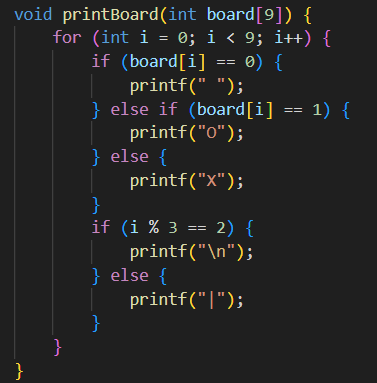
מערכות הפעלה מטלה 2

שקד שורץ – 214695108

ליאור שולמן – 326629094

**הסבר על הקוד**

ttt.cpp

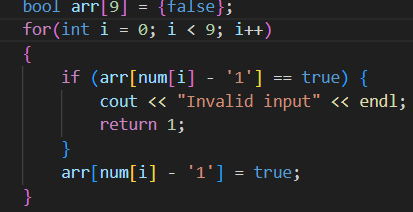


פונקציית הדפסה ללוח אין משמעות מעבר לנוחות.

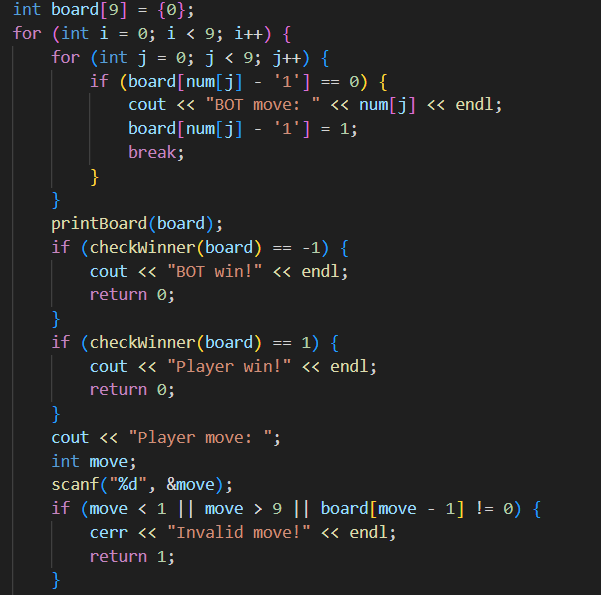
A computer screen with text on it

Description automatically generated

פונקצייה שמקבלת את הלוח ובודקת את כל האופציות לניצחון.



בדיקה של הקלט, האם מדובר ב9 מספרים שונים מ1 עד 9.

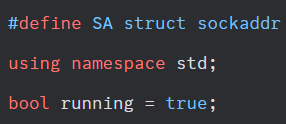


A screen shot of a computer program

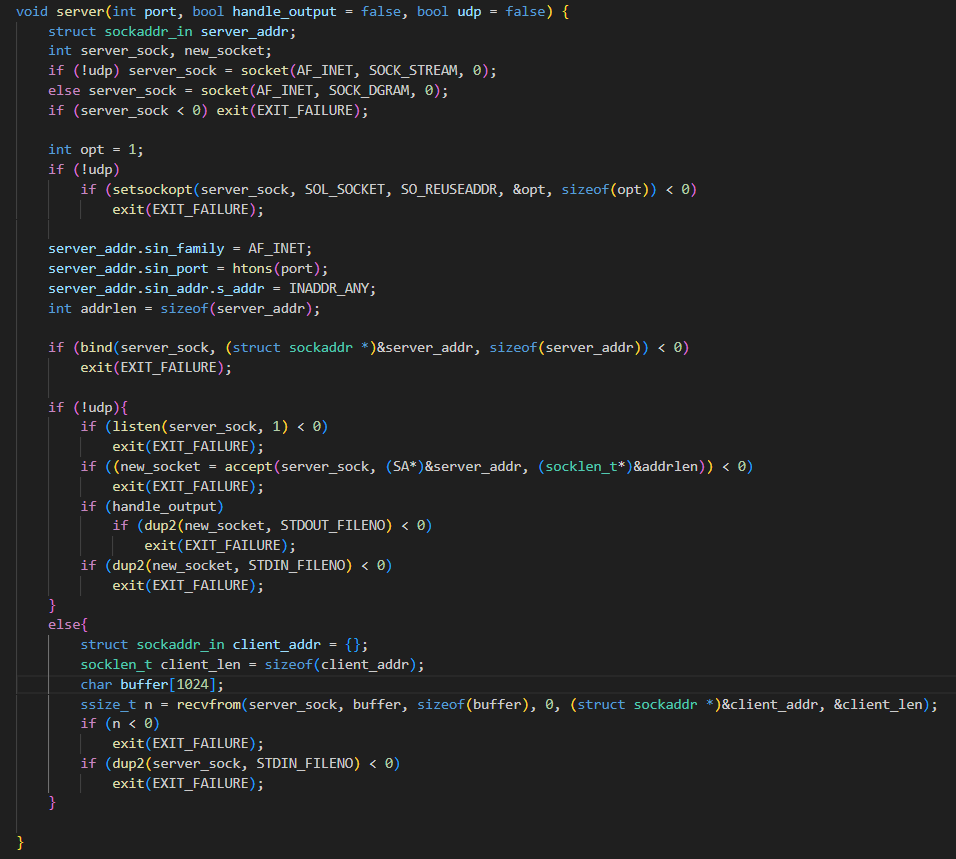
Description automatically generated

מערך של 9 שמריץ את המשחק קודם כל מבצע את המהלך של הבוט לפי האינפוט שהוא קיבל ואז בודק ניצחון אם לא אז ממשיך לשחקן באותו אופן ואז בודק ניצחון.

