4190.308: Computer Architecture Lab. I

Jaehoon Shim
Ikjoon Son
Seongyeop Jeong
(snucsl.ta@gmail.com)

Systems Software & Architecture Lab.

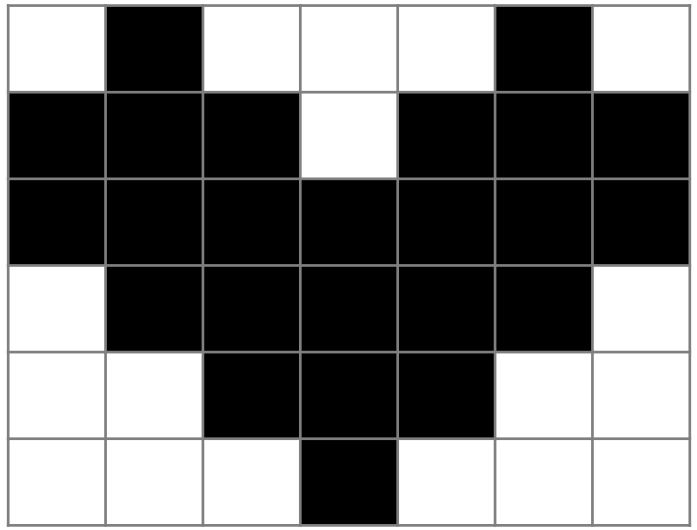
Seoul National University

Fall 2021

Run-length Encoding

What is Run-length Encoding?

- A form of lossless data compression in which runs of data are stored as single data value and count
  - Runs of data: sequence in which the same data value occurs in many consecutive data elements



W1 B1 W3 B1 W1

B3 W1 B3

B7

W1 B5 W1

W2 B3 W2

W3 B1 W3

Simplified Run-length Encoding

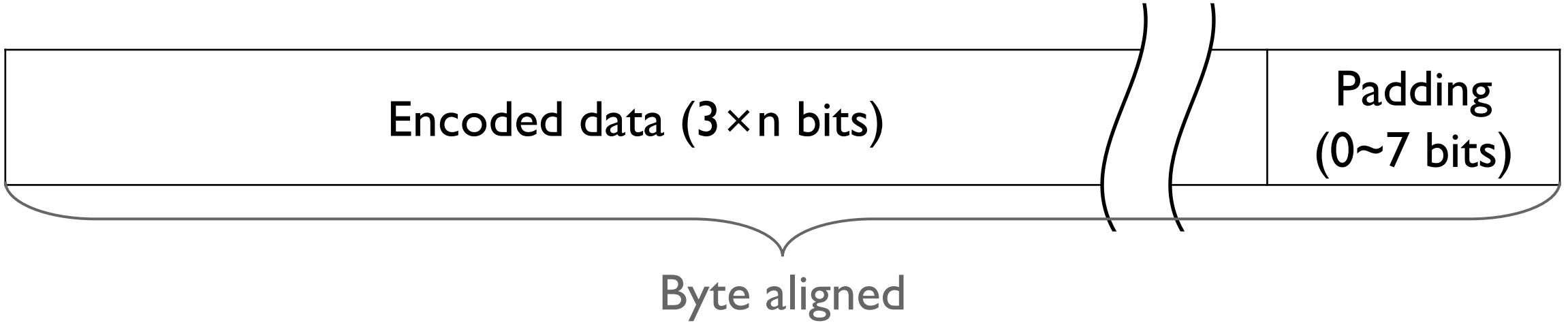
- Consider the input is a sequence of binary digits
  - Just count runs that consist of all 0's or all 1's
- We repeatedly store the length of the run with all 0's and then with all 1's
  - No need to store data value
- We allocate three bits to represent the length of each run

Steps for Simplified Run-length Encoding

1. Find a run with all 0's in the input. Then, emit the three bits that correspond to the length
2. Find a run with all 1's in the input. Then, emit the three bits that correspond to the length
3. Repeat step 1 and 2 until the end of the input

