Atitit.基于寄存器的虚拟机设计

真正的计算机中有许多不同种类的寄存器，但我们的虚拟机中只使用 4 个寄存器，分别如下：

1. PC 程序计数器，它存放的是一个内存地址，该地址中存放着 ****下一条**** 要执行的计算机指令。
2. SP 指针寄存器，永远指向当前的栈顶。注意的是由于栈是位于高地址并向低地址增长的，所以入栈时 SP 的值减小。
3. BP 基址指针。也是用于指向栈的某些位置，在调用函数时会使用到它。
4. AX 通用寄存器，我们的虚拟机中，它用于存放一条指令执行后的结果。

## **子函数调用**

这是汇编中最难理解的部分，所以合在一起说，要引入的命令有 CALL, ENT, ADJ 及LEV。

首先我们介绍 CALL <addr> 与 RET 指令，CALL 的作用是跳转到地址为 <addr> 的子函数，RET 则用于从子函数中返回。

为什么不能直接使用 JMP 指令呢？原因是当我们从子函数中返回时，程序需要回到跳转之前的地方继续运行，这就需要事先将这个位置信息存储起来。反过来，子函数要返回时，就需要获取并恢复这个信息。因此实际中我们将 PC 保存在栈中。如下：

手把手教你做一个 C 语言编译器（2）：虚拟机 - 博客 - 伯乐在线.htm