4190.308: Computer Architecture Lab. I

Sunmin Jeong, Injae Kang

(snucsl.ta@gmail.com)

Systems Software & Architecture Lab.

Seoul National University

Fall 2020

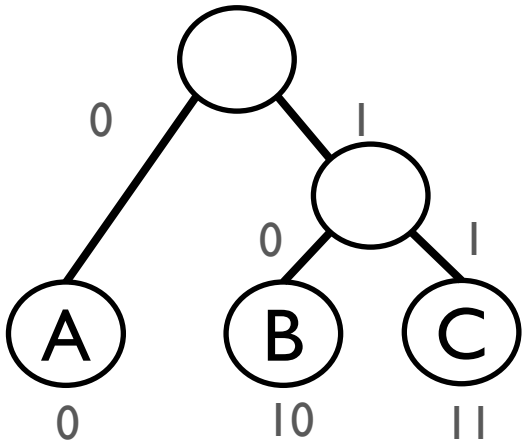
Huffman Coding

What is Huffman Coding?

- An algorithm used for lossless data compression.
- It represents patterns with high frequency as short binary code.
  - Note that you don't have to know the algorithm in this assignment!

A=0, B=10, C=11

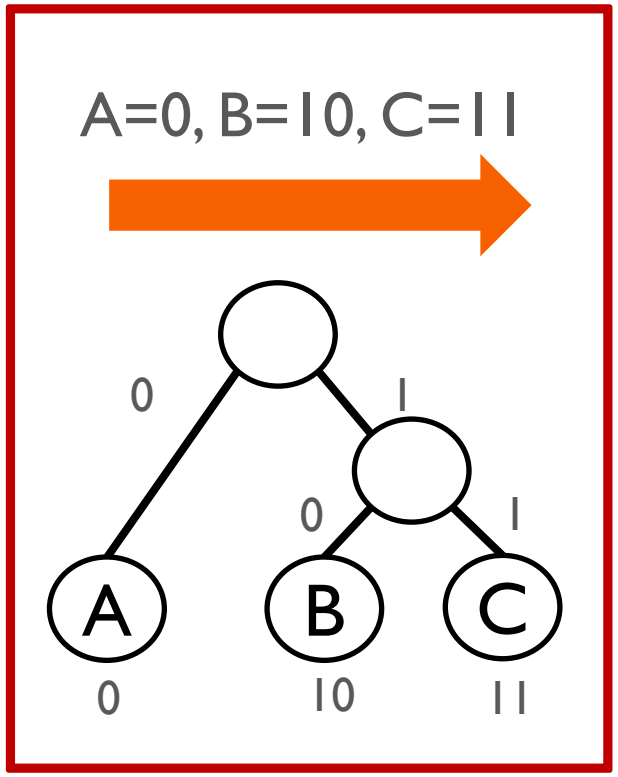
ABAACABBC → 0100011010101011



What is Huffman Coding?

- An algorithm used for lossless data compression.
- It represents patterns with high frequency as short binary code.
  - Note that you don't have to know the algorithm in this assignment!

ABAACABBC



01000110101011

You don't have to know
how to draw the tree.
We will give you an encoding table.

Simplified Huffman Coding

- Consider the input is a stream of 4-bit symbols
  - 0000 ~ 1111 or 0 ~ f
- You should count the frequency of symbols, rank them and get codes with a given table.

Rank	Code	Rank	Code
0	000	8	11000
1	001	9	11001
2	010	10	11010
3	011	11	11011
4	1000	12	11100
5	1001	13	11101
6	1010	14	11110
7	1011	15	11111

Simplified Huffman Coding

- Symbols with higher frequency gets higher rank(closer to 0).
- The smaller symbol wins for same frequency.
  - e.g., 0110 wins 1000 (0x6 < 0x8)

Rank	Code	Rank	Code
0	000	8	11000
1	001	9	11001
2	010	10	11010
3	011	11	11011
4	1000	12	11100
5	1001	13	11101
6	1010	14	11110
7	1011	15	11111

Simplified Huffman Coding

- To simplify, you don't have to consider the frequency of symbols in rank 8 to 15.
  - Regardless frequency, smaller symbols are higher in rank 8-15!

