CS 240 Programming in C

Introduction

Schedule

1. Course Introduction
2. History of C
3. History of C Language
4. Editor
5. Intro of Linux Command-line

Course Introduction

- The course lecture materials I am using are mostly based on the professor Ming Ouyang's past and Professor Haoyu Wang CS240 course materials. I am much grateful for his gracious help and guidance.
- However, I made some changes based on my preferences.

How to get better scores

- Programming in C is a fundamental course, it entails reading, coding, testing, many times maybe, to get good at it. Thus it is kind of demanding in terms of time and effort to be paid.
- However for getting good scores in this class is not hard if you taken it seriously. And there are some suggestions that I want to offer.

How to get better scores

- Generally speaking, if you understood course topics, the demo codes and done the homework well, you will get good grades.
- Specifically,

How to get better scores

- for coding new starters:
  spend more time to catch up those common and basic concepts and skills in programming area, like basic Linux command-line commands, basic file system concepts etc. Prioritise your problems, take it slowly instead of avoiding them completely.
- for coding 'veterans':
  finishing the homework by instructions first; and pays enough attention at least to know what topics will be on each exam and review them.

Homework Requirements

- It's better for you to write your code in some text editors for this class instead of IDEs.
- Submit your source codes in plain source code files according to homework instructions.
- Last year, some student submitted the whole project folders created by some IDE got zero score on their homework.

Discussion

- Announcements and discussions will be posted on Blackboard
- Check Blackboard and read your email regularly

Plagiarism

- Don't do it in any form or it will be reported.
- Test cheating will lead to fail.

Setup Environment

- Don't do it in any form or it will be reported.
- Test cheating will lead to fail.

Linux Server

- Access Linux server.
- Transfer files between your machine and sever.

