UNIVERSIDAD DE PUERTO RICO

RECINTO UNIVERSITARIO DE MAYAGÜEZ

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y DE COMPUTADORAS

DOINGNOTHING\_LANGUAGE

ICOM4036 (Programming Languages)

Prof. Wilson Rivera Gallego

Stephanie Vargas Villarini

Andres Ayala Garcia

Efrain Oliveras

Table of Contents

Introduction……………………………………………………………..…………………………………………………………….................1

Language Machine…………………………………………………..………………………………………………………………………......2

Language syntax…………………………………………………….………………………………………………………………………………3

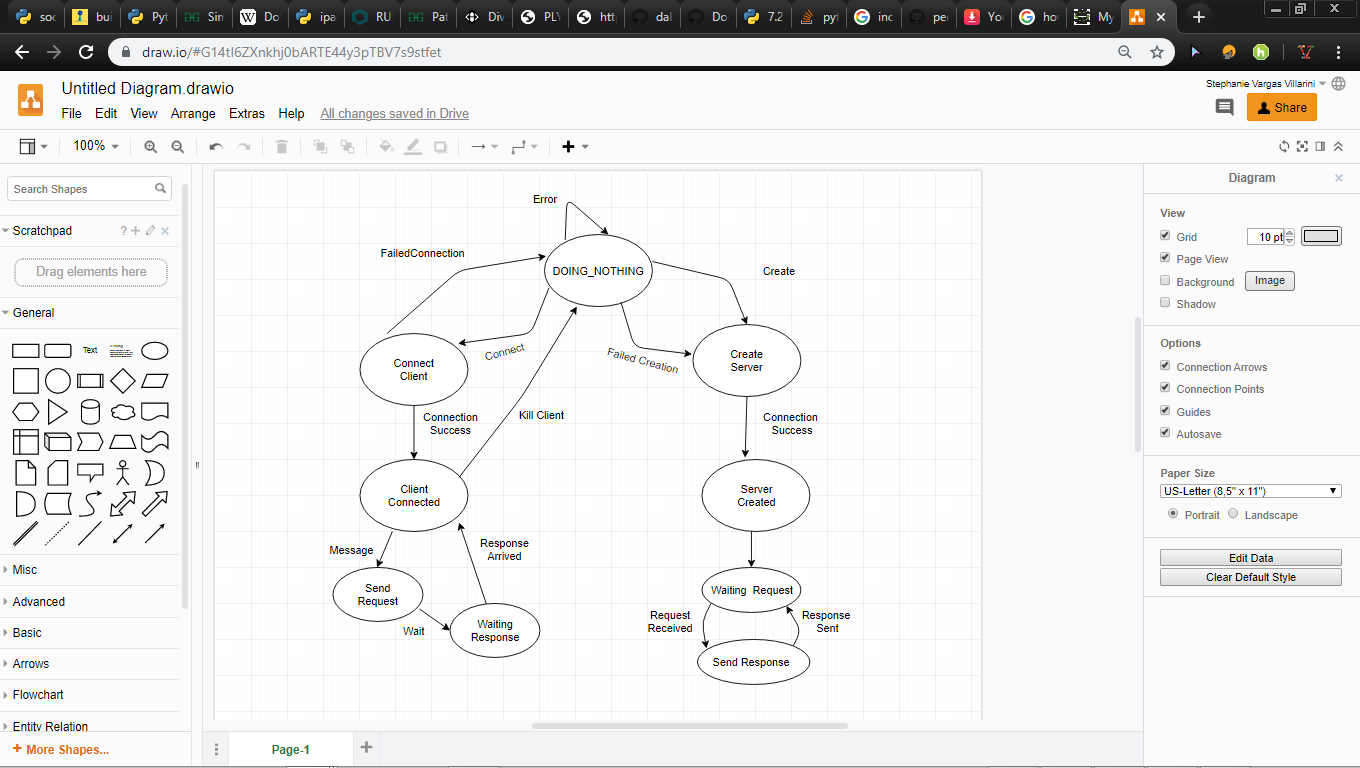
Example of running program………………………………………………………………………………………………………………….4

INTRODUCTION:

Learning how to like networking may be a difficult task, especially for ICOM/Software/CIIC students who like programming. This is in part because students have the idea that networking doesn't require designing, developing and programming, in addition that they find it really different from programming. Other important aspect to consider is that it is difficult to find certifications, classes, and laboratories that adjust to different parts of networking.