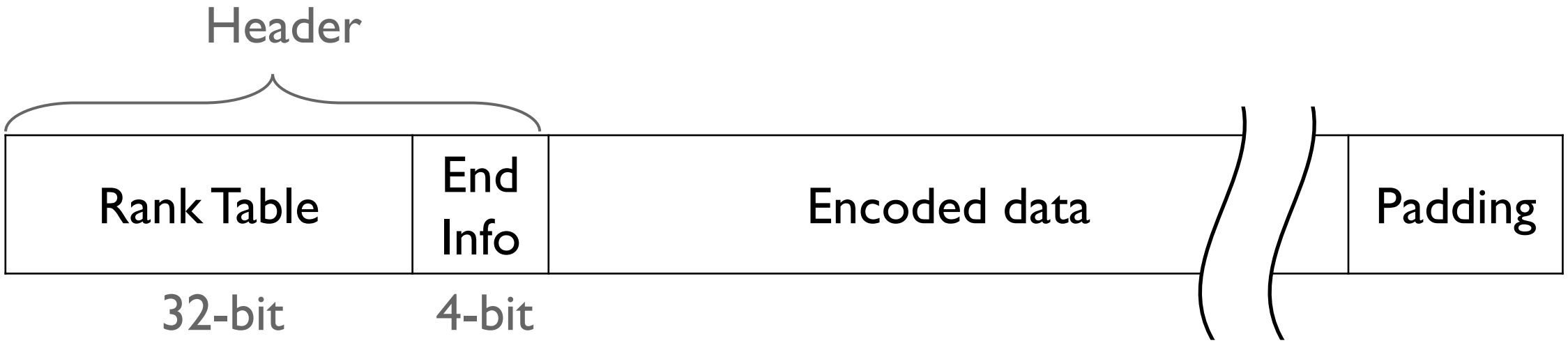
Rank	Code	Rank	Code
0	000	8	11000
1	001	9	11001
2	010	10	11010
3	011	11	11011
4	1000	12	11100
5	1001	13	11101
6	1010	14	11110
7	1011	15	11111

Higher frequency wins

Smaller symbol wins

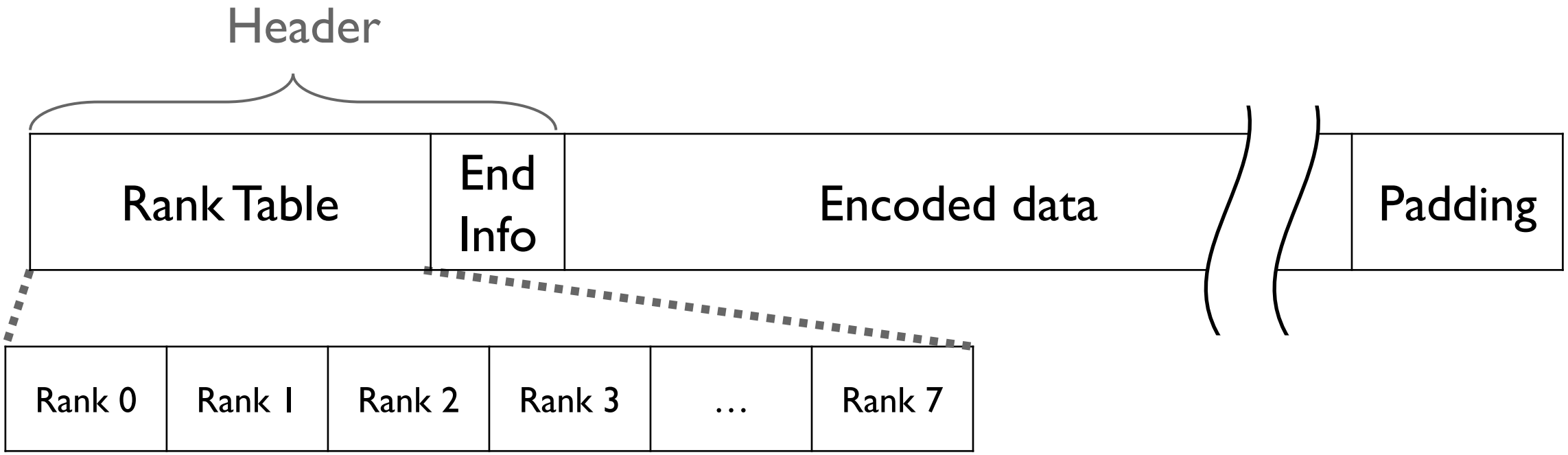
Simplified Huffman Coding

- You should write the output in the following format.
- Output consists of Header, Encoded data and Padding.
- Header consists of 32-bit Rank Table and 4-bit End Info.