Computer architecture

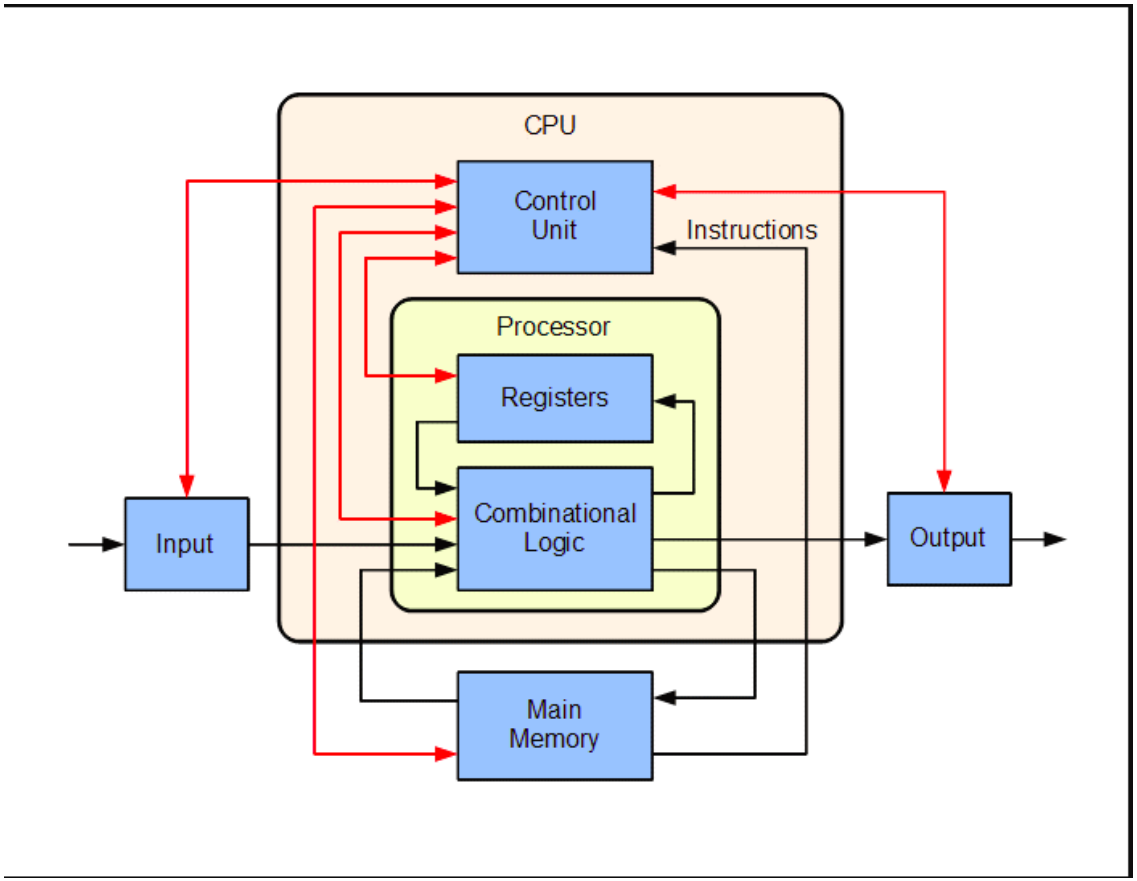

Figure: Block diagram of a basic computer with uniprocessor CPU. Black lines indicate data flow, whereas red lines indicate control flow. Arrows indicate the direction of flow.

`https://en.wikipedia.org/wiki/Computer_architecture`

Computer architecture

Now, let's getting into some basic computer principle.

- CPU executes on sequences of binary digits, called machine language or machine code or object code which are dictated of CPU instructions.
- Machine language is the only language a computer is capable of understanding.

Instruction set architecture

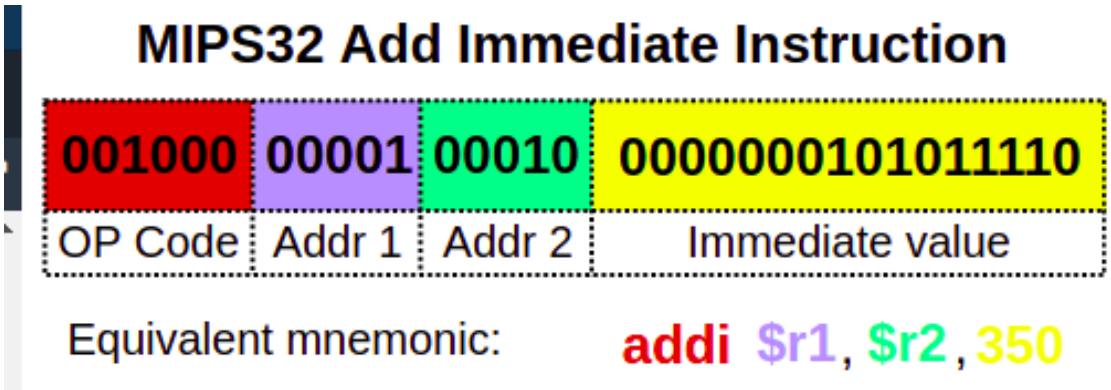

Figure: One instruction may have several fields, which identify the logical operation, and may also include source and destination addresses and constant values. This is the MIPS "Add Immediate" instruction, which allows selection of source and destination registers and inclusion of a small constant.

`https://en.wikipedia.org/wiki/Instruction_set_architecture`

Early tape program

- Take a look at Punched Tape Codes in early days, which means binary code on tape. `http://www.chilton-computing.org.uk/acl/literature/chapman/p015.htm`
- Well as time move on, there came "higher" abstracted programming languages which were written in English characters, but can be translated into machine code for executing.
- The text programs written are called source code. The processes of translating source code into machine code are called compilation.

Compiler

- The actual main processes of a modern compiler are working like these:

  source code $\rightarrow$ preprocessor $\rightarrow$ complier $\rightarrow$ assembler $\rightarrow$ object code $\rightarrow$ linker $\rightarrow$ executables

- However, in most cases we just say writing code, compiling it and running it.

