**READ ME**

1. Development Environment  
   version of operating system : Ubuntu 16.04  
   languages : python3  
   editor : vim  
   compile options: python3 server.py , python3 client.py  
   다른 프로젝트 이유로, 다른 사람의 vmware를 복제해서 사용하고 있습니다. Screen capture 상 기기의 이름이 다른 사람의 이름인 점 양해 부탁 드립니다.
2. How to run both sender and receiver programs  
   **0. Compile command**python3 server.py   
   python3 client.py **1. File system**  
   Directory *One* : server, client(User1), fileA.jpg  
   Directory *Two* : client(User2), fileB.pdf  
   Directory *Three* : cilent(User3), fileC.pdf  
   **2. Situation**1) User1, User2, User3 차례로 입장  
   2) 0, show menu   
   3) 1, register file  
    - 각 유저(User1, User2, User3)는 자신이 갖고 있는 파일(fileA, fileB, fileC)을 등록한다.  
   4) 2, show global file list  
    - 유저가 등록한 파일들이 제대로 잘 담겼는지 확인한다.  
    - 총 3개의 파일(User1/fileA.jpg, User2/fileB.pdf, User3/fileC.pdf)이 담겨있어야 한다.5) 3, Download file  
    - User1이 User2/fileB.pdf를 요청한다. 해당 파일(fileB.pdf)을 User2로부터 다운로드 받고, global file list를 업데이트한다.  
    - 업데이트된 global file list는 총 4개의 파일(User1/fileA.jpg, User1/fileB.pdf, User2/fileB.pdf, User3/fileC.pdf)을 포함한다.  
   5’) 3, Download file 한번 더 확인  
    - 이번엔 User3가 User1/fileB.pdf를 요청한다. 해당 파일(fileB.pdf)을 User1로부터 다운로드 받고, global file list를 업데이트한다.  
    - 업데이트된 global file list는 총 5개의 파일(User1/fileA.jpg, User1/fileB.pdf, User2/fileB.pdf, User3/fileB.pdf, User3/fileC.pdf )을 포함한다.  
   6) 4, Exit  
    - User2, User1, User3 순서대로 퇴장한다.   
    - server는 퇴장하는 User에 해당하는 파일을 global file list에서 제외시킨다.



1. Present how to design

|  |  |  |  |
| --- | --- | --- | --- |
| Server | | Client | |
| * User list(딕셔너리) Key : User ID value : socket for userID(Key) * File list(딕셔너리) key : User ID value : file list of userID(Key) | | * Do\_register(Boolean)   True: 등록하려는 파일명을 입력 받으려 할 때   * Do\_download(Boolean) True: 다운받으려는 파일명을 입력 받으려 할 때 | |
| 1. 유저가 입장할 때   UserID와 해당 socket을 User list에 mapping  연결된 다른 유저들에게 WELCOME 메시지를 보냄 | | 서버와 연결  UserID를 서버로 보냄 | |
| 1. Register file 요청   File list[user\_id].append(파일 이름)  연결된 다른 유저들에게 global file list가 업데이트됨을 알림 | | 1번 option 선택, Do\_register = true  파일 이름 입력  파일 이름과 user id 서버로 전송 | |
| 1. Global file list 요청   저장하고 있는 global file list를 string 형태로 변환 후, 전송 | | 2번 option 선택  서버에 global file list를 요청 | |
| 1. Download file 요청   파일을 가지고 있는 소켓에게 파일 크기 요청   전달받은 파일의 크기를 수신 소켓에게 알림 파일의 크기만큼 전달받은 file content를 수신 소켓에게 전송함   file list에 해당 파일 추가. global file list가 업데이트됨을 연결된 다른 소켓들에게도 알림 | **수신 소켓**  3번 option 선택  다운로드 요청할 유저와 파일이름 입력  파일 이름과 수신 user\_id 전송  파일의 크기 확인 전달받은 파일의 크기만큼 file content 수신  요청한 파일 저장  완료 후 다운로드 받은 파일에 대해, 서버에 등록한다 | | **송신 소켓**  파일의 크기를 서버에게 알려줌  파일을 1024씩 나눠서 전송함 |
| 1. Exit    소켓의 연결이 끊어짐을 인지 User list에서 해당 소켓의 유저 아이디를 지움 file list에서 해당 소켓의 유저에 속한 파일들을 지움 global file list가 업데이트됨을 연결된 다른 소켓들에게도 알림 | | 4번 option 선택  소켓 종료 | |