

C-- with others

基于flex, bison以及LLVM实现的多语言(c, js, pascal, lua, 主要实现了c子集部分)编译器, 可生成符合LLVM规范的IR, 代码可执行.

Authors

- Xie zhengping (826018427@qq.com)

Build

- ubuntu 16.04 (recommended)
- llvm-3.9
- jsoncpp

具体环境配置(完全的)

```
wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key | sudo apt-key add -
sudo apt -o Acquire::AllowInsecureRepositories=true update
sudo apt-get install clang-3.9 lldb-3.9
sudo apt-get install libjsoncpp-dev
sudo apt-get install flex
sudo apt-get install bison
sudo apt-get install zlib1g-dev
sudo apt-get install libncurses5-dev
```

build

```
./install
and
./install clean
```

使用

```
./compiler c_test.in
或者
./compiler psc_test.in等.
```

流程控制图产生于/tmp中,为dot文件.

可以查看dot文件的方式很多.

`./vis/disp.html` 查看AST.

(AST生成在c++代码中使用的直接路径,会由于环境改变而失效.如果想要正确使用,可以去main.cpp中更改路径,但是如果只为演示效果,下面的`disp.html`是可以直接浏览器查看的)

C Grammar Support

Details about grammar support can be found in bison source file `parse.y`

Type

- Integer
- String
- Bool
- Struct

Control flow

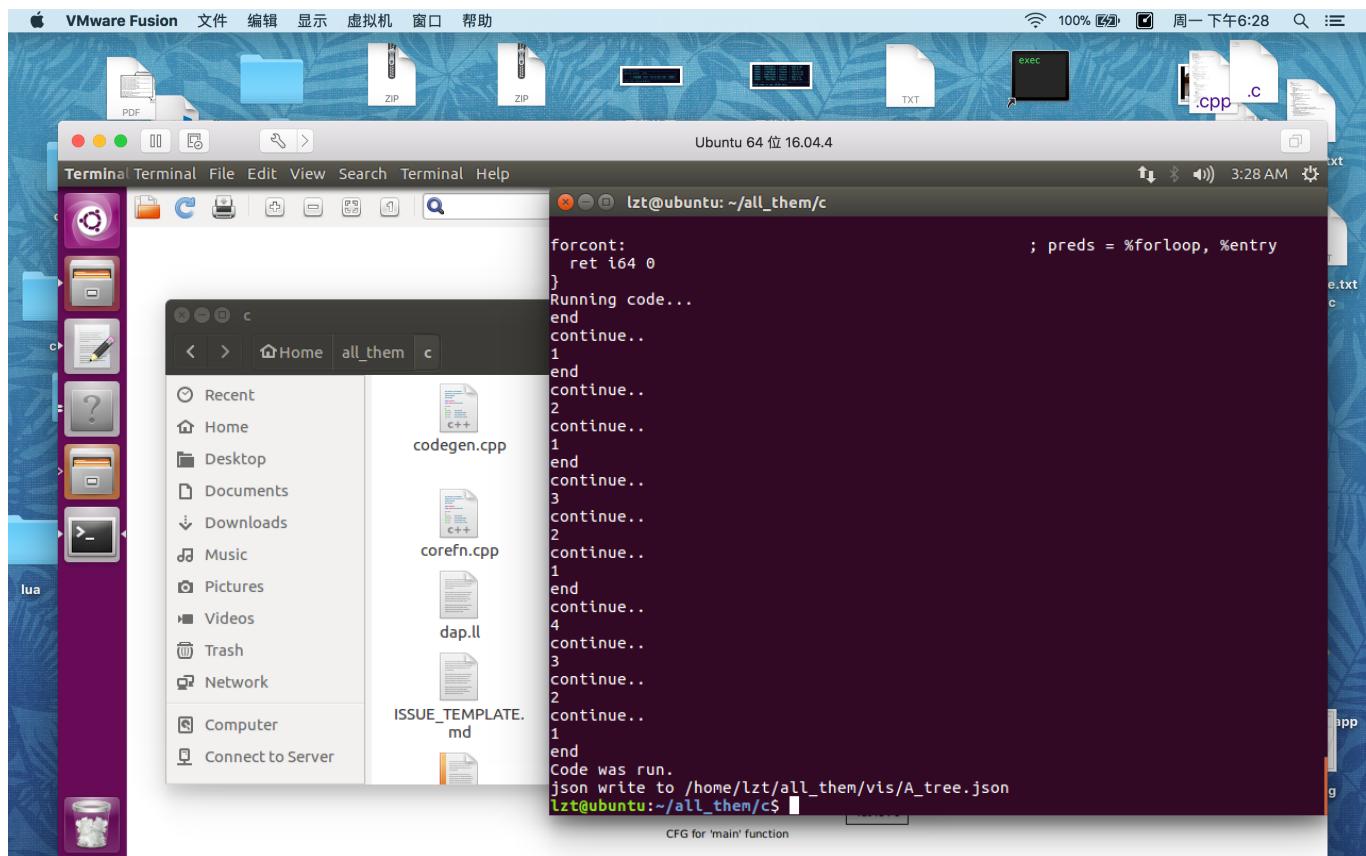
- if statement
- for statement

Others

- function declaration/execution
- binary operators
- Output functions

Example

c



```
extern void printNum(int x)
extern void printStr(string str)
extern int printf(string str)
struct sct
{
    int ia
    string stra
    string strb
}
int func(sct tmp)
{
    if(tmp.ia>0)
    {
        printStr(tmp.stra)
        printNum(tmp.ia)
        tmp.ia = tmp.ia-1
        func(tmp)
    }
    else
    {
        printStr(tmp.strb)
    }
    return 0
}
int main()
{
    int i
    for(i = 0;i<5;i=i+1)
    {
        sct t_sct
```

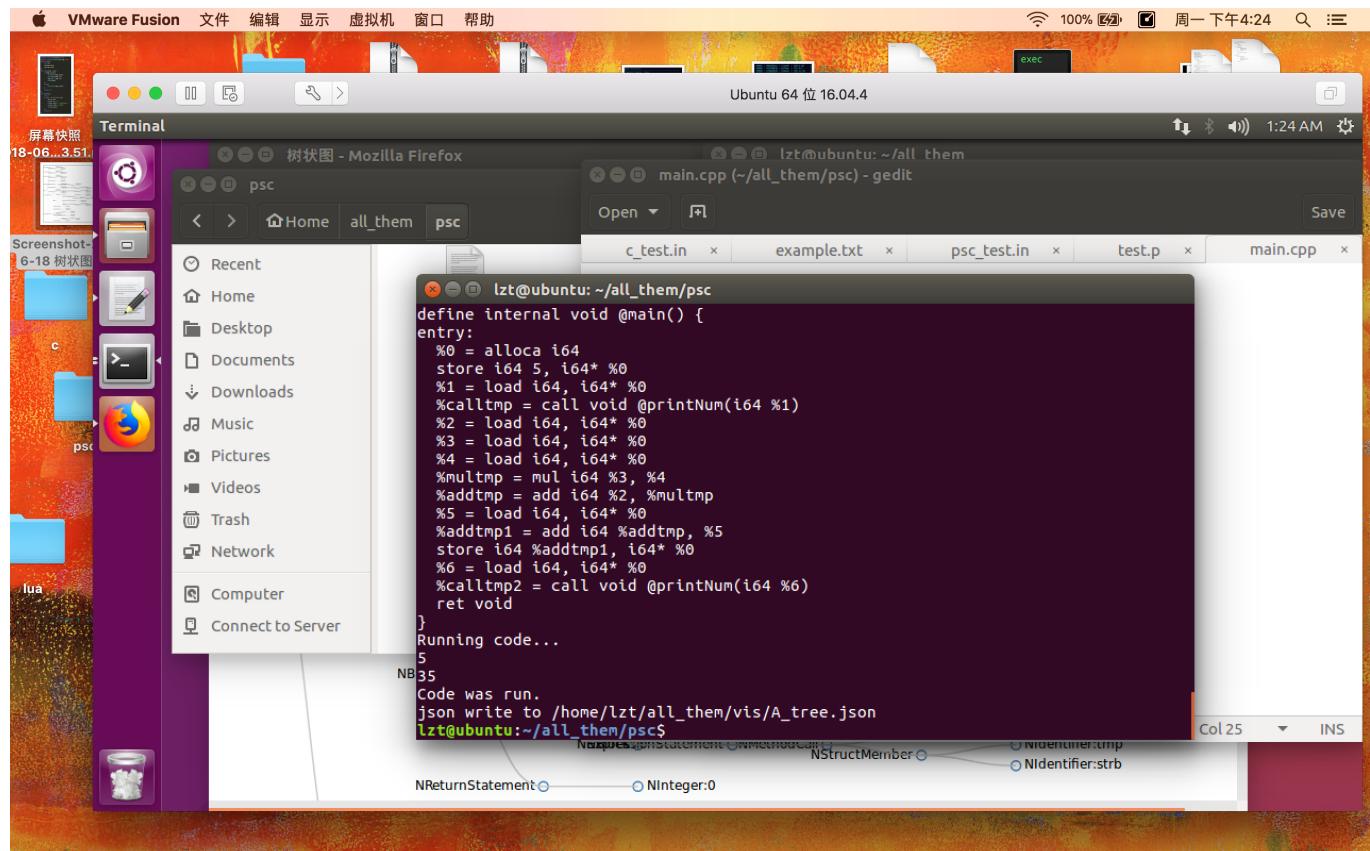
```

        t_sct.ia = i
        t_sct.stra = "continue.."
        t_sct.strb = "end"
        func(t_sct)
    }
    return 0
}

