

### מערכת הפעלה מטלה 3

#### שאלה 1 :

- התבקשנו לממש שרת שפותח תקשורת בTcp ומממש מחסן של אטומים מסוג oxygen , carbon ו hydrogen . השרת יקבל בקשות ע"י מחרוזת שמתקבלת מלקוח שמשמש כספק שמילא את מחסניו.
- השרת פתח socket להאזנה בפורט שהגיע לו כארגומנט וממתין בפונקציית select לפקודת connect מלקוח מסוים.
  - ברגע שנוזהה התחברות נחלץ את המידע הרלוונטי ונוסיף למשתנים של השרת, לאחר מכן נמתין להתחברות נוספת וכן הלאה.
  - בצד הלקוח (supplier) נקבל את הפורט וhostname
  - נפתח socket ונשלח פקודת connect אל השרת.
  - נבקש מהמשתמש להכניס את האטומים שהוא מספק ונכניס אותם לרשימה, נשלח את המידע כמחרוזת.
  - נדפיס את כל המידע שנמצא במחסן לאחר קבלת האטומים החדשים.
- הוראות הרצה :

1. להיכנס אל תיקיית tar1 ( לאחר בניית make רקורסיבי)

2. ./atom\_warehouse <port\_tcp> <port\_udp>

3. ./atom\_supplier localhost <port\_tcp>

בדוגמה זו נשלח :

ADD CARBON 30

hydrogen לא שלחתי

ADD OXYGEN 10

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar3$ ./atom_supplier 127.0.0.1 9999
Connected to server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 40
```

ניתן לראות שאחרי כל פקודה שהשרת מדבל הוא מדפיס סיכום מעודכן, לאחר מכן ממתיין למידע נוסף. כאשר הלקוח שלח את כל המידע הרלוונטי כרגע התקשורת נסגרת.

```
=== Current Stock ===
Total atoms      : 40
Carbon (C)       : 40
Hydrogen (H)     : 0
Oxygen (O)       : 0
=====
```

## שאלה 2:

הסבר הרצה:

4. להיכנס אל תיקיית tar2 ( לאחר בניית make רקורסיבי )
  5. ./molecule\_supplier <port\_tcp> <port\_udp>
  6. ./atom\_supplier localhost <port\_tcp>
  7. ./molecule\_requester localhost <port\_udp>
- שדרגנו את השרת בשאלה 1 אל שרת חדש בשם molecule\_supplier כך שבנוסף לקבלת מלאי חדש על tcp יוכל לקבל בקשות upd ליצירת מולקולות.
  - השרת החדש מקבל 2 ארגומנטים (פורט tcp ואז פורט udp)
  - יצרנו לקוח חדש שמטרתו לקבל בקשות מהטרמינל למולקולה מסוימת (חמצן, פחמן, גלוקוז, אלוכוהול), הלך יקבל כארגומנט את הפורט בו השרת מצפה לקבל חבילות udp.
  - פתחנו socket מסוג datagram על מנת להעביר חבילות על גבי udp ושלחנו את המולקולה הנדרשת ומספר המולקולות אל השרת.

```
while(true){
    std::cout << "What molecule do you want to create?";
    std::cout << "\n1 - WATER \n2 - CARBON_DIOXIDE \n3 - ALCOHOL \n4 - GLUCOSE \n5 - EXIT" << std::endl;
    int choice;
    std::cin >> choice;
    switch (choice) {
        case 1: // WATER
            std::cout << "Enter number of water molecules to create: ";
            std::cin >> water_count;
            break;
        case 2: // CARBON_DIOXIDE
            std::cout << "Enter number of carbon dioxide molecules to create: ";
            std::cin >> carbon_dioxide_count;
            break;
        case 3: // ALCOHOL
            std::cout << "Enter number of alcohol molecules to create: ";
            std::cin >> alcohol_count;
            break;
        case 4: // GLUCOSE
            std::cout << "Enter number of glucose molecules to create: ";
            std::cin >> glucose_count;
            break;
        default:
            std::cerr << "Invalid choice" << std::endl;
            close(sockfd);
            return 1;
    }
}
```

- בחירת בקשה של יצירת מולקולת water

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar2$ ./molecule_requester localhost 9998
Connected to server over UDP. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
1
Enter number of water molecules to create: 1
Server response: SUCCESSFULLY DELIVERED WATER
Do you want to deliver more? (y/press eny key to end)
y
```

- בצד השרת קיבלנו את הבקשה, פיצלנו אותה לפי רווחים וזיהינו איזה בקשה למולקולה קיבלנו ומה הכמות המבוקשת.

- נבחן את המלאי שיש בשרת ונראה האם ישנה כמות מספיקה של אטומים הנדרשים לבניית המולקולה. במידה וכן נחסיר את הכמות הנדרשת ונחזיר הודעת הצלחה, במידה ולא נחזיר הודעת שגיאה ונשאיר את המלאי כמו שהוא.

