CSE 344 FINAL SYSTEM PROGRAMMING

PIDESHOP PROJECT

Spring 2023-24, JUNE 15

Buket Gençer 210104004298

Table of Contents

1. Introduction

This report documents the development and testing of a simulation for a food production and delivery system, specifically a Pide Shop, as part of the CSE 344 System Programming course. The project involves creating a multi-threaded internet server that handles order placement, preparation, and delivery in a concurrent environment using various inter-process communication (IPC) mechanisms.

2. System Design and Components

Pide House

The Pide House contains:

- **Cooks**: Responsible for preparing the orders.
- **Oven**: Can hold up to 6 meals.
- **Manager**: Handles order assignments and deliveries.

Delivery Personnel

Delivery personnel wait outside the store and deliver prepared orders to customers. Each delivery person can carry up to 3 orders at a time.

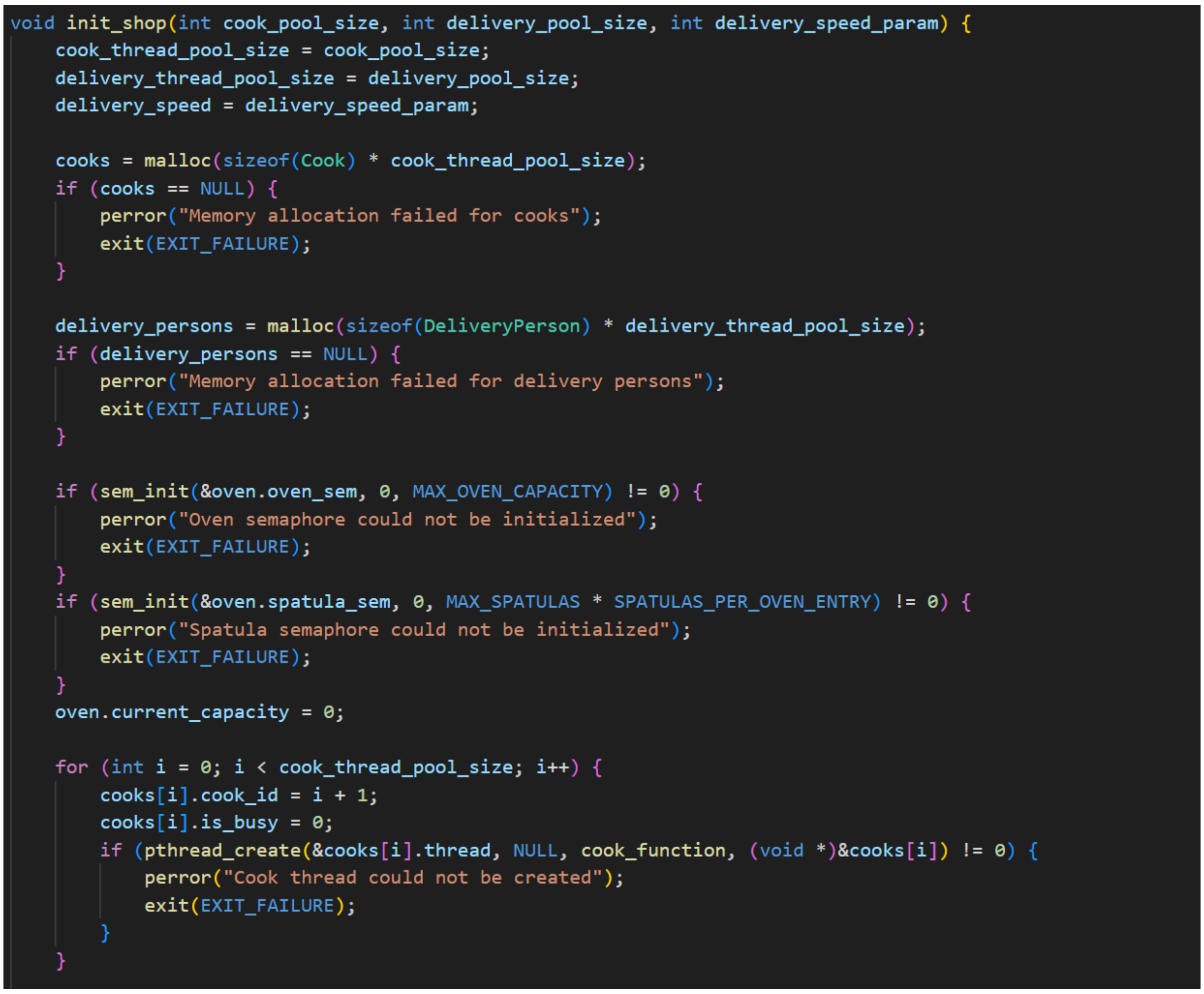
Customers

Customers place orders with their location details. The town is a p×qp \times qp×q km rectangle with the Pide Shop in the center.

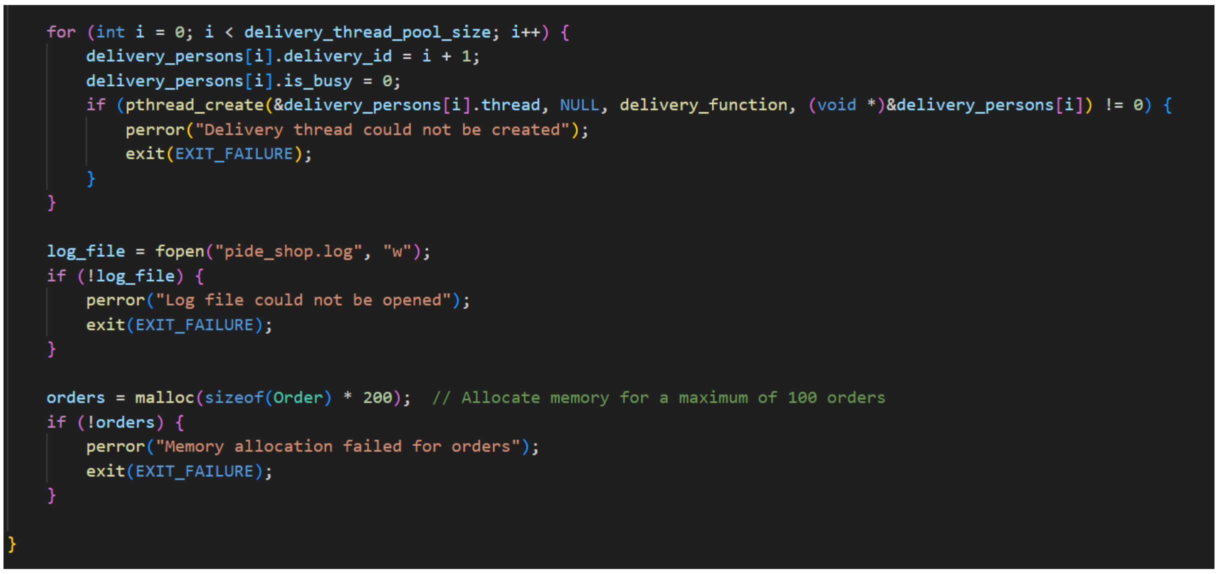
3. Implementation Details

Server Initialization

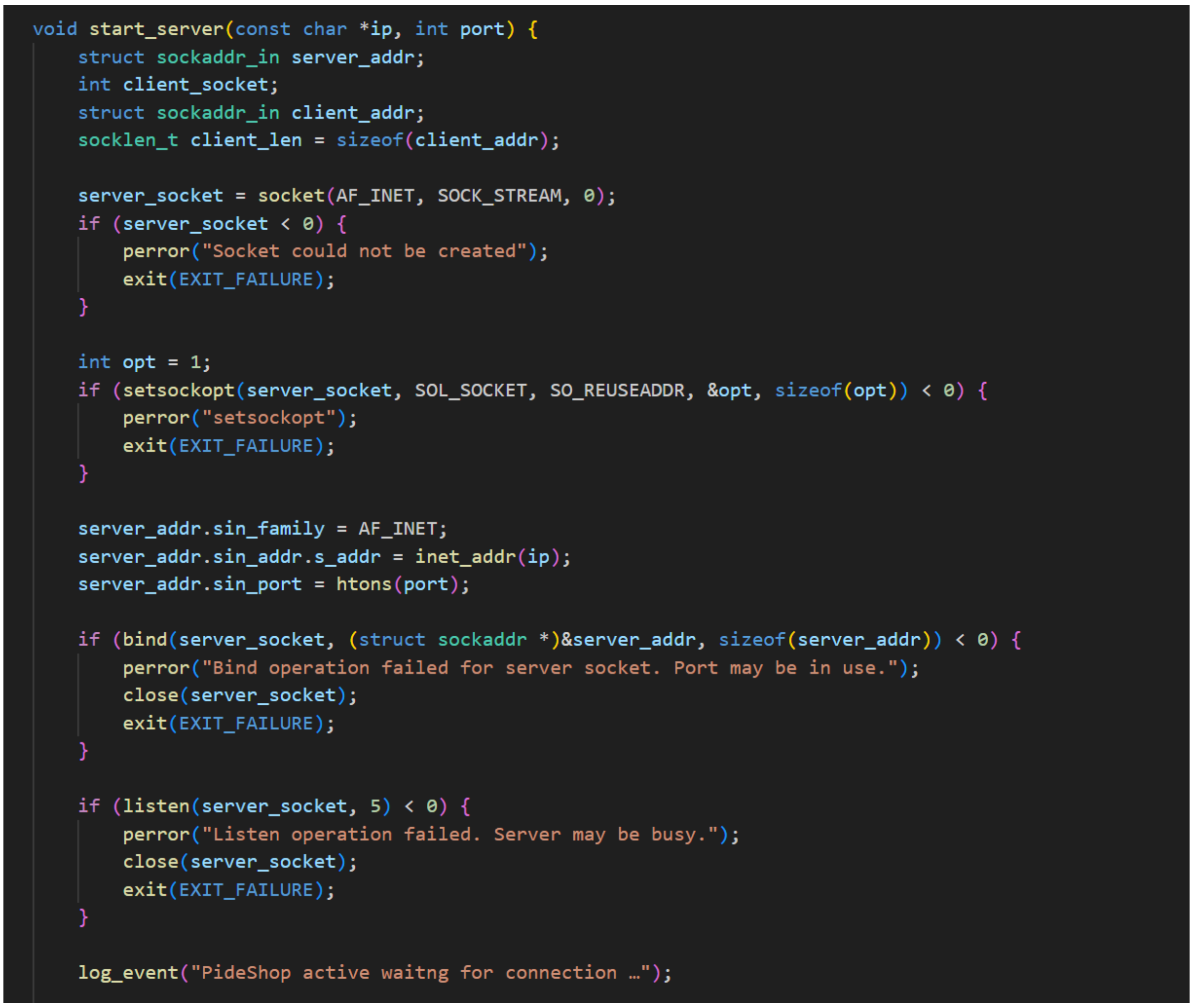
The server is initialized with the specified number of cook and delivery threads. It binds to a given IP and port to listen for incoming client connections.