Simplified Run-length Encoding Format

- You should write the output in the following format
- Output consists of encoded data and a padding
- Padding is used to make the final output aligned to bytes



Encoding Example (1)

- Input stream: "hi"
- Actual values to represent the given string are
  - In hexadecimal: 68 69
  - In binary: 01101000 01101001

Encoding Example (1)

1. Find a run with all 0's and count the length
   - Count the number of 0's that appear continuously
   - Write the length of the run to output stream

Input → 01101000 01101001

Encoded Output → 001

Encoding Example (1)

2. Find a run with all 1's and count the length
   - Count the number of 1's that appear continuously
   - Write the length of the run to output stream

Input → 011 01000 01101001

Encoded Output → 001 010

Encoding Example (1)

3. Repeat once

Input → 01101000 01101001

Encoded Output → 001 010 001 001

Encoding Example (1)

4. Repeat twice

Input → 01101000 01101001

Encoded Output → 001 010 001 001 100 010

Encoding Example (1)

5. Repeat three times

Input → 01101000 01101001

Encoded Output → 001 010 001 001 100 010 001 001

Encoding Example (1)

6. Repeat four times

Input → 01101000 01101001

Encoded Output → 001 010 001 001 100 010 001 001 010 001

Encoding Example (1)

7. Add padding bits(0's) to align in byte and return output size in # of bytes

Input → 01101000 01101001

Encoded Output → 00101000 10011000 10001001 01000100

Padding

Encoding Example (2)

- Input stream: 00000000 11110000
- When length of the run is longer than 7, it should be represented as a series of full runs

Input → 00000000 11110000

Encoded Output → 111

Encoding Example (2)

- Input stream: 00000000 11110000
- When length of the run is longer than 7, it should be represented as a series of full runs

Input → 00000000 11110000

Encoded Output → 111 000

Encoding Example (2)

- Input stream: 00000000 11110000
- When length of the run is longer than 7, it should be represented as a series of full runs

Input → 00000000 11110000

Encoded Output → 111 000 001

Encoding Example (2)

- Input stream: 00000000 11110000
- When length of the run is longer than 7, it should be represented as a series of full runs

Input → 00000000 11110000

Encoded Output → 111 000 001 100 100 0

(0xE0 0xC8)

Padding

Encoding Example (3)

- Input stream:
- Special case when the length of input is 0
- You don't have to do something → Just return 0

Decoding Example

- Encoded input stream: 11100000 11001000
- Check 3 bits and convert to run of data

Encoded Input → 111 000 001 100 100 0

Decoded Output → 0000000

Decoding Example

- Encoded input stream: 11100000 11001000
- Check 3 bits and convert to run of data

Encoded Input → 111 000 001 100 100 0

Decoded Output → 0000000 (skip)

Decoding Example

- Encoded input stream: 11100000 11001000
- Check 3 bits and convert to run of data

Encoded Input → 111 000 001 100 100 0

Decoded Output → 0000000 0

Decoding Example

- Encoded input stream: 11100000 11001000
- Check 3 bits and convert to run of data

Encoded Input → 111 000 001 100 100 0

Decoded Output → 0000000 0 1111 0000
(0x00 0xF0)

Specification

- All you need to do is to write two functions given in pa1.c
  - encode(), decode()
- *int encode(const char* const src, const int srclen,*
  char const dst, const int dstlen);*
  { /* fill this function */}
  - *src* points to the memory address of the input data
  - *srclen* is the length of input data (in bytes)
  - *dst* points to the memory address for encoded result
  - *dstlen* is the length of allocated space for result
  - It returns the length of the output (in bytes)
    - If the length of output is bigger than *dstlen*, return -1
      (In this case, contents of the output is ignored)
    - If *srclen* is zero, return zero

Specification

- *int decode(const char* const src, const int srclen,*
  char const dst, const int dstlen);*

