

CSED211: Lab. 1

Introduction & DataLab

조현욱 (gusdnr9779@postech.ac.kr)

박용곤 (nanimdo@postech.ac.kr)

POSTECH

2023.09.11

Table of Contents

- Lab introduction
- DataLab
- Homework

Lab Introduction

Communication with TAs

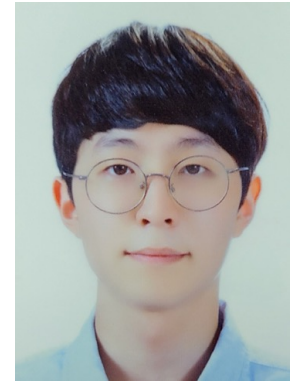
- To contact TAs, please use the 'Q&A' board, or 'LAB Q&A 게시판' board on PLMS
- Use of e-mails is limited to privacy-sensitive matters
 - E.g., excused absence, HW/lab scores...



조성준



백승훈



조현욱



김보석



박용곤



조승혁

About Lab Session

- Lab session will be held
 - TJ Park library 506 (청암학술정보관 506호)
- Evaluation
 - Quiz: 10%
 - Assignment: 90%
 - Lab report: 40%
 - Source code: 50%
- Schedule and percentile of each lab will be announced soon
 - You have a lab session next week (Datalab2)
 - Datalab1(2) will have 20%(40%) weight compared to other labs

About Lab Session (cont.)

- We will take **a brief quiz** (no pre-lab report)
- You have to submit **a zip file** with **your source file** & **the lab report** for a due date
 - Submit one zip file to the assignment board
 - Name of source code: **[student_#]_[your_name].c** (20231234_홍길동.c)
 - Name of the report: **[student_#]_[your_name].pdf** (20231234_홍길동.pdf)
 - Name of the zip file: **Lab[lab_#]_[student_#]_name.zip** (Lab1_20231234_홍길동.zip)
 - **Incorrect format of files would be regarded as 0 point**
- Contents of a final report: TBA
 - 1. What you did in the lab (method)
 - 2. Simple explanation of your source code (how you solved the problems)

```
unzip Lab1_20231234_홍길동.zip
Archive: Lab1_20231234_홍길동.zip
  creating: Lab1_20231234_홍길동/
  inflating: Lab1_20231234_홍길동/20231234_홍길동.pdf
  extracting: Lab1_20231234_홍길동/20231234_홍길동.c

~ (0.027s)
ls Lab1_20231234_홍길동 /
20231234_홍길동.c      20231234_홍길동.pdf
```

Lab Evaluation Policy

- Assignment late penalty
 - 10% penalty for every 24-hour
 - **No score** for submissions **over 72-hour**
 - E.g., When the deadline is 09/18 23:59 (midnight),
 - 09/19 00:00 ~ 09/19 11:59: 90% of the score
 - 09/20 00:00 ~ 09/20 11:59: 80%, 09/21 00:00 ~ 09/21 11:59: 70%,
 - Over 09/21 11:59: 0% (**zero point**)
- Attendance
 - **10% of total course score penalty** for one absence (except the Q&A session)
 - Being late for more than **15 minutes** is considered an absence
 - To check the attendance, students should sit in their **assigned sits**

Programming Account Registration

- Lab session and homework are processed on **Linux machine**
- Homework should work on a Linux machine
 - If it doesn't work on a Linux machine, it would be regarded as 0 point
- How to register a programming account
- How to install terminal programs such as PUTTY, and XShell and access the Linux server via the programs

Programming Account Registration (cont.)

- Login and enter the hemos (<https://hemos.postech.ac.kr/>)
- Click Server/File -> Programming Account



Programming Account Registration (cont.)

- Click the request and fill in the blanks

프로그래밍 실습 계정 신청 (Programming Account)

C 언어 등의 프로그래밍 및 기타 리눅스 실습을 수행할 수 있는 서버 환경에 대한 계정 신청 서비스입니다.

