Compiler Manual

Autores:

- Luis Miramontes –System Architect –Tenkoni
- Fernando Bustamante –System Integrator –Farlan11
- Antonio Molina –Project Manager –marmolinaa
- Sergio López –System Tester -sergioulises

Index

Objective.

Show how the compilator compis team works.

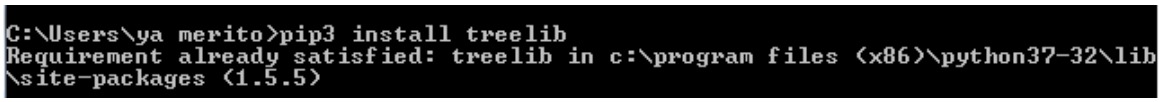
introduction.

Next through captures it is about making known how you as user can interact with our development program (compiler), it mentions what are the prerequisites to be able to clone and download the file from a version program (github), It also describes how to install, configure and run it on your computer from command lines.

User manual.

- **Installation prerequisites.**
- The compiler is designed to run in Unix operating systems (tested with OSX 10.11.4) using Python 3.6.5
- Install treelib

  Pip3 install treelib



- **Installation**.

  To install the compiler you need to clone the repository that is in github, for that you must do what is mentioned below.

First we open "git bash" and place ourselves (with the cd command) in the direction where we are going to work and write 'git clone and the repository address' as shown.

```
Sergio@Sergio MINGW64 ~ (master)
$ cd desktop

Sergio@Sergio MINGW64 ~/desktop (master)
$ git clone https://github.com/hiphoox/compilers2019_2.git
Cloning into 'compilers2019_2'...
remote: Enumerating objects: 234, done.
remote: Counting objects: 100% (234/234), done.
remote: Compressing objects: 100% (157/157), done.
remote: Total 1679 (delta 96), reused 187 (delta 75), pack-reused 1445
Receiving objects: 100% (1679/1679), 71.66 MiB | 2.06 MiB/s, done.
Resolving deltas: 100% (765/765), done.
```

With the command ls it is observed that a folder called compilers2019_2 /

```
compilers2019_2/
```

By entering this folder you will have access to all the files of the program.

- **Execution**.

  To run the file, the following command is used.

python compiler.py c_file_name [-h] [-o output_name] [-s | -t | -a]

In our compiler we handle different flags, each one is explained below.

We have [-h] which works to see the help as shown below.

```
C:\Users\ya merito\Desktop\compilers2019_2>python compiler.py -h
usage: compiler.py [-h] [-o OUTPUT] [-s | -t | -a] file

positional arguments:
  file                  name of the C file

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT, --output OUTPUT
                        Write output to <OUTPUT>
  -s, --ensamblador     Generates the assembler file
  -t, --tokens          Generates the token list and display it on the
                        terminal
  -a, --ast             Generates the AST and display it on the terminal

C:\Users\ya merito\Desktop\compilers2019_2>_
```

As we can see, it includes what all the flags are and then we will test each one of them. We will use a file called test.c. which will be the file that we will use as an example to execute our flags.

[-o output_name]

We execute

Python compiler.py test.c –o output_name

[-s assembly]

We execute

Python compiler.py test.c -s

```
C:\Users\ya merito\Desktop\compilers2019_2>python compiler.py test.c -s
Assembly succesfully generated
```

In this way we generate our assembler.

[-t tokens]

We execute

Python compiler.py test.c -t

```
C:\Users\ya merito\Desktop\compilers2019_2>python compiler.py test.c -t
['keyword_int', 'keyword_main', 'parentheses_open', 'parentheses_close', 'bracke
t_open', 'keyword_return', ('integer', 2), 'semicolon', 'bracket_close']
Token list succesfully generated
```

In this way we generate our list of tokens.

[-a ast]

We execute

Python compiler.py test.c -a

```
C:\Users\ya merito\Desktop\compilers2019_2>python compiler.py test.c -a
Program
└── Function:main
    └── Expression:return
        └── Constant:2

AST succesfully generated
```

In this way we generate our AST.

- **Tests**.

  To execute the tests, only the following command must be set.

We must place ourselves in the test folder with the CD command as shown below and then execute the following command.

python compiler_test.py

```
C:\Users\ya merito\Desktop\compilers2019_2>cd test

C:\Users\ya merito\Desktop\compilers2019_2\Test>python test_compiler.py
7 pruebas del generator ejecutadas correctamente
7 pruebas del Lexer ejecutadas correctamente
6 pruebas del parser ejecutadas correctamente
4 pruebas invalidas ejecutadas correctamente
.
----------------------------------------------------------------------
Ran 1 test in 0.420s

OK
```