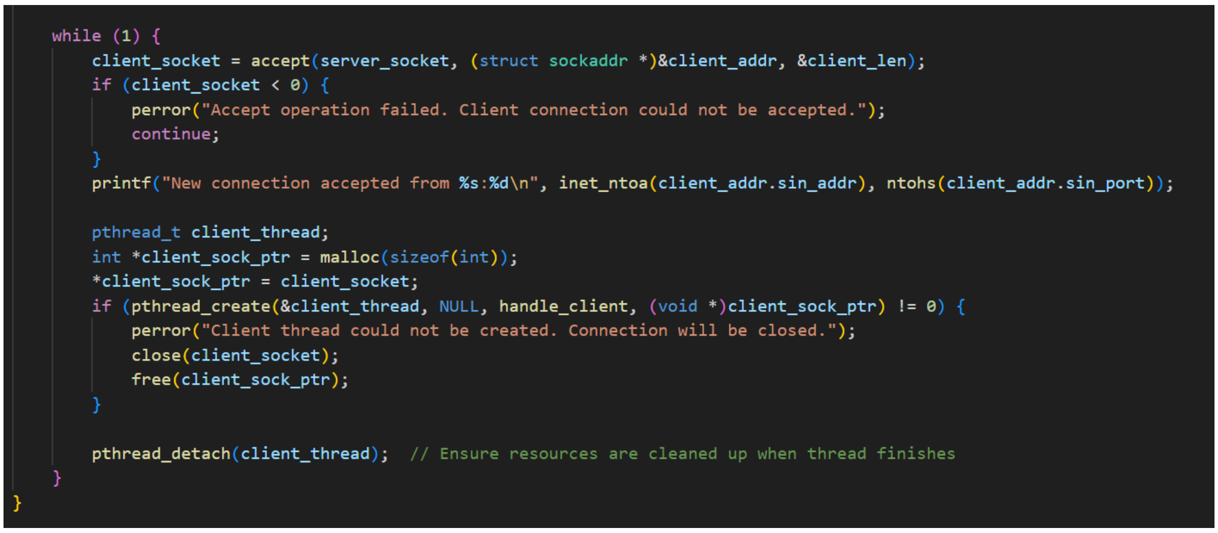
```c
void init_shop(int cook_pool_size, int delivery_pool_size, int delivery_speed_param) {
    cook_thread_pool_size = cook_pool_size;
    delivery_thread_pool_size = delivery_pool_size;
    delivery_speed = delivery_speed_param;

    cooks = malloc(sizeof(Cook) * cook_thread_pool_size);
    if (cooks == NULL) {
        perror("Memory allocation failed for cooks");
        exit(EXIT_FAILURE);
    }

    delivery_persons = malloc(sizeof(DeliveryPerson) * delivery_thread_pool_size);
    if (delivery_persons == NULL) {
        perror("Memory allocation failed for delivery persons");
        exit(EXIT_FAILURE);
    }

    if (sem_init(&oven.oven_sem, 0, MAX_OVEN_CAPACITY) != 0) {
        perror("Oven semaphore could not be initialized");
        exit(EXIT_FAILURE);
    }
    if (sem_init(&oven.spatula_sem, 0, MAX_SPATULAS * SPATULAS_PER_OVEN_ENTRY) != 0) {
        perror("Spatula semaphore could not be initialized");
        exit(EXIT_FAILURE);
    }
    oven.current_capacity = 0;

    for (int i = 0; i < cook_thread_pool_size; i++) {
        cooks[i].cook_id = i + 1;
        cooks[i].is_busy = 0;
        if (pthread_create(&cooks[i].thread, NULL, cook_function, (void *)&cooks[i]) != 0) {
            perror("Cook thread could not be created");
            exit(EXIT_FAILURE);
        }
    }
```



```c
    for (int i = 0; i < delivery_thread_pool_size; i++) {
        delivery_persons[i].delivery_id = i + 1;
        delivery_persons[i].is_busy = 0;
        if (pthread_create(&delivery_persons[i].thread, NULL, delivery_function, (void *)&delivery_persons[i]) != 0) {
            perror("Delivery thread could not be created");
            exit(EXIT_FAILURE);
        }
    }

    log_file = fopen("pide_shop.log", "w");
    if (!log_file) {
        perror("Log file could not be opened");
        exit(EXIT_FAILURE);
    }

    orders = malloc(sizeof(Order) * 200);  // Allocate memory for a maximum of 100 orders
    if (!orders) {
        perror("Memory allocation failed for orders");
        exit(EXIT_FAILURE);
    }

}
```



```c
void start_server(const char *ip, int port) {
    struct sockaddr_in server_addr;
    int client_socket;
    struct sockaddr_in client_addr;
    socklen_t client_len = sizeof(client_addr);

    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    if (server_socket < 0) {
        perror("Socket could not be created");
        exit(EXIT_FAILURE);
    }

    int opt = 1;
    if (setsockopt(server_socket, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt)) < 0) {
        perror("setsockopt");
        exit(EXIT_FAILURE);
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(ip);
    server_addr.sin_port = htons(port);

    if (bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("Bind operation failed for server socket. Port may be in use.");
        close(server_socket);
        exit(EXIT_FAILURE);
    }

    if (listen(server_socket, 5) < 0) {
        perror("Listen operation failed. Server may be busy.");
        close(server_socket);
        exit(EXIT_FAILURE);
    }

    log_event("PideShop active waitng for connection …");
```



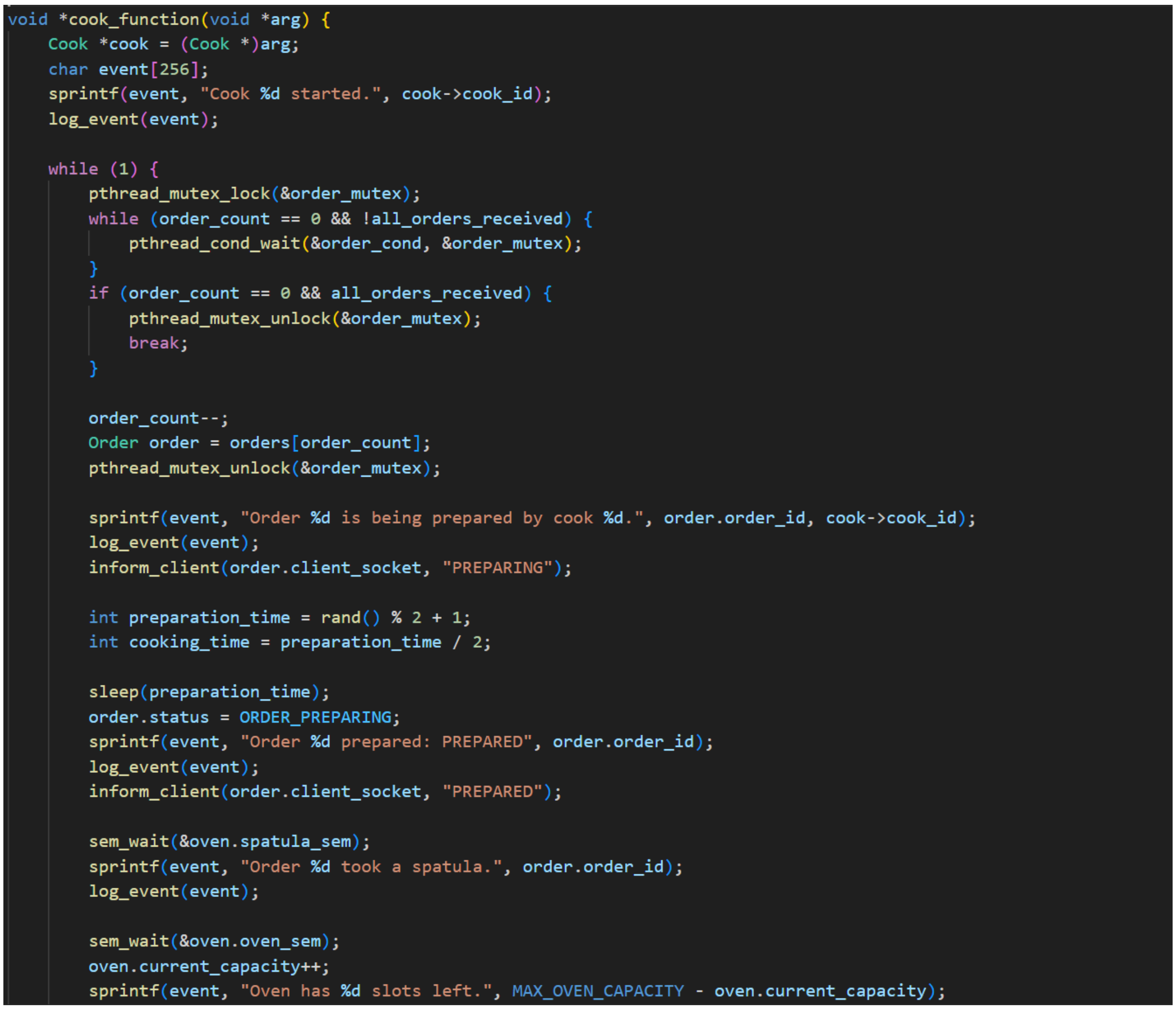
```c
    while (1) {
        client_socket = accept(server_socket, (struct sockaddr *)&client_addr, &client_len);
        if (client_socket < 0) {
            perror("Accept operation failed. Client connection could not be accepted.");
            continue;
        }
        printf("New connection accepted from %s:%d\n", inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));

        pthread_t client_thread;
        int *client_sock_ptr = malloc(sizeof(int));
        *client_sock_ptr = client_socket;
        if (pthread_create(&client_thread, NULL, handle_client, (void *)client_sock_ptr) != 0) {
            perror("Client thread could not be created. Connection will be closed.");
            close(client_socket);
            free(client_sock_ptr);
        }

        pthread_detach(client_thread);  // Ensure resources are cleaned up when thread finishes
    }
}
```

Order Processing

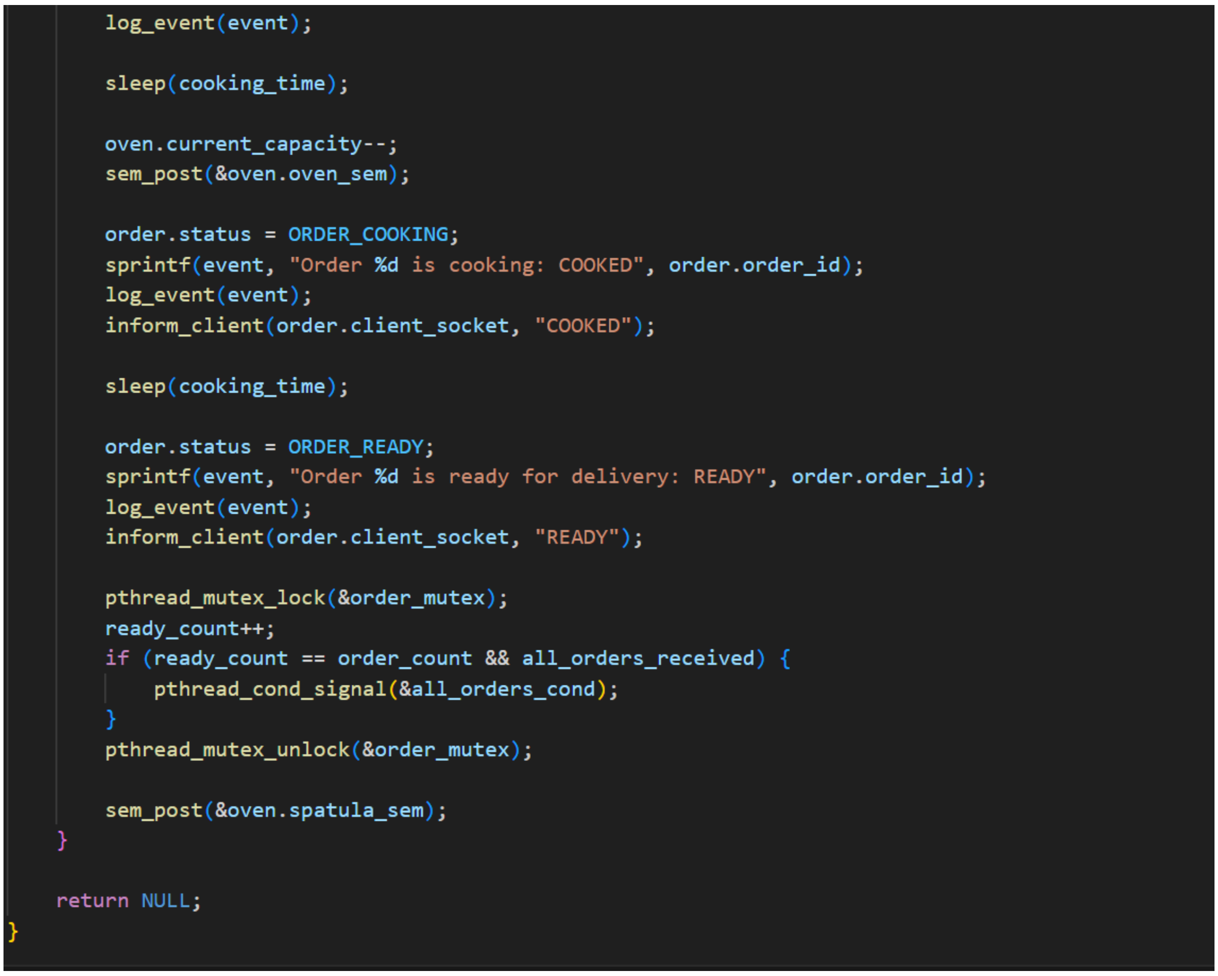
Orders are received from clients and processed in a FIFO manner. Each order goes through various stages: placed, preparing, cooked, ready for delivery, and delivered.