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar2$ ./molecule_supplier 9999 9998
Server listening over tcp on port 9999
Server listening over udp on port 9998
```

### קבלת בקשה 1 DELIVER WATER בצד השרת

```
=== Current Stock ===
Total atoms   : 30
Carbon (C)    : 10
Hydrogen (H)  : 10
Oxygen (O)    : 10
=====
Received UDP: DELIVER WATER 1

Delivered 1 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms   : 27
Carbon (C)    : 10
Hydrogen (H)  : 8
Oxygen (O)    : 9
=====
```

הדגמה של בקשה בה אין מספיק אטומים:  
DELIVER GLUCOSE 1

```
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
4
Enter number of glucose molecules to create: 1
Server response: FAILED TO DELIVER GLUCOSE
```

צד שרת

```
=== Current Stock ===
Total atoms   : 18
Carbon (C)    : 9
Hydrogen (H)  : 4
Oxygen (O)    : 5
=====
Received UDP: DELIVER GLUCOSE 1

Error: Not enough atoms to deliver 1 glucose molecules.
Failed to deliver glucose.
=== Current Stock ===
Total atoms   : 18
Carbon (C)    : 9
Hydrogen (H)  : 4
Oxygen (O)    : 5
=====
```

### שאלה 3 :

- הוראות הרצה :
- להיכנס אל תיקיית tar3 ( לאחר בניית make רקורסיבי )
- ./drinks\_bar <port\_tcp> <port\_udp>
- ./atom\_supplier localhost <port\_tcp>
- ./molecule\_requester localhost <port\_udp>
- כעת נאפשר לשרת לקבל קלט מהמקלדת במקביל לקלט udp השולח בקשות ליצירת מולקולות ולקלט tcp השולח בקשות לאספקת אטומים למחסן.
- הקלט מהמקלדת יהיה פקודה מדויקת של יצירת משקה הבנוי ממספר סוגי מולקולות בכמויות שונות.
- את האפשרות לעבוד המקביל נעשה בעזרת הכנסת fdn של stdin אל setn שעליו השרת מסתכל כל הזמן ובוחן האם ישנו מידע חדש שהגיע.
- נעזר במשתנה זמני שיקבל את מהספר האטומים הנוכחיים ונבחן לפי המתכון כמה משקאות מהסוג המבוקש ניתן להכין.

```
long long num_of_soft_drinks(){
    long long tmp_car=carbon_count;
    long long tmp_hyd=hydrogen_count;
    long long tmp_oxy=oxygen_count;
    long long count = 0;
    while(true){
        //water
        tmp_hyd -= 2;
        tmp_oxy -= 1;
        //carbon dioxide
        tmp_car -= 1;
        tmp_oxy -= 2;
        //glucose
        tmp_car -= 6;
        tmp_hyd -= 12;
        tmp_oxy -= 6;
        if(tmp_hyd < 0 || tmp_oxy < 0 || tmp_car < 0){
            break;
        }
        count++;
    }
    std::cout << "Number of soft drinks that can be created: " << count << std::endl;
    return count;
}
```

דוגמת הרצה בטרמינל:

```
=== Current Stock ===
Total atoms   : 180
Carbon (C)    : 60
Hydrogen (H)  : 60
Oxygen (O)    : 60
=====
GEN CHAMPAGNE
Received input: GEN CHAMPAGNE
Number of vodka that can be created: 3
GEN VODKA
Received input: GEN VODKA
Number of champagne that can be created: 7
GEN SOFT DRINK
Received input: GEN SOFT DRINK
Number of soft drinks that can be created: 4
```

## שאלה 4:

הוראות הרצה :

1. יש להיכנס אל תיקיית tar4 (לאחר בניית make רקורסיבי)
2. `./drinks_bar -T <tcp_port> -U <udp_port> [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]`
3. `./atom_supplier -h <hostname> -p <port>`
4. `./molecule_requester -h <hostname> -p <port>`
- התבקשנו לממש שעון שיסגור את השרת לאחר זמן מסוים בו אין אף פנייה אליו.
- נממש זאת באמצעות alarm עם 20 שניות טיימר, כמו כן נבצע שינוי בסיגנל SIGALRM כך שבמקום לצאת מהתוכנית יקפוץ לפונקציה שסוגרת תחילה את 2 החיבורים ואז יוצאת מהתוכנית.
- נעדכן את השעות ל 60 שניות נוספות בכל קבלת בקשה חדשה.
- כאשר הבקשה מהטרמינל של השרת :

```
Delivered 4 glucose molecules.
=== Current Stock ===
Total atoms   : 84
Carbon (C)    : 36
Hydrogen (H)  : 12
Oxygen (O)    : 36
=====
GEN CHAMPAGNE
Received input: GEN CHAMPAGNE
Number of vodka that can be created: 0
No activity for 60 seconds. Closing server.
```

כאשר הבקשה מהספק :

```
Server listening over tcp on port 9997
Server listening over udp on port 9996
New connection, socket fd: 5
=== Current Stock ===
Total atoms   : 30
Carbon (C)    : 30
Hydrogen (H)  : 0
Oxygen (O)    : 0
=====
No activity for 60 seconds. Closing server.
```

כאשר הבקשה מהלקוח המבקש מולקולות :

```
Received UDP: DELIVER WATER 2

Delivered 2 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms   : 59
Carbon (C)    : 25
Hydrogen (H)  : 16
Oxygen (O)    : 18
=====
No activity for 60 seconds. Closing server.
```

קטעי הקוד הרלוונטיים למימוש השעון:

```
signal(SIGINT, cleanup);
signal(SIGALRM, timeout_handler);
alarm(TIMEOUT);
```

```
void timeout_handler(int signum) {
    std::cout << "No activity for "<<TIMEOUT<<" seconds. Closing server." << std::endl;
    close(server_fd);
    close(udp_fd);
    exit(0);
}
```

- כמו כן נתאר את תהליך הוספת האופציות (דגלים), נאלץ את הקלט להגיע עם 2 ארגומנטים לפורט tcp וudp.
- על מנת לפרסר בצורה תקינה (תווים ומילים) נעזר בפונקציה getoptlong, נשלח לה מספר ארגומנטים:
- Argc, argv, את רשימת הדגלים, struct שמכיל את הכינוי הנוסף כמילה ארוכה בה אפשר לכוות כל דגל.
- נבחן איזה דגל הגיע ונעדכן בעזרת switch case את הארגומנט המתאים.
- בעזרת משתנים בוליאנים נאלץ לקבל את הקלט של Udp ו tcp.