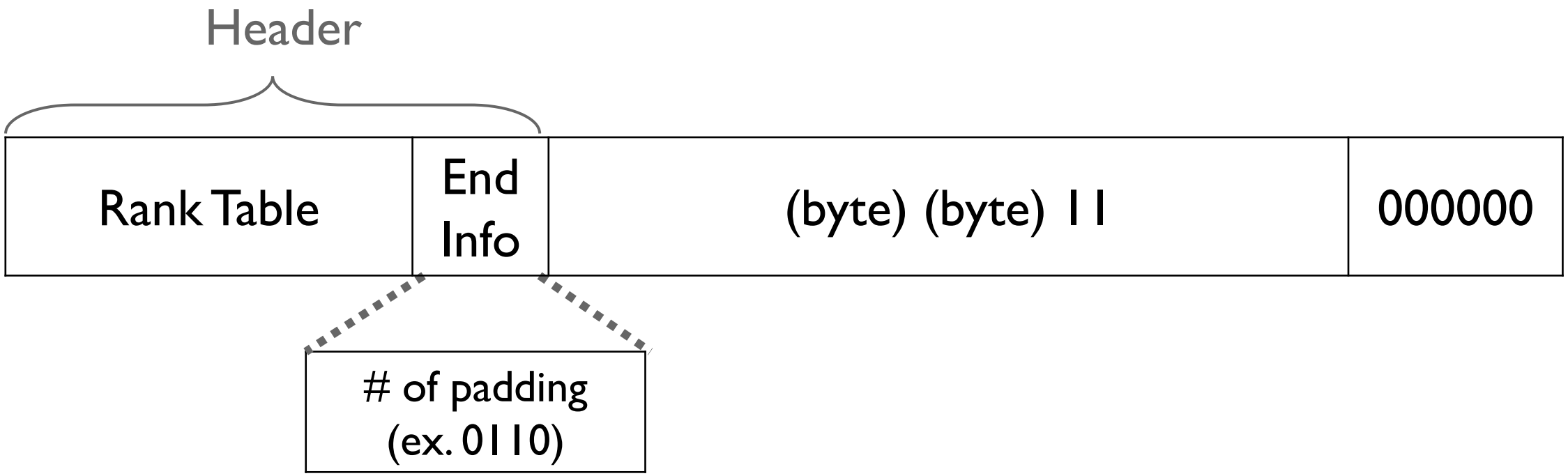
Simplified Huffman Coding

- Rank table has the information for the most frequent 8 symbols.
- You should write symbols sequentially on Rank Table.



Simplified Huffman Coding

- End Info records the # of padding bits for byte alignment.
  - e.g., If the last byte of Encoded data ends with 11, 6 padding bits(0's) are added



Example(1)

- Input stream(in ascii code) : 20 12 30 45 (in hexadecimal)

Symbol	0	1	2	3	4	5	Others
	0000	0001	0010	0011	0100	0101	-
Frequency	2	1	2	1	1	1	0
Rank	0	2	1	3	4	5	6-15
Code	000	010	001	011	1000	1001	

- Encoded Data : 001 000 010 001 011 000 1000 1001

Example(1)

- Input stream(in ascii code) : 20 12 30 45 (in hexadecimal)

Symbol	0	1	2	3	4	5	Others
	0000	0001	0010	0011	0100	0101	-
Frequency	2	1	2	1	1	1	0
Rank	0	2	1	3	4	5	6-15
Code	000	010	001	011	1000	1001	

- Encoded Data : 001 000 010 001 011 000 1000 1001

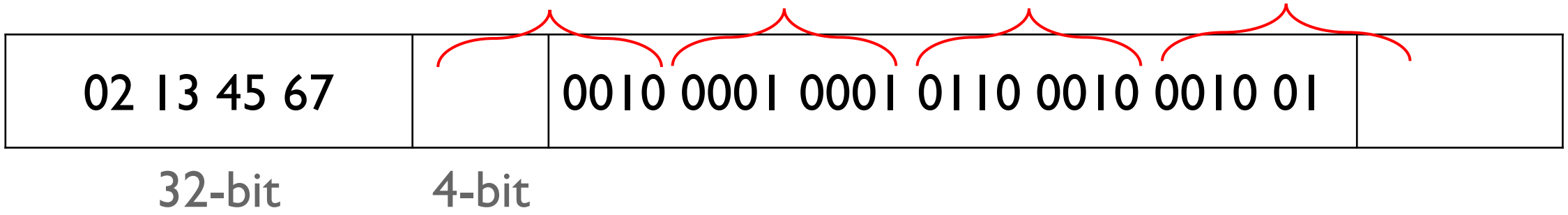
02 13 45 67		0010 0001 0001 0110 0010 0010 01	
32-bit	4-bit		

Example(1)

- Input stream(in ascii code) : 20 12 30 45 (in hexadecimal)

Symbol	0	1	2	3	4	5	Others
	0000	0001	0010	0011	0100	0101	-
Frequency	2	1	2	1	1	1	0
Rank	0	2	1	3	4	5	6-15
Code	000	010	001	011	1000	1001	

- Encoded Data : 001 000 010 001 011 000 1000 1001

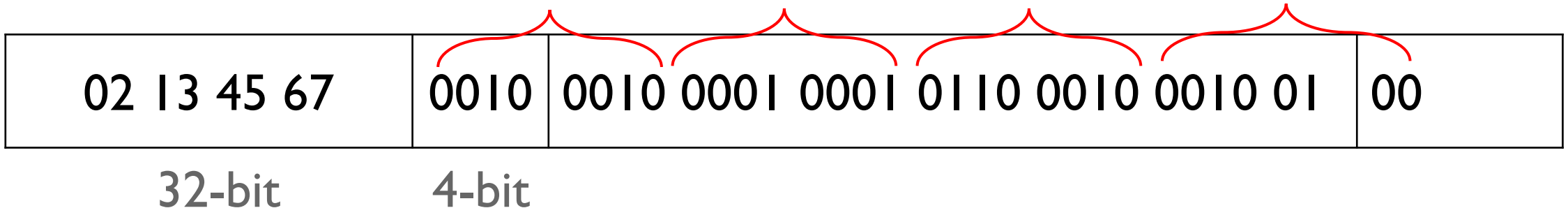


Example(1)