Cook Functionality

Cooks prepare the orders, manage oven usage, and update order status.



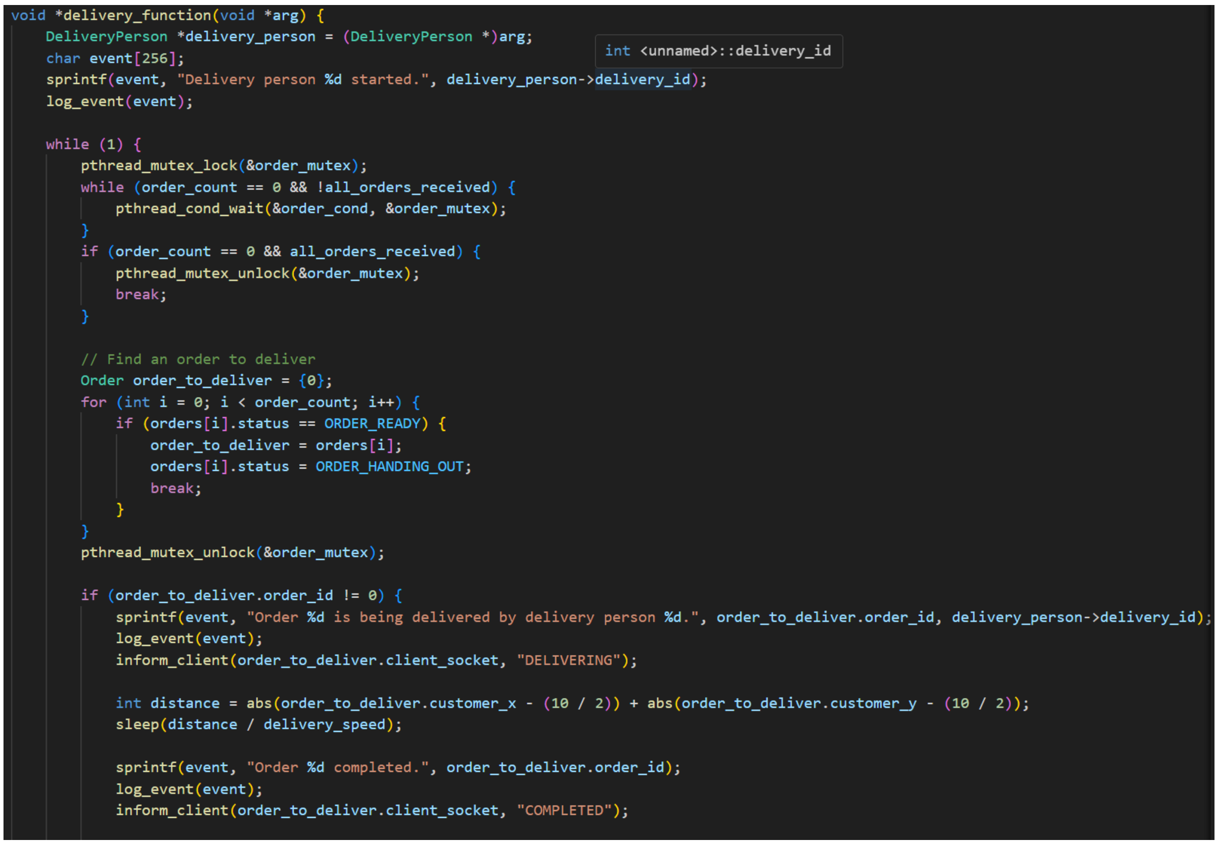
```c
void *cook_function(void *arg) {
    Cook *cook = (Cook *)arg;
    char event[256];
    sprintf(event, "Cook %d started.", cook->cook_id);
    log_event(event);

    while (1) {
        pthread_mutex_lock(&order_mutex);
        while (order_count == 0 && !all_orders_received) {
            pthread_cond_wait(&order_cond, &order_mutex);
        }
        if (order_count == 0 && all_orders_received) {
            pthread_mutex_unlock(&order_mutex);
            break;
        }

        order_count--;
        Order order = orders[order_count];
        pthread_mutex_unlock(&order_mutex);

        sprintf(event, "Order %d is being prepared by cook %d.", order.order_id, cook->cook_id);
        log_event(event);
        inform_client(order.client_socket, "PREPARING");

        int preparation_time = rand() % 2 + 1;
        int cooking_time = preparation_time / 2;

        sleep(preparation_time);
        order.status = ORDER_PREPARING;
        sprintf(event, "Order %d prepared: PREPARED", order.order_id);
        log_event(event);
        inform_client(order.client_socket, "PREPARED");

        sem_wait(&oven.spatula_sem);
        sprintf(event, "Order %d took a spatula.", order.order_id);
        log_event(event);

        sem_wait(&oven.oven_sem);
        oven.current_capacity++;
        sprintf(event, "Oven has %d slots left.", MAX_OVEN_CAPACITY - oven.current_capacity);
```



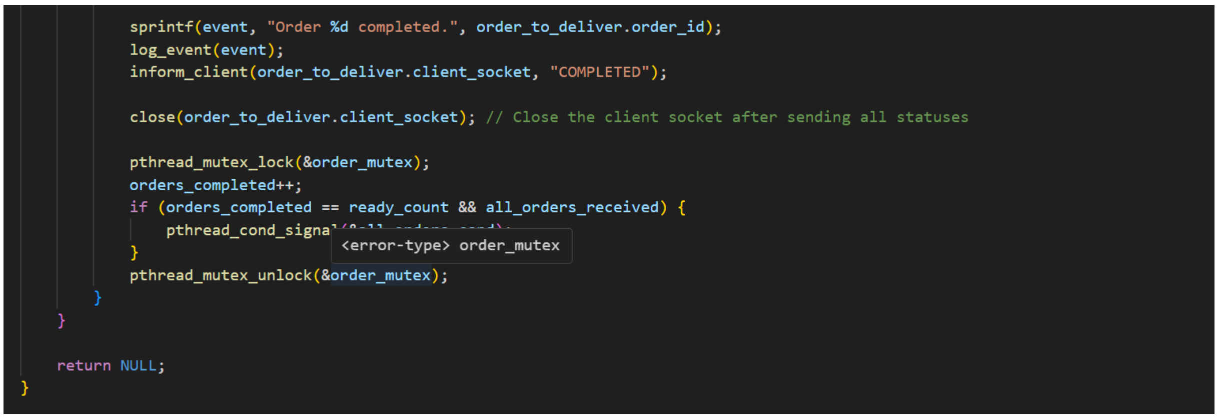
```
        log_event(event);

        sleep(cooking_time);

        oven.current_capacity--;
        sem_post(&oven.oven_sem);

        order.status = ORDER_COOKING;
        sprintf(event, "Order %d is cooking: COOKED", order.order_id);
        log_event(event);
        inform_client(order.client_socket, "COOKED");

        sleep(cooking_time);

        order.status = ORDER_READY;
        sprintf(event, "Order %d is ready for delivery: READY", order.order_id);
        log_event(event);
        inform_client(order.client_socket, "READY");

        pthread_mutex_lock(&order_mutex);
        ready_count++;
        if (ready_count == order_count && all_orders_received) {
            pthread_cond_signal(&all_orders_cond);
        }
        pthread_mutex_unlock(&order_mutex);

        sem_post(&oven.spatula_sem);
    }

    return NULL;
}
```

Delivery Functionality

Delivery personnel pick up ready orders and deliver them to customers based on their location.



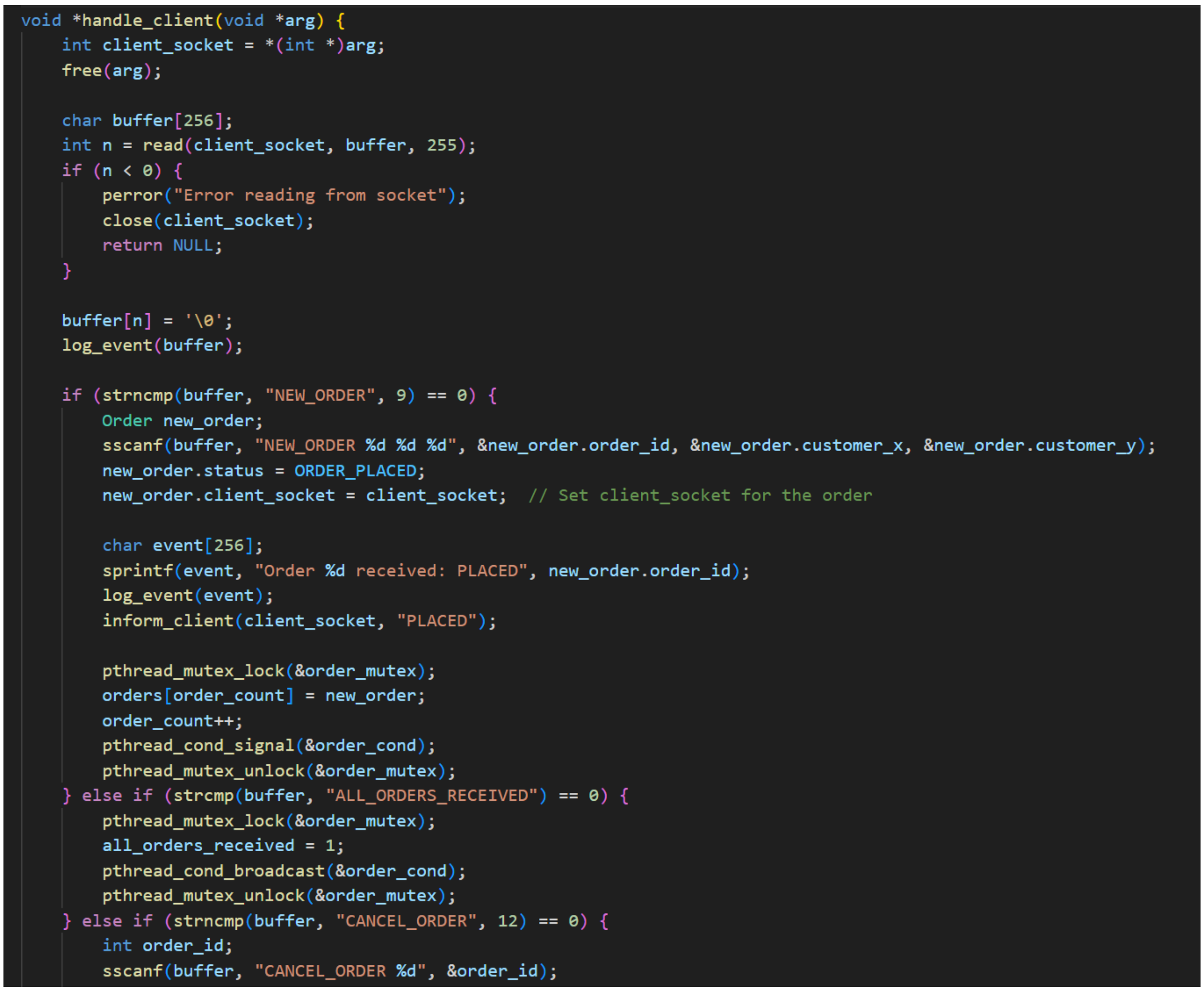
```c
void *delivery_function(void *arg) {
    DeliveryPerson *delivery_person = (DeliveryPerson *)arg;
    char event[256];
    sprintf(event, "Delivery person %d started.", delivery_person->delivery_id);
    log_event(event);

    while (1) {
        pthread_mutex_lock(&order_mutex);
        while (order_count == 0 && !all_orders_received) {
            pthread_cond_wait(&order_cond, &order_mutex);
        }
        if (order_count == 0 && all_orders_received) {
            pthread_mutex_unlock(&order_mutex);
            break;
        }

        // Find an order to deliver
        Order order_to_deliver = {0};
        for (int i = 0; i < order_count; i++) {
            if (orders[i].status == ORDER_READY) {
                order_to_deliver = orders[i];
                orders[i].status = ORDER_HANDING_OUT;
                break;
            }
        }
        pthread_mutex_unlock(&order_mutex);

        if (order_to_deliver.order_id != 0) {
            sprintf(event, "Order %d is being delivered by delivery person %d.", order_to_deliver.order_id, delivery_person->delivery_id);
            log_event(event);
            inform_client(order_to_deliver.client_socket, "DELIVERING");

            int distance = abs(order_to_deliver.customer_x - (10 / 2)) + abs(order_to_deliver.customer_y - (10 / 2));
            sleep(distance / delivery_speed);

            sprintf(event, "Order %d completed.", order_to_deliver.order_id);
            log_event(event);
            inform_client(order_to_deliver.client_socket, "COMPLETED");
```



