

מחסן המולקולות

בחלק זה של המטלה, ביצענו הרחבה לחלק 1:

בכך שאת השרת, שאותו עשינו בחלק 1, לא רק שהוא מקבל אטומים דרך **TCP**, אלא גם משרת בקשות ליצירת מולקולות דרך **UDP**.

ובכך, אנו רוצים להמשיך לשמר את מה שעשינו בחלק 1, ולכן כל הקובץ של **atom_supplier** יופיע גם בחלק 2, וניצור קובץ של לקוח נוסף **molecule_requester**, אשר תפקידו לבקש קבלת מולקולות, דרך תקשורת המבוססת **UDP**.

לכן, השרת שלנו נהפך למערכת דו שלבית:

לקוח אחד אחראי להזין אטומים למחסן, דרך **TCP**.

לקוח שני מבקש לקבל מולקולות, דרך **UDP**.

לשרת המשודרג שלנו נקרא **molecule_server**, וכעת נסביר מה הוספנו אליו על מנת לבצע האזנה בו-זמנית:

הוספנו יכולת **UDP** לשרת ← יצרנו socket נוסף מסוג **UDP** ששמו **udp_fd**:

```
// Create the UDP socket:
int udp_fd = socket(AF_INET, SOCK_DGRAM, 0);
if (udp_fd < 0) {
    perror("UDP socket failed");
    return 1;
}
```

נבצע לו **bind()** לכתובת **udp_addr** באותו הפורט של ה-**TCP**:

```
sockaddr_in udp_addr;
std::memset(&udp_addr, 0, sizeof(udp_addr));
udp_addr.sin_family = AF_INET;
udp_addr.sin_addr.s_addr = INADDR_ANY;
udp_addr.sin_port = htons(port); // same port as TCP
```

```
//bind to the UDP :
if (bind(udp_fd, (sockaddr*)&udp_addr, sizeof(udp_addr)) < 0) {
    perror("UDP bind failed");
    close(server_fd);
    close(udp_fd);
    return 1;
}
```

בנוסף, הוספנו את יכולת ה-**UDP** שלנו ל-**select()** לצד socket ה-**TCP**:

```
// Active waiting until one of the sockets will be ready for reading:
int activity = select(max_fd + 1, &readfds, nullptr, nullptr, nullptr);
if (activity < 0 && !shutdown_requested) {
    perror("Select error");
    break;
}
```

```
FD_SET(udp_fd, &readfds);
if (udp_fd > max_fd) {
    max_fd = udp_fd;
}
```

בהמשך נראה כיצד הלקוח שלנו (אשר מתקשר למולקולות ול-**UDP**) התממש, אבל כעת נניח וקיבלנו מהלקוח שלנו שורת פקודה בסגנון: **<AMOUNT> <MOLECULE> <DELIVER>**, ביצענו ניתוח של פקודה זו דרך **std::istringstream**

כאן הפקודה שקיבלנו מפורקת ל-3 רכיבים:
action שזה המילה DELIVER, molecule,
שזהו שם המולקולה, וה-amount, שזה
כמות הבקשות.

```
std::istringstream iss(command);
std::string action, molecule;
unsigned long long amount;

if (!(iss >> action >> molecule >> amount) || action != "DELIVER" || amount <= 0) {
    std::string response = "CANNOT_DELIVER";
    sendto(udp_fd, response.c_str(), response.size(), 0,
           (sockaddr*)&client_addr, client_len);
    continue;
}
```

אם השרת שלנו לא הצליח לקרוא את שלושת הערכים, או שהפעולה אינה DELIVER, או שהכמות ששמנו אינה חיובית ← הפקודה הינה שגויה, מחזיר ללקוח שלנו תשובה שלילית, ומדלג לבקשה הבאה בלולאה.