```
int timeout= 60;
static struct option long_options[] = {
    {"oxygen",    required_argument, 0, 'o'},
    {"carbon",    required_argument, 0, 'c'},
    {"hydrogen",  required_argument, 0, 'h'},
    {"timeout",   required_argument, 0, 't'},
    {"tcp-port",  required_argument, 0, 'T'},
    {"udp-port",  required_argument, 0, 'U'},
    {0, 0, 0, 0}
};
```

```
void updateStock(int argc, char* argv[], int& port_tcp, int& port_udp) {
    int opt;
    bool has_tcp = false, has_udp = false;
    while ((opt = getopt_long(argc, argv, "o:c:h:t:T:U:", long_options, nullptr)) != -1) {
        switch (opt) {
            if (std::atoi(optarg) <= 0) {
                std::cerr << "Error: any flag must be a positive integer." << std::endl;
                exit(EXIT_FAILURE);
            }
            case 'o':
                add_oxygen(std::atoi(optarg));
                break;
            case 'c':
                add_carbon(std::atoi(optarg));
                break;
            case 'h':
                add_hydrogen(std::atoi(optarg));
                break;
            case 't':
                timeout = std::atoi(optarg);
                break;
```

```

        case 'T':
            port_tcp = std::atoi(optarg);
            has_tcp = true;
            break;
        case 'U':
            port_udp = std::atoi(optarg);
            has_udp = true;
            break;
        default:
            std::cerr << "Usage: " << argv[0]
                << " -T <tcp_port> -U <udp_port> [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]\n";
            exit(EXIT_FAILURE);
    }

    if (!has_tcp || !has_udp) {
        std::cerr << "Error: Both -T (TCP port) and -U (UDP port) flags are required.\n";
        std::cerr << "Usage: " << argv[0]
            << " -T <tcp_port> -U <udp_port> [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]\n";
        exit(EXIT_FAILURE);
    }
}

```

נתאר מספר דוגמאות הרצה :

1. ללא דגלים אופציונאלית
2. עם חלק מהדגלים האפציונאליים
3. עם כל הדגלים האופציונאליים.
4. כל הדגלים עם מילים ארוכות.
5. ללא דגל חובה.
6. לא לפי הסדר.

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9997 -U 9996
=== Current Stock ===
Total atoms : 0
Carbon (C) : 0
Hydrogen (H) : 0
Oxygen (O) : 0
=====
Server listening over tcp on port 9997
Server listening over udp on port 9996
No activity for 60 seconds. Closing server.

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -U 9994 -t 20
=== Current Stock ===
Total atoms : 0
Carbon (C) : 0
Hydrogen (H) : 0
Oxygen (O) : 0
=====
Server listening over tcp on port 9995
Server listening over udp on port 9994
No activity for 20 seconds. Closing server.
libgcov profiling error:/home/itamar/projects_CPP/assignment2_OS/tar4/obj_files/drinks_bar.gcda:over
writing profile data with a different checksum

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -U 9994 -t 10 -c 20
=== Current Stock ===
Total atoms : 20
Carbon (C) : 20
Hydrogen (H) : 0
Oxygen (O) : 0
=====
Server listening over tcp on port 9995
Server listening over udp on port 9994
No activity for 10 seconds. Closing server.

```

```

itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -U 9994 -t 20 -c 20 -o 3 -h 15
=== Current Stock ===
Total atoms : 38
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 3
=====
Server listening over tcp on port 9995
Server listening over udp on port 9994
New connection, socket fd: 5
=== Current Stock ===
Total atoms : 45
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 10
=====
Received UDP: DELIVER WATER 2

Delivered 2 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms : 39
Carbon (C) : 20
Hydrogen (H) : 11
Oxygen (O) : 8
=====
No activity for 20 seconds. Closing server.

```

```

itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -U 9994 --timeout 10 -c 20 --oxygen 15 -h 15
=== Current Stock ===
Total atoms : 50
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Server listening over tcp on port 9995
Server listening over udp on port 9994
No activity for 10 seconds. Closing server.

```

```

No activity for 20 seconds. Closing server.
itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -t 20 -c 20 -o 3 -h 15
Error: Both -T (TCP port) and -U (UDP port) flags are required.
Usage: ./drinks_bar -T <tcp_port> -U <udp_port> [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]

```

```

itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./drinks_bar -T 9995 -t 10 -c 20 -U 9994 --oxygen 15 -h 15
=== Current Stock ===
Total atoms : 50
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Server listening over tcp on port 9995
Server listening over udp on port 9994

```

התאמת הלקוח והספק לעבוד עם אופציות:

```

itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./atom_supplier localhost 9995
Usage: ./atom_supplier -h <hostname> -p <port>
libgcov profiling error: /home/itamar/projects_CPP/assignment2_OS/tar4/obj_files/atom_supplier.gcda:overwriting an
existing profile data with a different checksum
itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./atom_supplier -h localhost -p 9995
Connected to server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 4

```

```

itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./molecule_requester localhost 9994
Usage: ./molecule_requester -h <hostname> -p <port>
libgcov profiling error: /home/itamar/projects_CPP/assignment2_OS/tar4/obj_files/molecule_requester.gcda:overwriting an
existing profile data with a different checksum
itamar@DESKTOP-2A3B0QL: ~/projects_CPP/assignment2_OS/tar4$ ./molecule_requester -h localhost -p 9994
Connected to server over UDP. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
5 - EXIT
Enter number of water molecules to create: 2
Server response: SUCCESSFULLY DELIVERED WATER
Do you want to deliver more? (y/press any key to end)

