# 实现技术

* + 1. Flex,bison（lex,yacc），c++,经过修改已经可以在aix，linux，windows上正确执行。

# 语法说明

## 变量声明总语法

init\_declarator\_list statement 或者 **statement**

init\_declarator\_list 表示变量声明

statement表示具体逻辑，包含赋值，函数，if，while，do while

## 变量声明

目前expr支持的变量类型包括：long、unsigned long、double、string，pvoid(相当于void\*)而且变量的声明必须放在开头。

其中string变量的命名方式为：XXXXX\_数字

例如：string str\_20 = “hello world”;

就表示分配20个字节来存储该字符串。

变量声明实例说明：

long x; long a; double b;

long x = 1234;

long x = (105+20)/23-2;

long x = fun(a,b) 其中参数a、 b不能为函数，可以为b=fun1(a);

## 函数入参说明

函数是程序的基本功能单元，其重要性不言而喻。

函数的两个要素是参数和返回值，在expr中参数只能为常量，变量以及变量和常量的运算表达式。参数不能为函数。

实例说明：

long a; long b; long c;

fun();

fun1(a);

fun2(a+b-c);

## 条件表达式

* + 1. 目前expr支持的条件表达式有：if(){}、while(){}、do{}while().不支持break、return、continue。
    2. 其中条件表达式的条件也必须是常量、变量以及常量和变量的运算表达式。

## 测试案例

long x=3; long y=4; if(3==x){x=y+1;}else{x=y+2;}

long x=3;if(x>3){x=x+1;}else{x=x+3;}

long x=3; {while(3==x){x=x+2;}}

long x=3;long y=9; {do{x=3+x\*3+(10\*4/3)-25+y;}while(3==x);}

# 编程接口

## 如何调用

try

{

WtExpr lk;

lk.SevVsize(1024,1024);//设置最大变量个数和常量个数

//TODO:注册自己的变量，或者是函数，如何注册详见

// FunctionMgr 类 和 ValueMgr 类

lk.InitFromFile("wt.c");//解析文件

lk.Dump(cout);//调试信息，输出执行语句

lk.Execute();//执行

printf("execute finish\n");

lk.m\_VMgr.Dump(cout);//输出所有变量信息（调试用）

}

catch(stl\_string & e)

{

printf("%s\n",e.c\_str());

}

wt.c文件内容如下：

long x=0;

string str\_32 ;

{

if(x==0)

{

x=9;

}

x = x\*2;

str\_32 ="12345" ;

x = atol("123"); //atol,strlen是我已经注册的系统级函数，可以自己注册自己的函数实现，函数最多支持6个参数

x = strlen(str\_32);

if(x==5)

{

x =10;

if(x==10)

{

x=7;

do

{

x = 9;

}

while(0);

}

}

}

## 基础数据结构

union DataValue

{

bool m\_t\_bool;

char m\_t\_char;

unsigned char m\_t\_uchar;

short m\_t\_short;

unsigned short m\_t\_ushort;

int m\_t\_int;

unsigned int m\_t\_uint;

long m\_t\_long ;

unsigned long m\_t\_ulong;

double m\_t\_double;

float m\_t\_float;

char\* m\_t\_pchar;

void\* m\_t\_ptr;

DataValue()

{

memset(this,0,sizeof(DataValue));

}

DataValue(unsigned int v)

{

m\_t\_uint = v;

}

DataValue( int v)

{

m\_t\_int = v;

}

DataValue(unsigned short v)

{

m\_t\_ushort = v;

}

DataValue( double v)

{

m\_t\_double = v;

}

DataValue( float v)

{

m\_t\_float = v;

}

DataValue(bool v)

{

m\_t\_bool = v;

}

DataValue(long v)

{

m\_t\_long = v;

}

DataValue(unsigned long v)

{

m\_t\_ulong = v;

}

DataValue(void \* p)

{

m\_t\_ptr = p;

}

DataValue( char \* p)

{

m\_t\_pchar = p;

}

DataValue(char p)

{

m\_t\_char = p;

}

DataValue(unsigned char p)

{

m\_t\_uchar = p;

}

};

## 如何注册变量

bool WtExpr::RegisterValue()

详见代码里面ValueMgr类的几个注册函数

## 如何注册函数

bool RegisterFun();

详见代码里面FunctionMgr类的几个注册函数

# 其他

系统函数我添加的很少，可以自行添加。有bug可以发wt19880111@163.com