- Input stream(in ascii code) : 20 12 30 45 (in hexadecimal)

Symbol	0	1	2	3	4	5	Others
	0000	0001	0010	0011	0100	0101	-
Frequency	2	1	2	1	1	1	0
Rank	0	2	1	3	4	5	6-15
Code	000	010	001	011	1000	1001	

- Encoded Data : 001 000 010 001 011 000 1000 1001



Example(1)

- Input stream(in ascii code) : 20 12 30 45 (in hexadecimal)

Symbol	0	1	2	3	4	5	Others
	0000	0001	0010	0011	0100	0101	-
Frequency	2	1	2	1	1	1	0
Rank	0	2	1	3	4	5	6-15
Code	000	010	001	011	1000	1001	

- Encoded Data : 001 000 010 001 011 000 1000 1001

02 13 45 67	2	2 11 62 24
32-bit	4-bit	

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.
- It is represented in ascii code as
  54 68 65 20 71 75 69 63 6b 20 62 72 6f 77 6e 20 66 6f 78 20 6a 75 6d
  70 73 20 6f 76 65 72 20 74 68 65 20 6c 61 7a 79 20 64 6f 67 2e

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

1. Count the frequency of each symbols.

Symbol	Frequency	Symbol	Frequency	Symbol	Frequency	Symbol	Frequency
0000	9	0100	3	1000	3	1100	1
0001	2	0101	6	1001	2	1101	1
0010	12	0110	23	1010	2	1110	2
0011	2	0111	15	1011	1	1111	4

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

2. Rank symbols with frequency

Symbol	Frequency	Symbol	Frequency	Symbol	Frequency	Symbol	Frequency
0000	9	0100	3	1000	3	1100	1
0001	2	0101	6	1001	2	1101	1
0010	12	0110	23	1010	2	1110	2
0011	2	0111	15	1011	1	1111	4

Rank	0	1	2	3	4	5	6	7
Symbol	0110	0111	0010	0000	0101	1111	0100	1000

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

3. Rank 8 to 15 → Rank them according to the values of symbols

Pattern	Frequency	Pattern	Frequency	Pattern	Frequency	Pattern	Frequency
0000	9	0100	3	1000	3	1100	1
0001	2	0101	6	1001	2	1101	1
0010	12	0110	23	1010	2	1110	2
0011	2	0111	15	1011	1	1111	4

Rank	8	9	10	11	12	13	14	15
Symbol	0001	0011	1001	1010	1011	1100	1101	1110

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

4. Encode the input stream with the given encoding table

Rank	Code	Rank	Code
0	000	8	11000
1	001	9	11001
2	010	10	11010
3	011	11	11011
4	1000	12	11100
5	1001	13	11101
6	1010	14	11110
7	1011	15	11111

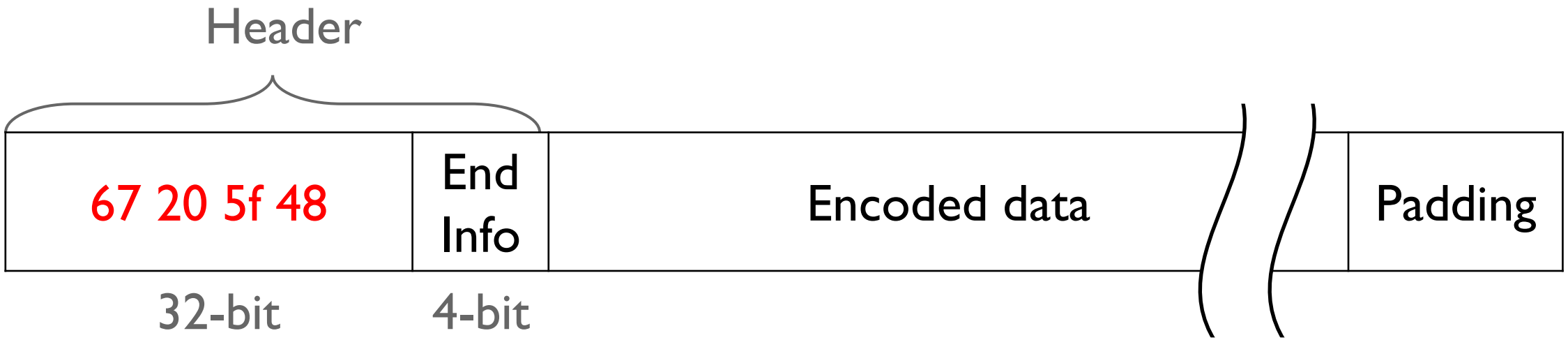
Input	T		h		e	
Symbol	0101	0100	0110	1000	0110	0101
Rank	4	6	0	7	0	4
Encoded data	1000	1010	000	1011	000	1000

Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

5. Write Rank 0-7 symbols in Rank Table

Rank	0	1	2	3	4	5	6	7
Symbol	0110	0111	0010	0000	0101	1111	0100	1000

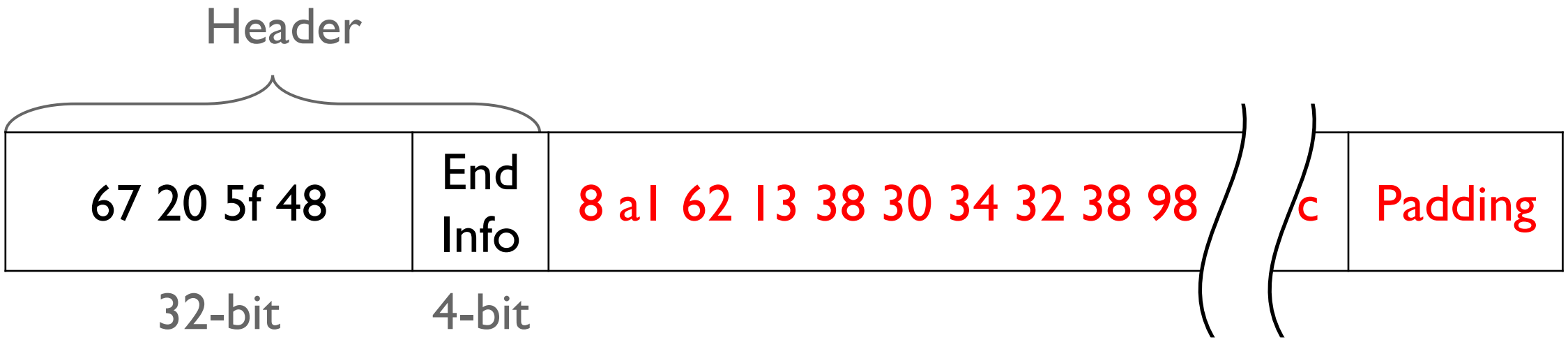


Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

6. Put the encoded data to buffer.

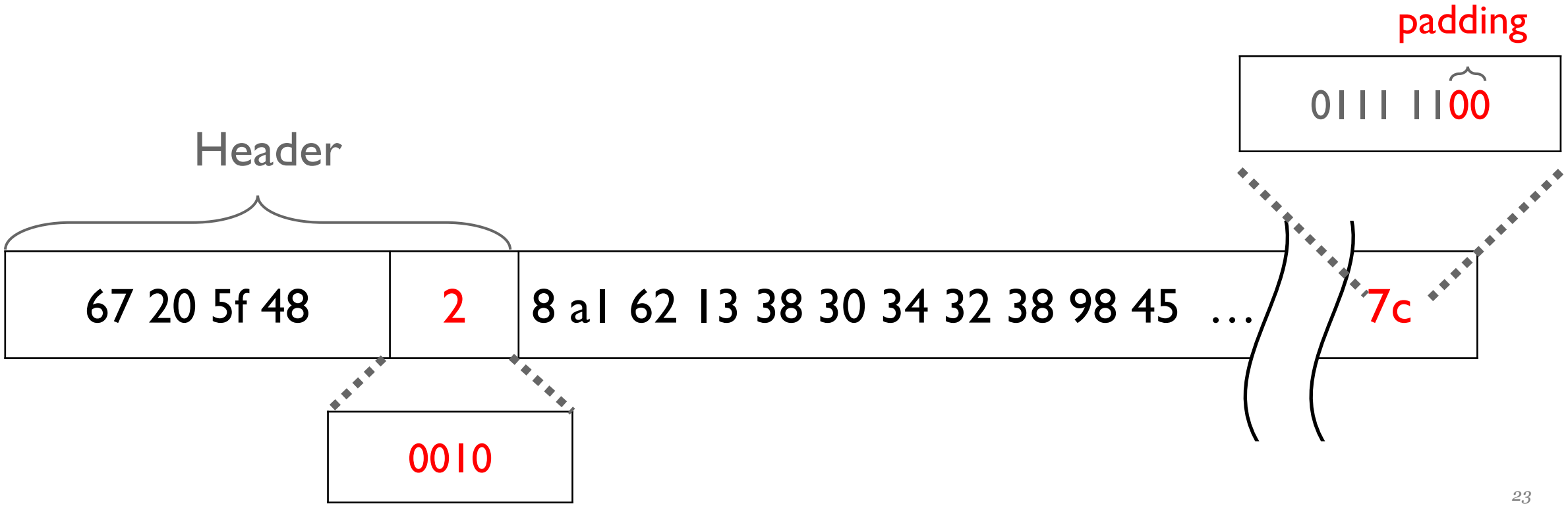
8 a1 62 13 38 30 34 32 38 98 45 09 24 7d 30 04 9b 4c 6c c0 f1 67 29 89 20 41 49 9a 16 21 31 d1 83 b3 a4 c5 09 05 7c



Example(2)

- Input stream : The quick brown fox jumps over the lazy dog.