Mync.cpp



משתנה גלובלי עבור UDP על מנת לעצור את הקוד בקבלת סיגנל.



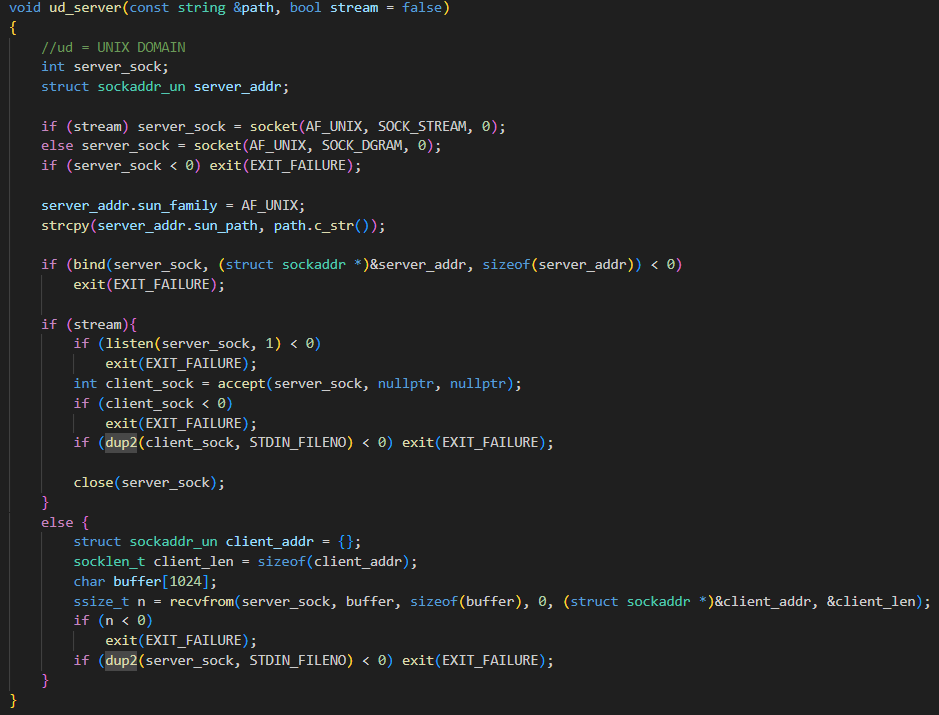
זה הקוד שפותח את שרתי ה UDP וה TCP הוא מבדיל בניהם בעזרת משתנה שהוא מקבל מהמיין  
קודם כל מגדיר את הסוקט מגדיר משתנים שנדרשים ואז מגדיר אותו בהתאם.  
במידה ומדובר ב TCP הוא מתחיל להאזין ואז מקבל את הלחיצת יד ואז בודק האם הוא הקלט ההתחלתי היה גם -i וגם -o אם כן אז הוא מגדיר שגם האינפוט וגם האאוטפוט יבוא מהטרמינל.  
אם לא אז הוא מגדיר שהאינפוט יבוא מהטרמינל.

במקרה של UDP הוא מגדיר את מה שהוא צריך ואז מבצע שליחה של תיאום ורק אז מגדיר שהאינפוט שישלח לסוקט יבוא מהטרמינל.

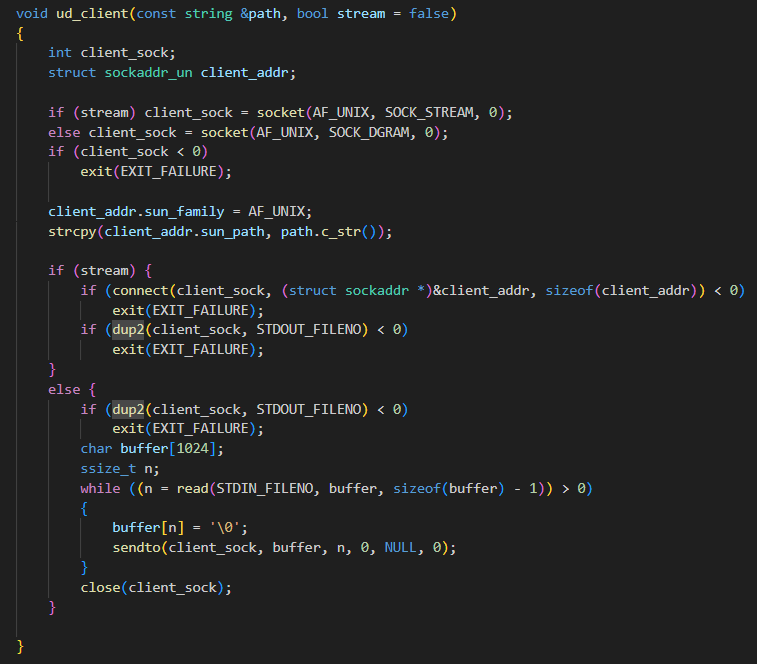
A computer screen shot of a program code

Description automatically generated

פונקציה שמייצרת את הסוקט של הUDP וה TCP אצל הלקוח עובדת באופן דומה לשרת חוץ מהחלק של ה DUP2 ששם מוגדר שהאאוטפוט שלו יהיה הפלט אצלו בטרמינל.