```

以下三种语言完成度很低.但是对其进行完善并不困难.

psc

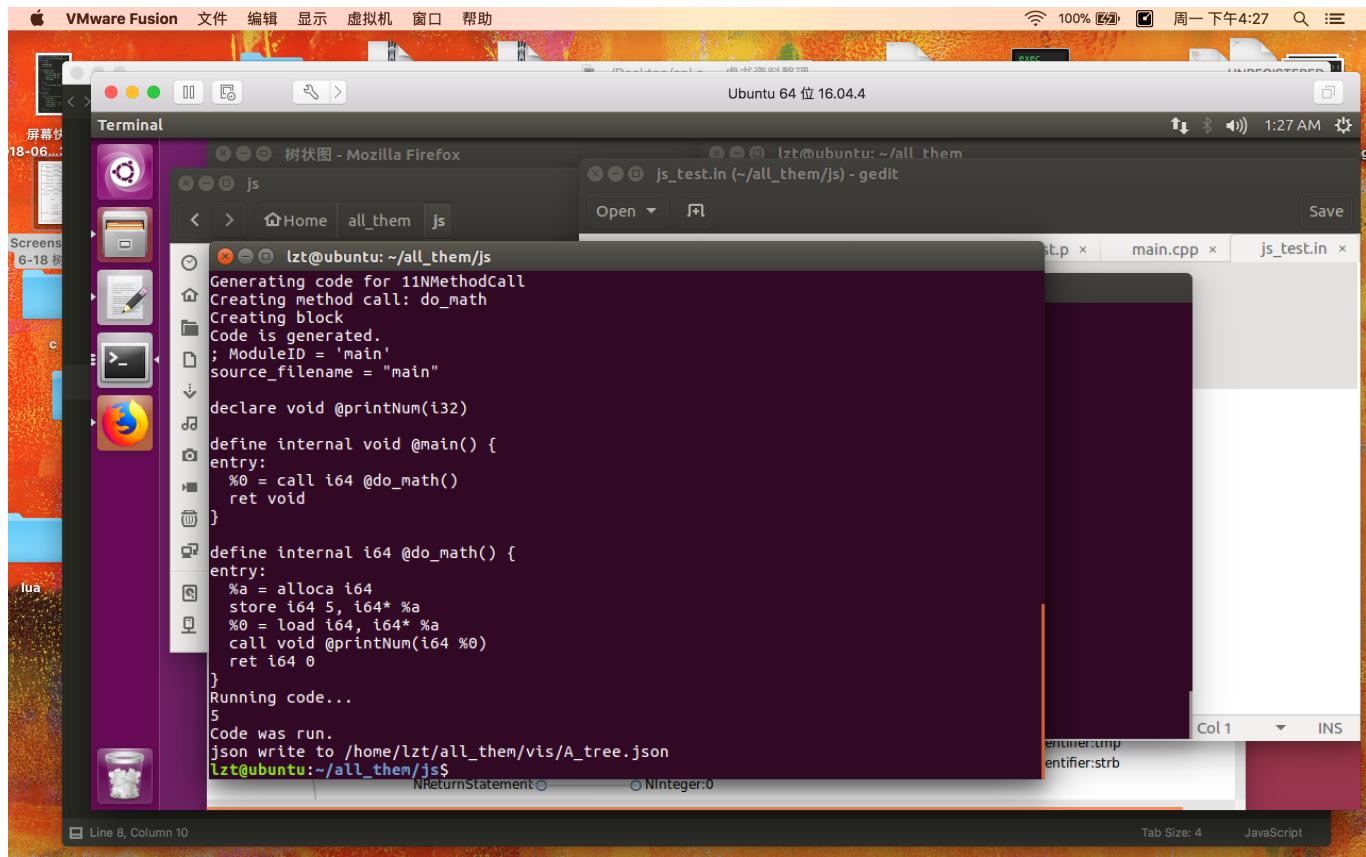


```

var
    i : int
begin
    i := 5
    printNum(i)
    i := i+i*i+i
    printNum(i)
end