```



## שאלה 5 :

הוראות הרצה :

1. יש להיכנס אל תיקיית tar5 (לאחר בניית make רקורסיבי)
2. `./drinks_bar [-T <tcp_port> -U <udp_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]`
3. `./atom_supplier [-h <hostname> -p <port> | -f <file path>]`
4. `./molecule_requester [-h <hostname> -p <port> | -f <file path>]`

- בשאלה זו התבקשנו לאפשר תקשורת באמצעות UDS ביצענו זאת באופן הבא :
- תחילה ביצענו אבחנה בין קבלת נתבי לקובץ כארגומנט או פורט וhostname תוך אילוץ להופעה של אחד משניהם וזריקת שגיאה כאשר ישנה הופעה של שניהם.
- בעת קבלת דגלים `-d` ו`-s` חילצנו את כתובת הקובץ אותם אנחנו ממפים לתקשורת ונאתחל משתנים גלובליים לטובת בדיקה וניתוב התקשורת לפונקציות שונות כמו כן נבחן סתירה בארגומנטים כמתבקש.

```
case 's':
    stream_path = optarg;
    has_stream_path = true;
    break;
case 'd':
    datagram_path = optarg;
    has_datagram_path = true;
    break;
default:
    std::cerr << "Usage: " << argv[0]
              << " [-T <tcp_port> -U <udp_port>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]\n";
    exit(EXIT_FAILURE);
}

if ((has_tcp || has_udp) && (has_stream_path || has_datagram_path)) {
    std::cerr << "Error: Cannot use both TCP/UDP and UDS options together.\n";
    std::cerr << "Usage: " << argv[0]
              << " [-T <tcp_port> -U <udp_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]\n";
    exit(EXIT_FAILURE);
}

if ((has_tcp || has_udp) && !(has_tcp || has_udp)) {
    std::cerr << "Error: Both -T (TCP port) and -U (UDP port) flags are required.\n";
    std::cerr << "Usage: " << argv[0]
              << " [-T <tcp_port> -U <udp_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]\n";
    exit(EXIT_FAILURE);
}
```

- ניתוב התקשורת לפונקציות שונות כשמדובר בuds

```
try {
    if (has_datagram_path || has_stream_path) {
        run_server_uds(has_datagram_path, has_stream_path);
    }
    else {
        run_server(port_tcp, port_udp);
    }
} catch (const std::runtime_error &e) {
    std::cout << "Error: " << e.what() << std::endl;
}
```

- בפונקציה `run_server_uds` ניצור 2 sockets מסוג `AF_UNIX` כאשר אחד ישמש עבור `datagram` והשני עבור `stream`.
- בשניהם נאתחל את המקום בזיכרון, נגדיר struct מסוג `socketaddr_un` ונאתחל את השדות שלו בהתאמה לקלט שקיבלנו. לאחר מכן נבצע `bind` בין `fd` של הקובץ שהתקבל בארגומנט לבין ה `socket` שנוצר על מנת שיהיה ניתן לכתוב אליו.
- ב `stream` נבצע גם `listen` על מנת שנוכל לקבוע תור בדומה ל `tcp` וב `datagram` נעשה זאת.

- החל מכאן העבודה היא זהה לקודם למעט השינוי ב fd ובstruct השונים שהותאמו לuds כבר בשלב בניית ה socket .

```
// Create a uds stream
if (!stream_path.empty()) {
    uds_stream_fd = socket(AF_UNIX, SOCK_STREAM, 0);
    if (uds_stream_fd < 0) throw std::runtime_error("UDS dgram socket failed");

    struct sockaddr_un stream_addr;
    memset(&stream_addr, 0, sizeof(stream_addr));
    stream_addr.sun_family = AF_UNIX;
    strncpy(stream_addr.sun_path, stream_path.c_str(), sizeof(stream_addr.sun_path) - 1);

    unlink(stream_path.c_str());
    if (bind(uds_stream_fd, (struct sockaddr*)&stream_addr, sizeof(stream_addr)) < 0)
        throw std::runtime_error("UDS stream bind failed");
    if (listen(uds_stream_fd, 10) < 0)
        throw std::runtime_error("UDS stream listen failed");

    std::cout << "Server listening over UDS stream at " << stream_path << std::endl;
}

// Create a uds datagram
if (!datagram_path.empty()) {
    uds_dgram_fd = socket(AF_UNIX, SOCK_DGRAM, 0);
    if (uds_dgram_fd < 0) throw std::runtime_error("UDS stream socket failed");

    struct sockaddr_un dgram_addr;
    memset(&dgram_addr, 0, sizeof(dgram_addr));
    dgram_addr.sun_family = AF_UNIX;
    strncpy(dgram_addr.sun_path, datagram_path.c_str(), sizeof(dgram_addr.sun_path) - 1);

    unlink(datagram_path.c_str());
    if (bind(uds_dgram_fd, (struct sockaddr*)&dgram_addr, sizeof(dgram_addr)) < 0)
        throw std::runtime_error("UDS datagram bind failed");

    std::cout << "Server listening over UDS datagram at " << datagram_path << std::endl;
}
```

- בלקוחות ביצענו התאמה דומה בפיצול הארגומנטים ויצירת הsockets , יש לשים לב שלקוח אחד עובד רק מעל dgram \ udp ולקוח אחד מעל stream\tcp וזה יבחר על פי הארגומנטים שנבחרו.