כעת, נבדוק את סוג המולקולה שלנו:

מים ← דרוש 2 מימן, וחמצן אחד לכל מולקולה
פחמן דו-חמצני ← דרוש פחמן אחד, ו-2 חמצן.
גלוקוז ← דרוש 6 פחמן, 12 מימן, ו-6 חמצן.
אלכוהול ← דרוש 2 פחמן, 6 מימן, ו-1 חמצן.
אחרת, מלבד המולקולות האלו ← נחזיר שאי אפשר לבצע שליחה.

```
// The requires molecules:
unsigned long long need_H = 0, need_O = 0, need_C = 0;
if (molecule == "WATER") {
    need_H = 2 * amount;
    need_O = 1 * amount;
} else if (molecule == "CARBON DIOXIDE") {
    need_C = 1 * amount;
    need_O = 2 * amount;
} else if (molecule == "GLUCOSE") {
    need_C = 6 * amount;
    need_H = 12 * amount;
    need_O = 6 * amount;
} else if (molecule == "ALCOHOL") {
    need_C = 2 * amount;
    need_H = 6 * amount;
    need_O = 1 * amount;
} else {
    std::string response = "CANNOT_DELIVER";
    sendto(udp_fd, response.c_str(), response.size(), 0,
           (sockaddr*)&client_addr, client_len);
    continue;
}
```

כעת, אחרי שידוע לנו כמה אטומים דרושים עבור כל מולקולה ספציפית, נבצע בדיקה ← משמע: אחרי ששמנו בדיקה של שליחת מולקולה עם כמות מסוימת, נלך ישר אל מחסן האטומים שלנו, זאת לבדוק האם יש לנו מספיק אטומים במלאי, במידה ויש ← נחסר ממחסן האטומים שלנו את הכמויות הרלוונטיות, אשר דרושות למולקולה שלנו + עדכון המלאי:

במידה ואין מספיק אטומים ← נודיע ללקוח מצב של CANNOT DELIVER.

```
if (hydrogen_count >= need_H && oxygen_count >= need_O && carbon_count >= need_C) {
    hydrogen_count -= need_H;
    oxygen_count -= need_O;
    carbon_count -= need_C;

    std::cout << "[INFO] UDP request received: DELIVER " << molecule << "\n";
    std::cout << "[INFO] Atoms: H=" << hydrogen_count
              << ", O=" << oxygen_count
              << ", C=" << carbon_count << "\n";
}
```

כעת, נדבר על המחלקה שהחדשה שבנינו, **molecule_requester**, אשר הינה מבצעת את החיבור של ה-UDP לטובת שליחת מולקולה כלשהי עם הכמות הדרושה.

ניצור שם כמו שעשינו בשרת המשודרג socket של UDP, ונגדיר את כתובת השרת.

נריץ לולאה אינסופית עד שהמשתמש שלנו יקליד EXIT, בדיוק כמו עם הלקוח הראשון.

נשלח את הפקודה שלנו דרך ה-UDP לשרת, אפשר להבין כי מדובר ב-UDP דרך הפקודה **sendto**, אם זה נכשל ← נדווח ונצא מהלולאה:

```
// Send the command to the server:
ssize_t sent = sendto(sock, command.c_str(), command.size(), 0,
                      (sockaddr*)&server_addr, sizeof(server_addr));
if (sent < 0) {
    perror("sendto failed");
    break;
}
```

לאחר מכן, נמתין לקבלת תשובה מהשרת שלנו, נמלא את התשובה אל תוך ה-buffer:

התשובה תהיה DELIVERED או CANNOT DELIVER

```
//Get response from the server:
char buffer[1024] = {0};
socklen_t len = sizeof(server_addr);
ssize_t received = recvfrom(sock, buffer, sizeof(buffer) - 1, 0,
                             (sockaddr*)&server_addr, &len);
if (received < 0) {
    perror("recvfrom failed");
    break;
}

buffer[received] = '\0';
std::cout << "[Server Reply] " << buffer << "\n";
```

כעת, נרצה הרצה מסודרת ומקיפה של כל חלק 2:

