

计算机系统基础

实验报告

PA 1

计算机科学与技术系
191220008 陈南曠

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？

结构体 (struct)：

C 语言允许用户自己指定这样一种数据结构，它由不同类型的数据组合成一个整体，以便引用，这些组合在一个整体中的数据是互相联系的，这样的数据结构称为结构体，它相当于其它高级语言中记录。

共用体/联合体 (union)：

共用体是一种特殊的数据类型，允许您在相同的内存位置存储不同的数据类型。您可以定义一个带有多成员的共用体，但是任何时候只能有一个成员带有值。共用体提供了一种使用相同的内存位置的有效方式。

struct 和 union 的区别：

1. 共用体和结构体都是由多个不同的数据类型成员组成，但在任何同一时刻，共用体只存放了一个被选中的成员，而结构体的所有成员都存在。
2. 对于共用体的不同成员赋值，将会对其它成员重写，原来成员的值就不存在了，而对于结构体的不同成员赋值是互不影响的。

结构体中所有变量是“共存”的——优点是“有容乃大”，全面；缺点是 struct 内存空间的分配是粗放的，不管用不用，全分配。

共用体中各变量是“互斥”的——缺点就是不够“包容”，即任何两个成员不会同时有效；但优点是内存使用更为精细灵活，也节省了内存空间。

部分 struct 改成 union 的原因：

我们希望以 `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 位也就是 `ax` 寄存器，`cpu.gpr[0]._8[0]` 访问到 `cpu.eax` 的低 8 位也就是 `al` 寄存器等。所以，需要将这两个变量使用一个地址存储，以达到访问同一个寄存器的目的，union 的功能十分符合。

2、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。

加法：

1) $0X7FDFFFFF + 0X7FDFFFFF$

2) 无； $0X00000001$ 是能表示的最小的非规格化数，是最小精度，无法通过加减法得到介于 $0 \sim 0X00000001$ 之间的数，使得产生阶码下移结果为零

乘法：

1) $0X7FDFFFFF * 0X7FDFFFFF$

2) $0X00000001 * 0X00000001$