- 이용 기간
 - 계정 생성 후 1년간 유효
 - 졸업 시에는 이용 불가
- 접속 방법
 - 접속 주소 : programming2.postech.ac.kr
 - 접속 포트 : SSH (2022), FTP (21)
 - 대학 외부에서 접속 시에는 POSTECH VPN(vpn.postech.ac.kr) 접속 후 이용
- 유의 사항
 - 계정 생성 시, 패스워드는 'id + !@#' 형태로 생성되며 최초 접속 후 본인이 반드시 변경해야 함 (eg, HEMOS-ID: gildong 일 경우, 최초 패스워드는 gildong!@#)

Request

List

프로그래밍 실습 계정 신청 (Programming Account)

신청자 정보 (Applicant)

Date	Time
2023/09/04	19:13:44
Name	HEMOS ID
조현욱	gusdnr9779
Position *	Department *
신분을 선택하세요.	컴퓨터공학과
Office Tel.	Cell Phone

신청 내용 (Details)

Usage *	Course (강의명)
사용 용도를 선택하세요.	
Request Details	

Submit

Terminal Programs (Putty and XShell)

- Through the terminal programs, students can access the Linux server
- IP: programming2.postech.ac.kr
- Port: 2022
- Putty
- Xshell

Install Putty

1. Search putty in Google.

Google

putty

전체 이미지 쇼핑 동영상 지도 더보기

검색결과 약 202,000,000개 (0.37초)

도움말: 이 검색을 영어 검색결과로 제한합니다. 언어별 필터링에 관해 자세히 알아보기

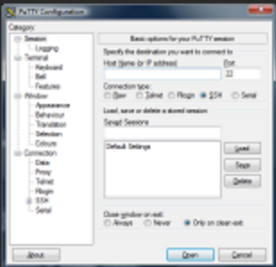
PuTTY

https://www.putty.org

Download PuTTY - a free SSH and telnet client for Windows

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with ...

2. Click “Download PuTTY”.



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with ...

