扩展后的中间指令描述

一、扩展后的中间指令的数据结构

```
typedef struct
{
    enum fct f;// function code
    int l;// level
    union
    {
        double d;
        int i;
        bool b;
        enum opc op;
    }v;// value
    enum type t;//type
} instruction;
```

Fct：lit=0, opr, lod, sto, cal, Int, jmp, jpc, lar, sar, jpe ,init

Opc：ret=0, neg, add, minuss, mult, divv, mod, andand, oror, eq, neq, lt, lte, gt, gte, readd, writee,writes, notnot

Type t：_INT=0,_FLOAT, _BOOL,_OPC

栈的数据结构

```
typedef struct
{
    union
    {
        double d;
        int i;
        bool b;
        enum opc op;
    }v;// displacement address
    enum type t;//type
}stack;
```

二、中间指令的功能描述

enum fct f	int l	Union v	enum type t;	功能
Lit				取常数 v 到栈顶，包括常数的类型 t
lod				取栈上 level 为 l 偏移为 v.i 的变量到栈顶，包括类型
sto				将栈顶上的数存入栈上 level 为 l 偏移为 v.i 的位置，以被存位置的数据类型为准。存后栈顶出栈
cal				栈顶+1 的位置存 base，栈顶+2 的位置存 b，栈顶+3 的位置村 p，栈底置栈顶+1，pc 指针置 v.i

Int				栈顶增加 v.i
jmp				Pc 指针置 v.i
jpc				栈顶值为 0 的情况下，置 pc 指针为 v.i，否则弹栈
lar				用于取数组元素，栈顶位置用作数组大小，栈顶-1 位置用作当前数组元素的位置。 在取数组时需要判断是否越界。 将层次 level 为 l，数组首地址为 v.i，数组偏移为 loc 的数组元素取至栈顶-1 位置，栈减小 1
sar				用于存数组元素，栈顶位置用作数组大小，栈顶-1 位置用作当前数组元素的位置。 在存数组时需要判断是否越界。 栈顶-2 位置是需要存的数，将它存入层次 level 为 l，数组首地址为 v.i，数组偏移为 loc 的位置。 栈减小 2
jpe				栈顶值为 1 的情况下，置 pc 指针为 v.i，否则弹栈
init				重置栈上 level 为 0，偏移为 i 的位置的数据类型，数据值在重置类型的时候进行类型转换
opr		v.op=ret		将栈顶、栈底、pc 指针恢复，返回值放置于栈顶位置。 v.l 存放的是，当前函数的参数个数
		neg		栈顶元素取负
		add		栈顶上两个元素想加存至栈顶-1 位置，栈减小 1. 要有数据类型自动转换比如 float+int=float
		minuss		栈减小 1，栈顶元素-栈顶+1 元素存至栈顶。 要有数据类型自动转换
		mult		栈减小 1，栈顶元素*栈顶+1 元素存至栈顶。 要有数据类型自动转换
		divv		栈减小 1，栈顶元素*栈顶+1 元素存至栈顶。 要有数据类型自动转换和除 0 错的处理
		mod		栈减小 1，栈顶元素%栈顶+1 元素存至栈顶。 参数和结果类型都必须为 int，除 0 错处理
		eq		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		neq		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		lt		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		gte		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		gt		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		lte		栈减小 1，栈顶元素和栈顶+1 元素的比较结果存入栈顶 栈顶类型 bool
		readd		L 存储的是数据类型，根据此类型，从命令行读入对应数据至栈顶

		writee		输出栈顶元素，栈减小 1 Bool 类型需要输出 true 和 false
		writes		输出编号为栈顶元素的字符串，字符串存储在中间代码中
		andand		栈减小 1，栈顶元素和栈顶+1 元素的与计算结果存入栈顶 栈顶类型 bool
		oror		栈减小 1，栈顶元素和栈顶+1 元素的或计算结果存入栈顶 栈顶类型 bool
		notnot		栈顶元素取非

三、中间代码在文件中的存储结构

中间代码在文件中分两段存储。分别是代码部分和数据部分。

代码部分存储了所有生成的代码，以二进制存储。数据部分存储了 c1 程序中出现的所有字符串。

代码部分和数据部分以一个长度为 instruction 的全为 0xff 的串分开。

数据部分内部，每个字符串之间以一个\0 分开

比如下图，中间的 ffff 将代码部分和数据部分分开，各数据部分之间以一个 00 分开