Educators are implementing more novel, simplified and interactive teaching techniques into their curriculums. This is with the purpose to engage and motivate students to learn. These initiatives, especially simplifying has shown effectiveness in motivating students to learn. With this in mind, it is proposed the creation and implementation of doingNothing, as a programming language for people to have a simple understanding of the basics of networking. The main goals for such programming language are the following: Implement simple and intuitive syntax. –Simplifying the learning of client-server communication.

Language Machine:



Language Syntax:

**syntax for connecting client:**

*Dot-decimal*

*Notation IPV4*

*number*

*Keyword Token Parenthesis Keyword Tokencomma*

connect <client\_name> to (port, ipv4address) or connect <client\_name> to port, ipv4address

*Tokencomma*

*Keyword UserDecides TokenParenthesis Keyword UserDecides number dot-decimal*

*notation IPV4*

*dot-decimal*

*notation IPV4*

**syntax for creating a server:**

create <server\_name> at (port , ipv4address) or create <server\_name> at port, ipv4address

**syntax for sending a message:**

send message “<message given by user>”

**syntax for response of message from server:**

send response message “<response message>”

**syntax for waiting response from server:**

wait response

**syntax for client requesting information from server:**

wait request

**syntax for exiting client/server or killing a specific port respectively:**

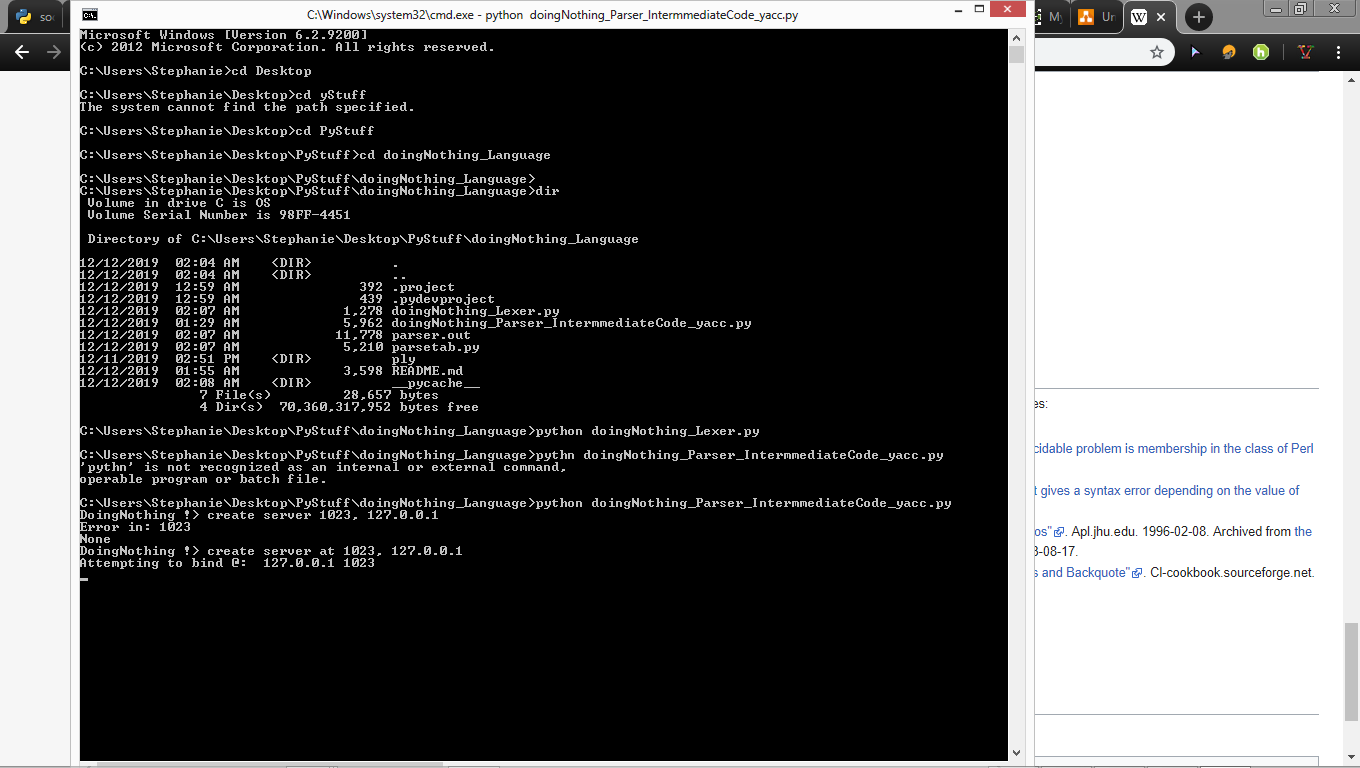
exit or kill <name\_given\_when\_creation\_was\_done>

