

## אופציות התחלה

נזכיר כי גם בחלק זה, העברנו את כל הקבצים שעשינו בחלק 3, לתוך התיקיה של חלק 4 (אופציות התחלה), ומשם פשוט המשכנו לשפץ את השרת שלנו.

כעת, המטרה שלנו בחלק זה הינו לקבל ערכים התחלתיים למספר האטומים מכל סוג, פורטים ל-TCP, ול-UDP, וגם פרק זמן מקסימלי (Timeout), שבו השרת ירד לבד אם לא קרה כלום.

כעת, המשכנו לשדרג את השרת שלנו, בכך שהוספנו כמה דברים:

האופציות שלנו הינן:

עבור ה-TCP  $T \leftarrow$ , ולכן הוספנו קלט משתנה `tcp_port`, ואנו קולטים את הפורט באמצעות `getopt`, וכך גם עבור ה-UDP  $U \leftarrow$ , ולכן הוספנו קלט משתנה `udp_port` ואנו קולטים גם את הפורט באמצעות `+getopt` נבדוק את התקינות שלהם, זאת על מנת לוודא שהמשתמש לא שכח להכניס את הפורטים  $\leftarrow$  בלי זה השרת לא יכול לפעול:

```
if (tcp_port == -1 || udp_port == -1) {
    std::cerr << "[ERROR] Both -T <tcp_port> and -U <udp_port> are required.\n";
    return 1;
}
```

```
int tcp_port=-1;
int udp_port=-1;
```

כעת, גם נקלוט את מספר האטומים הראשוני מכל סוג:

`h` עבור מימן, `c` עבור פחמן, `o` - `i` עבור חמצן.

ואנו מצבעים שימוש ב-`timeout`  $t \leftarrow$ , אנו מצבעים הגדרת משתנה `timeout_seconds`, זהו מנגנון שמוודא שהשרת שלנו לא ימשיך לרוץ לנצח, אלא ייפסק אוטומטית לאחר מספר שניות מוגדר מראש  $\leftarrow$  אם הוא אינו קיבל קלט חדש, או סתם נרצה לסגור אותו לאחר זמן:

```
int opt;
while ((opt = getopt(argc, argv, "o:c:h:t:T:U:")) != -1) {
    switch (opt) {
        case 'o': oxygen_count = std::stoull(optarg); break;
        case 'c': carbon_count = std::stoull(optarg); break;
        case 'h': hydrogen_count = std::stoull(optarg); break;
        case 't': timeout_seconds = std::stoi(optarg); break;
        case 'T': tcp_port = std::stoi(optarg); break;
        case 'U': udp_port = std::stoi(optarg); break;
        default:
            std::cerr << "Usage: " << argv[0] << " -T <tcp_port> -U <udp_port> [-t <timeout>] [-o <oxygen>] [-c <carbon>] [-h <hydrogen>]\n";
            return 1;
    }
}
```

אנו מבצעים שימוש ב-`std::stoull`, שזהו קיצור ל-`std::string to unsigned long long`, ושימוש ב-`std::stoi`, שזהו קיצור ל-`std::string to int`

ביצענו ניתוח לארגומנטים, פירשנו אותם והמרנו אותם לערכים נומריים כלשהם.

כאן אנחנו מממשים את ה-`timeout`, זאת בעזרת `SIGNALRM`:

המטרה הינה להפעיל `timeout` שיגרום לשרת להיסגר לאחר מספר השניות שנבחר.

```
if (timeout_seconds > 0) {
    signal(SIGALRM, handle_timeout);
    alarm(timeout_seconds);
}
```

כעת נראה הרצה מסודרת לכך, הוספנו ל-**makefile** שלנו הרצות עם ערכים מוגדרים  
לפורטים, מספר שניות מוגדר ב-**timeout**, ומספר אטומים לכל סוג:

```
1 CXX = g++
2 CXXFLAGS = -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage
3 LDFLAGS = -lgcov
4
5 SRCS = options_server.cpp molecule_requester.cpp atom_supplier.cpp
6 BINS = options_server molecule_requester atom_supplier
7
8 all: $(BINS)
9
10 options_server: options_server.cpp
11     $(CXX) $(CXXFLAGS) -o $@ $< $(LDFLAGS)
12
13 molecule_requester: molecule_requester.cpp
14     $(CXX) $(CXXFLAGS) -o $@ $< $(LDFLAGS)
15
16 atom_supplier: atom_supplier.cpp
17     $(CXX) $(CXXFLAGS) -o $@ $< $(LDFLAGS)
18
19 run_server:
20     ./options_server -t 60 -T 5555 -U 7777 -o 12 -c 12 -h 30
21
22 run_supplier:
23     ./atom_supplier 127.0.0.1 5555
24
25 run_requester:
26     ./molecule_requester 127.0.0.1 7777
27
28 clean:
29     rm -f $(BINS) *.gcno *.gda *.gcov
30
```

