Language 414

简介

这种特殊语言旨在满足北亚利桑那大学（NAU）EE414 和 EE514 课程的需求，减轻手动编写汇编指令的繁琐过程。起初，我曾尝试使用 C 语言的 "gcc"，但这并不是解决办法。课程使用的指令集与完整的 RISC-V 指令集之间的不完全一致。此外，"gcc "缺乏绑定变量和寄存器以直接存取寄存器的功能。

于是，我设计了 Language 414，作为为这门课量身定制的解决方案。尽管 Language 414 非常简单，甚至可以说是简陋，没有函数、for 循环和字符串支持等功能，但它大大简化了编写汇编指令的过程。

由于这个项目的时间很短（一个周末就完成了），可能会有一些错误没有被发现。如果您能协助发现或纠正这些潜在问题，我们将不胜感激。

变量

在使用变量之前不需要声明。变量储存在寄存器内。通常寄存器 x1-x31, t0-t6, s0-s11, 以及 a0-a7 用于储存变量。也可以通过修改位于 “Excuteable.py” 首部的列表来改变储存变量的寄存器。

例子:

```
i = 0
a = (i + 12 + 4)
```

变量的名称必须以字母开头，不支持下划线。

Assign

在默认情况下，编译器会自动为变量分配寄存器，但是你也可以通过使用 `assign` 语句手动指定变量的寄存器。例如，

```
assign SrcArray = reg(x8)
```

```
assign reg(x9) = target
```

While 循环

例子：

```
i = 0
while i < 12:
        a = (i + 12 + 4)
        i += 1
target = a
```

Break 语句可以手动打断循环

```
i = 0
target = 1
while i < 6:
        target = 1 * 2
        if i == 3:
                break
        i += 1
```

continue 语句可以打断这一次循环，并开始下一次

```
i = 0
target = 1
while i < 6:
        i += 1
        if i == 3:
                continue
        target = 1 * 2
```

Language 414 不支持在 while 循环中使用 else 语句。

If...Else

Language 414 支持通常的逻辑关系：

- 等于: `a == b`
- 不等于: `a != b`
- 小于: `a < b`
- 小于等于: `a <= b`
- 大于: `a > b`
- 大于等于: `a >= b`

同时也可以使用 `or` 和 `and` 关键字来复合逻辑关系。

if 语句 "是使用 if 关键字编写的。else 关键字捕捉前面条件没有捕捉到的任何内容。

```
a = 5
b = 6
if b >= a:
        target = 0
else:
        target = 1
```

Language 414 不支持 `else if` 关键字

缩进

和 Python 一样，Language 414 依赖于缩进（行首的空白）来确定代码的范围。编译器只会识别行首空白字符的数量，不会区分他们是 tab 还是空格。因此不要在代码中混合使用 tab 和空格来进行缩进。

数组

数组是形如 `A[234]`的表达式。在使用数组前需要手动指定数组的首地址。当使用数组时，程序会把数组表达式中的两个部分相加，然后加载该地址的内容。下面的例子将 5 储存在地址 1012 中。

```
A = 1000
A[12] = 5
```

下面的代码可以起到同样的作用。

```
A = 1000
12[A] = 5
```

```
12[1000] = 5
```

```
A = 1000
a = 12
A[a] = 5
```

注释

不同于大多数的编程语言，Language 414 的编译器不会忽视代码中的注释行。编译器会保留原代码中的注释，并在生成的汇编指令中呈现他们。

```
a = 5
b = 6
# Beginning of the If.
if b >= a and a < b:
        target = 0
else:
        target = 1
```

```
addi x1, x0, 5
addi x2, x0, 6
// Beginning of the If.
addi a7, x0, 0
blt x2, x1, BTMP0
addi a7, x0, 1
BTMP0:
```

注释必须写在单独的一行内，不能写在语句后写注释。不支持多行注释。

函数

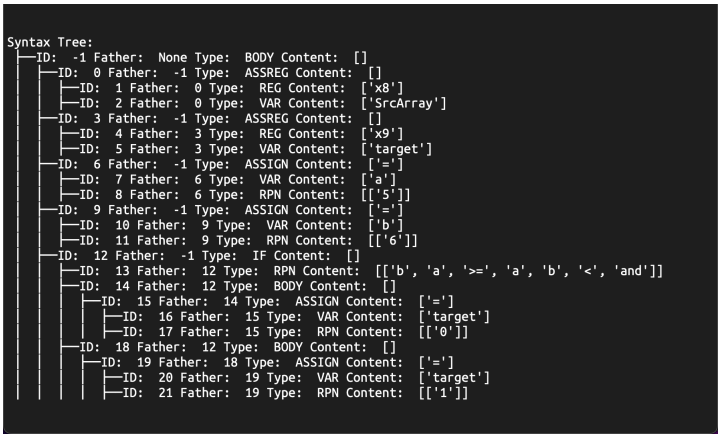
Language 414 不支持函数

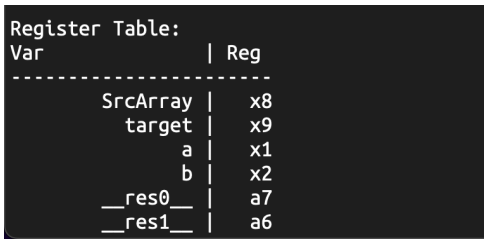
如何使用

将你的原代码放到和 Python 程序同一个文件夹内，然后运行 “main.py”。然后输入原代码文件名。

```
python3 main.py
Source File Name: test.414
```

程序会生成一个用于储存汇编指令的 “.out” 文件。语法树和变量寄存器对应表会被输出到终端中。

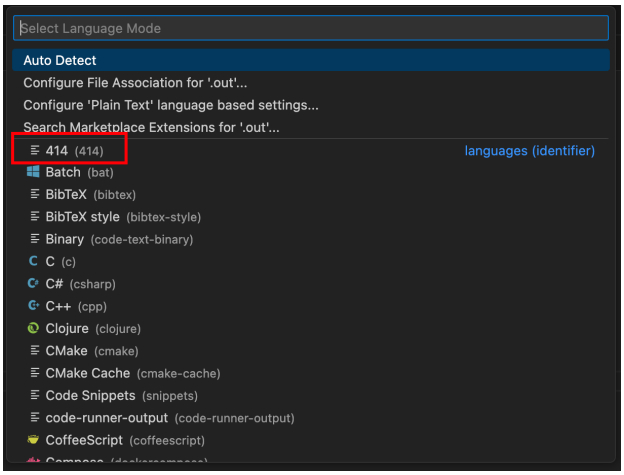




Visual Studio Code 语法高亮

可以通过手动安装Visual Code扩展来实现语法高亮。扩展仅用于语法高亮，不会检查语法错误。

要使用语法高亮，原代码的文件应当以 “.414” 结尾。你也可以在 VS Code 中手动选择 414 语言。



将 “language-414” 文件夹复制到 “~/.vscode/extensions” 来安装扩展。然后重启 VS Code