

22373407 王飞阳

编译第十八次作业

1.

(1)

栈式:

push A
push B
push C
Add
mul
push D
push B
push C
Add
Div
Sub
pop X

累加器式:

Load B
~~Add~~ C
Store T₁
Load A
Mul T₁
Store T₁
Load B
Add C
Store T₂
Load D
Div T₂
Store T₂
Load T₁
Sub T₂
Store X

寄存器-内存式:

Load R₀ B
Add R₁ R₀ C
~~Load R₂ A~~
Mul R₂ R₁ A
~~Add R₄ R₀ C~~
~~Load R₅ D~~
~~Div R₃ R₂ R₁~~
~~Sub R₄ R₂ R₃~~
~~Store R₄ X~~
Div R₃ D R₁
Sub R₄ R₂ R₃
Store R₄ X

寄存器-寄存器式:

Load R₀ B
Load R₁ C
Add R₂ R₁ R₀
Load R₃ A
Mul R₄ R₃ R₂
~~Add R₅ R₀ R₁~~
Load R₅ D
~~Div R₆ R₄ R₅~~
~~Sub R₇ R₆ R₇~~
~~Store R₇ X~~
Div R₆ R₅ R₂
Sub R₇ R₄ R₆
Store R₇ X

(2)

周期:

栈式:

$$3 \times 7 + 5 = 26$$

累加器式:

$$15 \times 3 = 45$$

寄存器-内存式:

~~$$6 \times 3 + 3 \times 2 = 24$$~~
$$5 \times 3 + 1 \times 1 = 16$$

寄存器-寄存器式:

~~$$5 \times 3 + 5 \times 1 = 20$$~~
$$5 \times 3 + 4 \times 1 = 19$$



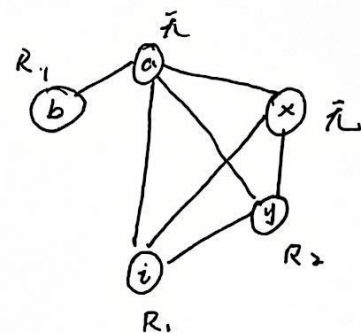
4.

将 ss 分配到栈中, 输出 0, 0. 因为栈上变量生命周期只有在函数调用期间有效, 故每次进 foo() 时 ss 会被重新初始化为 0

将 ss 分配到静态数据区, 输出 0, 1. 因为静态数据区变量生命周期为整个程序运行期, 故只会被初始化一次, 并在第一次调用 foo() 后自增.

5. 移出 b, 分配己 b
移出 a, 不分配己 b ~~a~~ x
移出 x, 不分配己 b a x x
移出 y, 分配己 b a x x y
移出 i, 分配己 b a x x y i

设寄存器为 R₁, R₂, 则



如结果为: R₁: b, i R₂: y

6.

四元式: $t_1 = B + C$
 $t_2 = A \times t_1$
 $t_3 = D \times t_1$
 $t_4 = t_2 - t_3$

X86 汇编代码:

MOV EAX, [ESP+14H]	MOV ECX, [ESP+18H]
ADD EAX, [ESP+18H]	ADD ECX, [ESP+18H]
PUSH EAX	PUSH ECX
MOV ECX, [ESP+10H]	MOV EDX, [ESP+1CH]
POP EAX	POP ECX
IMUL EAX, ECX	IMUL ECX, EDX
PUSH EAX	PUSH ECX
PUSH EAX	POP ECX
MOV ECX, [ESP+14H]	POP EAX
POP EAX	SUB EAX, ECX

(EAX 保存最终结果)