Script Language

- Some language called script language, like bash, python, integrates compiling and executing processes together so that source code can be directly executed.
- For example, python hello.py

C Language History

Let's talk a little bit about the history of C.

- C is a by product of the UNIX operating system, which was devised in the early 1970s by Ken Thompson and Dennis M. Ritchie from Bell Labs (ATT).
- `https://en.wikipedia.org/wiki/Ken_Thompson`
- `https://en.wikipedia.org/wiki/Dennis_Ritchie`

C Language History - Origin

- UNIX was first written in assembly language which were painful to debug and hard to maintenance.
- Thompson decided that a higher-level language was needed for the further development of UNIX, so he designed a small language call B.
- Then Rithie joined the chat and began to develop an extended version of B, which he called NB ("New B") at first and then C, since they became so different.
- Then the UNIX were wholly rewritten in C, which gave the an important benefit to UNIX which is able to run on many other machines as well.

C Language History - Standardization

- After that C got much popularity and continued to evolve.
- The book The C Programming Language (The first edition of KR) written by Brian Kernighan and Dennis Rithcie became the bible of C programmers, which acted as a status quo standard during that time.
- `https://en.wikipedia.org/wiki/Brian_Kernighan`
- However the first KR was fuzzy about some language features which causes problems that numerous dialects of C came about.
- To address this problem, in 1983, the American National Standards Institute (ANSI) formed a committee to make an standard specification of C, which later is referred to as ANSI C, C89. And the first version is call KR C.
- The textbook that we are using, the second edition of KR book, is actually about ANSI C, which is demonstrated on its cover.

C Language History - Standardization

Version	Standard	Publication Date
K&R	n/a	1978-02-22
C89	ANSI X3.159-1989	1989-12-14
C90	ISO/IEC 9899:1990	1990-12-20
C95	ISO/IEC 9899/AMD1:1995	1995-03-30
C99	ISO/IEC 9899:1999	1999-12-16
C11	ISO/IEC 9899:2011	2011-12-15

C-Family programming language

- C has had huge influence on modern-day programming languages, many of which borrow heavily from it, or directly derived from it, like C++, Java, Perl etc.
- `https://en.wikipedia.org/wiki/List_of_C-family_programming_languages`

C and OS

- Since it's birth, C is closely related to operating system, because it is "higher" language that closely related to hardware.
- Linux is directive derivative of Unix, which is written mostly in C, with some parts in assembly.
- Microsoft's Windows kernel and Mac OS X kernel are also developed mostly in C, with some parts in assembly language.
- Every program and driver in a Mac, as in Windows and Linux computers, is running on a C-powered kernel.
- Besides the world's most popular databases, including Oracle Database, MySQL, MS SQL Server, and PostgreSQL, are coded in C and C++.

Speed of C program

- C program is fast since it is close to hardware.
- A speed comparison.
  `https://leetcode.com/problems/two-sum/submissions/`
- For all implementations of the same algorithm in different languages, the C implementation is the fastest.

C Compilers

- `https://www.educba.com/best-c-compilers/`
- There are many compilers available for C. GCC stands out to be one of the best as of now.
- We will use GCC in this class. It comes with Linux or Mac OS. And you can install it on Windows.

Editors on Linux

- `vi`
  - Installed by default on typical Linux installations
  - Popular among system admins
  - `vim` is a variant
  - A tutorial
    `https://www.tutorialspoint.com/unix/unix-vi-editor.htm`
- `emacs`
  - Installed by this command:
    `sudo yum install emacs`
  - A lot bigger and a little slower than `vi`
  - A guided tour `http://www.gnu.org/software/emacs/tour/`
- On our department server, there is also a simpler editor Nano.
- Vi/vim and emacs are professional and come well along with linux system.
- However they are time consuming to learn and frustrating to use for beginners.

Editor Demo

Demo

End

Begin to read the first chapter, and try the examples on your home machine or server!