נבצע **makefile** מסודר לטובת קמפול והרצת החלק שלנו(כולל ביצוע **gcov** בתוך **makefile** שלנו):

```
MoleculesRequest > makefile
1 CXX = g++
2 CXXFLAGS = -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage
3 LDFLAGS = -lgcov
4
5 SRCS = molecule_server.cpp molecule_requester.cpp atom_supplier2.cpp
6 BINS = molecule_server molecule_requester atom_supplier
7
8 all: $(BINS)
9
10 molecule_server: molecule_server.cpp
11     $(CXX) $(CXXFLAGS) -o $@ $(LDFLAGS)
12
13 molecule_requester: molecule_requester.cpp
14     $(CXX) $(CXXFLAGS) -o $@ $(LDFLAGS)
15
16 atom_supplier: atom_supplier2.cpp
17     $(CXX) $(CXXFLAGS) -o $@ $(LDFLAGS)
18
19 clean:
20     rm -f $(BINS) *.gcno *.gda *.gcov
21
```

כעת, נבצע את ההרצה שלנו:

ראשית נפתח את השרת, ונכניס למחסן האטומים שלנו אטומים דרך הלקוח הראשון שלנו(TCP):

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ make
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_server molecule_server.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_requester molecule_requester.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o atom_supplier atom_supplier2.cpp -lgcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./molecule_server 12345
Server is listening on port 12345...

roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./atom_supplier 127.0.0.1 12345
Connected to server at 127.0.0.1:12345
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD CARBON 25
[Server Message]
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD OXYGEN 30
[Server Message]
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD HYDROGEN 40
[Server Message] (
Enter command (e.g., ADD HYDROGEN 3 or EXIT):

roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ make
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_server molecule_server.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_requester molecule_requester.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o atom_supplier atom_supplier2.cpp -lgcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./molecule_server 12345
Server is listening on port 12345...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=0, O=0, C=25
[INFO] Atoms: H=0, O=30, C=25
[INFO] Atoms: H=40, O=30, C=25
```

כעת, נצא מהלקוח הראשון דרך פקודת EXIT ונפתח את הלקוח השני לטובת הוספת מולקולות(UDP):

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./atom_supplier 127.0.0.1 12345
Connected to server at 127.0.0.1:12345
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD CARBON 25
[Server Message]
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD OXYGEN 30
[Server Message]
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD HYDROGEN 40
[Server Message] (
Enter command (e.g., ADD HYDROGEN 3 or EXIT): EXIT
Closing connection. Bye!
```

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./molecule_requester 127.0.0.1 12345
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER WATER 1
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER ALCOHOL 2
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER GLUCOSE 3
[Server Reply] CANNOT DELIVER
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER GLUCOSE 1
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): EXIT
Exiting.
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$
```

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ make
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_server molecule_server.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_requester molecule_requester.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o atom_supplier atom_supplier2.cpp -lgcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./molecule_server 12345
Server is listening on port 12345...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=0, O=0, C=25
[INFO] Atoms: H=0, O=30, C=25
[INFO] Atoms: H=40, O=30, C=25
[INFO] Client disconnected.
[INFO] UDP request received: DELIVER WATER
[INFO] Atoms: H=38, O=29, C=25
[INFO] UDP request received: DELIVER ALCOHOL
[INFO] Atoms: H=26, O=27, C=21
[INFO] UDP request received: DELIVER GLUCOSE
[INFO] Atoms: H=14, O=21, C=15
```

כעת, נבצע סגירה גם לשרת, זאת דרך SHUTDOWN מאחד הלקוחות(כמו שעשינו בחלק 1):

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd MoleculesRequest
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ make
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_server molecule_server.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o molecule_requester molecule_requester.cpp -lgcov
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o atom_supplier atom_supplier2.cpp -lgcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$ ./molecule_server 12345
Server is listening on port 12345...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=0, O=0, C=25
[INFO] Atoms: H=0, O=30, C=25
[INFO] Atoms: H=40, O=30, C=25
[INFO] Client disconnected.
[INFO] UDP request received: DELIVER WATER
[INFO] Atoms: H=38, O=29, C=25
[INFO] UDP request received: DELIVER ALCOHOL
[INFO] Atoms: H=26, O=27, C=21
[INFO] UDP request received: DELIVER GLUCOSE
[INFO] Atoms: H=14, O=21, C=15
[INFO] New client connected: 127.0.0.1
[INFO] Shutdown command received from client.
[INFO] Server shut down.
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/MoleculesRequest$
```