내야 한다.



# 제약조건(dlc) 검사하기

- `./dlc bits.c` 성공적으로 수행한 경우
  - 아래와 같이 Compilation Successful 메시지 보이거나,

```
root@in9-VirtualBox:~/test# ./dlc bits.c
/usr/include/stdc-predef.h:1: Warning: Non-includable file <command-line> included from includable file /usr/include/stdc-predef.h.

Compilation Successful (1 warning)
root@in9-VirtualBox:~/test#
```

- 메시지 보이지 않음

```
unj a@unj a- desktop: ~/data lab$ ./dlc bits.c
unj a@unj a- desktop: ~/data lab$
```

- dlc test가 성공적으로 수행되었다고, 정답은 아님!





# 제약조건(dlc) 검사하기

- Illegal constant 제약에 걸렸을 경우

```
Compilation Failed: 1 error, 1 warning
root@in9-VirtualBox:~/test# ./dlc bits.c
/usr/include/stdc-predef.h:1: Warning: Non-includable file <command-line> inclu
ed from includable file /usr/include/stdc-predef.h
dlc:bits.c:175:isZero: Illegal constant (0xFFFFFFFF) (only 0x0 - 0xff allowed)

Compilation Successful (1 warning)
root@in9-VirtualBox:~/test#
```

```
* Examples: isZero(
* Legal ops: ! ~ & '
* Max ops: 2
* Rating: 1
*/
int isZero(int x) {
    return 0xFFFFFFFF;
}
```



# 제약조건(dlc) 검사하기

- Illegal operator 제약에 걸렸을 경우

```
Compilation successful (1 warning)
root@in9-VirtualBox:~/test# ./dlc bits.c
/usr/include/stdc-predef.h:1: Warning: Non-includable file <comma>
ed from includable file /usr/include/stdc-predef.h.
dlc:bits.c:175:isZero: Illegal operator (-)
bits.c:175: 2 cannot dereference non-pointer type

Compilation Failed: 1 error, 1 warning
root@in9-VirtualBox:~/test#
```

```
/*
 * Examples: isZero(5) = 0, isZero(0) = 1
 * Legal ops: ! ~ & ^ | + << >>
 * Max ops: 2
 * Rating: 1
 */
int isZero(int x) {
    return -x;
}
```

# 제약조건(dlc) 검사하기

- 제약된 operator 수를 초과하였을 경우

```
Compilation Successful (1 warning)
root@in9-VirtualBox:~/test# ./dlc bits.c
/usr/include/stdc-predef.h:1: Warning: Non-includable file <command-line> included from includable file /usr/include/stdc-predef.h.
dlc:bits.c:176:isZero: Warning: 4 operators exceeds max of 2

Compilation Successful (1 warning)
root@in9-VirtualBox:~/test#
```

```
* Examples: isZero(5)
* Legal ops: ! ~ & '
* Max ops: 2
* Rating: 1
*/
int isZero(int x) {
    return !!!!x;
```

# 제약조건(dlc) 검사하기

- 제약조건(dlc)가 실행이 되지 않는 경우
  - # chmod 777 dlc

```
sp@sp: ~/sysprog$ ./dlc bits.c
bash: ./dlc: 허가 거부
sp@sp: ~/sysprog$ chmod 777 dlc
```

# 정답 맞춰보기

- `make btest`
  - `bits.c`에 입력된 정보를 기반으로 검산기능을 가진 실행파일 `btest`를 생성
- `./btest -a`
  - 모든 문제에 대한 검산결과를 출력
- `./btest -f [funcName]`
  - 특정함수에 대한 결과출력
- `./btest -g`
  - 점수득점상태를 한눈에 확인 가능
- `| grep score`를 통해서 검산과정을 제외한 점수 득점 결과만 출력 가능
  - `ex> # ./btest -a | grep score`



# 이상적인 정답 화면

- ./dlc bits.c 연산 결과 (제약 조건 검사)

```
root@in9-VirtualBox:~/test# ./dlc bits.c
/usr/include/stdc-predef.h:1: Warning: Non-includable file <command-line> included from includable file /usr/include/stdc-predef.h.

Compilation Successful (1 warning)
root@in9-VirtualBox:~/test#
```

- make btest

```
root@in9-VirtualBox:~/test# make btest
gcc -c libmath.c
gcc -O -Wall -static libmath.o -o btest bits.c btest.c decl.c tests.c
root@in9-VirtualBox:~/test#
```