7. Put padding bits(0's) to align in byte and write the # of padding bits to End Info.



Example(3)

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

Specification

- All you need to do is to write a function given in pa1.c
- *int encode(const char *inp, int inbytes, char *outp, int outbytes)*
  { /* fill this function */}
  - *inp* points to the memory address of the input data
  - *inbytes* is the length of input data (in bytes)
  - *outp* points to the memory address for storing result
  - *outbytes* 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 *outbytes*, return -1
      (In this case, contents of the output is ignored)
    - Note that it should include the length of header

Specification

- You can't use any libraries in pa1.c, including <stdio.h>
  - If you used libraries for debugging, please make sure to erase it before submission.

- Your solution should finish within 5 seconds.

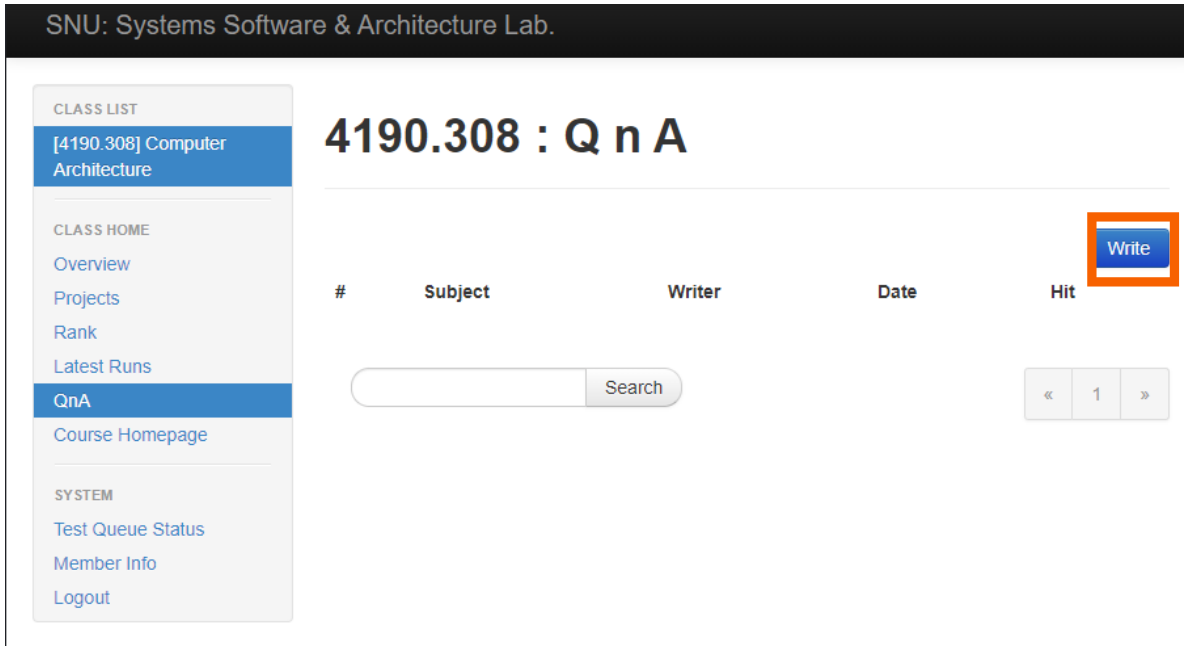
Submission

- Due: 11:59PM, September 20 (Sunday)
  - 25% of the credit will be deducted for every single day delay.