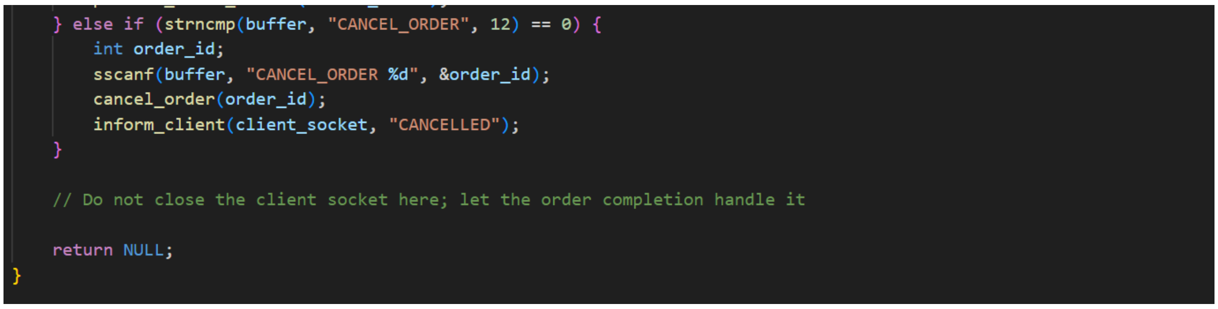
```c
            sprintf(event, "Order %d completed.", order_to_deliver.order_id);
            log_event(event);
            inform_client(order_to_deliver.client_socket, "COMPLETED");

            close(order_to_deliver.client_socket); // Close the client socket after sending all statuses

            pthread_mutex_lock(&order_mutex);
            orders_completed++;
            if (orders_completed == ready_count && all_orders_received) {
                pthread_cond_signal(&all_orders_cond);
            }
            pthread_mutex_unlock(&order_mutex);
        }
    }

    return NULL;
}
```

Client Handling

Clients connect to the server to place orders. Each client runs in a separate thread to handle communication and status updates.



```c
void *handle_client(void *arg) {
    int client_socket = *(int *)arg;
    free(arg);

    char buffer[256];
    int n = read(client_socket, buffer, 255);
    if (n < 0) {
        perror("Error reading from socket");
        close(client_socket);
        return NULL;
    }

    buffer[n] = '\0';
    log_event(buffer);

    if (strncmp(buffer, "NEW_ORDER", 9) == 0) {
        Order new_order;
        sscanf(buffer, "NEW_ORDER %d %d %d", &new_order.order_id, &new_order.customer_x, &new_order.customer_y);
        new_order.status = ORDER_PLACED;
        new_order.client_socket = client_socket;  // Set client_socket for the order

        char event[256];
        sprintf(event, "Order %d received: PLACED", new_order.order_id);
        log_event(event);
        inform_client(client_socket, "PLACED");

        pthread_mutex_lock(&order_mutex);
        orders[order_count] = new_order;
        order_count++;
        pthread_cond_signal(&order_cond);
        pthread_mutex_unlock(&order_mutex);
    } else if (strcmp(buffer, "ALL_ORDERS_RECEIVED") == 0) {
        pthread_mutex_lock(&order_mutex);
        all_orders_received = 1;
        pthread_cond_broadcast(&order_cond);
        pthread_mutex_unlock(&order_mutex);
    } else if (strncmp(buffer, "CANCEL_ORDER", 12) == 0) {
        int order_id;
        sscanf(buffer, "CANCEL_ORDER %d", &order_id);
```



```c
    } else if (strncmp(buffer, "CANCEL_ORDER", 12) == 0) {
        int order_id;
        sscanf(buffer, "CANCEL_ORDER %d", &order_id);
        cancel_order(order_id);
        inform_client(client_socket, "CANCELLED");
    }

    // Do not close the client socket here; let the order completion handle it

    return NULL;
}
```

4. IPC Mechanisms Used

Mutexes

Mutexes are used to synchronize access to shared resources such as order lists.

```
pthread_mutex_t order_mutex = PTHREAD_MUTEX_INITIALIZER;
```

Condition Variables

Condition variables are used to signal state changes, such as when a new order is placed or all orders are completed.

```
pthread_cond_t order_cond = PTHREAD_COND_INITIALIZER;
pthread_cond_t all_orders_cond = PTHREAD_COND_INITIALIZER;
```

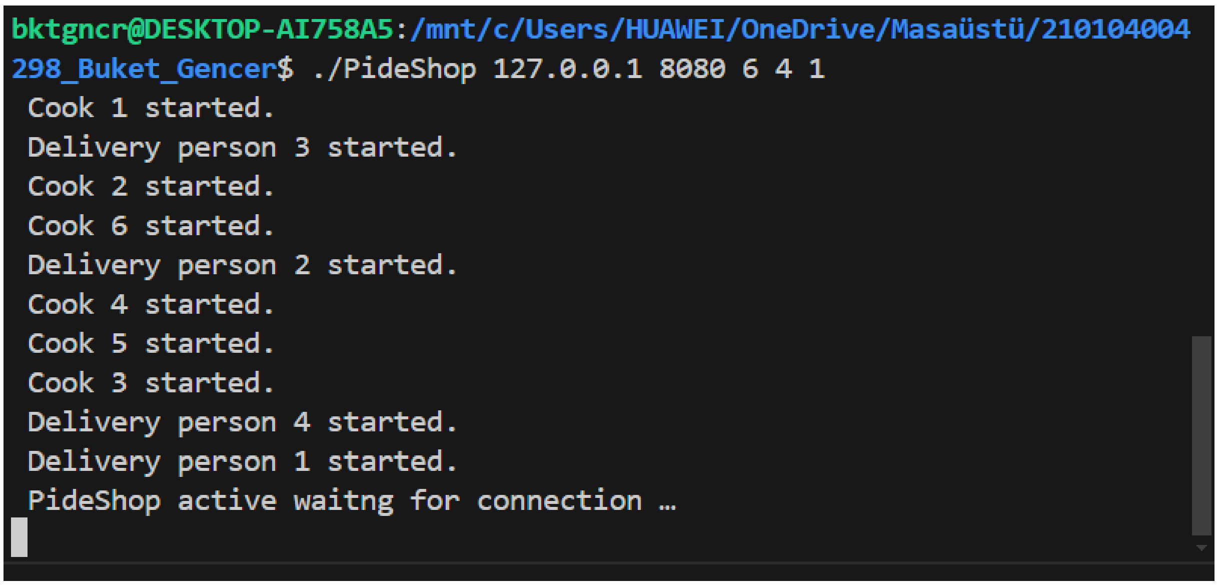
Semaphores

Semaphores manage the capacity of the oven and the availability of spatulas.

```
sem_t oven_sem;
sem_t spatula_sem;
```

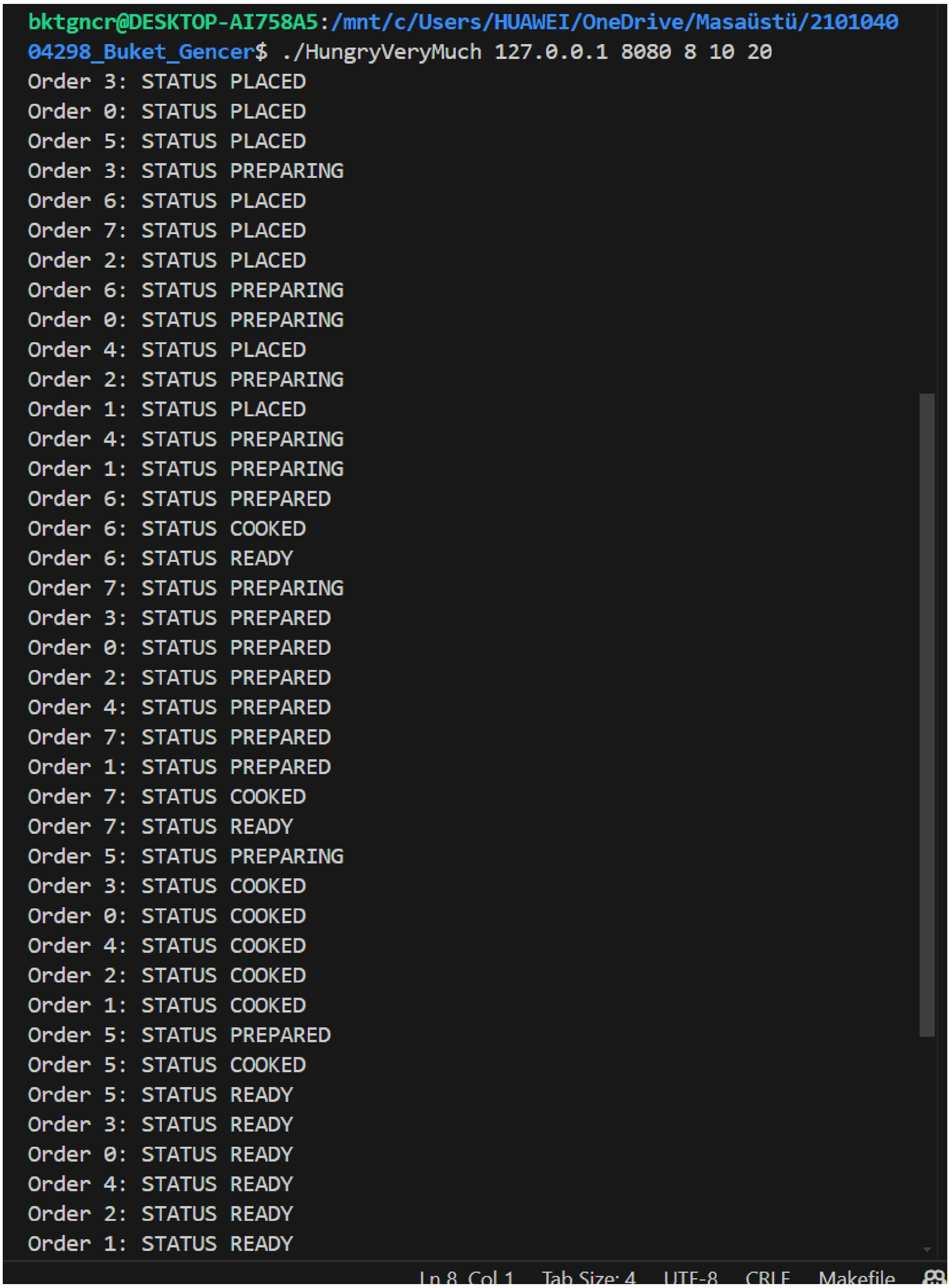
5. Test Cases and Results

a. Testing basic functionality with a small number of clients.

