**PyDi**

HELP FILE

**Project:** Distributed File System  
**Language:** Python  
**Operating System:** Windows, Linux, Mac OS

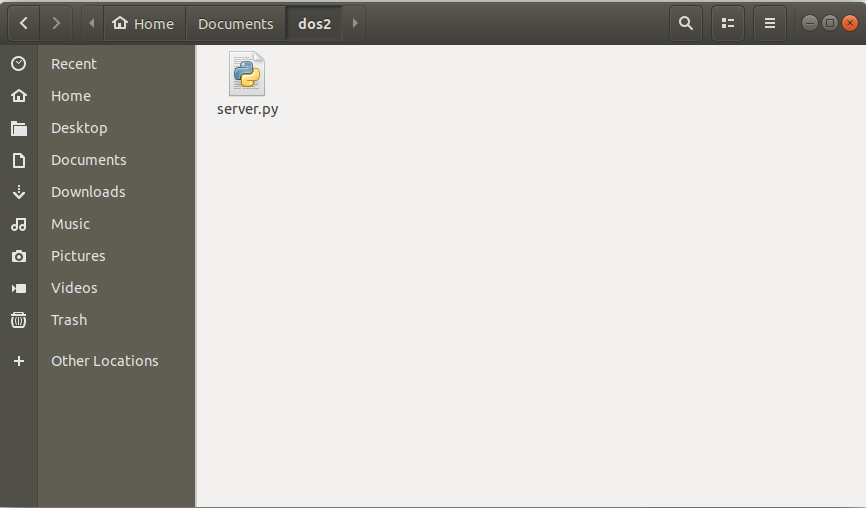
Setup and Installation

**Server - Side**

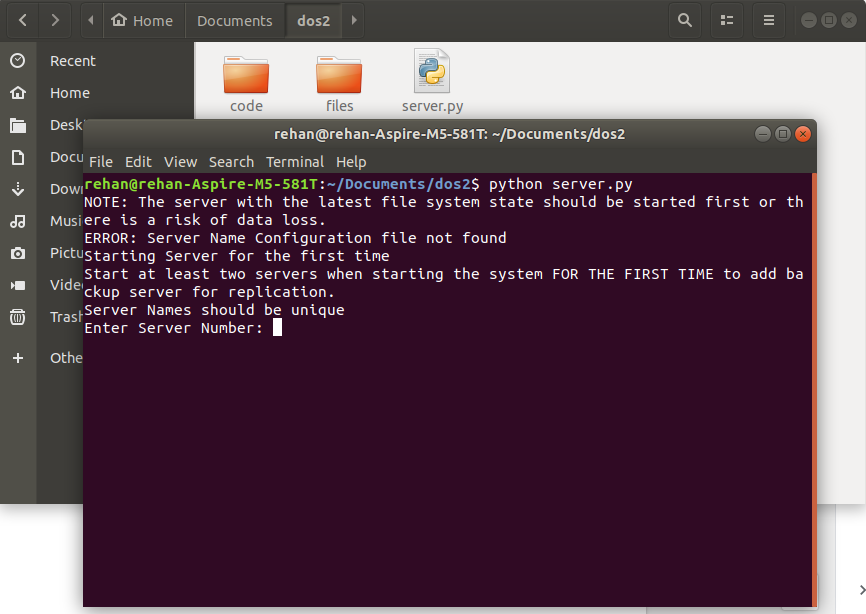
**PREREQUISITES**

* Python3
* Python3 libaries
  + *pickle*

**INSTALLATION**

1. [Download the server.py file](https://gist.github.com/reallyrehan/76ff02a6af8fd6a1bdfc6890ca52283d) and save it in any folder.  
   
2. Right-click in the folder and click on **Open in Terminal**.
3. Type the following code,  
   **python server.py**Or if you have both, Python2 and Python3 installed,  
   **python3 server.py**

When starting a server for the first time, you should see the following screen.



*The* ***ERROR*** *is to warn you that no previous Server configuration file was found, so this server will start as a new server and will start the initial configuration process.*

**INITIAL CONFIGURATION**

The first time you start up your system, you will be asked for a server number.

1. Each server gets its own number, and it should be **unique** in order for the system to work properly and interact with each other.
2. The first time you start the system, you should start all the servers (or at least two servers) at least once for the servers to recognize each other.

After the initial configuration process, the server will save the configuration details and won’t prompt you again.

**IMPORTANT THINGS**

* Each Server should get its own number, from 0 to 10000 or so on. (No **limit** on the number as long it can be stored in an **integer**).
* The Server that is the **last to be turned off** should be the **first to turn on** in order to preserve Server state since only the latest server would have the latest state of files and the directory structure. Otherwise, there is a chance of **data loss**.

The system has been tested with up to 3 servers, but with a replication factor of **2**, it’ll maintain access to 100% of the files on the system even with 2 servers up and running. The system also works perfectly with one server, but will lose access to around 20-40% of the files.

To **EXIT** the server, use CTRL+C to send an interrupt or type ‘exit’ on the console, which will save the file structure and recent changes and kill all the threads being run by the Server script.

**KNOWN ISSUES**

* CTRL+C doesn’t work properly on Windows’ command prompt. Moreover, server ‘exit’ is implemented using signals which doesn’t work in Windows.

**Client - Side**

**PREREQUISITES**

* Python3
* Python3 libaries
  + *Pickle*
  + *Tkinter*
  + *webbrowser*

**INSTALLATION**

1. [Download the client.py file](https://gist.github.com/reallyrehan/76ff02a6af8fd6a1bdfc6890ca52283d) and save it in any folder.
2. Right-click in the folder and click on **Open in Terminal**.
3. Type the following code,  
   **python client.py**Or if you have both, Python2 and Python3 installed,  
   **python3 client.py**

That’s it.

***Note****: To upload a file, place the file in the same directory as the client.py file.*