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

Submission

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

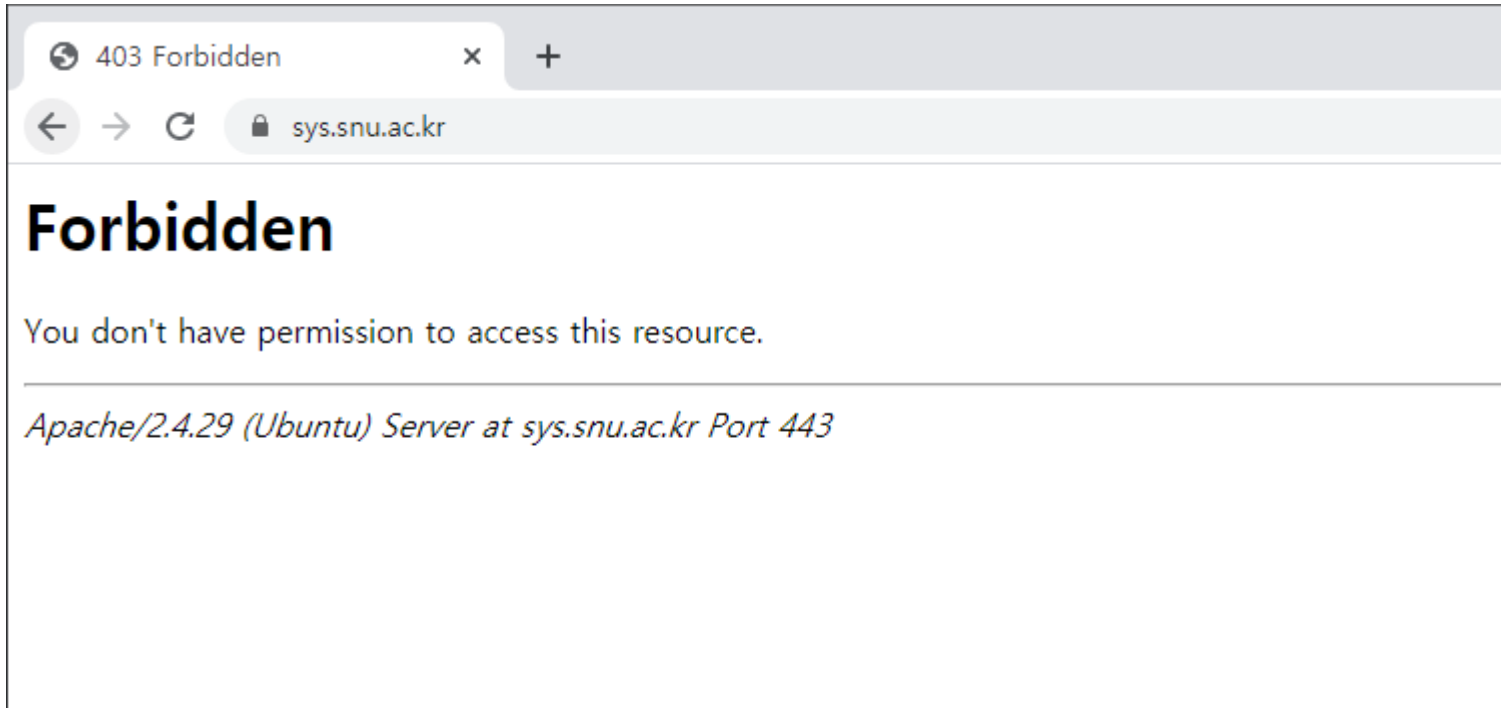


<Scoring ratio for last semester>
Projects 40%
Project #1 5%
Project #2 8%
Project #3 11%
Project #4 16%
※It is not for this semester.

Register an account to server

How to register an account to server?

- You should make an account for http://sys.snu.ac.kr to submit your assignments.
- Only allowed IP can access the server.



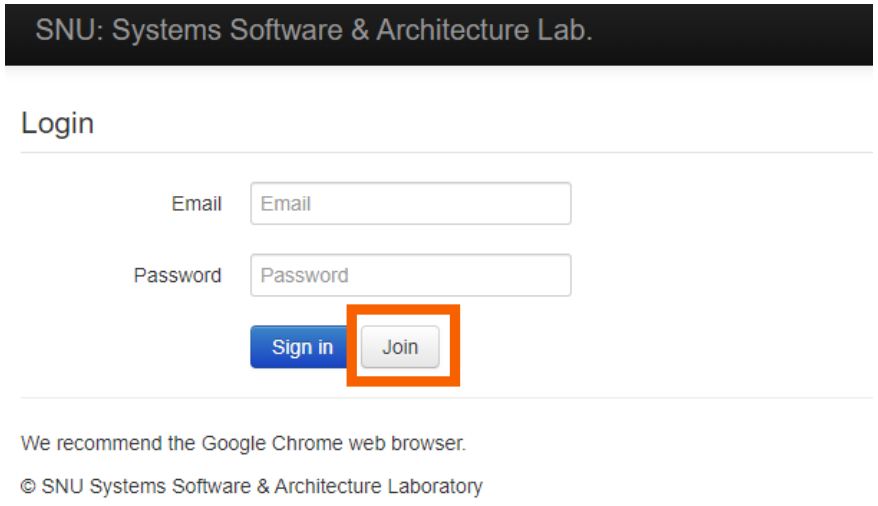
How to register an account to server?

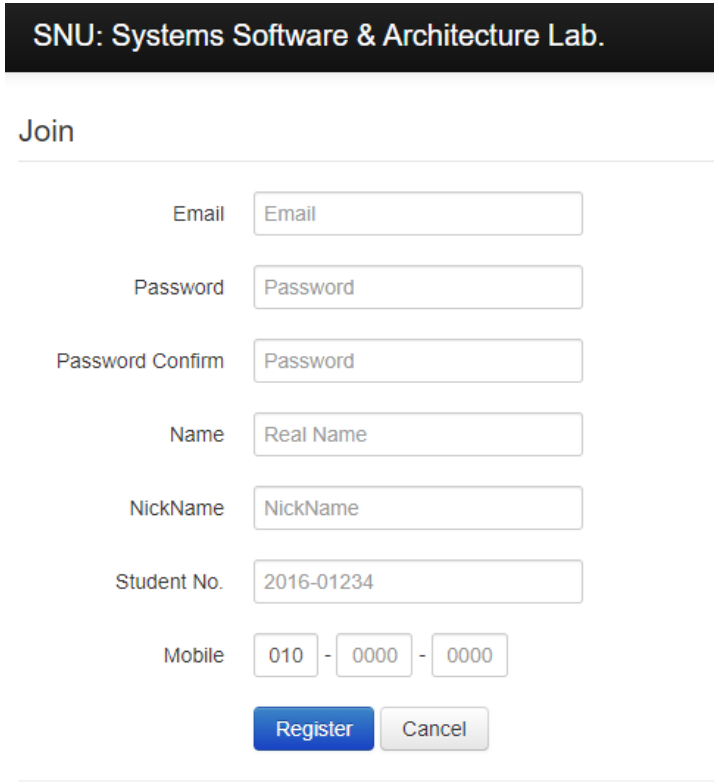
- Please send an email to snucsl.ta@gmail.com
  with your name, student number and IP address.