כעת נציג מספר דוגמאות הרצה :

1. קלט חסר
2. קלט סותר
3. קלט של hostname + port
4. קלט של file כולל עבודה במקביל עם הלקוח והספק וכולל עצירת שעון.
1. חסר -d

```
ttamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_US/tar$ ./drinks_bar -c 20 -o 15 -h 15 -s ./tmp/stream.txt
Error: Both -s (stream path) and -d (datagram path) flags are required for UDS mode.
Usage: ./drinks_bar [-T <tcp_port> -U <udp_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]
```

2. יש גם tcp + udp וגם -d + -s

```
ttamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_US/tar$ ./drinks_bar -T 9995 -U 9994 -c 20 -o 15 -h 15 -s ./tmp/stream.txt -d ./tmp/dgram.txt
Error: Cannot use both TCP/UDP and UDS options together.
Usage: ./drinks_bar [-T <tcp_port> -U <udp_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>]
```

### 3. בדיקה כי הדרישות הקודמות עדיין עובדות:

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar5$ ./drinks_bar -T 9991 -U 9990 -c 20 -o 15 -h
=== Current Stock ===
Total atoms : 50
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Server listening over tcp on port 9991
Server listening over udp on port 9990
New connection, socket fd: 5
=== Current Stock ===
Total atoms : 60
Carbon (C) : 30
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Received UDP: DELIVER WATER 1
Delivered 1 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms : 57
Carbon (C) : 30
Hydrogen (H) : 13
Oxygen (O) : 14
=====
Client disconnected, socket fd: 5
```

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar5$ ./atom_supplier -h localhost -p 9991
Connected to server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 10

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 4
```

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar5$ ./molecule_requester -h localhost -p 9990
Connected to server over UDP. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
5 - EXIT
1
Enter number of water molecules to create: 1
Server response: SUCCESSFULLY DELIVERED WATER
Do you want to deliver more? (y/press any key to end)
n
libgcov profiling error:/home/itamar/projects_CPP/assignment2_05/tar5/obj_files/molecule_requester.gcda:overwriti
ng an existing profile data with a different checksum
```

### 4. כעת נציג קבלת תוספת מהספק, בקשה של מולקולה מהלקוח ובקשה של משקה מהטרמינל.

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar5$ ./drinks_bar -c 20 -o 15 -t30 -h 15 -s ./tmp/stream.txt
-d ./tmp/dgram.txt
=== Current Stock ===
Total atoms : 50
Carbon (C) : 20
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
New UDS stream connection, socket fd: 5
=== Current Stock ===
Total atoms : 60
Carbon (C) : 30
Hydrogen (H) : 15
Oxygen (O) : 15
=====
Received UDS datagram: DELIVER CARBON_DIOXIDE 2
Delivered 2 carbon dioxide molecules.
=== Current Stock ===
Total atoms : 54
Carbon (C) : 28
Hydrogen (H) : 15
Oxygen (O) : 11
=====
GEN SOFT DRINK
Received input: GEN SOFT DRINK
Number of soft drinks that can be created: 1
Client disconnected, socket fd: 5
No activity for 30 seconds. Closing server.
```

```
itamar@DESKTOP-2A3B00L:~/projects_CPP/assignment2_OS/tar5$ ./atom_supplier -f ./tmp/stream.txt
Connected to UDS stream server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 10

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 4
```

```
itamar@DESKTOP-2A3B00L:~/projects_CPP/assignment2_OS/tar5$ ./molecule_requester -f ./tmp/dgram.txt
Connected to UDS datagram server. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
5 - EXIT
2
Enter number of carbon dioxide molecules to create: 2
Server response: SUCCESSFULLY DELIVERED CARBON DIOXIDE
```

## שאלה 6 : (הערות הסבר בקוד כתובות בעיקר בחלק זה)

- התבקשנו לנהל שמירה של המלאי בקובץ שיסונכרן בין כלל התהליכים של הברמנים.
- במידה ולא הגיע הדגל f- נעבוד על פי הערכים המתקבלים באופציות כמו בסעיפים הקודמים (בעבודה על גבי פורטים וגם uds)
- במידה והתקבל הדגל f- והקובץ לא קיים ניצור אותו ונתחיל לסנכרן אותו בכל שינוי שיתבצע כך שכל ברמן אחר יראה ב"זמן אמת" את המידע העדכני על המלאי.
- את הכתיבה לקובץ וקריאה ממנו נבצע בעזרת הפונקציה mmap באופן זה נוכל להקצות חלק מקובץ (או קובץ שלם) בינארי לטובת זיכרון. באופן זה נאפשר לכלל התהליכים לגשת לאותו קובץ ולהתעדכן במידע (ולעדכן בעצמם)
- כמו כן עלינו לוודא שאין גישה כפולה לקובץ, כלומר שאין 2 ברמנים שונים שמוסיפים אטומים אך לא מעודכנים שהברמן האחר גם מוסיף ובעצם משאירים ערך שגוי בקובץ. את הסנכרון הזה נבצע באמצעות fcntl בכך שבכל כתיבה וקריאה תחילה ננעל את הקובץ ואת נבצע את הפעולה.
- הוראות הרצה :
- יש להיכנס אל תיקיית tar6 (לאחר בניית make רקורסיבי)
- ./drinks\_bar [-T <tcp\_port> -U <udp\_port> | -s <UDS stream file path> -d <UDS datagram file path>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>] [-t <timeout>] [-f <stock file path>]
- ./atom\_supplier [-h <hostname> -p <port> | -f <file path>]
- ./molecule\_requester [-h <hostname> -p <port> | -f <file path>]
- נציג כאן את הדרך לזיהוי האם הקובץ קיים כדי לדעת האם לסנכרן אליו או ממנו.

```
case 'f':
    save_file = optarg;
    load_from_file = true;
    struct stat st;
    if (stat(save_file.c_str(), &st) != 0) {
        new_file = true;
        int fd = open(save_file.c_str(), O_RDWR | O_CREAT, 0666); // Open the file for reading
        if (fd < 0) {
            std::cerr << "Error: Could not open file " << save_file << std::endl;
            return;
        }
    }
    break;
```