```

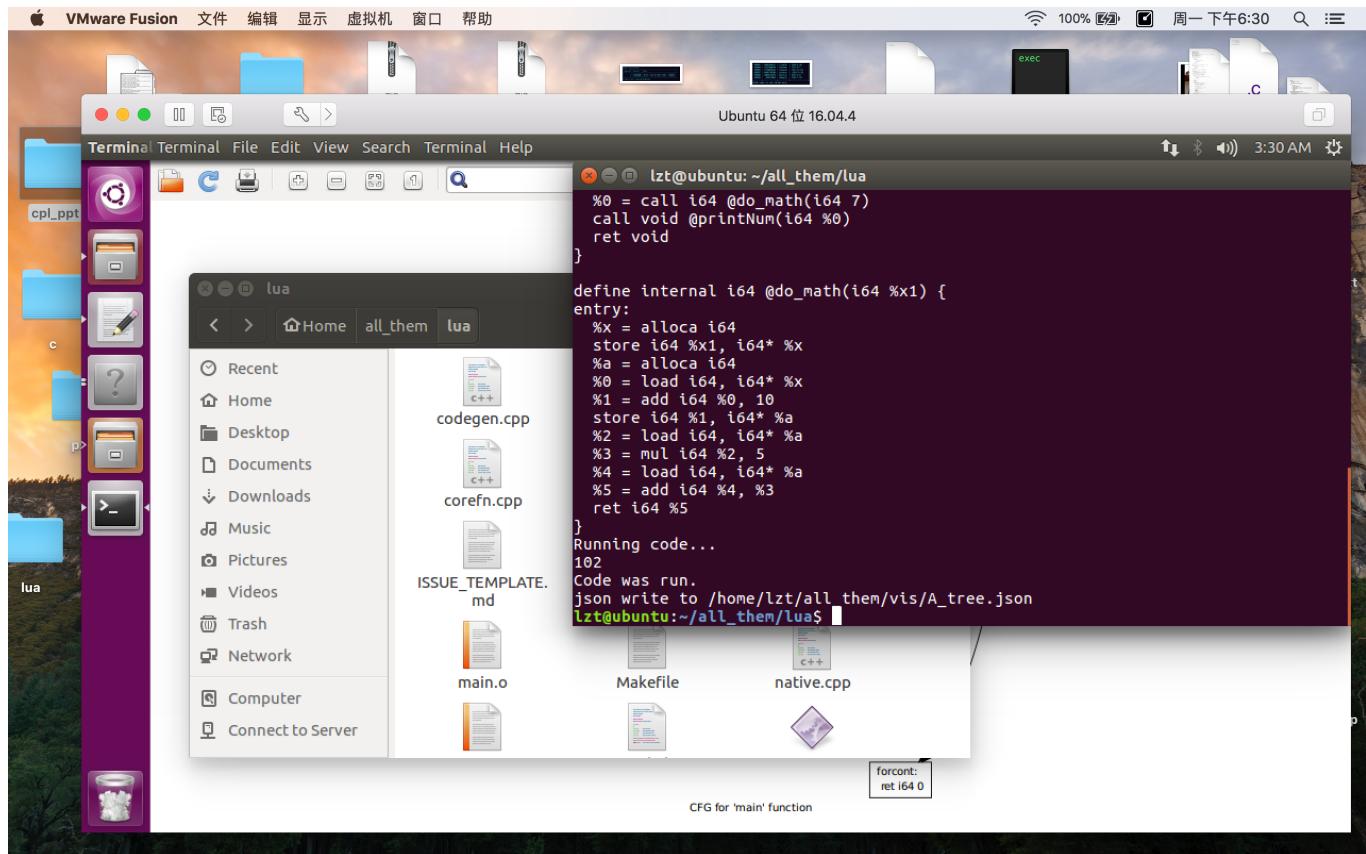
js



```
function do_math() {
    var a = 5
    printNum(a)
    return 0
}

do_math()
```

lua



```

function do_math(x)
    a = x+10
    return a+a*5
end

printNum(do_math(7))

```

TODO list

将多个编译器的后端合为一体
(他们之间的diff小于三行,但苦于没有找到较好的连接方式)

其他三个语言的完善.

图形界面.

错误提示.

References

- LLVM官网教程

<http://llvm.org/docs/tutorial/index.html>

- 使用 LLVM 框架创建一个工作编译器，这里有IR的教程

<http://www.ibm.com/developerworks/cn/opensource/os-createcompilerllvm1/>

- 一个完整的flex+bison+llvm的例子

<http://www.cppblog.com/woaidongmao/archive/2009/11/11/100693.aspx>

- 例子源代码

https://github.com/lsegal/my_toy_compiler

- 另外一个例子

<http://my.oschina.net/linlifeng/blog/97457>