  { /* fill this function */}
  - *src* points to the memory address of the input encoded data
  - *srclen* is the length of input data (in bytes)
  - *dst* points to the memory address for decoded result
  - *dstlen* is the length of allocated space for result
  - It returns the length of the output (in bytes)
    - If the length of output is bigger than *dstlen*, return -1
      (In this case, contents of the output is ignored)
    - If *srclen* is zero, return zero

Restrictions

- Contents of the memory outside of the allocated buffer should not be corrupted
- You are not allowed to use any array
- You are not allowed to use any library functions
- Your solution should finish within a reasonable time

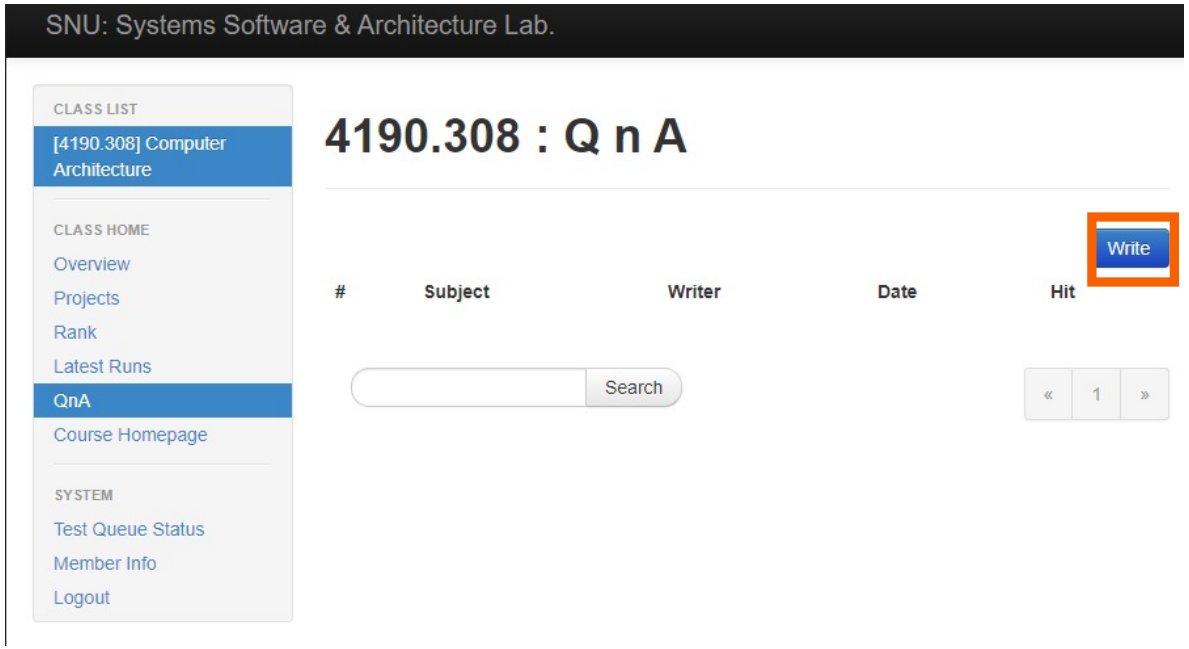
Submission

- Due: 11:59PM, September 26 (Sunday)
  - 25% of the credit will be deducted for every single day delay

- Only submit the pa1.c file to the submission server
  - You don't have to write a report in this assignment

Slip Days

- You can use up to 4 slip days during this semester
  - To use slip days, please post how many slip days you want to use on QnA board
- We highly recommend to save slip days for next projects!