- נציג את חלק הקוד שמבצע נעילה לקובץ ומיפוי למידע כאשר קוראים מהקובץ אל struct של האטומים :

```
771
772 void loadFromFile(std::string save_file){
773
774     int fd = open(save_file.c_str(), O_RDWR | O_CREAT, 0666);
775     if (fd < 0) {
776         std::cerr << "Error: Could not open file " << save_file << std::endl;
777         exit(EXIT_FAILURE);
778     }
779     if (ftruncate(fd, sizeof(Stock)) == -1) { // Set the file size to the size of Stock
780         std::cerr << "Error: Could not set file size" << std::endl;
781         close(fd);
782         exit(EXIT_FAILURE);
783     }
784     struct flock fl; // File lock structure
785     fl.l_type = F_WRLCK; // Set the lock type to write lock
786     fl.l_whence = SEEK_SET; // Set the lock starting point to the beginning of the file
787     fl.l_start = 0; // Start locking from the beginning of the file
788     fl.l_len = 0; // Lock the entire file
789     fcntl(fd, F_SETLK, &fl); // Apply the lock
```

```

790 // Map the file to memory
791 Stock* mapped_stock = (Stock*)mmap(NULL, sizeof(Stock), PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
792 if (mapped_stock == MAP_FAILED) {
793     std::cerr << "Error: mmap failed" << std::endl;
794     close(fd);
795     exit(EXIT_FAILURE);
796 }
797
798 my_stock->atom_count = mapped_stock->atom_count;
799 my_stock->carbon_count = mapped_stock->carbon_count;
800 my_stock->hydrogen_count = mapped_stock->hydrogen_count;
801 my_stock->oxygen_count = mapped_stock->oxygen_count;
802
803 fl.l_type = F_UNLCK; // Set the lock type to unlock
804 fcntl(fd, F_SETLK, &fl); // Release the lock
805 munmap(mapped_stock, sizeof(Stock)); // Unmap the memory
806 close(fd);

```

כעת נציג מספר מקרים לטובת בדיקה שהתרגילים הקודמים עדיין עובדים וגם כי האפשרויות הנוספות ממומשות כראוי :

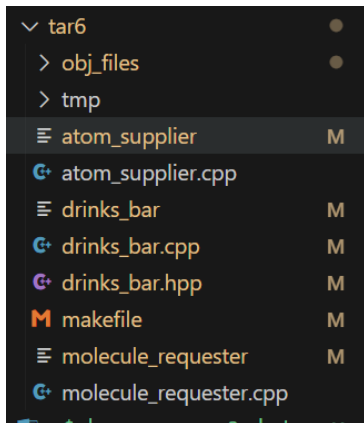
1. הרצה עם uds כולל אופציות וללא file
2. הרצה עם uds כולל אופציות עם file חדש
3. הרצה עם uds כולל אופציות ו file קיים, הוספת של אטומים, בקשה של מולקולות ובקשה של משקאות.
4. בדיקה שאחרי סעיף 3 המלאי מתעדכן ע"פ הקובץ.
5. הרצה עם tcp + udp על מנת לבדוק שהאפשרויות הקודמות עדיין עובדות.
6. הרצת מספר ברמנים במקביל וביצוע פעולת תוספת אטומים באחד והצגה כי הקובץ מסונכרן ביניהם בזמן אמת.

.1

```

ltamar@DESKTOP-2A3BQQL:~/projects_CPP/assignment2_05/tar6$ ./drinks_bar -s ./tmp/stream.txt -d ./tmp/dgram.txt -
o 10 -c 30 -h 15
false
=== Current Stock ===
Total atoms : 55
Carbon (C) : 30
Hydrogen (H) : 15
Oxygen (O) : 10
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
New UDS stream connection, socket fd: 5

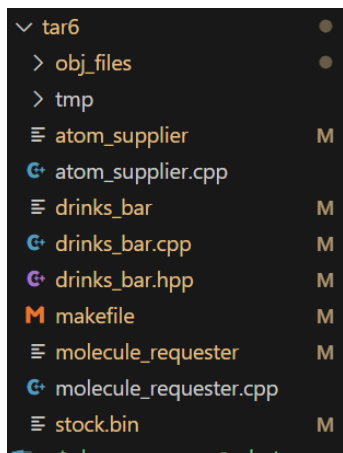
```



```

itamar@DESKTOP-2A3B00L: ~/projects_CPP/assignment2_05/tar6$ ./drinks_bar -s ./tmp/stream.txt -d ./tmp/dgram.txt -o 10 -c 30 -f stock.bin
true
=== Current Stock ===
Total atoms : 40
Carbon (C) : 30
Hydrogen (H) : 0
Oxygen (O) : 10
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
New UDS stream connection, socket fd: 6
=== Current Stock ===
Total atoms : 55
Carbon (C) : 45
Hydrogen (H) : 0
Oxygen (O) : 10
=====
Client disconnected, socket fd: 6
No activity for 60 seconds. Closing server.
itamar@DESKTOP-2A3B00L: ~/projects_CPP/assignment2_05/tar6$