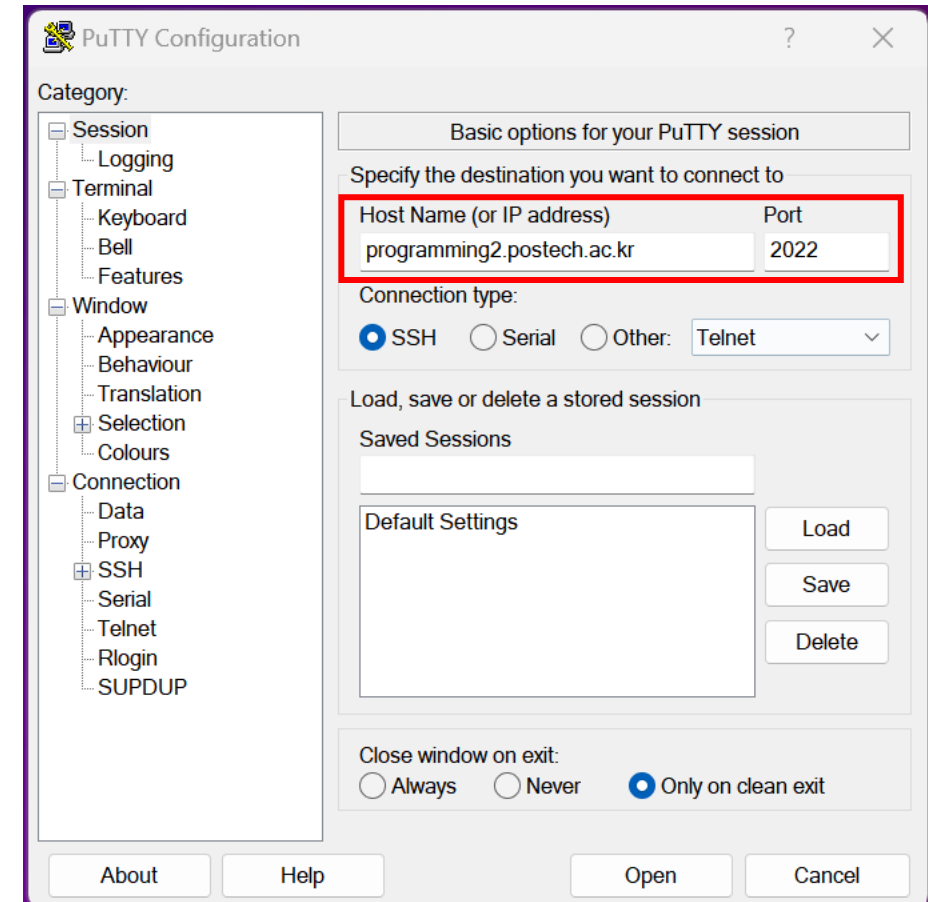
Download PuTTY

3. Download putty.exe corresponding to your OS.

Alternative binary files		
The installer packages above will provide versions of all of these (except PuTTYtel and PuTTYtel64). (Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry .)		
putty.exe (the SSH and Telnet client itself)		
64-bit x86:	putty.exe	(signature)
64-bit Arm:	putty.exe	(signature)
32-bit x86:	putty.exe	(signature)