כעת נראה את ההרצה:

לאחר ביצוע **make**, נבצע **make run\_server** עם הגדרה של ערכים לפי רצוננו ב-**make**,  
כמובן שגם אפשר לכתוב בשורת הפקודה ערכים לפי רצוננו ולא ערכים מוגדרים מראש  
כמו שעשינו ב-**makefile**

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make run_server
./options_server -t 60 -T 5555 -U 7777 -o 12 -c 12 -h 30
Server is listening on TCP port 5555 and UDP port 7777...
```

כעת, נבצע חיבור TCP עם הלקוח עבור מחסן האטומים עם פורט 5555, ונוסיף אטומים עם  
כמות כלשהי:

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules$ cd Options
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ ./atom_supplier 127.0.0.1 5555
Connected to server at 127.0.0.1:5555
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD HYDROGEN 10
Server returned count: 40
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD CARBON 4
Server returned count: 16
Enter command (e.g., ADD HYDROGEN 3 or EXIT): ADD OXYGEN 2
Server returned count: 14
Enter command (e.g., ADD HYDROGEN 3 or EXIT): EXIT
Closing connection. Bye!
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$
```

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make run_server
./options_server -t 60 -T 5555 -U 7777 -o 12 -c 12 -h 30
Server is listening on TCP port 5555 and UDP port 7777...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=40, O=12, C=12
[INFO] Atoms: H=40, O=12, C=16
[INFO] Atoms: H=40, O=14, C=16
[INFO] Client disconnected.
```

כעת, נבצע חיבור UDP עם הלקוח שרוצה להעביר מולקולות, עם הפורט 7777, ונוסיף מולקולות:

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ ./molecule_requester 127.0.0.1 7777
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER WATER 2
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER GLUCOSE 1
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER ALCOHOL 2
[Server Reply] DELIVERED
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): DELIVER ALCOHOL 3
[Server Reply] CANNOT_DELIVER
Enter command (DELIVER <MOLECULE> <AMOUNT> | EXIT): EXIT
Exiting.
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$
```

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make run_server
./options_server -t 60 -T 5555 -U 7777 -o 12 -c 12 -h 30
Server is listening on TCP port 5555 and UDP port 7777...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=40, O=12, C=12
[INFO] Atoms: H=40, O=12, C=16
[INFO] Atoms: H=40, O=14, C=16
[INFO] Client disconnected.
[INFO] UDP request received: DELIVER WATER
[INFO] Atoms: H=36, O=12, C=16
[INFO] UDP request received: DELIVER GLUCOSE
[INFO] Atoms: H=24, O=6, C=10
[INFO] UDP request received: DELIVER ALCOHOL
[INFO] Atoms: H=12, O=4, C=6
```

במידה והשרת שלנו אינו מקבל שום קלט מאיזשהו לקוח לאחר 60 שניות (זאת כי הגדרנו את timeoutn שלנו להיות לאחר 60 שניות), אזי מופעל הtimeoutn ויש shutdown, ובכך השרת שלנו נסגר:

```
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make clean
rm -f options_server molecule_requester atom_supplier *.gcno *.gcda *.gcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make
g++ -std=c++17 -Wall -Wextra -fprofile-arcs -ftest-coverage -o options_server options_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_supplier.cpp -lgcov
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$ make run_server
./options_server -t 60 -T 5555 -U 7777 -o 12 -c 12 -h 30
Server is listening on TCP port 5555 and UDP port 7777...
[INFO] New client connected: 127.0.0.1
[INFO] Atoms: H=40, O=12, C=12
[INFO] Atoms: H=40, O=12, C=16
[INFO] Atoms: H=40, O=14, C=16
[INFO] Client disconnected.
[INFO] UDP request received: DELIVER WATER
[INFO] Atoms: H=36, O=12, C=16
[INFO] UDP request received: DELIVER GLUCOSE
[INFO] Atoms: H=24, O=6, C=10
[INFO] UDP request received: DELIVER ALCOHOL
[INFO] Atoms: H=12, O=4, C=6
[INFO] Timeout reached. Shutting down.
[INFO] Server shut down.
roy3177@LAPTOP-QCUJB7RP:~/Counting molecules/Options$
```