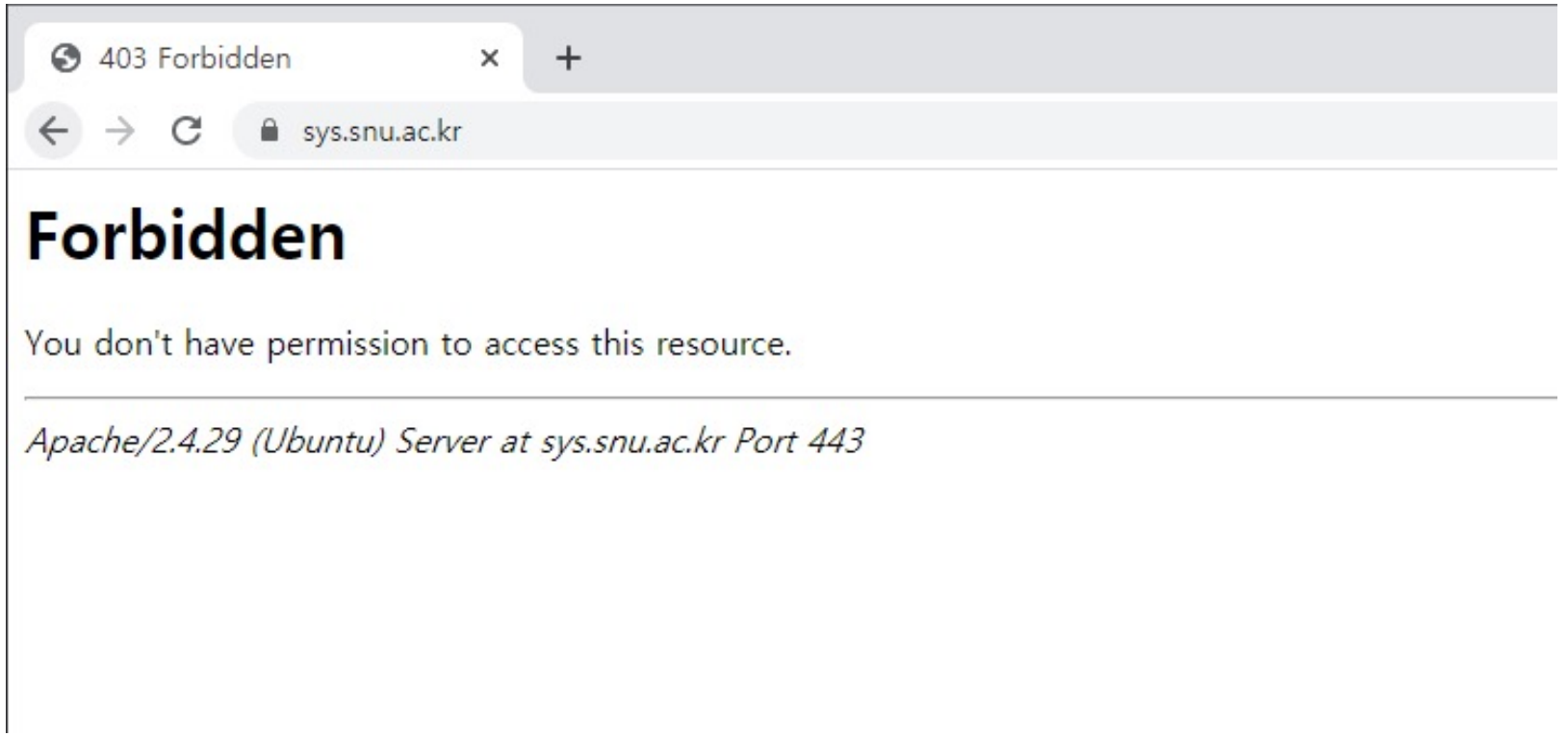


<Scoring ratio for last semester>
Projects 40%
Project #1 5%
Project #2 8%
Project #3 13%
Project #4 14%
※ It is not for this semester.

