

CS2323 Computer Architecture (Sep-Nov 2020)

Assignment 1: Program Code Converter

Marks: 50

This assignment must be done individually.

Start Date: 24-09-2020

Due Date: 11-10-2020, 11:55 pm

NOTE: For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due.

1. Problem Description

Write code in Java (or C++ or C) to do the following:

- Convert a MIPS assembly language program to binary language program for MIPS. **[25 points]**
- Convert a binary language program for MIPS to the MIPS assembly language program. **[25 points]**

Along with your code, you need to provide five sample MIPS assembly language program code as: prog1.asm, prog2.asm, prog3.asm, prog4.asm and prog5.asm. Try to keep these programs as diverse as possible so that they test different aspects of assemble language programming.

Your program should prompt the user to enter a MIPS program code filename, say for example prog2.asm. Your code then should read the content of file prog2.asm and then generate a binary language program code called prog2bin.txt. For readability purpose, along with the binary code also print the hexadecimal equivalent and the corresponding MIPS code in bracket. For example,

add \$t0, \$s1, \$s2 → 00000010001100100100000000100000 (02324020, add \$t0, \$s1, \$s2)

2. What to Submit

You should submit your files as a .zip file through Microsoft Teams. Name it as **YourRollNumber_Assign1.zip**. You need to submit the following files:

1. **Assembler.java/cpp/c**: It converts MIPS assembly code to binary machine code
2. **ConvertAssembly.java/cpp/c**: It converts the binary machine code to MIPS assembly code
3. Five sample MIPS program code as: prog1.asm, prog2.asm... prog5.asm. Put them in a sub-folder called **sample_data**.

3. References

MIPS opcode and functions: <http://alumni.cs.ucr.edu/~vladimir/cs161/mips.html>

MIPS register conventions: Register 1, called \$at, is reserved for the assembler, and registers 26-27, called \$k0-\$k1, are reserved for the operating system.

Name	Register number	Usage
\$zero	0	The constant value 0
\$v0-\$v1	2-3	Values for results and expression evaluation
\$a0-\$a3	4-7	Arguments
\$t0-\$t7	8-15	Temporaries
\$s0-\$s7	16-23	Saved
\$t8-\$t9	24-25	More temporaries
\$gp	28	Global pointer
\$sp	29	Stack pointer
\$fp	30	Frame pointer
\$ra	31	Return address

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

Assembler

Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
100011001111001000000000000000100
10101100111100100000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
```