

代码生成设计文档

本次的代码生成主要分为两个部分，一部分是生成中间代码，一部分是生成最终的mips代码。

中间代码

```
const std::string opDict[] = {
    "=",
    "+", "-", "*", "/",
    "push", "call", "ret",
    "beq", "bne", "bgt", "bge", "blt", "ble",
    "[]=", "=[]",
    ":",
    "goto",
    "printfs", "printfi", "printfc",
    "scanfi", "scanfc",
    "alloc", "recycle",
    "put", "get"
};
```

以上为中间代码生成的运算符，+ - */ =分别对应相应的运算符，push代表把op1压入op2函数的栈中，call代表调用函数，ret代表函数返回。

beq所在的一行代表对应的比较跳转，用于if，while和switch中，进行对应分支的跳转，改行的选择取决于条件的比较符号，在condition()中进行了翻译，[]=和=[]分别对应数组初始化赋值和从数组中取值。：代表生成一个label，而goto代表跳转到对应label。

printfs，printfi，printfc分别对应输出字符串，输出数字，输出字符，scanfi和scanfc分别代表读入数字和读入字符，alloc和recycle分别代表在进入函数时为数组申请内存空间和从函数中返回时释放内存空间，本质上是修改sp的值，put和get 分别代表将函数返回值放入v0和从v0中拿出函数返回值。

生成中间代码时因为循环的存在需要自己生成label，为了防止和已有的变量或函数名冲突，在自身生成的label前加上符号'\$'，以提供给循环语句使用。

因为本次生成的是四元式，对于较长表达式需要分步计算其值，在计算过程中可能存在寄存器数量不够的情况，因此需要生成临时变量以提供使用，同样为了防止重名在生成的变量tmp前加上符号'\$'。

大部分的中间代码依据语法分析生成即可，其中主要涉及的难点为表达式、数组和函数，表达式的解决方法为层层深入，将下一层的结果放入tmp内以供上一层使用。数组的难点为，本次的中间代码是四元式，但是存在二维数组，因此将二维数组视作连续的一维数组，即通过hash的方式将一维下标乘以一维长度加上二维下标得到二维数组对应一维的位置，并且乘以4以后就可以直接得到相对数组基地址的偏移。

函数的难点较多，主要是参数压栈的过程中存在参数为函数需要连续进入函数的情况，因此在读入参数时并不即时压栈，将所有的参数名存入list当中，然后当进行函数调用时申请空间然后统一压栈，这样避免了内存空间冲突。在函数调用时统一规定返回值存入v0，并且取返回值只从v0当中取出，在函数返回时即时将v0当中的值存入函数调用的左端变量或者tmp变量当中也避免了返回值丢失。

Mips代码生成

```
enum mipsOperator {
    MIPS_DATA, MIPS_TEXT,
    MIPS_SPACE, MIPS_ASCIIIZ, MIPS_WORD, MIPS_WORDARR,
    MIPS_SYSCALL,
    MIPS_LABEL,
    MIPS_SW, MIPS_LW,
    MIPS_LA, MIPS_LI,
    MIPS_ADDU, MIPS_SUBU, MIPS_ADDIU, MIPS_SUBIU,
    MIPS_MUL, MIPS_DIV,
    MIPS_MFHI, MIPS_MFLO,
    MIPS_JAL, MIPS_JR, MIPS_JUMP,
    MIPS_BEQ, MIPS_BNE, MIPS_BGT, MIPS_BGE, MIPS_BLT, MIPS_BLE
};
```

以上为使用到的指令，对应数字在数组opDict当中翻译为对应的指令，便于使用和理解并且与中间代码相区分，在使用寄存器时也直接使用如reg[V0]，reg[SP]即翻译为对应的寄存器。

genMips当中根据输入的操作符号和操作数单条的生成mips代码并输出到mips.txt内，主要的内容为遍历符号表和中间代码来生成相应的mips代码。

首先会扫描全局符号表，为全局符号表内的变量生成.word，因为是全局变量，内存可以固定，数组采用.space开空间的形式，数组对应的label代表了数组的基地址，然后扫描字符串表，将字符串写入.asciiiz当中便于之后的输出，局部变量拥有自己的offset，通过offset和sp的运算找到对应的地址进行存取值。

扫描中间代码时会进入analysisInter当中，为不同的中间代码以及中间代码对应操作数的不同情况进行分析并对应的使用genMipsCode生成mips代码，因此Mips文件的核心部分就在analysisInter部分，因为已经进行过中间代码的生成，分析时也基本只需要根据操作类型来生成能达到对应效果的mips即可。