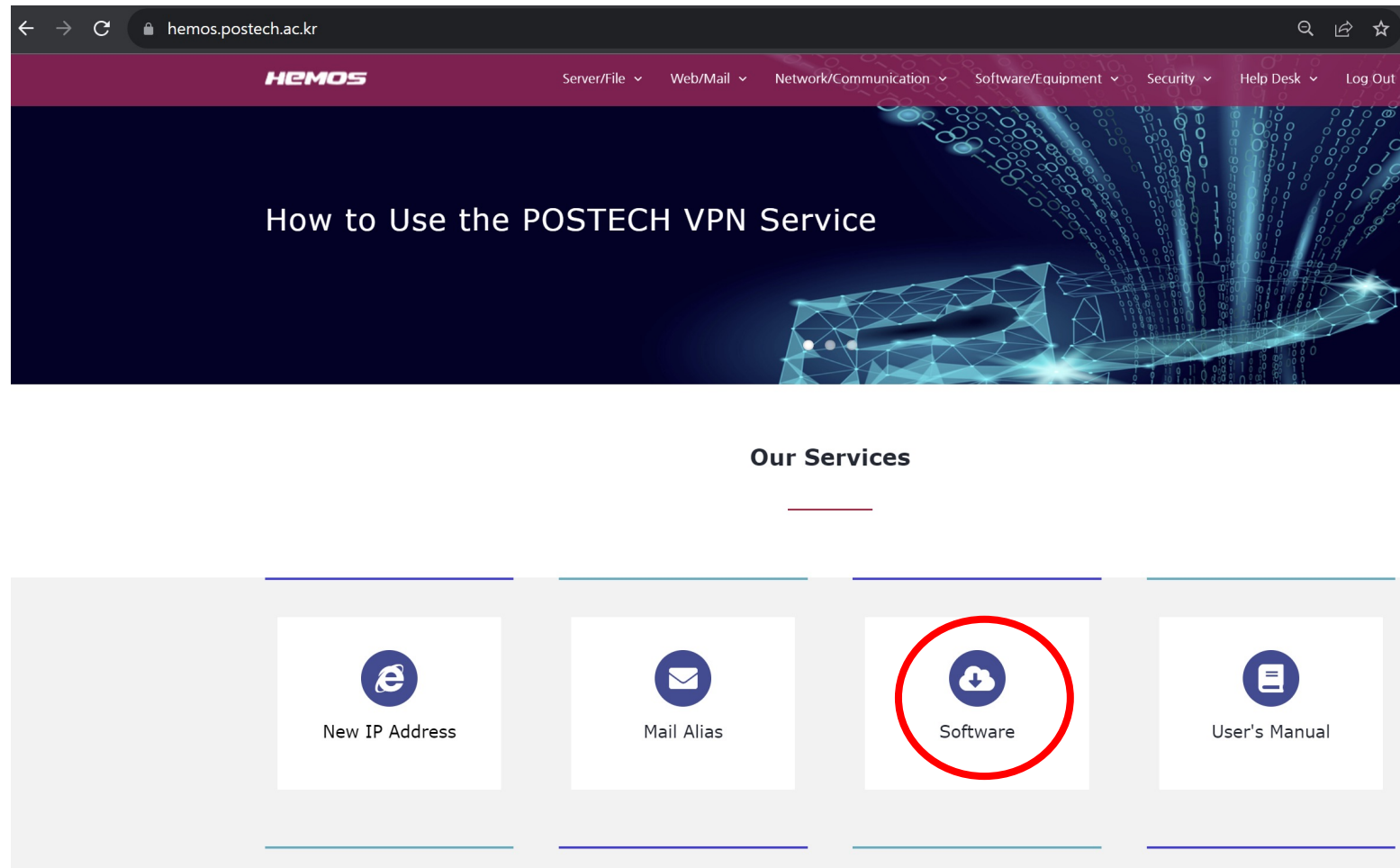
Executing Putty

- Execute downloaded “putty.exe”
- Fill the IP at Host Name and Port using SSH then you can access the Linux server



Install XShell

- Access hemos and click the software



Install XShell

- Click TOOL(Utility) and download Xmanager Enterprise

소프트웨어 배포 (Software Library)

The screenshot displays the 'Software Library' interface. On the left is a vertical sidebar with category buttons: ERP, GRAPHIC, LANGUAGES, LIBRARY, OA, OS, PROMOTION VIDEO, SIEMENS, STATISTICS_AI, TOOL(UTILITY), VACCINE, and WEB. The 'TOOL(UTILITY)' button is highlighted. The main content area features a search bar with 'xmanager' entered. Below the search bar are two dropdown menus: 'Order By:' set to 'Publish Date' and 'Order:' set to 'Descending Order'. A breadcrumb trail shows 'Home > Search Result For xmanager'. The search results list 'Xmanager Enterprise' as the top item, with a file size of 166.35 MB and 2986 downloads. This result is enclosed in a red rectangular box.

ERP
GRAPHIC
LANGUAGES
LIBRARY
OA
OS
PROMOTION VIDEO
SIEMENS
STATISTICS_AI
TOOL(UTILITY)
VACCINE
WEB

xmanager

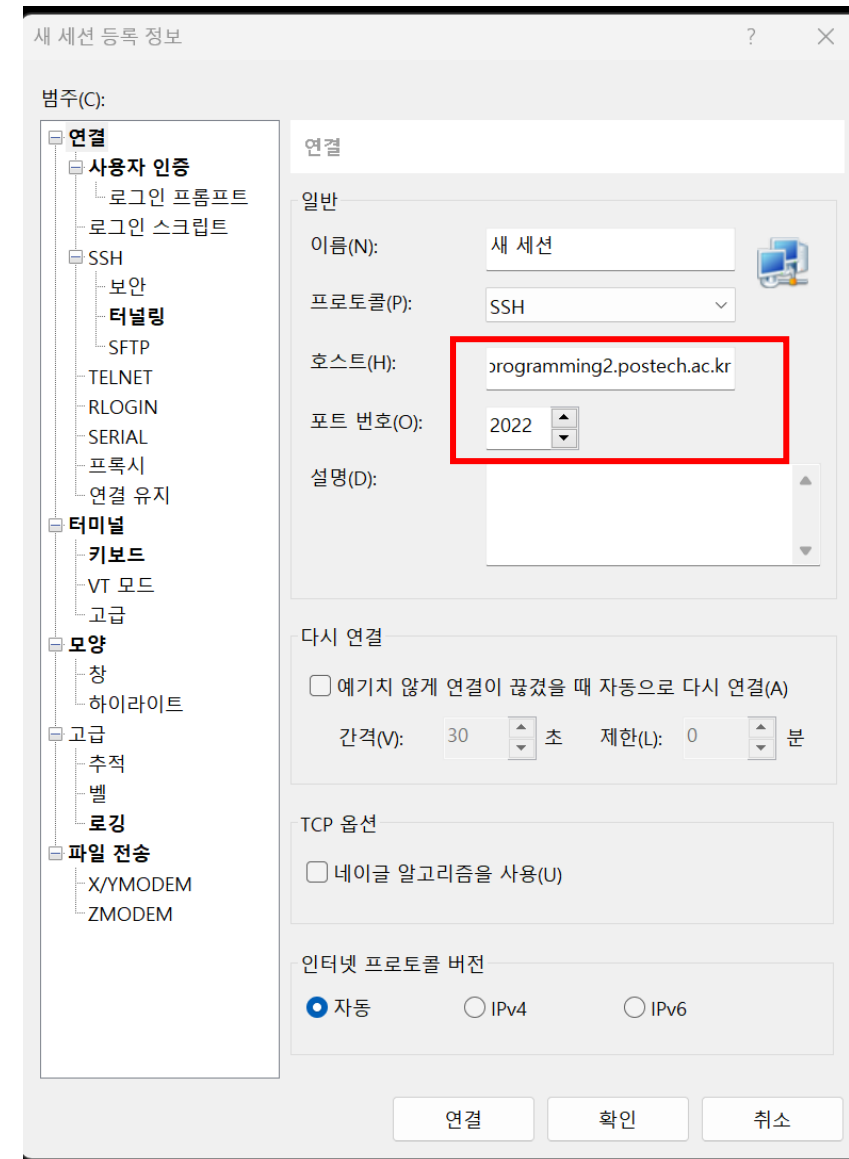
Order By: Publish Date Order: Descending Order

