National Research University Higher School of Economics
Faculty of Computer Science
Bachelor’s Program “HSE University and University of London Double Degree Program in Data Science and Business Analytics”

Introduction to Programming

Workshop #26

Thu 15.04.2021

Julio Carrasquel

NATIONAL RESEARCH
UNIVERSITY

Binary numbers and bitwise operations. Why to study them?

Binary numbers and bitwise operations. Why to study them?

To understand how computers *think* at the most bottom level.

- **How the data of a program is stored in the memory**
- **How instructions of a program are stored and executed**
- **How integers and other data types are represented**
- **...**

Assembly in MARS simulator [not for evaluation!]

- We know that C++ programs are transformed to *assembly* language in the *compilation procedure*.
- Instructions in *assembly* language are easily transformed to binary numbers which are stored in the memory.

Assembly in MARS simulator [not for evaluation!]

- We know that C++ programs are transformed to *assembly* language in the *compilation procedure*.
- Instructions in *assembly* language are easily transformed to binary numbers which are stored in the memory.

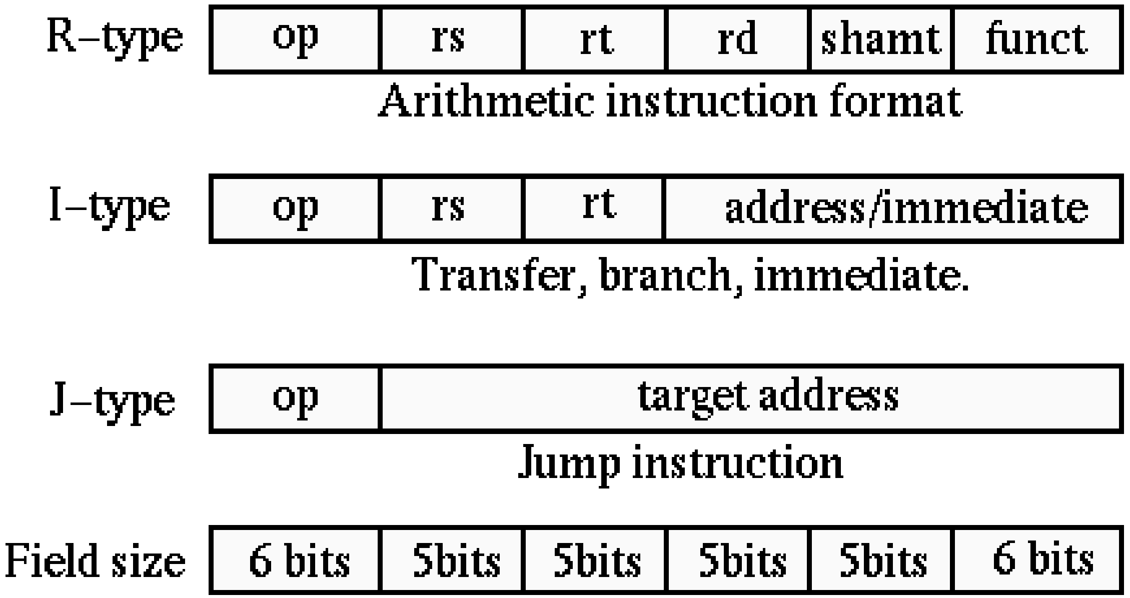
- MARS (*MIPS Assembler and Runtime Simulator*) is an IDE for programming in MIPS assembly language:

  - Download at: http://courses.missouristate.edu/kenvollmar/mars/

  - Needs *Java* installed in your computer to run: https://www.oracle.com/java/technologies/javase-downloads.html

- MIPS is an example of a *computer architecture*: data types, main memory, instructions, etc.

Examples of instructions in MIPS assembly language



R–type	op	rs	rt	rd	shamt	funct
	Arithmetic instruction format					
I–type	op	rs	rt	address/immediate		
	Transfer, branch, immediate.					
J–type	op	target address				
	Jump instruction					
Field size	6 bits	5bits	5bits	5bits	5bits	6 bits

op: operation
rs: source register
rt: target register

Testing – Google Test

Google Test Library

- Google Test (also known as *gtest*) is a *unit testing* library for C++.

- *Unit testing* is software method to check if individual units of source code (functions, classes) behave as expected, for example, if a function produces the expected result.

Useful links

Basic concepts in the official page:
https://google.github.io/googletest/primer.html

Download library [download repository in your computer]:
https://github.com/google/googletest

Video: Configure QtCreator + Google Test
https://youtu.be/hMriR-fFkZo

Video: Configure CLion + Google Test
https://youtu.be/M067vFQG7ZA

Exercise

- **Download the Google Test library.**

- **Create a project in your IDE adding the Google Test library.**

- **In the main.cpp file, code a function to calculate the factorial of a number.**

- **Try to test some result of the factorial function as follows:**

```
TEST(MyTestSuite, MyTestCase)
{
    EXPECT_EQ(120, factorial(5));
}
```

- **Remember you need to include these libraries...**

```
#include <gtest/gtest.h>
#include <gmock/gmock-matchers.h>
```

- **...and to change the main() function as follows:**

```
int main(int argc, char *argv[])
{
    testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```