Account Registration

How to Access the Submission Server

- http://sys.snu.ac.kr
- Need to access via in-school IP or authorized IP



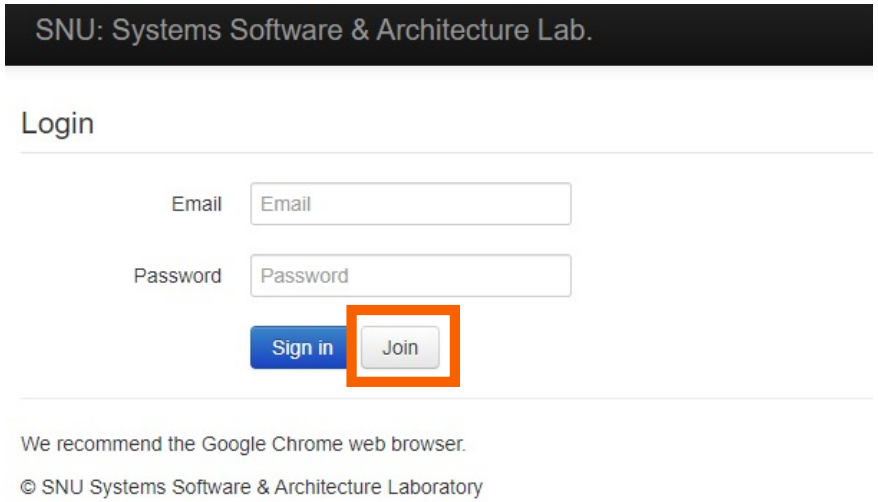
How to Get My IP Authorized?

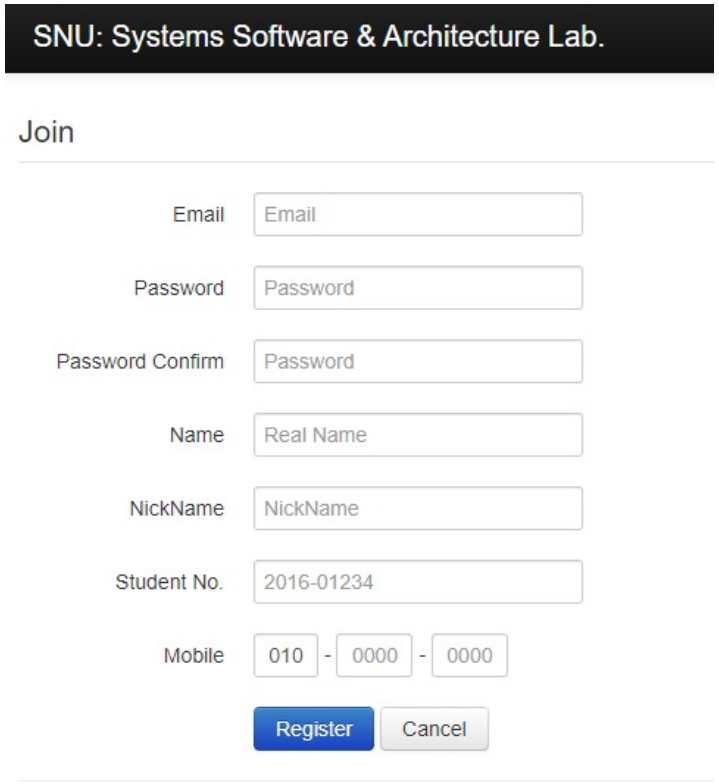
- Please send an email to snucsl.ta@gmail.com with your name, student number, and IP address

- To get your IP address, search "what is my ip" in Google
  - Note that virtual IP addresses are not valid
    - 10.0.0.0 ~ 10.255.255.255
    - 172.16.0.0 ~ 172.31.255.255
    - 192.168.0.0 ~ 192.168.255.255

How to Register an Account?