- 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 to server?

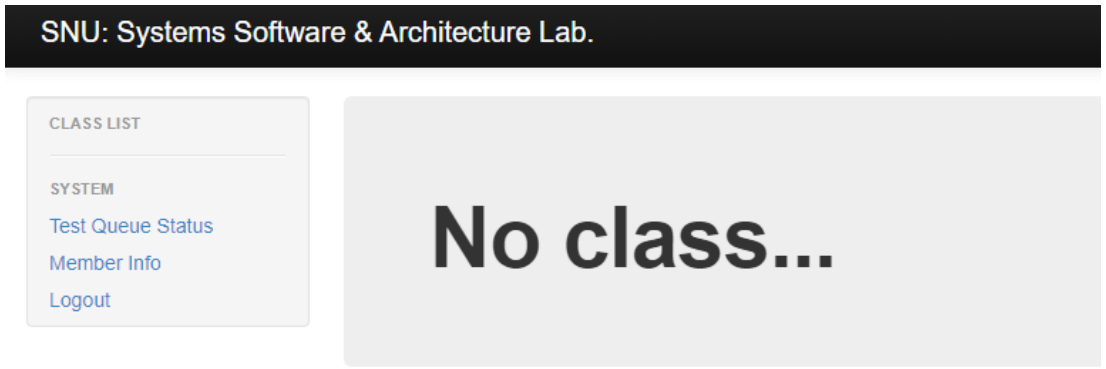
- Then, you can access http://sys.snu.ac.kr.
- Please make an account with correct name and student number.

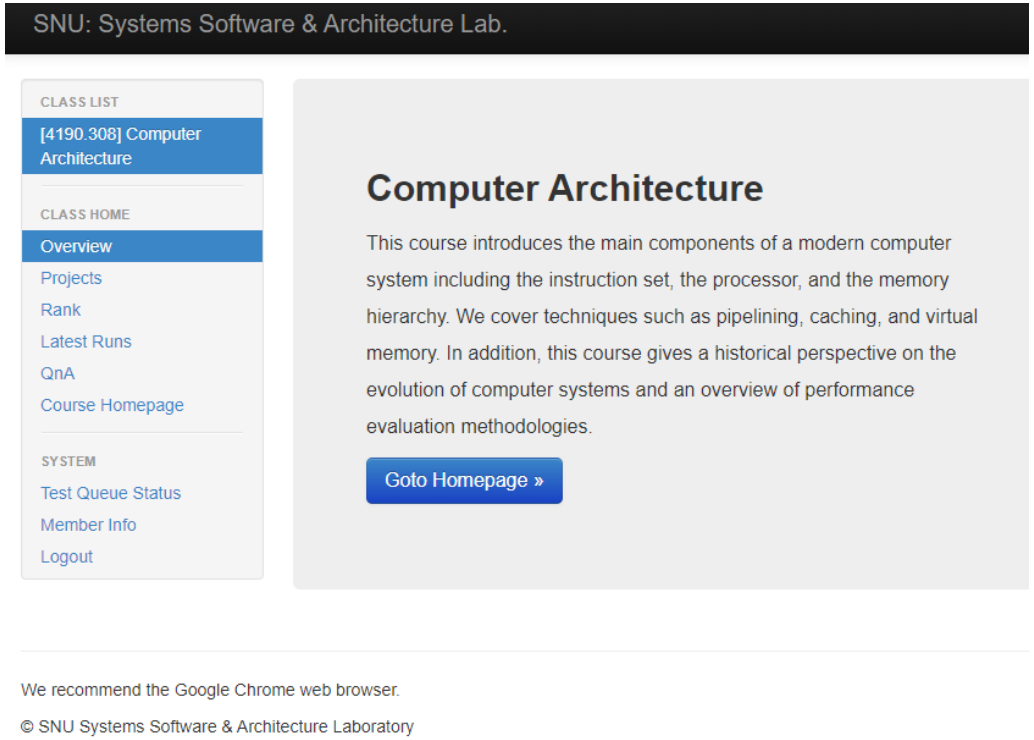




How to register an account to server?

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





Thank you!

- If you have any question about the assignment, feel free to ask us in email or KakaoTalk(TBA).

- This file will be uploaded after the lab session☺