

# local stack slot allocation

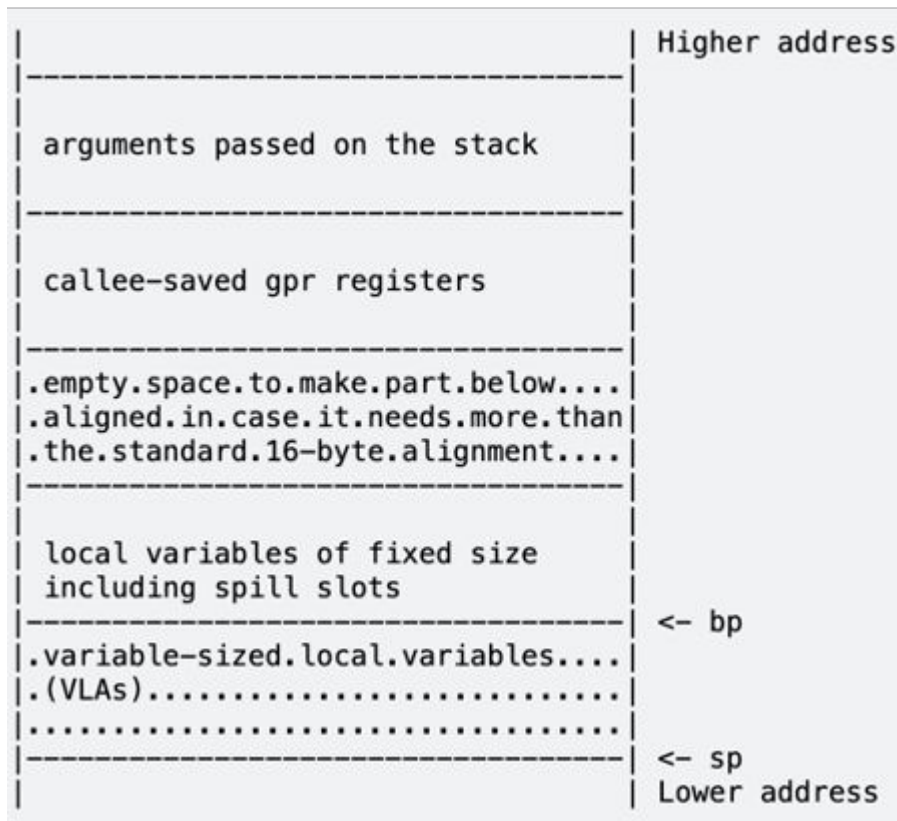
PLCT实习生:陆旭凡

# 为了解决什么问题？

<https://github.com/riscv/riscv-gcc/issues/193>

RISC-V load/store指令的offset位只有12比特，因此如果offset的值大于有符号12比特所能表示的范围，需要用一个临时寄存器来拼接出这个数值。这样就导致在访问一个栈对象的时候，引入了几条指令。

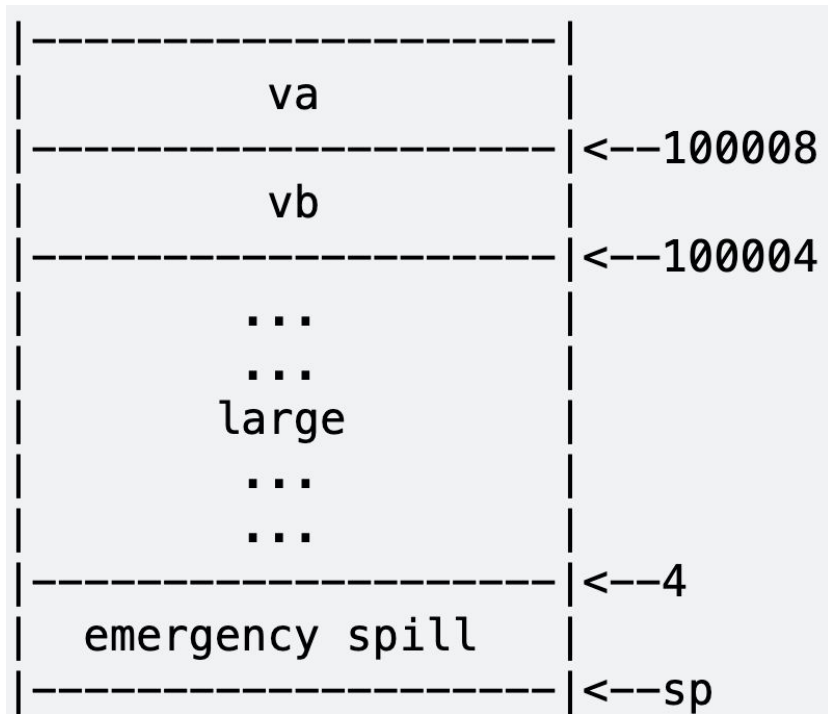
# llvm-riscv 栈帧布局



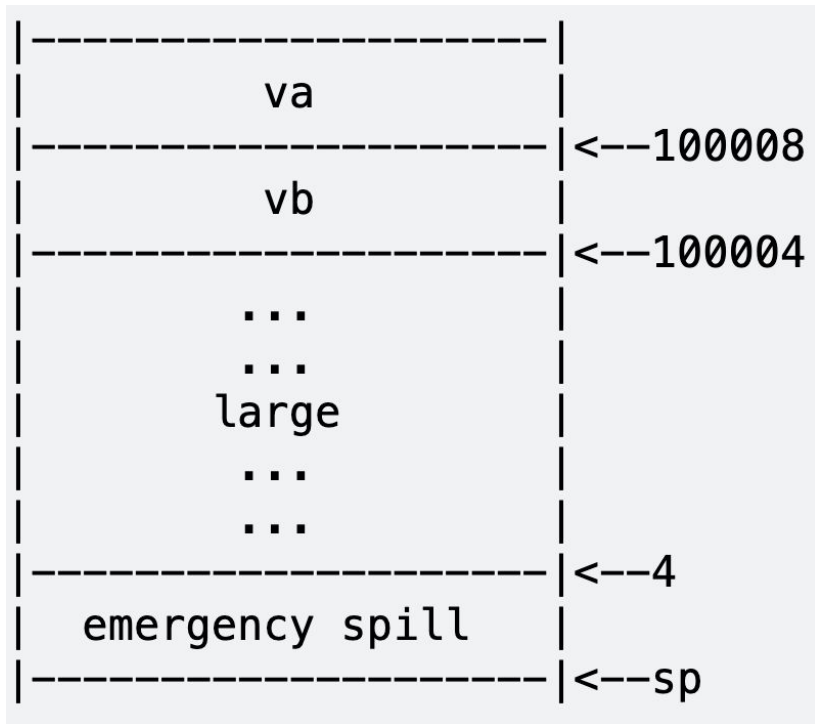
# Test file

```
define void @use_frame_base_reg() {  
    %va = alloca i8, align 4  
    %va1 = alloca i8, align 4  
    %large = alloca [ 100000 x i8 ]  
    %argp.cur = load volatile i8, i8* %va, align 4  
    %argp.next = load volatile i8, i8* %va1, align 4  
    ret void  
}
```

## 栈帧示意图



## 栈帧示意图



从示意图上可以看出，va与vb的offset仅仅相差4，因此我们可以将1000004作为基地址存入一个寄存器，然后当访问vb时，可以直接用该寄存器。而访问va时，可以用该寄存器加上Offset 4。这样就减少了对大整数的重复拼接。