```



```

itamar@DESKTOP-2A3B00L: ~/projects_CPP/assignment2_05/tar6$ ./atom_supplier -f ./tmp/stream.txt
Connected to UDS stream server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 15

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./drinks_bar -s ./tmp/stream.txt -d ./tmp/dgram.txt -
o 10 -c 30 -f stock.bin
false
=== Current Stock ===
Total atoms : 55
Carbon (C) : 45
Hydrogen (H) : 0
Oxygen (O) : 10
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
New UDS stream connection, socket fd: 5
=== Current Stock ===
Total atoms : 70
Carbon (C) : 45
Hydrogen (H) : 15
Oxygen (O) : 10
=====
=== Current Stock ===
Total atoms : 85
Carbon (C) : 45
Hydrogen (H) : 15
Oxygen (O) : 25
=====
Received UDS datagram: DELIVER WATER 2

```

```

Delivered 2 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms : 79
Carbon (C) : 45
Hydrogen (H) : 11
Oxygen (O) : 23
=====
GEN CHAMPAGNE
Received input: GEN CHAMPAGNE
Number of champagne that can be created: 0
No activity for 60 seconds. Closing server.

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./atom_supplier -f ./tmp/stream.txt
Connected to UDS stream server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 2
Enter number of hydrogen atoms to add: 15

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 3
Enter number of oxygen atoms to add: 15

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 4

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./molecule_requester -f ./tmp/dgram.txt
Connected to UDS datagram server. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
5 - EXIT
1
Enter number of water molecules to create: 2
Server response: SUCCESSFULLY DELIVERED WATER

```



#### 4. אחרי טעינה מחדש :

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar6$ ./drinks_bar -s ./tmp/stream.txt -d ./tmp/dgram.txt -o 10 -c 30 -f stock.bin
false
=== Current Stock ===
Total atoms : 79
Carbon (C) : 45
Hydrogen (H) : 11
Oxygen (O) : 23
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
```

.5

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar6$ ./drinks_bar -T 9999 -U 9998 -o 10 -c 30 -f stock.bin
false
=== Current Stock ===
Total atoms : 79
Carbon (C) : 45
Hydrogen (H) : 11
Oxygen (O) : 23
=====
Server listening over tcp on port 9999
Server listening over udp on port 9998
New connection, socket fd: 5
=== Current Stock ===
Total atoms : 89
Carbon (C) : 55
Hydrogen (H) : 11
Oxygen (O) : 23
=====
Received UDP: DELIVER WATER 2

Delivered 2 water molecules.
Successfully delivered water.
=== Current Stock ===
Total atoms : 83
Carbon (C) : 55
Hydrogen (H) : 7
Oxygen (O) : 21
=====
GEN VODKA
Received input: GEN VODKA
Number of vodka that can be created: 0
No activity for 60 seconds. Closing server.
```

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar6$ ./atom_supplier -h localhost -p 9999
Connected to server. Enter commands:
```

```
Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 10
```

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_05/tar6$ ./molecule_requester -h localhost -p 9998
Connected to server over UDP. Enter commands:
What molecule do you want to create?
1 - WATER
2 - CARBON_DIOXIDE
3 - ALCOHOL
4 - GLUCOSE
5 - EXIT
1
Enter number of water molecules to create: 2
Server response: SUCCESSFULLY DELIVERED WATER
Do you want to deliver more? (y/press any key to end)
n
```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./drinks_bar -s ./tmp/stream.txt -d ./tmp/dgram.txt -
-o 10 -c 30 -f stock.bin
false
=== Current Stock ===
Total atoms   : 47
Carbon (C)    : 37
Hydrogen (H)  : 0
Oxygen (O)    : 10
=====
Server listening over UDS stream at ./tmp/stream.txt
Server listening over UDS datagram at ./tmp/dgram.txt
New UDS stream connection, socket fd: 5
=== Current Stock ===
Total atoms   : 50
Carbon (C)    : 40
Hydrogen (H)  : 0
Oxygen (O)    : 10
=====
Client disconnected, socket fd: 5
No activity for 60 seconds. Closing server.
libgcov profiling error:/home/itamar/projects_CPP/assignment2_OS/tar6/obj_files/drinks_bar.gcda:overwriting an e
xisting profile data with a different checksum

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./atom_supplier -f ./tmp/stream.txt
Connected to UDS stream server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 3

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./drinks_bar -s ./tmp/stream1.txt -d ./tmp/dgram1.txt
-o 10 -c 20 -f stock.bin
false
=== Current Stock ===
Total atoms   : 47
Carbon (C)    : 37
Hydrogen (H)  : 0
Oxygen (O)    : 10
=====
Server listening over UDS stream at ./tmp/stream1.txt
Server listening over UDS datagram at ./tmp/dgram1.txt
New UDS stream connection, socket fd: 5
=== Current Stock ===
Total atoms   : 53
Carbon (C)    : 43
Hydrogen (H)  : 0
Oxygen (O)    : 10
=====
Client disconnected, socket fd: 5
No activity for 60 seconds. Closing server.

```

```

itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ ./atom_supplier -f ./tmp/stream1.txt
Connected to UDS stream server. Enter commands:

Choose atom to add:
1. Add Carbon
2. Add Hydrogen
3. Add Oxygen
4. Exit
Enter your choice: 1
Enter number of carbon atoms to add: 3

```

## : Code coverage

חילצנו דוח עבור תרגיל 6 בלבד מאחר והוא מכיל בתוכו את כל שאר התרגילים, כלל קבצי ה gcov נמצאים בתיקיית gcov\_tar6 לאחר כל ההרצות הרלוונטיות.

הוראות הרצה :

אחראי המערכת המוקדש נריץ את כל התוכניות (לפי הוראות ההרצה מעלה) ונבצע את הפעולות הבאות :

1. הרצת drinks\_bar עם uds ועם file שלא קיים לאחר מכן עם ports
2. הרצת atom\_supplier
3. ביצוע כלל האפשרויות של atom\_supplier
4. הרצת gcov atom\_supplier.cpp
5. פתיחת הקובץ שנוצר בשם atom\_supplier.cpp.gcov

```
itamar@DESKTOP-2A3B0QL:~/projects_CPP/assignment2_OS/tar6$ gcov atom_supplier.cpp
File '/usr/include/c++/13/bits/move.h'
Lines executed:66.67% of 6
Creating 'move.h.gcov'

File '/usr/include/c++/13/bits/new_allocator.h'
Lines executed:100.00% of 2
Creating 'new_allocator.h.gcov'

File '/usr/include/c++/13/bits/basic_string.h'
Lines executed:100.00% of 14
Creating 'basic_string.h.gcov'

File '/usr/include/c++/13/bits/charconv.h'
Lines executed:65.38% of 26
Creating 'charconv.h.gcov'

File '/usr/include/c++/13/bits/allocator.h'
Lines executed:100.00% of 2
Creating 'allocator.h.gcov'

File 'atom_supplier.cpp'
Lines executed:94.50% of 109
Creating 'atom_supplier.cpp.gcov'

File '/usr/include/c++/13/bits/char_traits.h'
Lines executed:0.00% of 4
Creating 'char_traits.h.gcov'

Lines executed:87.12% of 163
```

6. הרצת molecule\_requester
7. ביצוע כלל האפשרויות עבורו
8. הרצת gcov molecule\_requester.cpp
9. פתיחת הקובץ שנוצר בשם molecule\_requester.cpp.gcov

```

itamar@DESKTOP-2A3BOQL:~/projects_CPP/assignment2_OS/tar6$ gcov molecule_requester.cpp
File '/usr/include/c++/13/bits/move.h'
Lines executed:66.67% of 6
Creating 'move.h.gcov'

File '/usr/include/c++/13/bits/new_allocator.h'
Lines executed:100.00% of 2
Creating 'new_allocator.h.gcov'

File '/usr/include/c++/13/bits/basic_string.h'
Lines executed:100.00% of 14
Creating 'basic_string.h.gcov'

File '/usr/include/c++/13/bits/charconv.h'
Lines executed:100.00% of 26
Creating 'charconv.h.gcov'

File '/usr/include/c++/13/bits/allocator.h'
Lines executed:100.00% of 2
Creating 'allocator.h.gcov'

File 'molecule_requester.cpp'
Lines executed:94.41% of 143
Creating 'molecule_requester.cpp.gcov'

File '/usr/include/c++/13/bits/char_traits.h'
Lines executed:0.00% of 4
Creating 'char_traits.h.gcov'

Lines executed:92.89% of 197

```

10. נבצע את שאר הפעולות ש `drinks_bar` יכול לבצע כדי לגשת לכל השורות בקוד שלו:

```

itamar@DESKTOP-2A3BOQL:~/projects_CPP/assignment2_OS/tar6$ gcov drinks_bar.cpp
drinks_bar.cpp:source file is newer than notes file 'drinks_bar.gcno'
(the message is displayed only once per source file)
drinks_bar.hpp:source file is newer than notes file 'drinks_bar.gcno'
File 'drinks_bar.cpp'
Lines executed:88.83% of 609
Creating 'drinks_bar.cpp.gcov'

File 'drinks_bar.hpp'
No executable lines
Removing 'drinks_bar.hpp.gcov'

File '/usr/include/c++/13/bits/new_allocator.h'
Lines executed:100.00% of 2
Creating 'new_allocator.h.gcov'

File '/usr/include/c++/13/bits/allocator.h'
Lines executed:100.00% of 2
Creating 'allocator.h.gcov'

File '/usr/include/c++/13/bits/basic_string.tcc'
Lines executed:66.67% of 36
Creating 'basic_string.tcc.gcov'

File '/usr/include/c++/13/bits/stl_iterator_base_types.h'
Lines executed:100.00% of 1
Creating 'stl_iterator_base_types.h.gcov'

File '/usr/include/c++/13/bits/stl_iterator_base_funcs.h'
Lines executed:100.00% of 3
Creating 'stl_iterator_base_funcs.h.gcov'

File '/usr/include/c++/13/bits/basic_string.h'
Lines executed:92.86% of 14
Creating 'basic_string.h.gcov'

File '/usr/include/c++/13/bits/move.h'
Lines executed:0.00% of 2
Creating 'move.h.gcov'

File '/usr/include/c++/13/bits/char_traits.h'
Lines executed:28.00% of 25
Creating 'char_traits.h.gcov'

File '/usr/include/c++/13/ext/string_conversions.h'
Lines executed:88.24% of 17
Creating 'string_conversions.h.gcov'

File '/usr/include/c++/13/bits/ios_base.h'
Lines executed:100.00% of 11
Creating 'ios_base.h.gcov'

File '/usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h'
Lines executed:100.00% of 2
Creating 'c++config.h.gcov'

Lines executed:85.77% of 724

```

gcov drink\_bar.cpp .11  
make coverage .12 (אופציונלי לסדר)  
פתח את קובץ ה gcov.cpp. עבור כל אחד מהקבצים לעיל.