Note:

When I say user decides is that he/she specifies what he/she wants, also he/she gives the port number and IPV4 he/she wants. Keywords and Tokens are all required by the compiler for it to not give error. The or in the middle means these instruction can be given in any of the two ways.

Program example using doingNothing language:

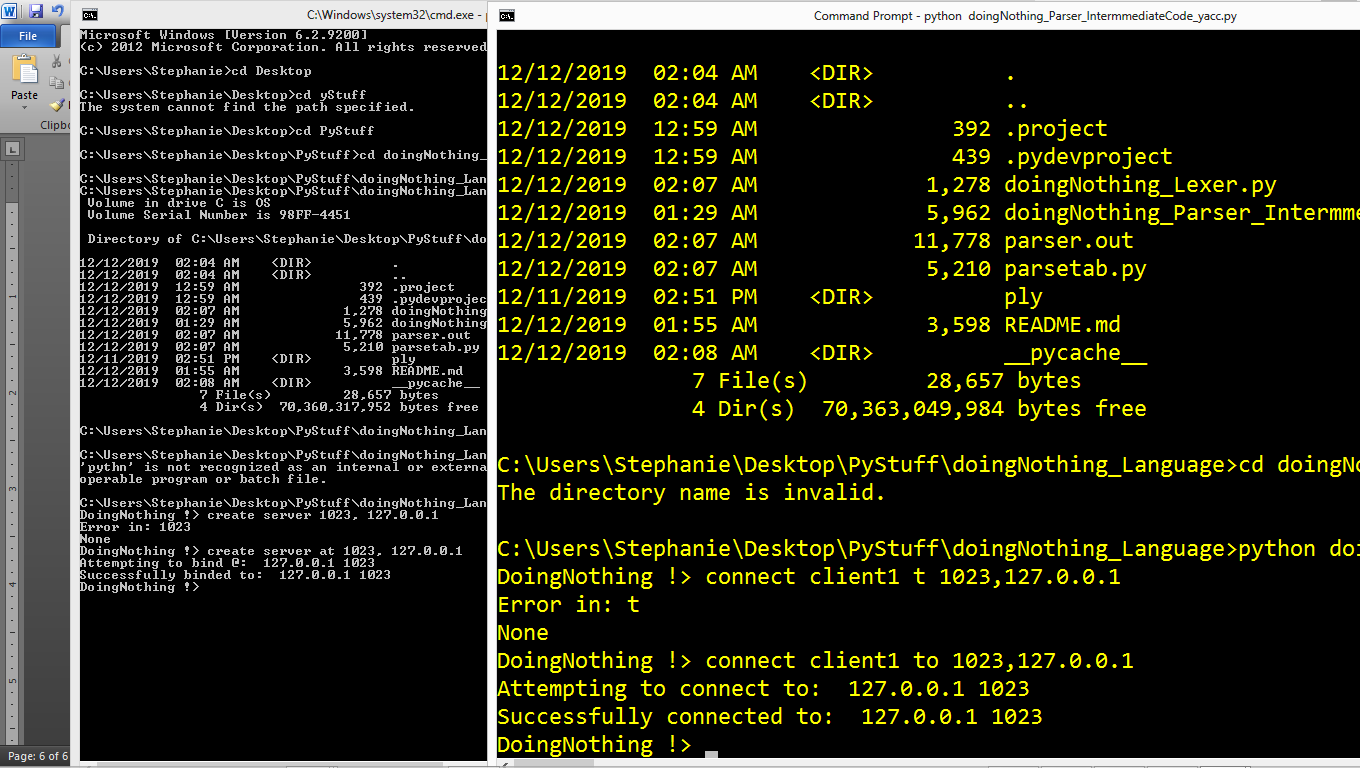
Note: This language needs two terminals one for the client other for the server.



