Protocol

1, Command
REG: register new user account
LOGIN: login a user account
LOGOUT: logout a user account
PASS: change password to a user account
VIEW: view all data for a user given type
SEND: send emails
GET: get data
PUT: put data
DEL: delete data
MOVE: move data
DIR: create new folder
SPUT: notify master when PUT done
SDEL: notify master when DEL done
SMOVE:notify master when MOVE done
SDIR: notify master when DIR done
FINISH: notify master when any read operation done
STATE: get state info of each server
TABLE: get simple bigtable info
SHUTDOWN: shutdown a server
LOG: get the log

2, Format
/*
fileRelativeAddr means the relative address of that file in user's root directory
fileType = {"FOLDER", "DRIVE", "EMAIL"}
*/
   1)Frontend -> Master
      GET username + "#" + fileRelativeAddr + "#" + fileType
      PUT username + "#" + fileRelativeAddr + "#" + fileType
      LOGIN username + "#" + password
      LOGOUT username
      REG username + "#" + password
      PASS username + "#" + password
      DEL username + "#" + fileRelativeAddr + "#" + fileType
      MOVE username + "#" + oldFileRelativeAddr + "#" + newFileRelativeAddr+"#" +
fileType
      VIEW username + "#" + fileType
      DIR username + "#" + dirRelativeAddr + "#" + fileType
      SEND username + "#" + receiverEmailAddr

   2)Master -> Frontend
      A, Reply GET
         +OK URL

        -ERR username doesn't exist

        -ERR file not found

    B, Reply PUT

        +OK URL

        -ERR username doesn't exist

    C, Reply LOGIN

        +OK

        -ERR username doesn't exist

        -ERR user already logged in

        -ERR password incorrect

    D, Reply Reg

        +OK

        -ERR username already registered

        -ERR password format incorrect

    E, Reply PASS

        +OK

        -ERR username doesn't exist

    F, Reply DEL

        +OK URL

        -ERR username doesn't exist

        -ERR file not found

    G, Reply MOVE

        +OK URL

        -ERR username doesn't exist

        -ERR file doesn't exist

        // handle dir and file separately

    H, Reply VIEW

        +OK listOfFiles(fileRelativeAddr+"#" + fileType for each file separated by "#")

        -ERR username doesn't exist

    I, Reply DIR

        +OK URL

        -ERR username doesn't exist

        -ERR directory already exists

    J, Reply LOGOUT

        +OK

        -ERR username doesn't exist

        -ERR username not logged in

    H, Reply SEND

        /*

        flag = {REMOTE, LOCAL}, REMOTE means @gmail.com, LOCAL means

        @pennclould.com

        */

        +OK flag + "#" + serverAddr

        -ERR username doesn't exist

3) Frontend -> Storage

GET username + "#" + fileRelativeAddr + "#" + fileType
PUT username + "#" + fileRelativeAddr + "#" + fileType + "#" + fileContent
DEL username + "#" + fileRelativeAddr + "#" + fileType
MOVE username + "#" + oldFileRelativeAddr + "#" + newFileRelativeAddr +"#" + fileType
DIR username + "#" + dirRelativeAddr

4) Storage -> Frontend
    A,Reply GET:
       +OK fileContent
       -ERR username doesn't exist
       -ERR file doesn't exist
    B,Reply PUT:
       +OK
       -ERR username doesn't exist
    C,Reply DEL
       +OK
       -ERR username doesn't exist
    D, Reply MOVE
       +OK
       -ERR username doesn't exist
    E, Reply DIR
       +OK
       -ERR username doesn't exist

5)Storage -> Master
    A, On Finishing PUT
       SPUT username + "#" + fileRelativeAddr + "#" + fileType + "#" + URL
    B, On Finishing DEL
       SDEL username + "#" + fileRelativeAddr + "#" + fileType + "#" + URL
    C, On Finishing MOVE
       SMOVE username + "#" + oldFileRelativeAddr + "#" + newFileRelativeAddr +"#" + fileType + "#" + URL
    D, On Finishing DIR
       SDIR username + "#" + dirRelativeAddr + "#" + fileType + "#" + URL
    E, On Finishing others
       FINISH URL

6) Master -> Storage
    A, Reply SPUT
       +OK
       -ERR username doesn't exist
    B, Reply SDEL
       +OK
       -ERR username doesn't exist
    C, Reply SMOVE

+OK
-ERR username doesn't exist
D, Reply SDIR
+OK
-ERR username doesn't exist
E, Reply FINISH
+OK
-ERR

7) Admin -> Master
TABLE
STATE

8) Master -> Admin
A, Reply TABLE
+OK username + "#" + fileRelativeAddr + "#" + fileType + "#" + serverAddr
-ERR
B, Reply STATE
// each flag is 1 or 0, indicating the corresponding server down or not
+OK flag1 + "#" + flag2 + "#" + "flag3" + …
-ERR

9) Admin -> Frontend or Backend Servers
SHUTDOWN