Home > Search Result For xmanager

Xmanager Enterprise
166.35 MB 2986 downloads

Executing XShell

- Press Alt + N
- Fill the IP (Host) and Port



DataLab

Linux Commands Basic

- **ls**: list information of files in the directory
 - Option -a (all files), l (detail info)
- **cd dir_location**: change directory
 - cd /subdir
- **cp original new**: copy file
 - cp a.out new.out
- **mv old new**: move file
 - mv a.out new.out
- **mkdir dir_name**: make directory
 - mkdir homework1

Linux Commands Basic (cont.)

- **rm file**: remove file
 - **rm a.out**
- **./executable_file**: execute a file
 - **./a.out**
- **gcc**: compile a source file to executable file
 - **gcc test.c -> a.out / gcc -o output test.c -> output**
- **chmod option file**: change mode of a file
 - **chmod +x a.out**

Linux Commands Basic (cont.)

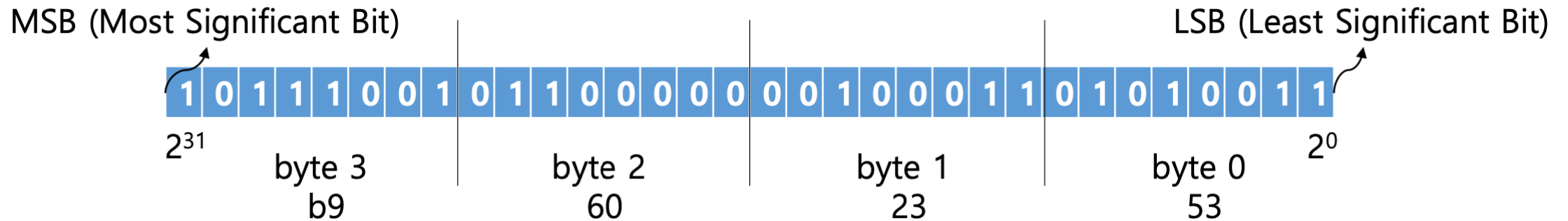
- **more file / cat file**: see contents of a file
 - **cat(or more) a.out**
- **tar option file**: compress & decompress file
 - **tar -xvf compressed.tar** (for decompression)
 - **tar -cvf compressed.tar datafile**
- Commands in this slide will help you to do your assignments
- For more information, google 'linux commands'
 - Ex) <https://www.digitalocean.com/community/tutorials/linux-commands>

Vim Command

- Use vim when open for writing and reading your program on Linux
- Usage: **vi (m) 'your_file'** -> code -> (esc):wq
- For the vim commands and configurations, please refer to the below websites
 - <https://www.cs.colostate.edu/helpdocs/vi.html>
 - <http://vimconfig.com/>

Bit and Byte

- Most systems use 4 bytes to represent an integer data type
- 1 byte = 8 bits
- Example
 - Int a = 0xb9602353



Bitwise Operation

- A bitwise operation operates on one or more bit patterns or binary numerals *at the level of their individual bit*
- Bitwise operators (\sim , $\&$, $|$, \wedge)
- Shift operators (\ll , \gg)
- Bitwise assignment operators ($\&=$, $|=$, $\wedge=$, $\ll=$, $\gg=$)