- ./btest -a | grep score 연산결과  
– (-g 또는 grep score로도 확인 가능)

```
sp@sp: ~/sysprog$ ./btest -a | grep score
Test isZero score: 1.00/1.00
Test bitAnd score: 0.00/1.00
Test minusOne score: 0.00/1.00
Test bitOr score: 0.00/1.00
Test tmin score: 0.00/1.00
Test bitXor score: 0.00/2.00
Test evenBits score: 0.00/2.00
Test copyLSB score: 0.00/2.00
Test getByte score: 0.00/2.00
Test isNotEqual score: 0.00/2.00
Test reverseBytes score: 0.00/3.00
Test conditional score: 0.00/3.00
Test bitMask score: 0.00/3.00
Test isNonNegative score: 0.00/3.00
Test sum8 score: 0.00/3.00
Test sum2c score: 0.00/4.00
Test bang score: 0.00/4.00
Test abs score: 0.00/4.00
Overall correctness score: 1.00/42.00
```



# 과제 목록

함수이름	과제내용	Use Ops	Max ops	Rating
isZero	x가 0이면 1 리턴	! ~ & ^   + << >>	2	2
bitOr	(x   y) 를 ~ 와 & 만 이용해서 표현	~ &	8	2
bitAnd	(x & y) 를 ~ 와   만 이용해서 표현	~	8	2
minusOne	-1 값을 표현	! ~ & ^   + << >>	2	2
negate	입력된 양수의 x를 음수로 변환하여 리턴	! ~ & ^   + << >>	5	2
bitXor	^ 을 ~ 와 & 만 이용해서 표현	~ &	14	3
isPositive	x가 0보다 큰지 검사	! ~ & ^   + << >>	8	3
getBytes	x로부터 n 바이트만큼 추출	! ~ & ^   + << >>	6	3
isNotEqual	x, y 값을 비교하여 서로 다르면 1, 같으면 0 리턴	! ~ & ^   + << >>	6	3
evenBits	짝수 번째 비트를 1로 설정	! ~ & ^   + << >>	8	3



# 과제 목록

함수이름	과제내용	Use Ops	Max ops	Rating
reverseBytes	x 의 Byte 순서를 뒤바꿈	! ~ & ^   + << >>	35	3
conditional	x ? y : z	! ~ & ^   + << >>	16	4
isGreater	x가 y보다 큰지 검사	! ~ & ^   + << >>	24	4
bang	!x 를 '!'를 사용하지 않고 표현	! ~ & ^   + << >>	12	4





# 주의할 점

- `stdio.h` 를 포함시키면 에러가 남
  - `printf`문 등의 standard 함수 사용을 금지
- 설명된 파일 이외의 파일은 수정 금지
- 제출 파일
  - “make handin TEAM=학번” 명령을 수행
    - ex ) # make handin TEAM=20XX12345
  - 위 명령에서 문제 있을 경우 “cp bits.c 학번-ver-bits.c”
    - ex ) # cp bits.c 20XX12345-1-bits.c
  - 학번은 bits.c 에 입력한 학번과 같아야 함
  - 명령 수행 후 생성 된 “학번-버전-bits.c” 파일을 제출
    - 해당 파일로 자동 테스트를 돌려 채점
    - 이 때 파일명 잘못으로 인한 문제발생 시, 책임은 제출자에게 있음
  - 풀이과정을 pdf 형식으로 제출
    - 파일명은 “학번-이름-datalab”
    - ex) “20xx12345-홍길동-datalab.pdf”

# 주의할 점

- 파일 제출 후, 수정된 파일을 다시 제출하려 한다면
  - “make handin TEAM=학번 VERSION=2” 명령 수행
  - 또는 “cp bits.c 학번-ver-bits.c” 명령 수행
  - 매회 제출시마다 VERSION 의 숫자를 증가시켜 제출
  - 평가는 마지막 버전만 이루어짐
- 제출방법
  - E-Mail 제출 : [2017sslabor@gmail.com](mailto:2017sslabor@gmail.com)
  - 제목은 반드시 [2017\_시스템프로그래밍#1] 으로 시작
  - 제목에 학번과 이름을 표기
    - [2017\_시스템프로그래밍#1]20XX12345\_홍길동
  - 위에 주어진 제출 방법을 따르지 않는 과제는 채점되지 않으며, 미제출 처리된다.

# Q&A

- 과제는 수업게시판에서 9/28(목) 18:00부터 9/29(금) 18:00 까지 접근 가능
- 질의응답은 수업게시판을 이용
- 과제 기한 연장은 없음