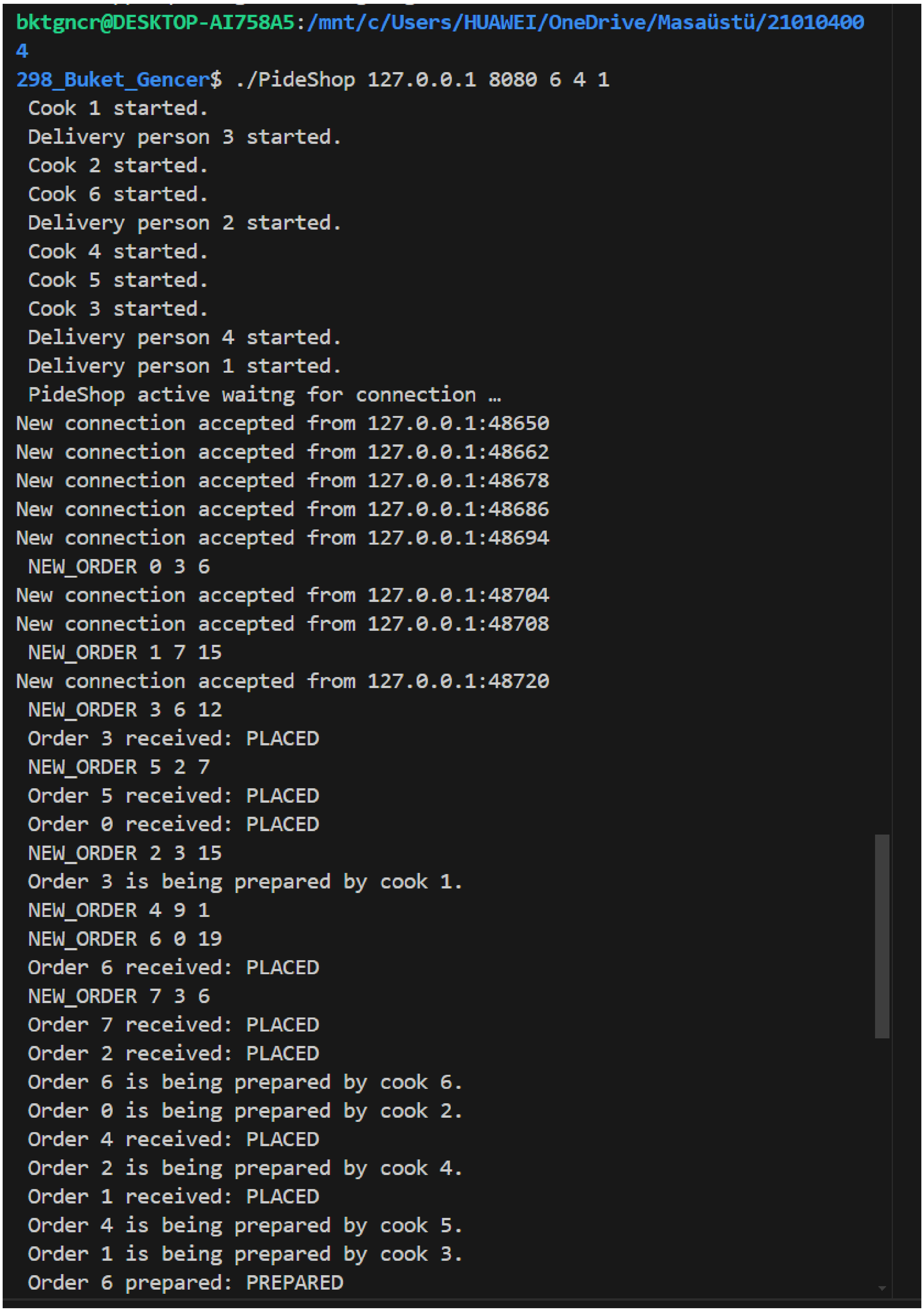


```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/210104004
298_Buket_Gencer$ ./PideShop 127.0.0.1 8080 6 4 1
 Cook 1 started.
 Delivery person 3 started.
 Cook 2 started.
 Cook 6 started.
 Delivery person 2 started.
 Cook 4 started.
 Cook 5 started.
 Cook 3 started.
 Delivery person 4 started.
 Delivery person 1 started.
 PideShop active waitng for connection …
```

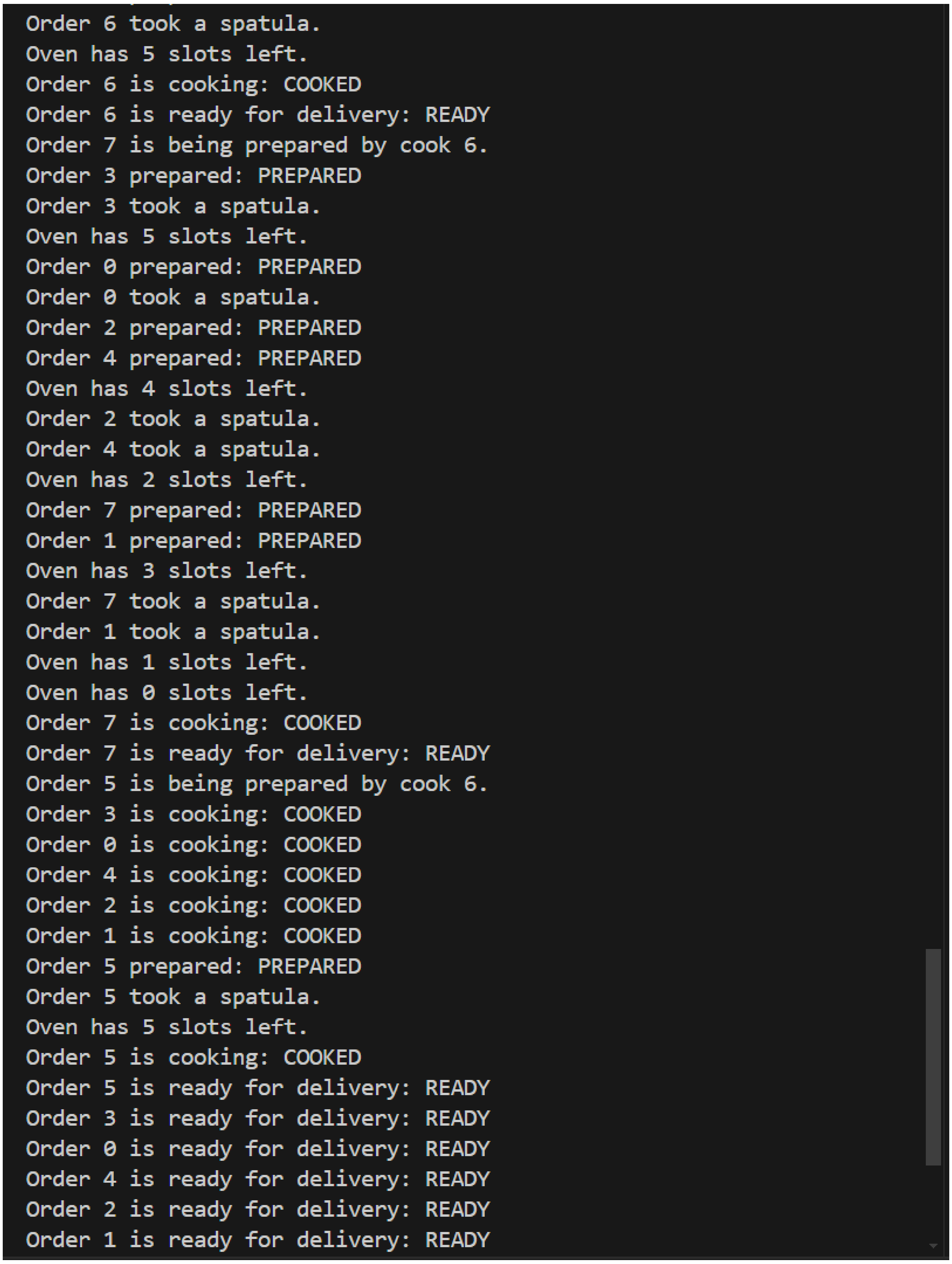
All orders is deliverd after starting client side:



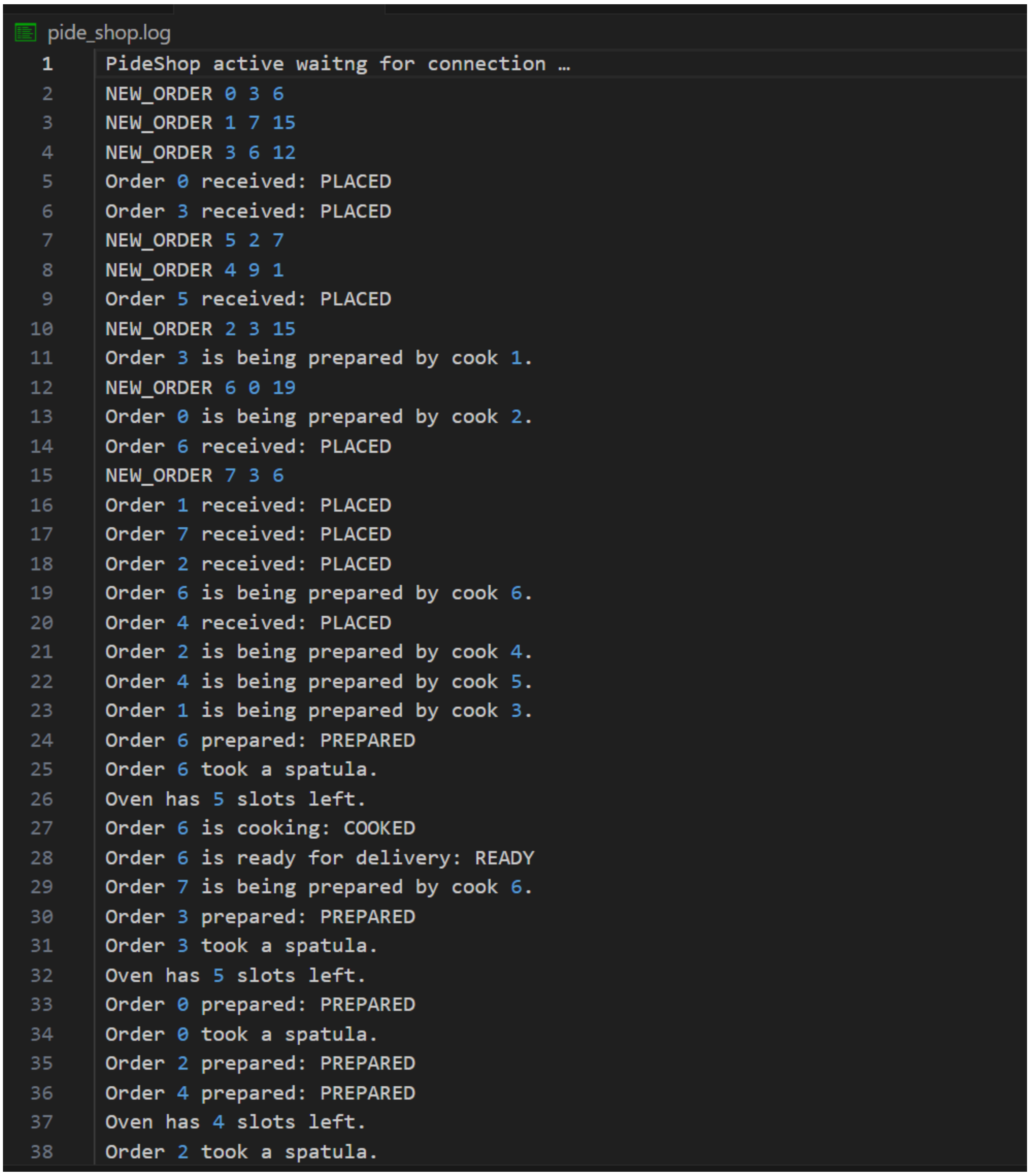
```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 8 10 20
Order 3: STATUS PLACED
Order 0: STATUS PLACED
Order 5: STATUS PLACED
Order 3: STATUS PREPARING
Order 6: STATUS PLACED
Order 7: STATUS PLACED
Order 2: STATUS PLACED
Order 6: STATUS PREPARING
Order 0: STATUS PREPARING
Order 4: STATUS PLACED
Order 2: STATUS PREPARING
Order 1: STATUS PLACED
Order 4: STATUS PREPARING
Order 1: STATUS PREPARING
Order 6: STATUS PREPARED
Order 6: STATUS COOKED
Order 6: STATUS READY
Order 7: STATUS PREPARING
Order 3: STATUS PREPARED
Order 0: STATUS PREPARED
Order 2: STATUS PREPARED
Order 4: STATUS PREPARED
Order 7: STATUS PREPARED
Order 1: STATUS PREPARED
Order 7: STATUS COOKED
Order 7: STATUS READY
Order 5: STATUS PREPARING
Order 3: STATUS COOKED
Order 0: STATUS COOKED
Order 4: STATUS COOKED
Order 2: STATUS COOKED
Order 1: STATUS COOKED
Order 5: STATUS PREPARED
Order 5: STATUS COOKED
Order 5: STATUS READY
Order 3: STATUS READY
Order 0: STATUS READY
Order 4: STATUS READY
Order 2: STATUS READY
Order 1: STATUS READY
```



```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4
298_Buket_Gencer$ ./PideShop 127.0.0.1 8080 6 4 1
 Cook 1 started.
 Delivery person 3 started.
 Cook 2 started.
 Cook 6 started.
 Delivery person 2 started.
 Cook 4 started.
 Cook 5 started.
 Cook 3 started.
 Delivery person 4 started.
 Delivery person 1 started.
 PideShop active waitng for connection …
New connection accepted from 127.0.0.1:48650
New connection accepted from 127.0.0.1:48662
New connection accepted from 127.0.0.1:48678
New connection accepted from 127.0.0.1:48686
New connection accepted from 127.0.0.1:48694
 NEW_ORDER 0 3 6
New connection accepted from 127.0.0.1:48704
New connection accepted from 127.0.0.1:48708
 NEW_ORDER 1 7 15
New connection accepted from 127.0.0.1:48720
 NEW_ORDER 3 6 12
 Order 3 received: PLACED
 NEW_ORDER 5 2 7
 Order 5 received: PLACED
 Order 0 received: PLACED
 NEW_ORDER 2 3 15
 Order 3 is being prepared by cook 1.
 NEW_ORDER 4 9 1
 NEW_ORDER 6 0 19
 Order 6 received: PLACED
 NEW_ORDER 7 3 6
 Order 7 received: PLACED
 Order 2 received: PLACED
 Order 6 is being prepared by cook 6.
 Order 0 is being prepared by cook 2.
 Order 4 received: PLACED
 Order 2 is being prepared by cook 4.
 Order 1 received: PLACED
 Order 4 is being prepared by cook 5.
 Order 1 is being prepared by cook 3.
 Order 6 prepared: PREPARED
```

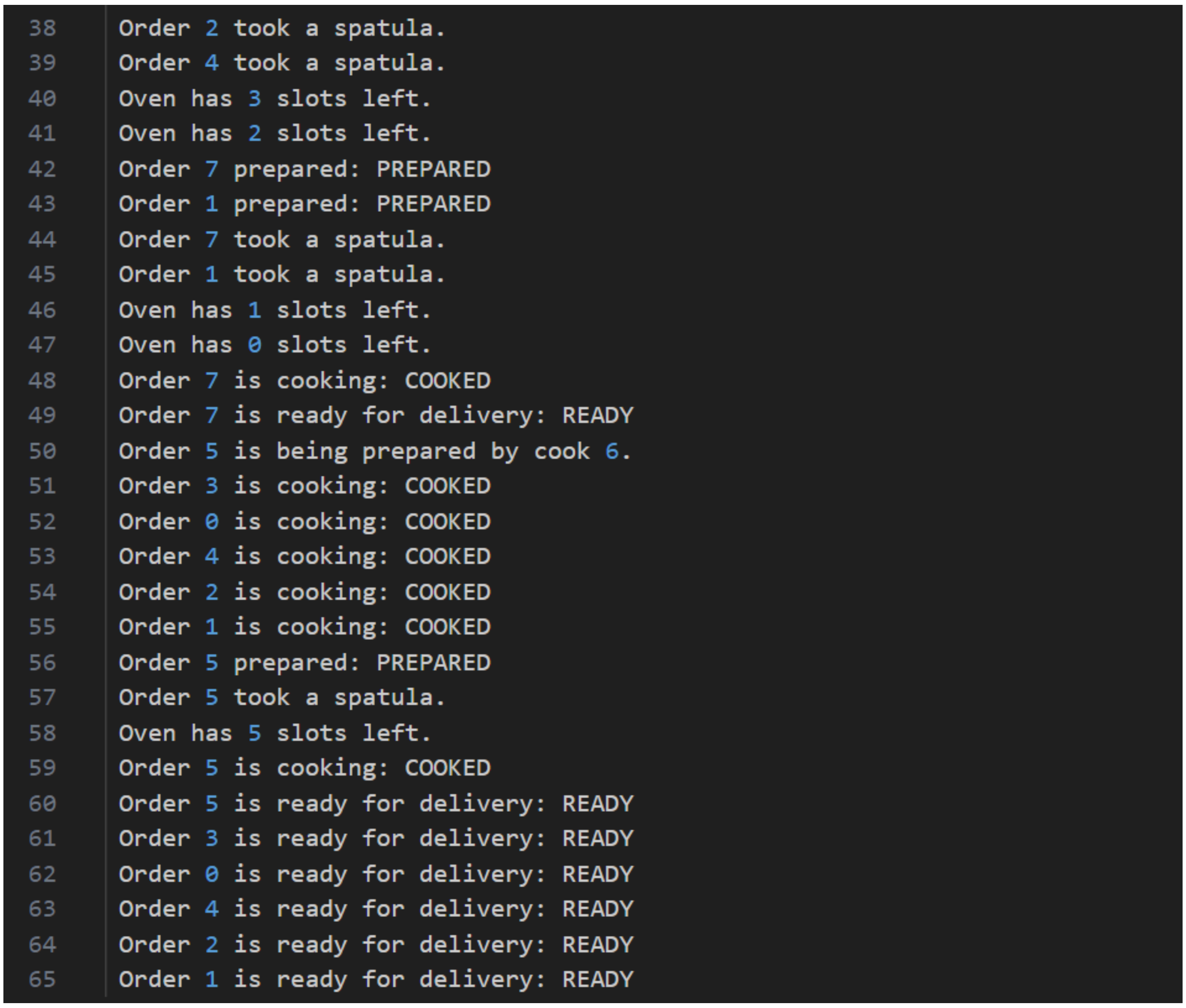

```
Order 6 took a spatula.
Oven has 5 slots left.
Order 6 is cooking: COOKED
Order 6 is ready for delivery: READY
Order 7 is being prepared by cook 6.
Order 3 prepared: PREPARED
Order 3 took a spatula.
Oven has 5 slots left.
Order 0 prepared: PREPARED
Order 0 took a spatula.
Order 2 prepared: PREPARED
Order 4 prepared: PREPARED
Oven has 4 slots left.
Order 2 took a spatula.
Order 4 took a spatula.
Oven has 2 slots left.
Order 7 prepared: PREPARED
Order 1 prepared: PREPARED
Oven has 3 slots left.
Order 7 took a spatula.
Order 1 took a spatula.
Oven has 1 slots left.
Oven has 0 slots left.
Order 7 is cooking: COOKED
Order 7 is ready for delivery: READY
Order 5 is being prepared by cook 6.
Order 3 is cooking: COOKED
Order 0 is cooking: COOKED
Order 4 is cooking: COOKED
Order 2 is cooking: COOKED
Order 1 is cooking: COOKED
Order 5 prepared: PREPARED
Order 5 took a spatula.
Oven has 5 slots left.
Order 5 is cooking: COOKED
Order 5 is ready for delivery: READY
Order 3 is ready for delivery: READY
Order 0 is ready for delivery: READY
Order 4 is ready for delivery: READY
Order 2 is ready for delivery: READY
Order 1 is ready for delivery: READY
```

LOG FİLE:



