程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

DEVELOPMENT TOOLS | SEC204

https://tutorcs.com

OVERVIEW

- We'll briefly discuss ~~nt~~ tools
    - Assembler
    - Linker
    - Debugger
    - Compiler
    - Object code disassembler

- Then we'll do practical activities
    - Create your own cheat sheet for gcc, gdb, objdump
    - Create a simple program
    - Observe the fetch, decode, execute cycle when running a program

ASSEMBLER, LINKER

ASSEMBLER

- Converts the Assembly language source code to instruction code for the processor
  - Instruction code varies for different assemblers (even for the same architecture)
    - Assembler Directives instruct the assembler how to construct the instruction code program. They are unique to the individual assembler

- Examples include
  - MASM
  - NASM
  - **GAS**
  - HLA

LINKER

- Links objects: resolves all defined functions and memory address labels declared in the program code

  - For external functions (eg printf) usually a second step is required to link the assembly object code with other external dynamic libraries and allow the executable to run on the host system

  - ld command

    `ld -o test test.o`
    Creates executable file `test` from the object file `test.o`

DEBUGGER

- Very important. Runs the progr~~~~~~~~~~own controlled "sandbox"

- Runs the program in a controlled environment, specifying runtime parameters

- Stops the program at any point within the program

- Examines data elements, such as memory locations or registers

- Changes elements in the program while it is running to help bug removal

- GDB

  ```
  gcc -gstabs -o test test.c
  ```

- `gdb test`

- Compiles `test.c` using debugging information to create executable file `test`. Then opens it in debugging mode with gdb.

GDB COMMANDS

- `break` – Set a breakpoint
- `watch` – Set a watchpoint [cause execution] when a variable reaches the specific value
- `info` – observe system elements, such as registers, the stack, memory
- `x` – examine memory location
- `print` – Display variable values
- `Run` – Start execution
- `list` – List specified functions or lines
- `next` – Step to the next instruction in the program
- `step` – Step to the next instruction in the program
- `cont` – Continue executing the program from the stopped point
- `until` – Run the program until it reaches the specified source code line (or greater)

COMPILER, OBJECT CODE DISASSEMBLER

- COMPILER
  - Converts high level code into ____ language and then into instruction code for the processor to execute.
  - gcc
  - `gcc -o test test.c`
    Creates executable file `test` from C language program `test.c`

- OBJECT CODE DISASSEMBLER
  - Takes a full executable program (or an object code file) and displays the instruction codes that will be ran by the processor
    - Some disassemblers convert the instructions into Assembly language syntax (mnemonics)
  - `gcc -c test.c`
    `objdump -d test.o`
    Creates an object file by compiling `test.c`. Then display the disassembled object code file with `objdump`.

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

PRACTICAL ACTIVITIES
Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

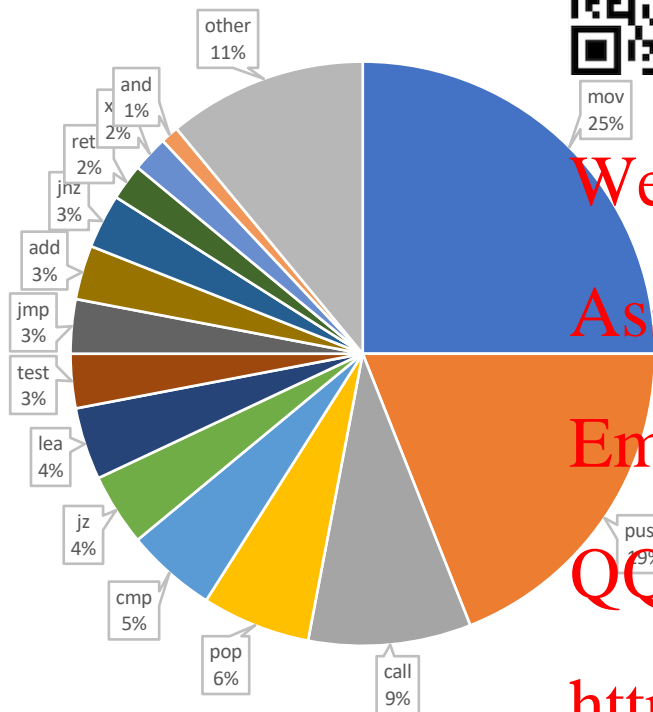# ASSEMBLY TASK

Opcode Breakdown in Goodware

- mov 25%
- push 9%
- call 9%
- pop 6%
- cmp 5%
- jz 4%
- lea 4%
- test 3%
- jmp 3%
- add 3%
- jnz 3%
- ret 2%
- x 2%
- and 1%
- other 11%

Remember these Assembly instructions?

☐ Find out what the first 5 can do

☐ You can use the internet or the IA 32 cheat sheet on DLE

REFERENCE: DATA TYPES IN C

In C: char
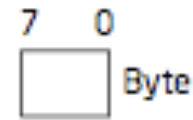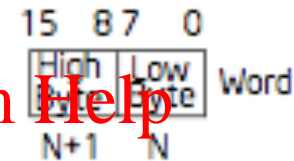
short

int/long
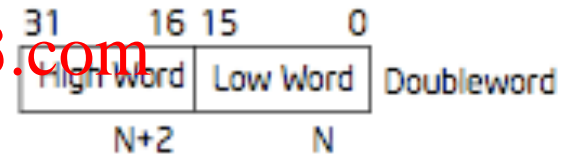
double/long long
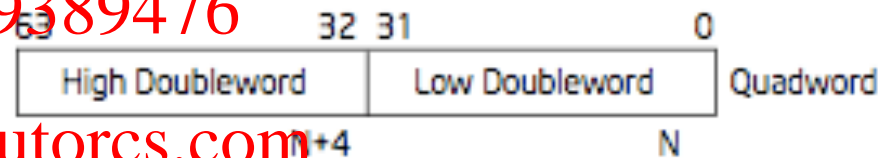
CREATE A SIMPLE PROGRAM
RUN IT STEP BY STEP

- We will create a simple program that calculates sum=x + y, where x, y, sum are numbers (short)
- Then we will compile it, and run it
- Then we will enter debug mode and run it step by step to observe the fetch decode execute cycle.
  - Watch register values after each step
- Then we will disassemble the program to view the assembly code

- C Reference:
  - https://www.tutorialspoint.com/cprogramming/index.htm
- Remember to create a cheat sheet file with commands you use as you go along. This will help you in future labs.