Waiting for client or device for communication

Command to start the language

Wrong syntax so it created error

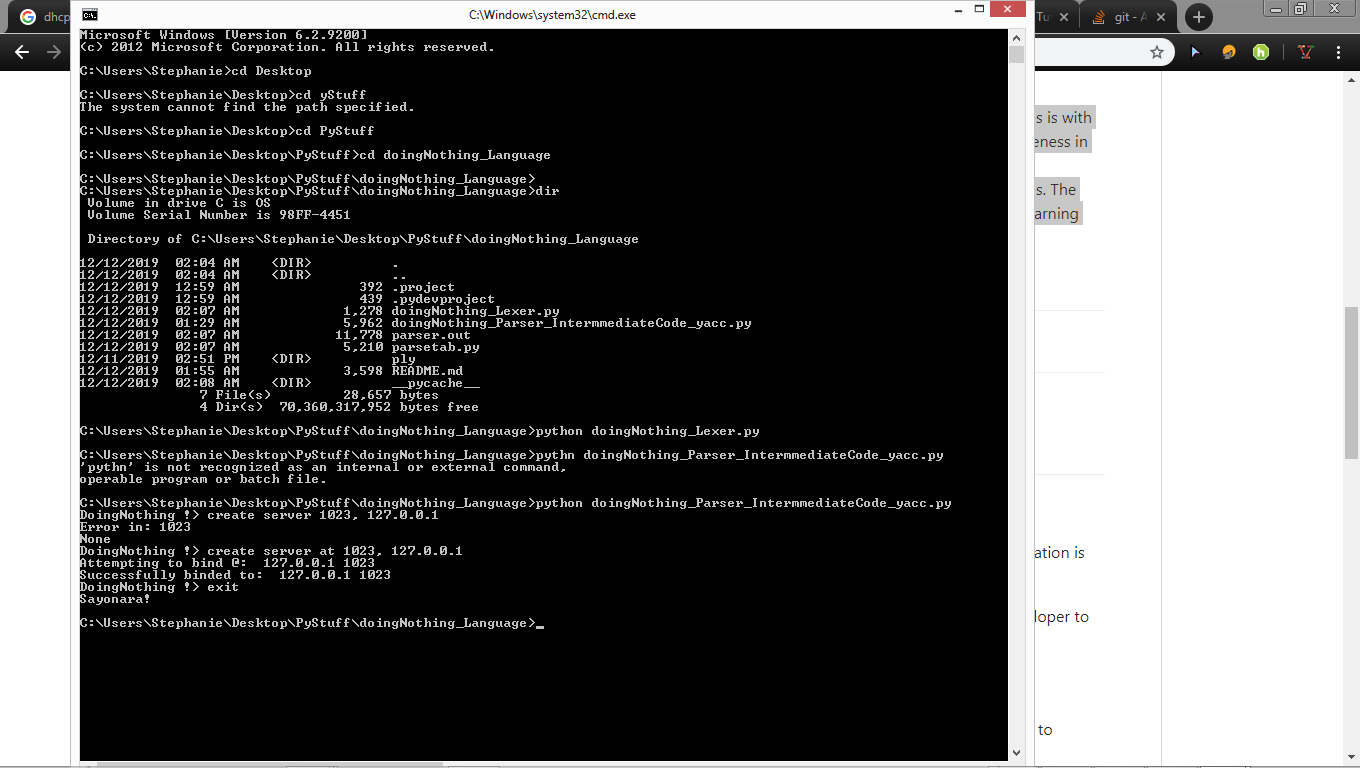
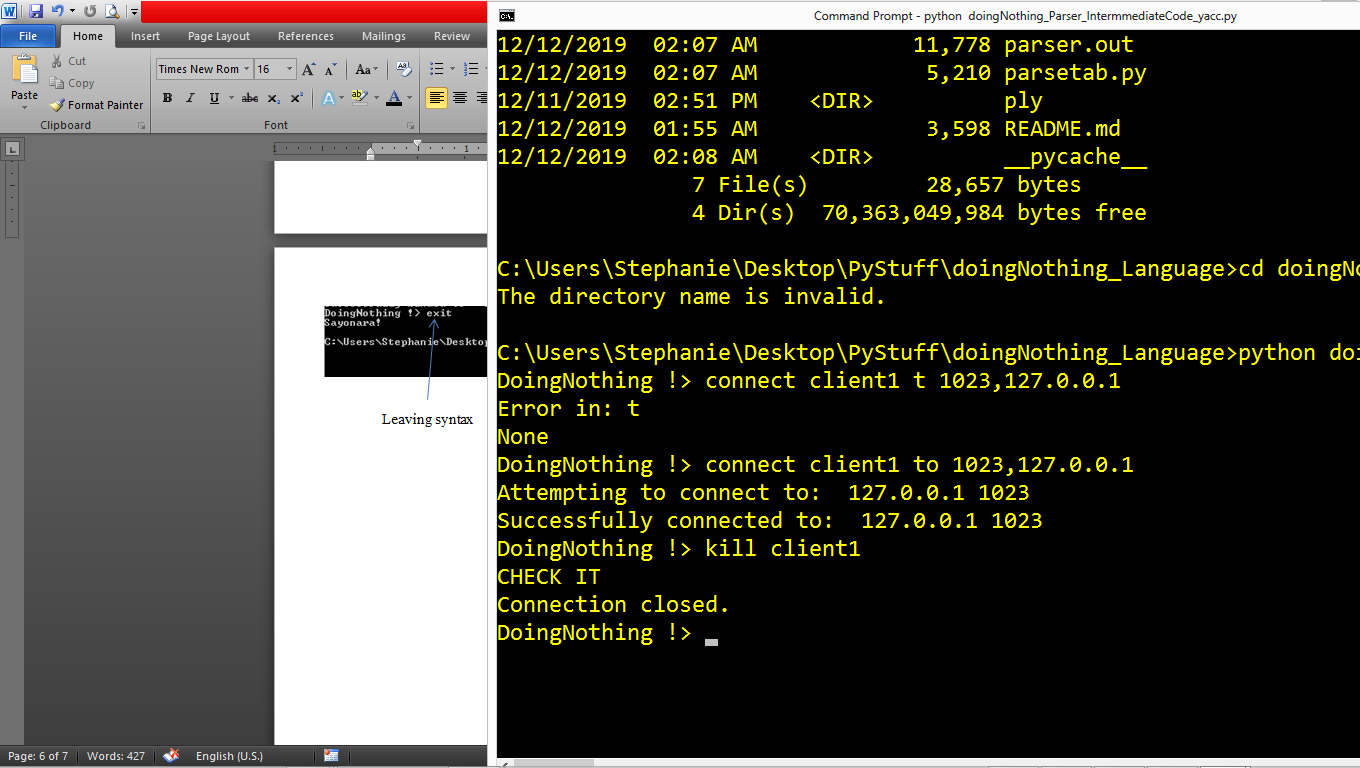


Wrong syntax so it created error

Server binded to client communication was established.

Message that connection succeded

Command to connect



Killing a Port

Leaving syntax

Note:

Once you kill a port it can’t be used again part of the language to make it fun!

Future Work:

* Successfully send a message.
* Connect multiple clients to multiple servers (In other words developing multiple communication).
* Allowing a client to have multiple open sockets in order to request many operations to the server.
* Create an external server, and make the user pick between connecting to a local server or an external server.