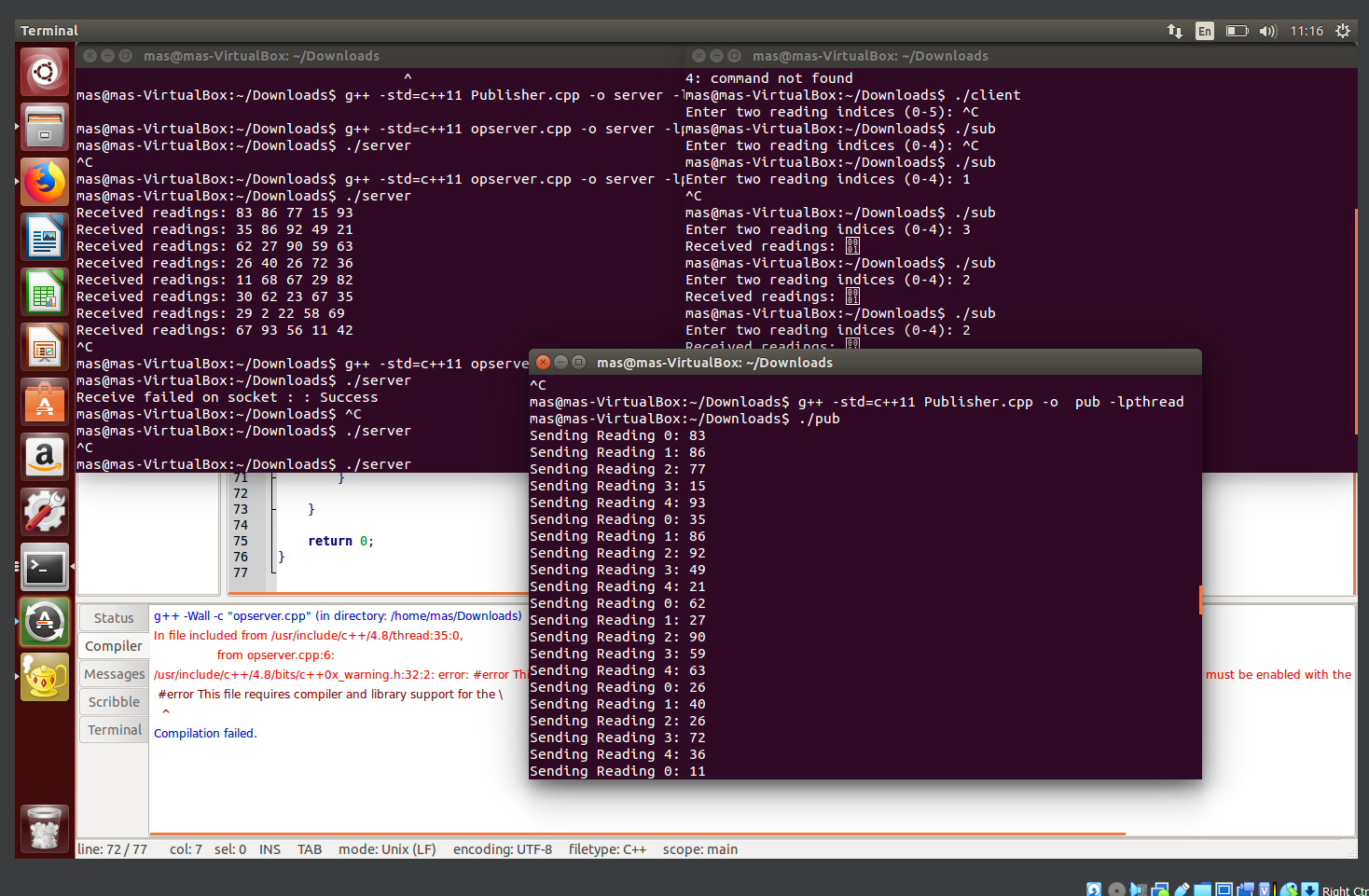
Mohamed Abshar Shihab

398505

CE-43 B

OPEN LAB



**Publisher code:**  
#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string.h>

#include <vector>

#include <thread>

#include <chrono>

#include <string>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

using namespace std;

void publishReadings(int fd) {

while (true) {

string readings;

for (int i = 0; i < 5; i++) {

string reading = to\_string(rand() % 100);

readings += reading + " "; // Generate readings

cout << "Sending Reading " << i << ": " << reading << endl;

}

send(fd, readings.c\_str(), readings.size(), 0);

this\_thread::sleep\_for(chrono::seconds(5)); // Wait for 5 seconds

}

}

int main() {

// Your existing publisher code here...

}

**Server Code:**  
#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string.h>

#include <vector>

#include <thread>

#include <chrono>

#include <string>

#include <sstream>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

using namespace std;

vector<string> readings(5, "0"); // Initialize readings

int main() {

int fd = socket(AF\_INET, SOCK\_STREAM, 0);

if (fd == -1) {

perror("Socket creation failed : ");

exit (-1);

}

struct sockaddr\_in s\_addr;

s\_addr.sin\_addr.s\_addr = INADDR\_ANY;

s\_addr.sin\_family = AF\_INET;

s\_addr.sin\_port = htons(80);

if (bind(fd, (struct sockaddr\*)(&s\_addr), sizeof(s\_addr)) == -1) {

perror("Bind failed on socket : ");

exit(-1);

}

int backlog = 4;

if (listen(fd, backlog) == -1) {

perror("listen failed on socket: ");

exit(-1);

}

struct sockaddr\_in c\_addr;

socklen\_t cliaddr\_len = sizeof(c\_addr);

int connfd = accept(fd, (struct sockaddr\*)&c\_addr, &cliaddr\_len);

if (connfd < 0) {

perror("accept failed on socket: ");

exit(-1);

}

char buffer[1000];

while (1) {

bzero(buffer, sizeof(buffer));

if (recv(connfd, buffer, 1000, 0) > 0) {

string received(buffer);

istringstream iss(received);

for (string s; iss >> s; ) {

readings.push\_back(s);

}

cout << "Received readings: " << received << endl;

}

else {

perror("Receive failed on socket : ");

break;

}

bzero(buffer, sizeof(buffer));

if (recv(connfd, buffer, 1000, 0) > 0) {

int reading1 = atoi(buffer);

int reading2 = atoi(buffer + 1);

string response = readings[reading1] + " " + readings[reading2];

send(connfd, response.c\_str(), response.size(), 0);

}

}

return 0;

}

**Subscriber code:**

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

using namespace std;

int main() {

int fd = socket(AF\_INET, SOCK\_STREAM, 0);

if (fd == -1) {

perror("Socket creation failed : ");

exit (-1);

}

struct sockaddr\_in s\_addr;

s\_addr.sin\_family = AF\_INET;

s\_addr.sin\_port = htons(80);

inet\_aton("127.0.0.1", &s\_addr.sin\_addr);

if (connect(fd, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1) {

perror("Connect failed on socket : ");

exit(-1);

}

char buffer[1000];

cout << "Enter two reading indices (0-4): ";

cin.getline(buffer, 1000);

send(fd, buffer, strlen(buffer), 0);

char response[1000];

recv(fd, response, 1000, 0);

cout << "Received readings: " << response << endl;

return 0;

}