Usage

Introduction

This script `node.py` implements Ricart-Agarwala(token passing version) algorithm for mutual exclusive. And the critical data is just a python dictionary(the key is a string, the value is an integer).

The system is communicated by sending and receiveing messages based on UDP protocal.

Requirement: `python3`

Running script

```
python3 node.py (bind ip):(bind port) [-t]
```

The ip and port should be what you want use to receive message. And at the same time, the program will use `(bind ip):(bind port)-1` as send address. `[-t]` paremeter is used to set token. It's user's responsbility to initial the node correctly. **YOU SHOULD ONLY CREATE ONE TOKEN.**

Specially, if you want to run a local node with a token which only used in one pc, you should input like

```
python3 node.py 127.0.1.1:8000 -t
```

Command

You can input some commands to login, logout, get data or change data.

The data is stored in local memory, but when it is changed, the changes will be broadcasted to the whole nodes network.

Every command should start with `/`.

/login

This command is used to login to other nodes network. The address your input should be any node in the network you want to join. As soon as it log into the network, other nodes in networks will add it into the network and it will get initial information(nodes, data) from the node you input.

FORM:

```
/login hostname:port
```

/logout

This command is used to logout from the network. After that, the information stored in local memory will be deleted.

```
/logout
```

/get_data

Critical Section. This command is used to get data.

FORM:

```
/get_data s
```

EXAMPLE:
If you want get `data['ttt']`, just input:

```
/get_data ttt
```

/change_data

Critical Section. This command is used to get data. The value should only be integer.

FORM:

```
/change_data s v
```

EXAMPLE:
If you want change `data['ttt']` to 999, just input:

```
/change_data ttt 999
```

Message

Message form:

```
(hostname:port)@logical_time@message_text
```

As you can see, every message will include self's receiving address and its logical_time.

Message text

/login

```
/login hostname:port
```

/logout

```
/logout
```

/init

```
/init hostsdata jsondata tokendata requestdata
```

/data

```
/data x 5
```

/token

```
/token
```

/request

```
/request
```

Known issues

- If the node which is not holding the token crashes, it doesn't matter. The rest of nodes should work fine, but they will try to sending useless messages to the 'dead' node.
- If the node holding the token crashes, there is no effective solution to save the system automatically. A possible solution is to create a new node with a token and let it join into the network.