编译原理

MiniDecaf 编译器实验报告 -- STAGE 5

2021010706 岳章乔

一、 思考题

step 11:

1.

在现有的基础上，编译器在翻译函数内数组定义，生成 `MALLOC Ti, size` 这种中间代码，对应的是

```
addi sp, sp -size
mv <Ti>, sp
```

后者是立即数。

现在如果大小可变，那么在翻译数组定义的时候，可以把 MALLOC 后面的操作数改为是一个变量，那么就有

```
MALLOC Ti, Tj
```

对应指令

```
imuli <Tj>, <Tj>, -4
add sp, sp, <Tj>
mv <Ti>, sp
```

step 12:

1.

这个跟数组寻址模式有关系。实际上，对于定长数组

```
T arr[M1][M2]...[Mn]
```

对其元素进行引用，如

```
arr[i1][i2]...[in]
```

其对应地址为

```
arr + sizeof(T) * Mn(M(n-1)(...(M3(M2*i1 + i2) + i3)...) + i(n-1)) +in
```

其也等价于

$$\text{arr} + \text{sizeof}(T) \cdot \sum_{k=1}^{n} i_k \left(\prod_{j=k+1}^{n} N_j \right)$$

无论是哪种方式求值，其都与 N1 无关，因此不需要记录第一维信息。

在本编译器的实现中，当函数数组参数没有指定第一维的时候，会把第一维填充为 1，以通过构造符号表阶段的类型检查。