Project1 Environment

E-mail: jylee@ics.kaist.ac.kr
bsjang@ics.kaist.ac.kr

C SIMULATION

◆ C Simulation Procedure

```
sckim@dell2:~/midproj_template$ ls
input_file  krpiss.c  krpiss.h
sckim@dell2:~/midproj_template$ gcc krpiss.c -o krpiss
```

ex) C/C++

ex) gcc

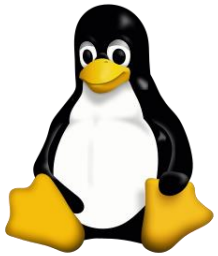
ex) cmd

```
void process(unsigned int cmd, unsigned int inst) {
  printf("Write your ISS program here\n");
  exit(0);
  /*cmd 0: A process when a command 's' is entered*/
  /*cmd 1: A process when a command 'r' is entered*/
}
```

```
sckim@dell2:~/midproj_template$ ./krpiss input_file output_file
============================================================
                    KRP 2.0 Instruction Set Simulator
============================================================
Request:
>> s
Write your ISS program here
```

C SIMULATION

- **Tool Environment**
  - Linux
  - GNU compiler collection (GCC)



- **IP Address:**
  - Assigned IP
    - Check the file below
    - Computer_architecture_server.xlsx

- **ID:**
  - Your student number

- **Default password:**
  - 0000
  - Change it as soon as possible.

C SIMULATION

◆ Server access

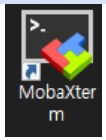
- If your computer is connected to network outside KAIST, it has to be connected to the server through KVPN.
  - For more details, visit KVPN.kaist.ac.kr

- Install MobaXterm
  - It provide terminals for various protocols including SSH and SSH file transfer.

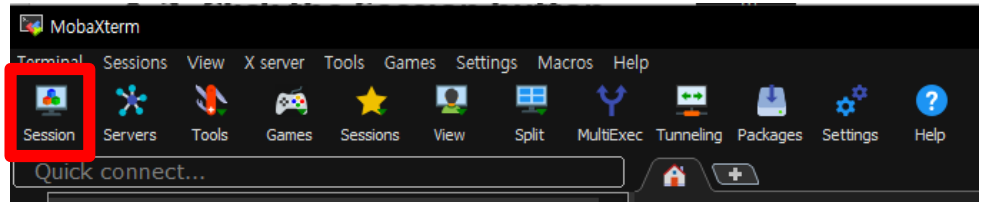
  - https://mobaxterm.mobatek.net/download-home-edition.html
  - Any edition is okay.

How to access server?

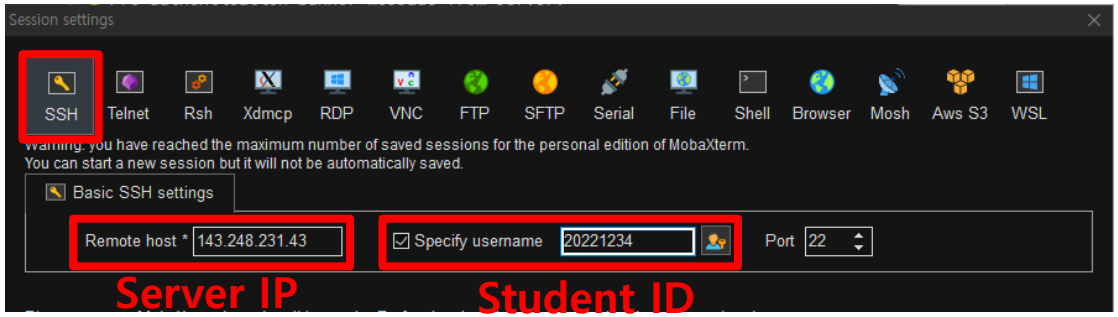
- 1. Execute MobaXterm



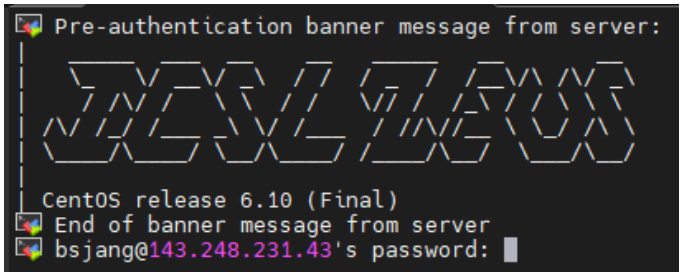
- 2. Click the Session button



- 3.Session Setting → OK



- 4. Log in (init PW : 0000)



C SIMULATION

- You can do the simulations on your own server.

- But we will compile and evaluate your codes on the notified server.
  - **Make sure your compiled output works well on the notified server.**
  - **Claims for errors due to environmental differences will not be accepted.**

- Make a directory for your code submission on the server.
  - ~/project1_submission
    - You must also upload your zip file on KLMS.
  - You can create a directory you want.

