

הסבר על המערכת

אז מה עשינו?

במערכת שלנו ישנם שלושה קבצים – tests.py, Client.py, Server.py.

קובץ Client.py:

בקובץ זה ישנה מחלקה בשם Client ופונקציית main היוצרת Instance של המחלקה.

במחלקה הזו ישנן שדות כמו:

- name: המייצג את שם המשתמש.
- port: המייצג את ה Port שאותו בחר המשתמש.
- server_name: המייצג את כתובת השרת – מוגדר ב init.
- server_address: המייצג את כתובת ה IP וה Port של השרת.
- dir_files: המייצג את רשימת הקבצים בצ'אט.
- flag: המשמש בשביל לבדוק האם השם של המשתמש כבר תפוס.
- is_Alive: הבודק האם הקליינט מחובר (על מנת לסגור באופן חלק את ה Thread ים הקיימים).
- output: המייצג את כל ההדפסות שיוצאות למסך (על מנת לוודא תקינות בבדיקות).
- client_socket_TCP: המייצג את קשר ה TCP עם השרת.
- client_socket_UDP: המייצג את קשר ה UDP עם השרת.

במחלקה הזו ישנן גם פונקציות (נמנה את הפונקציות העיקריות):

- init: המאתחלת את המשתמש עם הנתונים הספציפיים הקשורים אליו, מפעילה את הטרדים ופותחת את ה connection ים.
- get_port: קולט Port ממשתמש ובודק את תקינותו וזמינותו מול השרת.
- is_valid: הבודק את תקינות שם הקובץ שהתקבל מהמשתמש מול dir_files.
- switcher: המנתב לפונקציה הרלוונטית על הפקודה המתקבלת מהמשתמש.
- menu: המציגה למשתמש את הפקודות האפשריות לביצוע בצ'אט.
- receive_half_file: מקבלת מהשרת דרך קשר UDP חצי מהקובץ בצ'אט שולחת אקים על כל צ'אט שהתקבל ומאמתת את תקינות הפאקטה על ידי חישוב Checksum והשוואתו ל Checksum שהתקבל ב Chunk מהשרת.
- receive_UDP_messages: פונקציה המופעלת כל הזמן ברקע ע"י טרד שלישי ותפקידה להאזין לקבלת תגובה מהשרת ע"י קשר UDP ולהגיב בהתאם.
- receive_TCP_messages: פונקציה המופעלת כל הזמן ברקע ע"י טרד שני ותפקידה להאזין לקבלת תגובה מהשרת ע"י קשר TCP ולהגיב בהתאם.
- actions: פונקציה אשר מופעלת כל הזמן ברקע ע"י טרד ראשון ותפקידה לקבל את הפקודה ולהמירה ל int אשר מועבר ל switcher. עם זאת, הפונקציה דואגת לוודא את התקינות של הפקודה הנכנסת.
- checksum: מחשבת checksum.

עבור כל אחת מהפעולות שניתן לבצע ישנה פונקציה ששולחת לשרת את התבנית המתאימה ומקבלת נתונים במידה וצריך.

נדגיש שבמחלקה זו ישנם שלושה טרדים המופעלים כל הזמן ודואגים להתנהלות הביצועים בצורה מקבילה ותקינה עם התחשבות בפרוטוקול המייצג.

קובץ Server.py:

בקובץ זה ישנן 2 מחלקות: מחלקה בשם Server ופונקציית main היוצרת Instance של המחלקה ומחלקה נוספת בשם Chunk המייצגת חבילה.

נתחיל עם מחלקת Server:

במחלקה הזו ישנן שדות כמו:

- server_port: מוגדר להיות 50000 כמו שהתבקשנו ליישם במטלה.
- client_socket_TCP: המייצג את קשר ה-TCP עם הלקוחות.
- client_socket_UDP: המייצג את קשר ה-UDP עם הלקוחות.
- names: מהווה מילון אשר מייצג כמפתח Port וכערך את שם הלקוח של הלקוחות המחוברים.
- users: מהווה מילון אשר מייצג כמפתח את שם הלקוח וכערך את IP PORT ו-TCP Connection.
- online_users: הסופר כמה לקוחות מחוברים יש בצ'אט.
- files: מהווה מילון שהמפתח בו הוא שם הקובץ והערך הוא הגודל של הקובץ ומי המשתמש שמוריד אותו.

במחלקה הזו ישנן גם פונקציות (נמנה את הפונקציות העיקריות):

- init: המאתחלת את המשתמש עם הנתונים הספציפיים הקשורים אליו, מפעילה את הטרדים ופותחת את ה-connection ים.
- multi_threaded_client: מאזינה להודעות הנשלחות מהמשתמשים ומנתבת את ההודעות ל-actions.
- actions: מנתבת את ההודעות לפונקציה הרלוונטית על פי הפקודה שהתקבלה ושולחת את השדות הנדרשים.
- send_and_ACK: מקבלת הודעה לשליחה, מגדירה timeout ושולחת הודעה כל עוד לא התקבל ACK בזמן המוגדר.
- send_file: מקבלת את שם הקובץ, פותחת את הקובץ ומחלקת אותו לצ'אנקים על פי מחלקת Chunk ומזינה את השדות הדרושים.
- checksum: מחשבת checksum.

עבור כל אחת מהפעולות שניתן לבצע ישנה פונקציה ששולחת תגובה למשתמש הספציפי עם הנתונים שביקש על פי התבנית המתאימה.

נמשיך עם מחלקת Chunk:

מחלקה זו מייצגת פאקטה משום כך יש בה רק שדות:

- Checksum – המייצגת את תוצאת חישוב ה-checksum.
- data – המידע הנשלח מהקובץ.
- id – מספור החבילה.

קובץ tests.py:

במחלקה זו אנו ממשים בדיקות היקפיות על סמך unittesting. כל הבדיקות נעשות באותו האופן:

תחילה פתחנו ב-thread נפרד את instance של Server ולאחר מכן שלחנו לפונקציית test_scenario את האינפוטים שמכניס המשתמש והחלק הרלוונטי של ההדפסות שאנו מצפים לקבל (שאותן וידאנו אל מול שדה output בקליינט).

פונקציית test_scenario: בשביל להכניס את האינפוטים בצורה אוטומטית למערכת השתמשנו בספריית MOCK אשר מחקה לקוח פיקטיבי ושורת אינפוטים רצופים שהוא מכניס. כאשר האינפוטים הסתיימו וידאנו שה output מכיל את החלק הרלוונטי של ההדפסות שקיבלנו בפונקציה.

ניתן לראות תרשימים מפורטים על פעילות המערכת בקובץ 3 and 2 part Final Ex.