& Operator (And)

- Bitwise AND operator
- It works on the bits of the operands
- *Do not confuse with* logical AND “&&”

bit a	bit b	a & b (a AND b)
0	0	0
0	1	0
1	0	0
1	1	1

- $$\begin{array}{r} 11001010 \\ \& \\ 10011000 \\ \hline = 10001000 \end{array}$$

~ Operator (Not)

- Bitwise NOT operator (Complement operator)
- Invert the bits for every bit 1 the result is bit 0, and conversely for every bit 0 we have a bit 1
- *Do not confuse with* logical negation “!”
- $\sim 1011 = 0100$

bit a	$\sim a$ (complement of a)
0	1
1	0

| Operator (Or)

- Bitwise OR operator
- It works on the bits of the operands
- *Do not confuse with* logical OR “||”

bit a	bit b	a b (a OR b)
0	0	0
0	1	1
1	0	1
1	1	1

- 11001010
|
10011000
= 11011010

^ Operator (XOR)

- Bitwise XOR (exclusive OR) operator
- The result is zero when two bits are same

- $$\begin{array}{c} 11001010 \\ \wedge \end{array}$$

- $$\begin{array}{c} 10011000 \\ = 01010010 \end{array}$$

bit a	bit b	a ^ b (a XOR b)
0	0	0
0	1	1
1	0	1
1	1	0

<<, >> Operator (Shift)

- <<: Left shift
 - $001100 \ll 1 \rightarrow 011000$ ($\times 2^1$)
 - $110110 \ll 3 \rightarrow 110000$ ($\times 2^3$)
- >>: Right shift
 - $001100 \gg 1 \rightarrow 000110$ ($/2^1$)
 - $110110 \gg 3 \rightarrow 000110$ ($/2^3$)
- Logical shift: the blanks will be filled by **0s**
- Arithmetic shift: the blanks will be filled with the **sign bit**

Logical/Arithmetic Shift

- Left shift is a logical shift
- Right shift depends on shift type
- For example,
 - Int $x = -8$ (1111 1000), $x \gg 2 = ?$
 - If it is a logical shift, 0011 1110
 - If it is an arithmetic shift, 1111 1110 ($:-2$)
- When shifting an unsigned value, \gg operator in C is a logical shift
- When shifting a signed value, \gg operator is an arithmetic shift

Bitwise Assignment Operators

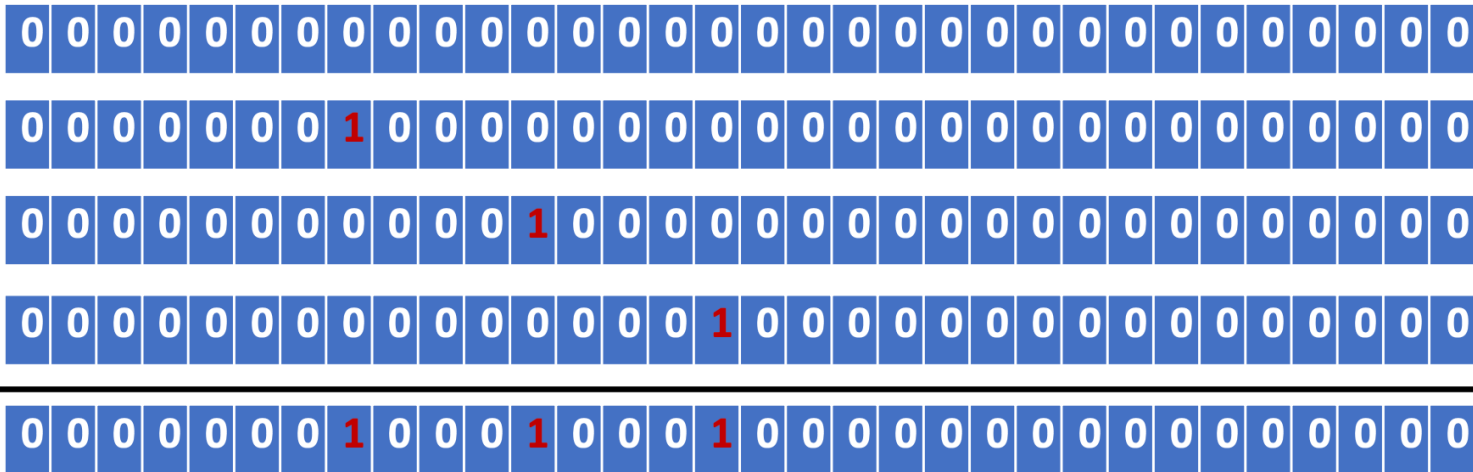
$a \&= b$	\leftrightarrow	$a = a \& b$
$a = b$	\leftrightarrow	$a = a b$
$a \wedge= b$	\leftrightarrow	$a = a \wedge b$
$a \ll= b$	\leftrightarrow	$a = a \ll b$
$a \gg= b$	\leftrightarrow	$a = a \gg b$