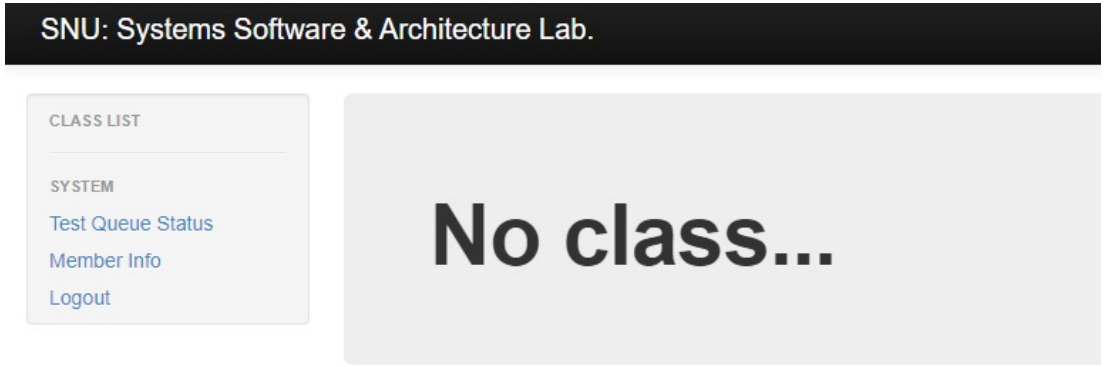
- You should make an account to submit your assignments
- Please make an account with correct name and student number

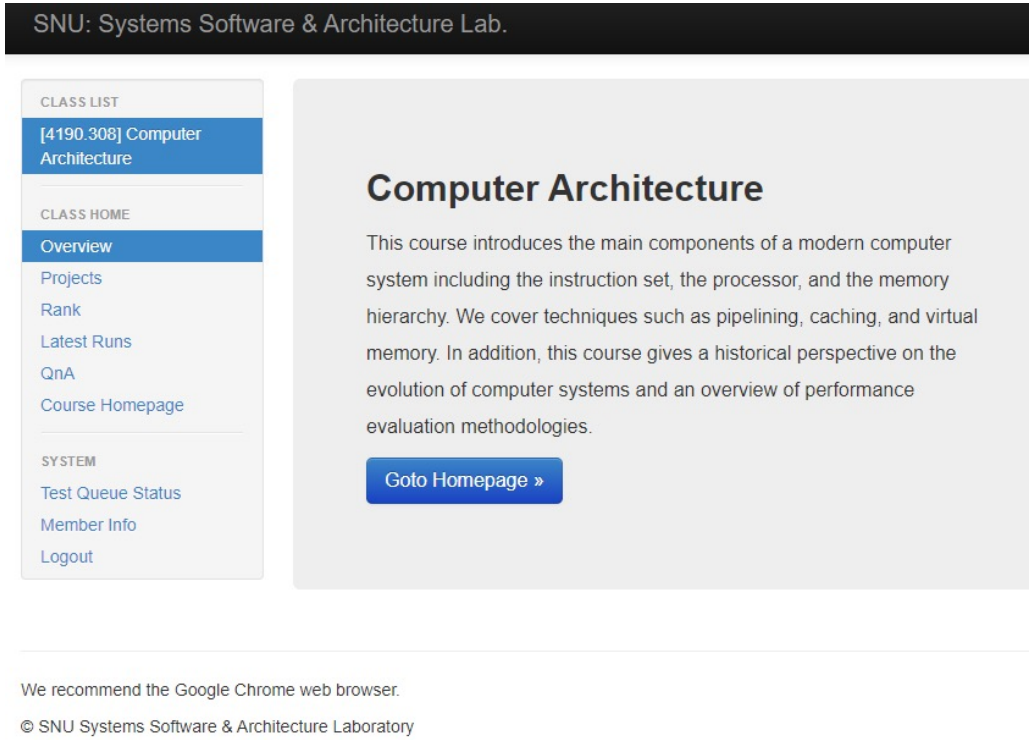




How to Register an Account?

- Don't worry if there's "No class" on the screen
- We will join you to class as soon as possible





Thank You!

- Don't forget to read detailed description before you start your assignment
- If you have any question about the assignment, feel free to ask via email or KakaoTalk open chat
- This file will be uploaded after the lab session☺