Programming Assignment #2: Numbers and Bit Manipulations

Prof. Jae W. Lee (jaewlee@snu.ac.kr)
Department of Computer Science and Engineering
Seoul National University

TA: Yunho Jin, Jeonghun Gong

Goal of this project

Understand and be familiarized with bit representation

- You are to implement several simple functions.
  - e.g., bit or, negation, byte swapping.
- Limited types and number of bit operators are allowed.

Experimental setup

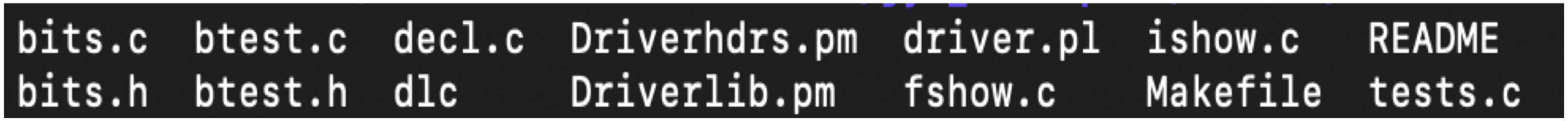
You will use GCC on Linux: Two options

- Option 1: Use Virtual Machine from PA1
  - Download update.sh from eTL and execute it.

- Option 2: Use your own Linux box
  - Install GCC and gcc-multilib
  - If there’s any dependency issue, please notify TA.

Experimental setup

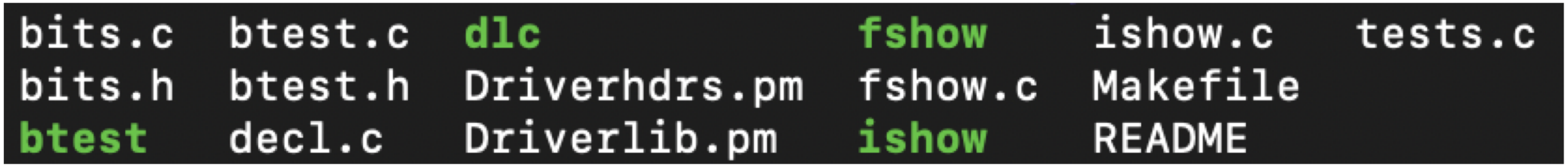
Dowload and unzip PA2.zip file from etl.



```
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
```

make and add execution permission to ‘dlc’

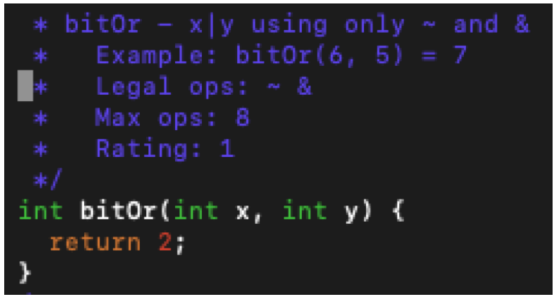
- make
- $> chmod +x ./dlc



```
bits.c  btest.c  dlc            fshow    ishow.c   tests.c
bits.h  btest.h  Driverhdrs.pm  fshow.c  Makefile
btest   decl.c   Driverlib.pm   ishow    README
```

Problems

Fill in the functions in bits.c



```
 * bitOr - x|y using only ~ and &
 *   Example: bitOr(6, 5) = 7
 *   Legal ops: ~ &
 *   Max ops: 8
 *   Rating: 1
 */
int bitOr(int x, int y) {
  return 2;
}
```

There are some rules to make it more interesting

- See INTEGER CODING RULES in Line 26 of bits.c
- FLOATING POINT CODING RULES are in Line 91 of bits.c
- Each problem has its own additional rules

Helper programs

`./dlc bit.c`

- Check whether you followed the rules correctly.

`./btest`

- Test your functions for correctness.
- `make btest` to test your new bits.c code.
- recompile `btest` whenever bits.c is changed

`./ishow & ./fshow`

- Given hex representation, show int/unsigned int/float value.
- Given int/unsigned int/float value, show hex representation.

Submission

Write-up

- Briefly describe your implementation (no more than 5 pages).
- Filename: [student_id].pdf (example: 2019-12345.pdf)
- Please submit it in **PDF** format. Other formats are not accepted.

Compress your source code and write-up into a single zip file.

- Compress bits.c and your report.
- Filename should be [student_id].zip (example: 2019-12345.zip).
- Please submit it in **ZIP** format. Other formats are not accepted.

Submission deadline: by 23:59 on October 14, 2019