בפרוטוקול הזה הנתונים שמועברים הם שונים ולכן קיימת פונקצייה שונה.  
פה מדובר בתקשורת תוך מחשבית שיש לה שתי אופציות stream ו – datagram,  
stream שקול ל TCP ו datagram שקול ל UDP חוץ מזה אותו הרעיון לגבי הסרבר והלקוח ממקודם באופן של השימוש בDUP2

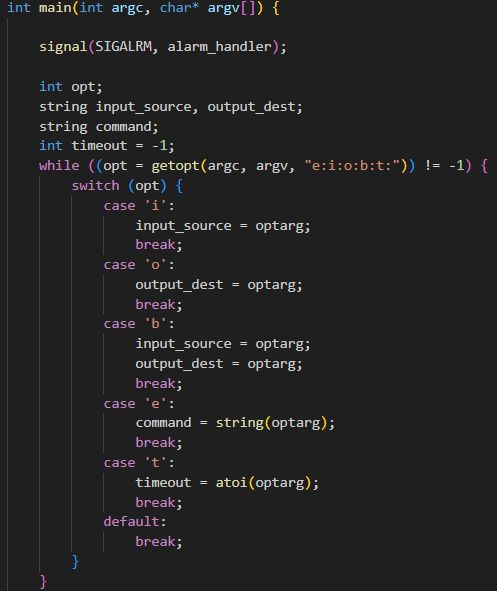


באופן דומה זה הלקוח.

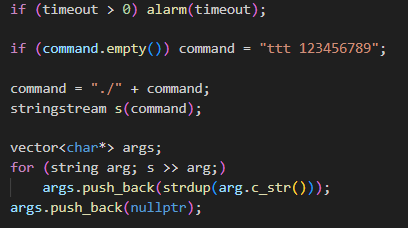
A computer screen with white text

Description automatically generated

פעולה שמשנה את המשתנה הגלובלי בשימוש רק ב UDP הפעולה הזאת היא לטובת סגירת כל התהליכים לפני סיום התוכנית.



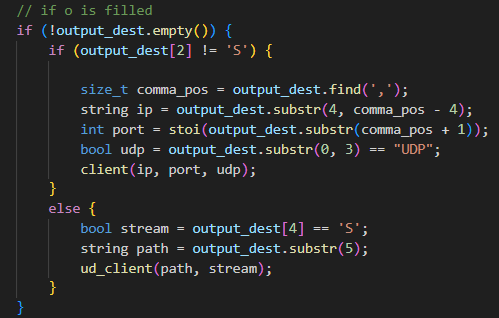
קליטת ארגומנטים שימוש בעצה שקיימת בהנחיות



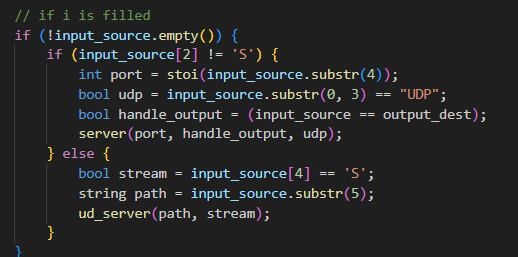
נגדיר אלרם בשביל UDP ואז נעביר את הפקודה לרשימה כדי להשתמש ב execvp



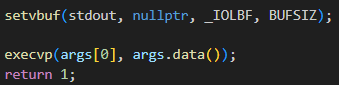
פיצול התהליך ולתהליך בן



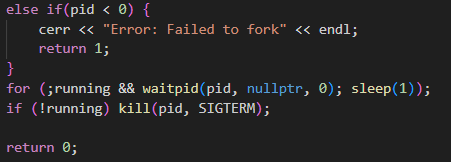
בודק באיזה פרוטוקול צריך להשתמש בעזרת פירוק הארגומנט לתתי סטרינגים וגם מחלץ את הפורט והכתובת במידת הצורך (אם מדובר ב UDP או TCP ) וקורא לפונקצייה שבונה את הלקוח אחרת (UDS) בודק האם זה stream או datagram ואז קורא לפעולה שבונה לקוח.



באופן דומה גם עבור השרת.



פעולה שמפצלת את השורות באאוטפוט ומריצה את הפקודה שהתקבלה בארגומנטים ומחזירה 1 בשביל להרוג את התהליך.



בדיקה אם הפיצול נכשל ולולאה שאם הטיימר נגמר להרוג את התהליך.

tcp gcov  


Udp gcov



Uds gcov

