

PA1

1.1

C 语言中的 struct 和 union 关键字都是什么含义,寄存器 结构体的参考实现为什么把部分 struct 改成了 union?

struct 是一种构造类型, 它是由若干“成员”组成的。每一个成员可以是一个基本数据类型或者又是一个构造类型。

union 是一种特殊的类, 也是一种构造类型的数据结构。

一个“union”内可以定义多种不同的数据类型, 一个被说明为该“union”类型的变量中, 允许装入该“union”所定义的任何一种数据, 这些数据共享同一段内存, 以达到节省空间的目的。在本实验中希望以 cpu.eax 形式访问的和以 cpu.gpr[0]_32 形式访问的是同一个模拟寄存器, 同时 cpu.gpr[0]_16 能够访问到 cpu.eax 的低 16 位, cpu.gpr[0]_8[0]访问到 cpu.eax 的低 8 位等, 所以通过 union 结构将 eax 和 ax 以及 al、ah 寄存器实现, 同时存储 32 位数作为一个寄存器, 可以通过数组联合调用寄存器。

1.3

在实验报告中, 回答以下问题: 为浮点数加法和乘法各找两个例子: 1) 对应输入是规格化或非规格化数, 而输出产生了阶码上溢结果为正(负)无穷的情况; 2) 对应输入是规格化或非规格化数, 而输出产生了阶码下溢结果为正(负)零的情况。是否都能找到? 若找不到, 说出理由。

1) 浮点数加法: 0 11111111 111 0000 0000 0000 0000 0000 0 11111111 111 0000 0000 0000 0000 0000

浮点数乘法: 0 11111111 110 0000 0000 0000 0000 0000 0 11111111 110 0000 0000 0000 0000 0000

2) 浮点数加法: 不能找到。

3) 浮点数乘法: 0 00000001 000 0000 0001 0000 0000 0000 0 00000001 000 0000 0001 0000 0000 0000

运行结果:

```
zyh@debian:~/pa2017$ ./nemu/nemu
```

```
===== reg test =====
```

```
reg_test()      pass
```

```
===== alu test =====
```

```
alu_test_add()  pass
```

```
alu_test_adc()  pass
```

```
alu_test_sub()  pass
```

```
alu_test_sbb()  pass
```

```
alu_test_and()  pass
```

```
alu_test_or()   pass
```

```
alu_test_xor()  pass
```

```
alu_test_shl()  pass
```

```
alu_test_shr()  pass
```

```
alu_test_sal()  pass
```

```
alu_test_sar()  pass
```

```
alu_test_mul()  pass
```

```
alu_test_div()  pass
```

```
alu_test_imul() pass
```

```
alu_test_idiv() pass
```

```
===== fpu test =====
```

```
fpu_test_add()  pass
```

```
fpu_test_sub()  pass
```

```
fpu_test_mul()  pass
```

```
fpu_test_div()  pass
```

```
zyh@debian:~/pa2017$ █
```