

一、使用说明

将写好的汇编语句放在文件 file.txt 中，输入命令“asm.exe file.txt”后，程序会在目录下生成一个 hex_file.txt 文件，里面的内容就是汇编语句的十六进制表示形式。

汇编语句的基本语法见 test.txt 文件，一些需要提前说明的地方有：

1. 支持 C 语言形式的//单行注释和/*多行注释*/。
2. 支持二进制、十进制、十六进制的数字，二进制数字以 0b(数字零)或 0B 开头，十六进制数字以 0x 或 0X 开头(a-f 的字母可大写也可小写)，无前缀的为十进制数字。
3. 支持负数，在数字前面加上减号即可表示负数，如“-0xa”表示十进制的负十。可以在 addi、load 等指令中使用。
4. 语句末尾有没有分号都无所谓。

二、指令集说明

指令	指令意义	Op[31:26]	Func[25:20]	[19:15]	[14:10]	[9:5]	[4:0]
add	寄存器加法	000000	000001	00000	rd	rs	rt
and	寄存器与	000001	000001	00000	rd	rs	rt
or	寄存器或	000001	000010	00000	rd	rs	rt
xor	寄存器异或	000001	000100	00000	rd	rs	rt
srl	逻辑右移	000010	000010	shift	rd	00000	rt
sll	逻辑左移	000010	000011	shift	rd	00000	rt
addi	立即数加法	000101	16 位 immediate			rs	rt
andi	立即数与	001001	16 位 immediate			rs	rt
ori	立即数或	001010	16 位 immediate			rs	rt
xori	立即数异或	001100	16 位 immediate			rs	rt
load	取整数数据字	001101	16 位 offset			rs	rt
store	存整数数据字	001110	16 位 offset			rs	rt
beq	相等则跳转	001111	16 位 offset			rs	rt
bne	不相等则跳转	010000	16 位 offset			rs	rt
jump	无条件跳转	010010	26 位 address				

1. add/and/or/xor rd, rs, rt; // rd = rs op rt
2. sll/srl rd, rt, shift; // rd = rt 左或右移 shift
3. addi rt, rs, imm; // rt = rs + SignedExt(imm), 此处 imm 可为正数，也可为负数
4. andi/ori/xori rt, rs, imm; // rt = rs op ZeroExt(imm), 此处的 imm 只能是正数
5. load rt, offset(rs); // rt = Memory[rs+offset], 此处的 offset 可为正数，也可为负数
6. store rt, offset(rs); // Memory[rs+offset] = rt

7. beq rs, rt, offset; // if(rs==rt) { PC = PC+4+offset<<2; }

8. bne rs, rt, offset; // if(rs!=rt)

9. jump address; // 无条件跳转，目标地址的高四位是 PC+4 的高四位，然后依次是 address 和两个 0。

10. nop; // 32 个 0。

Author: 张 斌, 949868575@qq.com