TEST ENVIRONMENT

- **krpassembler**
  - Assembler for KRP 2.0
  - Translate assembly language to machine language format

  - Input files:
    - You can make your own assembly language.
    - test_codes (provided)
      - example, test1, and test2
  - Output files:
    - output_log
      - Log of assembly language
    - vectors.bin → input_file of ISS
      - Binary file of instructions

TEST ENVIRONMENT

◆ Execution example

Assembly language

```
ADDI r1, r2, #0x10000
ADDI r1, r2, SHL(#0x200, #0x10)
ANDI r1, r2, #2
ANDI r1, r2, LSR(#4, #0x8)
```

```
sckim@dell2:~/assembler$ ls
kass  test_vectors
sckim@dell2:~/assembler$ ./kass ./test_vectors/example
sckim@dell2:~/assembler$ ls
kass  output_log  test_vectors  vectors.bin
sckim@dell2:~/assembler$
```

Execution

Output files

```
1. ADDI type 1                          → Instruction type
   ra: 1, rb: 2, imm: 65536             → Register and immediate information
   insn: 00450000                       → Instruction
2. ADDI type 2
   ra: 1, rb: 2, imm: 512, mode: 0, shamt: 16
   insn: 08450010
```

This is for your easy debugging

TEST ENVIRONMENT

◆ **Execution example**

```
sckim@dell2:~/assembler$ ls
kass  test_vectors
sckim@dell2:~/assembler$ ./kass ./test_vectors/example
sckim@dell2:~/assembler$ ls
kass  output_log  test_vectors  vectors.bin
sckim@dell2:~/assembler$
```

Output files

```
^@^@E^@^P^@E^H^B^@D (^BD
~
~
```

:%!xxd

Can be translated in hex format by vim editor

```
00000000: 0000 4500 1000 4508 0200 4420 2802 4428
00000010: 0b00 9612 4414 961a 0b00 4032 e706 003a
00000020: 0030 4440 0040 8648 0000 015b 0070 8055
00000030: 0030 4468 00c0 9662 0060 8770 0400 4480
00000040: 2030 4480 0400 4478 2030 4478 0400 4488
00000050: 2030 4488 0400 4490 2030 4490 0100 0298
00000060: 0260 0898 0100 44a0 04b0 0ca1 0000 20a8
00000070: 1400 40b0 e803 7eb8 0800 e2b8 0c00 40c0
00000080: 0400 7ec8 0400 88c8 2800 40d0 1400 40d8
00000090: 0000 00e8 0000 00f0 0000 00f8 0a
```

Address$_{(16)}$ Instructions$_{(16)}$

Stored in little-endian

Instruction: 08450010

This is for ISS input_file

ISS EXECUTION

◆ Example

```
sckim@dell2:~/midproj_template$ ./krpiss ../assembler/vectors.bin output_file
================================================================================
                        KRP 2.0 Instruction Set Simulator
================================================================================
Request:
>> s
PC :00000000
IR :00450000
IE :0
IPC:00000000
```

intput_file

or

Insert a command
$ ./run.sh

TEST ENVIRONMENT

- test_codes
  - test1

```
MOVI r0, ROR(#-9, #4)
MOVI r1, #90
MOVI r2, #15
MOVI r3, #3
MOVI r4, SHL(#2, #1)
MOVI r5, ASR(#-8, #1)
MOVI r6, #-12
MOVI r7, #-100
MOVI r8, #22
MOVI r9, #-33
ADD r10, r7, r1
ADDI r11, r8, #-22
SUB r12, r4, r5
NOT r13, r8
SUB r14, r9, r7
ADD r15, r5, r6
NEG r16, r15
```

```
@r0 = 7F_FF_FF_FF 2147483647
@r1 = 00_00_00_5A 90
@r2 = 00_00_00_0F 15
@r3 = 00_00_00_03 3
@r4 = 00_00_00_04 4
@r5 = FF_FF_FF_FC -4
@r6 = FF_FF_FF_F4 -12
@r7 = FF_FF_FF_9C -100
@r8 = 00_00_00_16 22
@r9 = FF_FF_FF_DF -33
@r10 = FF_FF_FF_F6 -10
@r11 = 0
@r12 = 00_00_00_08 8
@r13 = FF_FF_FF_E9 -23
@r14 = 00_00_00_43 67
@r15 = FF_FF_FF_F0 -16
@r16 = 00_00_00_10 16
```

Assembly language

Expected result

TEST ENVIRONMENT

- test_codes
  - test2

```
MOVI r0, #0
MOVI r1, #131071
ORI r2, r2, #255
NOT r3, r2
SHL r4, r3, #16
LSR r5, r4, #2
ASR r6, r4, #2
XOR r7, r6, r1
AND r8, r3, r7
OR r9, r3, r8
```

Assembly language

```
@r0 = 0
@r1 = FF_FF_FF_FF, -1
@r2 = 00_00_00_FF, 255
@r3 = FF_FF_FF_00, -256
@r4 = FF_00_00_00, -16777216
@r5 = 3F_C0_00_00, 1069547520
@r6 = FF_C0_00_00, -4194304
@r7 = 00_3F_FF_FF, 4194303
@r8 = 00_3F_FF_00, 4194048
@r9 = FF_FF_FF_00, -256
```

Expected result