- Example

- $x \ll= 2$
- $y |= (1 \ll 24)$
- $a \&= b \gg 3$

Bitwise Operation Example

- Change the values of bit 24, bit 20, bit 16 to 1
 - $x = 0x0;$
 - $x |= (1<<24) | (1<<20) | (1<<16);$



Quiz

Homework

Lab Homework 1

- Due: 09/18 23:59 (midnight)
- Upload a zip file which contains your source file and report
 - Explain your answer in the report
 - File name format (again): [student_#]_[name].c / .pdf, Lab[lab_#]_[student_#]_[name].zip
- Please refer to “writeup_lab1” and “README” for further description

Homework Instruction

- Use the minimum number of operators as you can
- You are allowed to use only the following:
 - 1. Integer constants 0 through 255(0xFF)
 - 2. Function arguments and local variables
 - 3. Unary integer operations: !, ~
 - 4. Binary integer operations: &, ^, |, +, <<, >>

Homework Instruction (cont.)

- You are expressly forbidden to:
 - 1. Use any control constructs: if, do, while, for, switch
 - 2. Define or use any macros
 - 3. Call any functions
 - 4. Use any other operations: &&, ||, -, or ? :
 - 5. Use any data type other than int (cannot use arrays, structs, or unions)

Homework: Problem 1

- bitNor – $\sim(x|y)$ using only \sim and $\&$
- Example: `bitNor(0x6,0x5) = 0xFFFFFFFF8`
- Legal ops: \sim , $\&$
- Max ops: 8

```
int bitNor(int x, int y) {  
    // to be implemented  
}
```

Homework: Problem 2

- isZero – return 1 if $x == 0$, and 0, otherwise
- Examples: $\text{isZero}(5) = 0$, $\text{isZero}(0) = 1$
- Legal ops: $!$, \sim , $\&$, \wedge , $|$, $+$, \ll , \gg
- Max ops: 2

```
int isZero(int x) {  
    // to be implemented  
}
```

Homework: Problem 3

- addOK - Determine if we can compute $x + y$ without overflow
- Example: $\text{addOK}(0x80000000, 0x80000000) = 0$,
 $\text{addOK}(0x80000000, 0x70000000) = 1$
- Legal ops: `!, ~, &, ^, |, +, <<, >>`
- Max ops: 20

```
int addOK(int x, int y) {  
    // to be implemented  
}
```

Homework: Problem 4

- `absVal` – absolute value of `x`
- Example: `absVal(-1) = 1`
- You may assume $-T_{\max} \leq x \leq T_{\max}$
- Legal ops: `!`, `~`, `&`, `^`, `|`, `+`, `<<`, `>>`
- Max ops: 10

```
int absVal(int x) {  
    // to be implemented  
}
```


Homework: Problem 5

- logicalShift - shift x to the right by n, using a logical shift
- Can assume that $0 \leq n \leq 31$
- Examples: `logicalShift(0x87654321,4) = 0x08765432`
- Legal ops: `~, &, ^, |, +, <<, >>`
- Max ops: 20

```
int logicalShift(int x, int n) {  
    // to be implemented  
}
```

How to Do?

- Download “bits.c” file and implement each function
- Please read “README” file before starting your homework
- Follow the rules
 - Use only legal operations, and follow the definition of the function
- Start homework as early as possible

Q & A