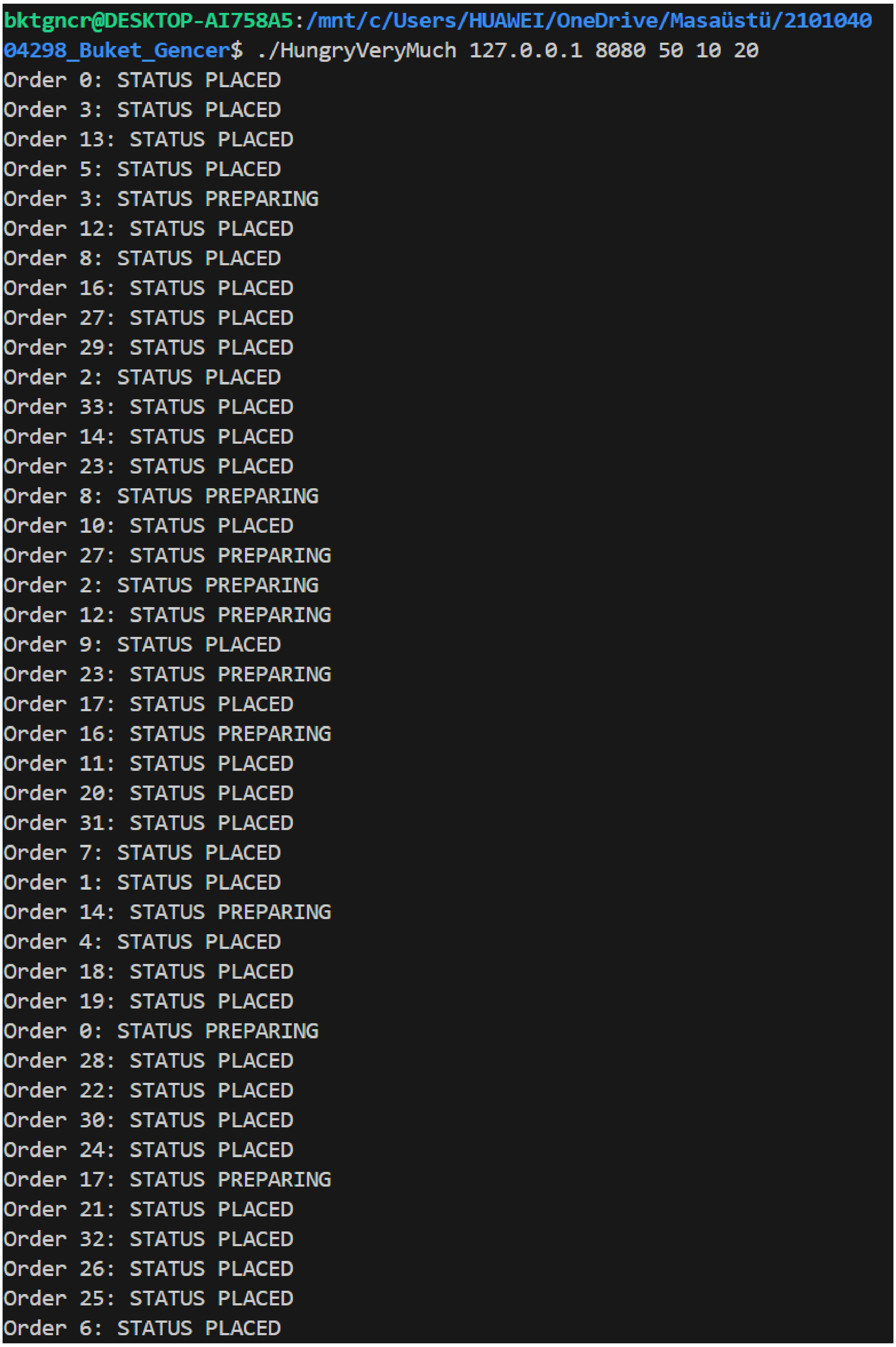
pide_shop.log

```
PideShop active waitng for connection …
NEW_ORDER 0 3 6
NEW_ORDER 1 7 15
NEW_ORDER 3 6 12
Order 0 received: PLACED
Order 3 received: PLACED
NEW_ORDER 5 2 7
NEW_ORDER 4 9 1
Order 5 received: PLACED
NEW_ORDER 2 3 15
Order 3 is being prepared by cook 1.
NEW_ORDER 6 0 19
Order 0 is being prepared by cook 2.
Order 6 received: PLACED
NEW_ORDER 7 3 6
Order 1 received: PLACED
Order 7 received: PLACED
Order 2 received: PLACED
Order 6 is being prepared by cook 6.
Order 4 received: PLACED
Order 2 is being prepared by cook 4.
Order 4 is being prepared by cook 5.
Order 1 is being prepared by cook 3.
Order 6 prepared: PREPARED
Order 6 took a spatula.
Oven has 5 slots left.
Order 6 is cooking: COOKED
Order 6 is ready for delivery: READY
Order 7 is being prepared by cook 6.
Order 3 prepared: PREPARED
Order 3 took a spatula.
Oven has 5 slots left.
Order 0 prepared: PREPARED
Order 0 took a spatula.
Order 2 prepared: PREPARED
Order 4 prepared: PREPARED
Oven has 4 slots left.
Order 2 took a spatula.
```

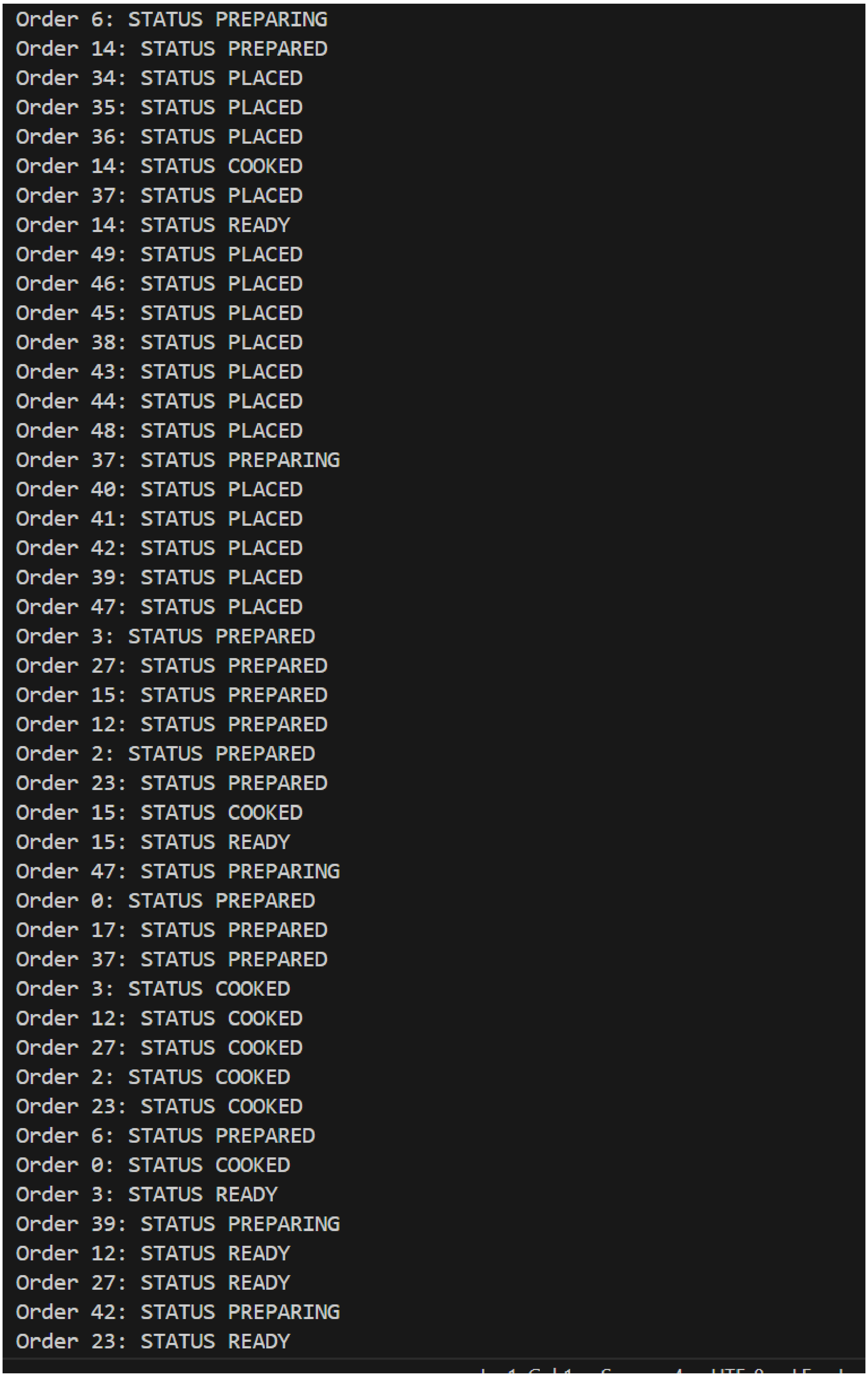


```
Order 2 took a spatula.
Order 4 took a spatula.
Oven has 3 slots left.
Oven has 2 slots left.
Order 7 prepared: PREPARED
Order 1 prepared: PREPARED
Order 7 took a spatula.
Order 1 took a spatula.
Oven has 1 slots left.
Oven has 0 slots left.
Order 7 is cooking: COOKED
Order 7 is ready for delivery: READY
Order 5 is being prepared by cook 6.
Order 3 is cooking: COOKED
Order 0 is cooking: COOKED
Order 4 is cooking: COOKED
Order 2 is cooking: COOKED
Order 1 is cooking: COOKED
Order 5 prepared: PREPARED
Order 5 took a spatula.
Oven has 5 slots left.
Order 5 is cooking: COOKED
Order 5 is ready for delivery: READY
Order 3 is ready for delivery: READY
Order 0 is ready for delivery: READY
Order 4 is ready for delivery: READY
Order 2 is ready for delivery: READY
Order 1 is ready for delivery: READY
```

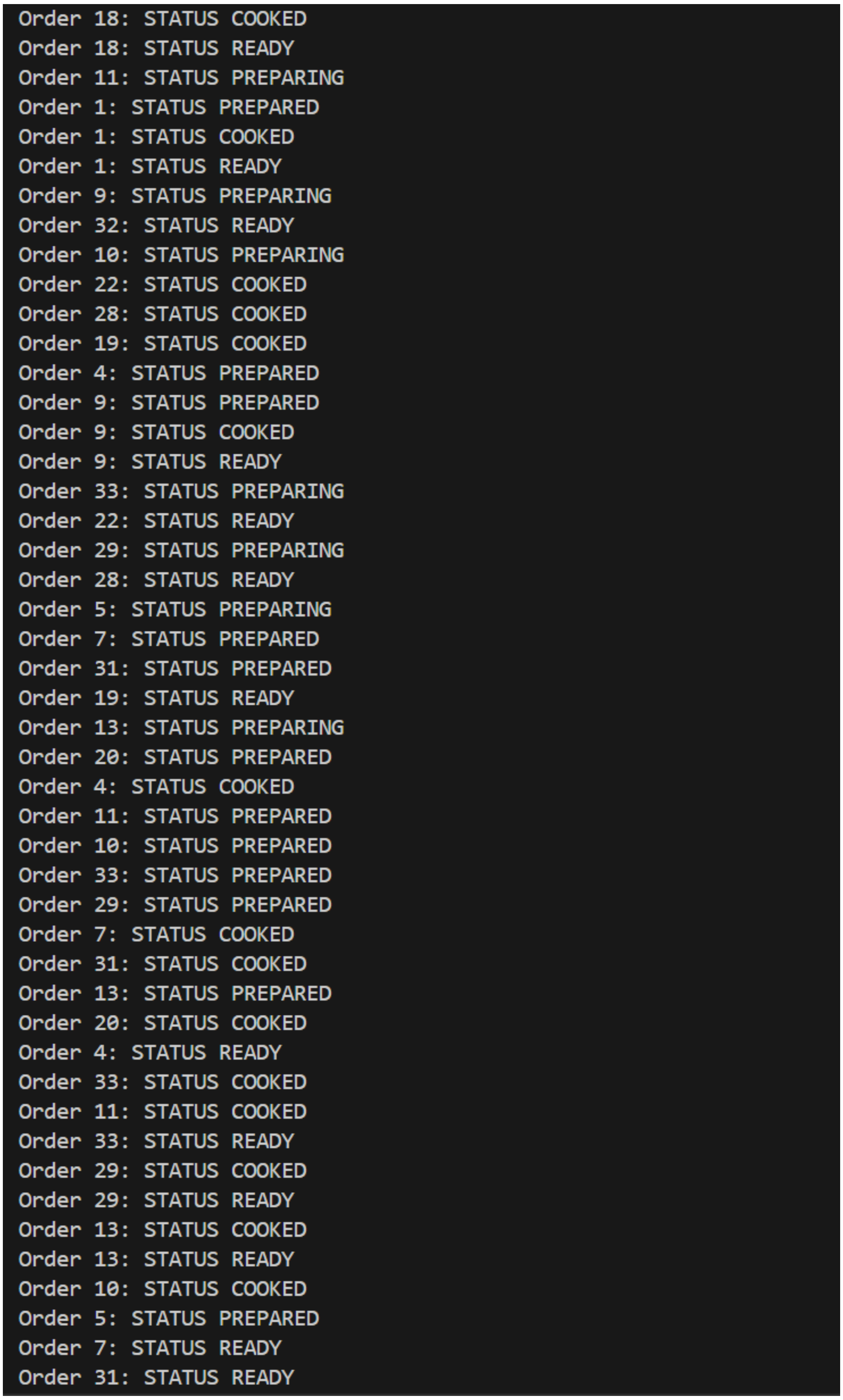
b. Test with 50 , 100 , 200 client:



