

前

9). (1) 可跳转范围大约为 2^{19} + 1 个指令

(2) 偏移量为 -2^{11} 到 2^{11}

地址址空间 -2^{11} 到地址空间 $+2^{11}$

(3) lui a0, addr

addi a0, a0, addr

jalr ra, a0, 0

(4). (1). 该指令必须是在寄存器之间进行操作的指令，而不是涉及内存或外设的指令。

(2). 指令的操作数必须是寄存器，而不是立即数或内存地址。

(3). 指令的操作数必须在 RVC 指令的寄存器编码范围内，即寄存器编号不大于 15。

(4). 指令的立即数操作数必须在指定的范围内，如 ADDI 指令的立即数范围为 $-112 \sim 196$ 。

LUI 指令的立即数操作数为 $[-32768, 32767]$

(5). 指令的位移量操作数必须是 2 的倍数，并且在指定的范围内，如 BEQ 指令的位移量范围为 $[-256, 255]$ 。（以 2 字节为单位）

18).

$$\text{平均长度 } \bar{p}_i l_i = 0.25 \times 2 + 0.2 \times 3 + 0.15 \times 3$$

$$+ 0.2 \times 2 + 0.15 \times 3 + 0.05 \times 3$$

$$= 0.5 + 0.6 + 0.45 + 0.4 + 0.6$$

$$= 2.55$$

$$\text{熵 } H = - \sum p_i \log_2 p_i \approx$$

$$\approx 2.466$$

$$\text{信息冗余} = 1 - \frac{H}{\sum p_i l_i} = 0.033$$

(9). (1) 当函数嵌套调用过深或者递归调用没有正确的结束条件时，每次调用都会往栈中压入一些信息，栈的深度会不断增加。如果栈的深度超过了操作系统为程序分配的栈的大小，就会发生栈的溢出。

(2) 优化代码，避免过深的函数嵌套和递归调用。

增加操作系统为程序分配的栈大小。

采用非递归算法。

20). $r_a(F_1)$

$t_o(F_1)$

~~$r_a(F_2)$~~ $r_a(F_2)$

~~$r_a(F_2)$~~ $s_o(F_1)$

$t_o(F_2)$

$t_p(F_2)$

~~$s_o(F_2)$~~ $r_a(F_3)$

~~$s_o(F_3)$~~

~~$r_a(F_3)$~~