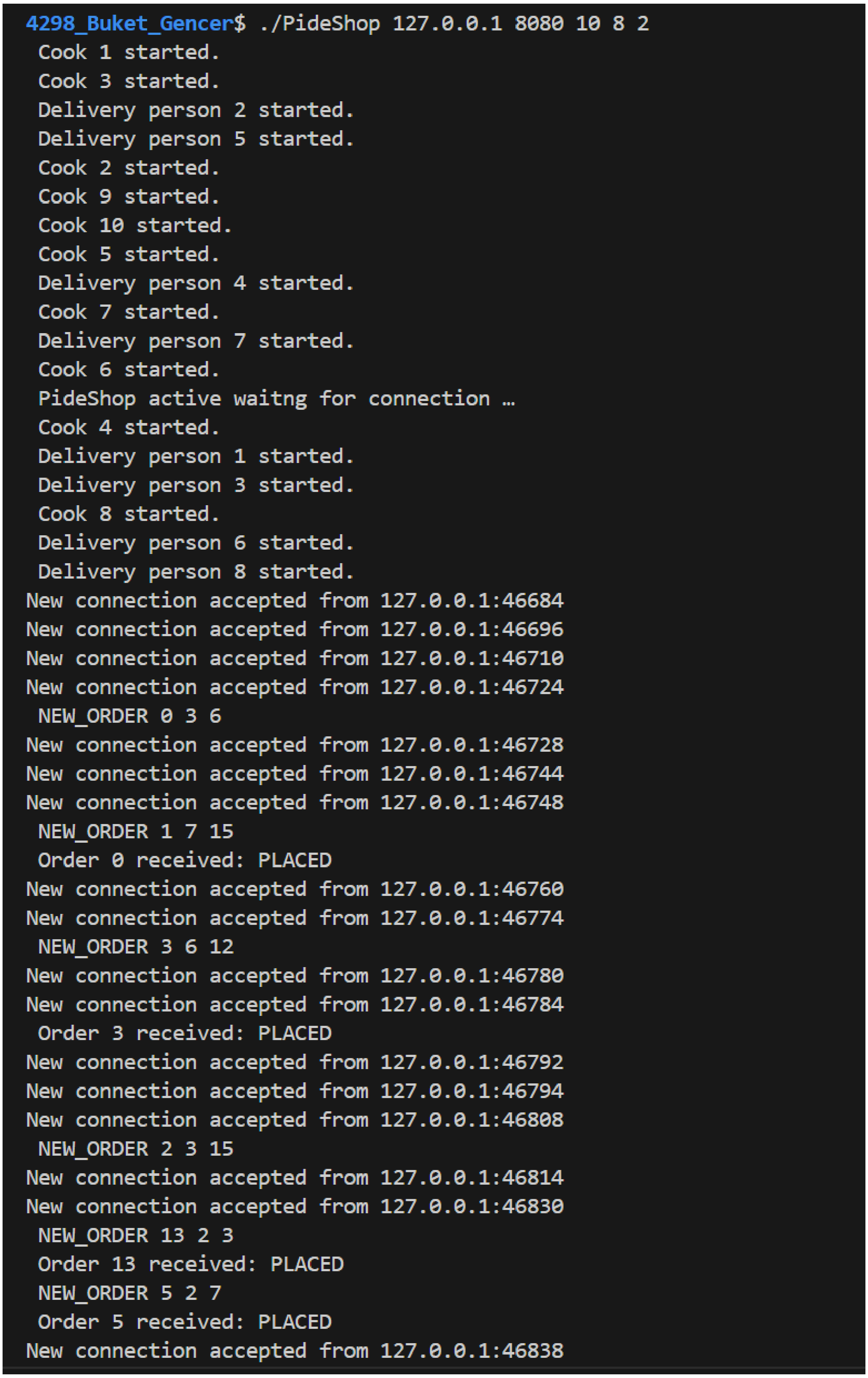
```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 50 10 20
Order 0: STATUS PLACED
Order 3: STATUS PLACED
Order 13: STATUS PLACED
Order 5: STATUS PLACED
Order 3: STATUS PREPARING
Order 12: STATUS PLACED
Order 8: STATUS PLACED
Order 16: STATUS PLACED
Order 27: STATUS PLACED
Order 29: STATUS PLACED
Order 2: STATUS PLACED
Order 33: STATUS PLACED
Order 14: STATUS PLACED
Order 23: STATUS PLACED
Order 8: STATUS PREPARING
Order 10: STATUS PLACED
Order 27: STATUS PREPARING
Order 2: STATUS PREPARING
Order 12: STATUS PREPARING
Order 9: STATUS PLACED
Order 23: STATUS PREPARING
Order 17: STATUS PLACED
Order 16: STATUS PREPARING
Order 11: STATUS PLACED
Order 20: STATUS PLACED
Order 31: STATUS PLACED
Order 7: STATUS PLACED
Order 1: STATUS PLACED
Order 14: STATUS PREPARING
Order 4: STATUS PLACED
Order 18: STATUS PLACED
Order 19: STATUS PLACED
Order 0: STATUS PREPARING
Order 28: STATUS PLACED
Order 22: STATUS PLACED
Order 30: STATUS PLACED
Order 24: STATUS PLACED
Order 17: STATUS PREPARING
Order 21: STATUS PLACED
Order 32: STATUS PLACED
Order 26: STATUS PLACED
Order 25: STATUS PLACED
Order 6: STATUS PLACED
```



```
Order 6: STATUS PREPARING
Order 14: STATUS PREPARED
Order 34: STATUS PLACED
Order 35: STATUS PLACED
Order 36: STATUS PLACED
Order 14: STATUS COOKED
Order 37: STATUS PLACED
Order 14: STATUS READY
Order 49: STATUS PLACED
Order 46: STATUS PLACED
Order 45: STATUS PLACED
Order 38: STATUS PLACED
Order 43: STATUS PLACED
Order 44: STATUS PLACED
Order 48: STATUS PLACED
Order 37: STATUS PREPARING
Order 40: STATUS PLACED
Order 41: STATUS PLACED
Order 42: STATUS PLACED
Order 39: STATUS PLACED
Order 47: STATUS PLACED
Order 3: STATUS PREPARED
Order 27: STATUS PREPARED
Order 15: STATUS PREPARED
Order 12: STATUS PREPARED
Order 2: STATUS PREPARED
Order 23: STATUS PREPARED
Order 15: STATUS COOKED
Order 15: STATUS READY
Order 47: STATUS PREPARING
Order 0: STATUS PREPARED
Order 17: STATUS PREPARED
Order 37: STATUS PREPARED
Order 3: STATUS COOKED
Order 12: STATUS COOKED
Order 27: STATUS COOKED
Order 2: STATUS COOKED
Order 23: STATUS COOKED
Order 6: STATUS PREPARED
Order 0: STATUS COOKED
Order 3: STATUS READY
Order 39: STATUS PREPARING
Order 12: STATUS READY
Order 27: STATUS READY
Order 42: STATUS PREPARING
Order 23: STATUS READY
```

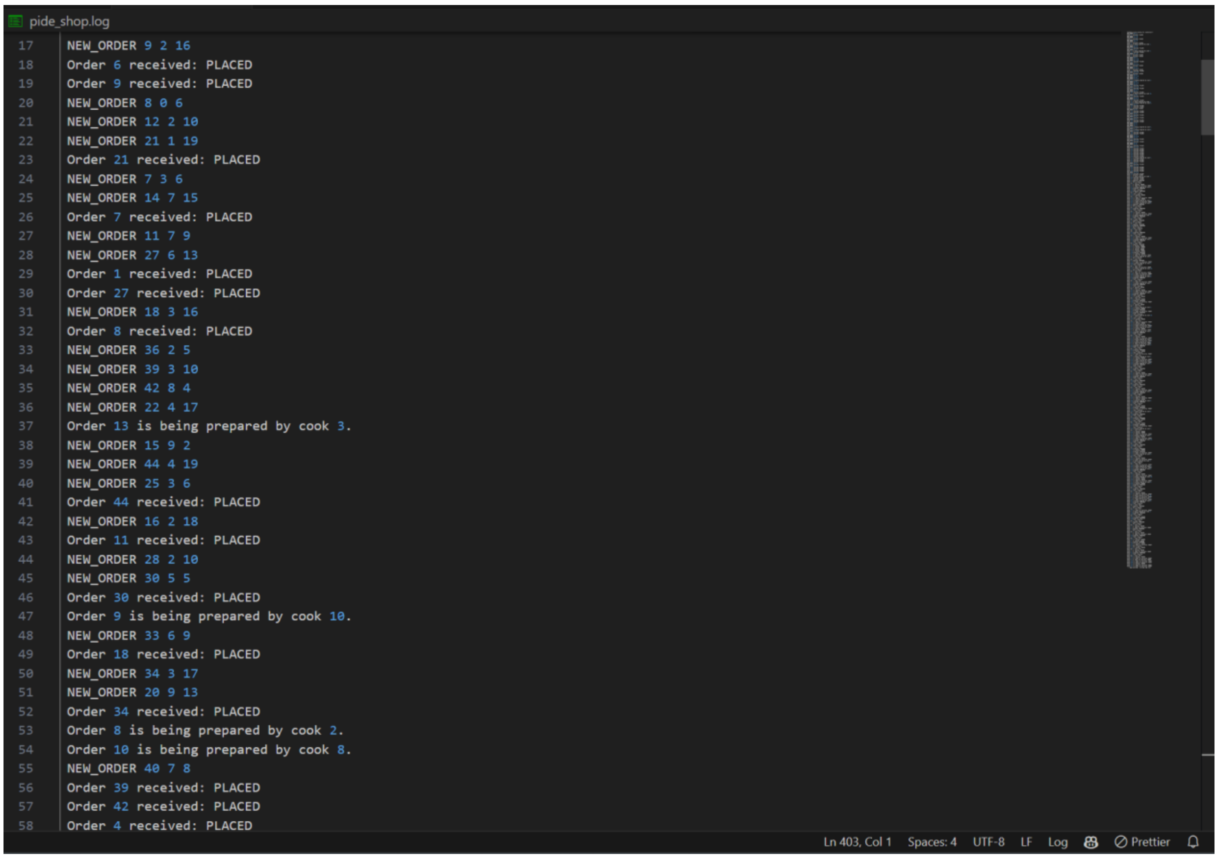

```
Order 18: STATUS COOKED
Order 18: STATUS READY
Order 11: STATUS PREPARING
Order 1: STATUS PREPARED
Order 1: STATUS COOKED
Order 1: STATUS READY
Order 9: STATUS PREPARING
Order 32: STATUS READY
Order 10: STATUS PREPARING
Order 22: STATUS COOKED
Order 28: STATUS COOKED
Order 19: STATUS COOKED
Order 4: STATUS PREPARED
Order 9: STATUS PREPARED
Order 9: STATUS COOKED
Order 9: STATUS READY
Order 33: STATUS PREPARING
Order 22: STATUS READY
Order 29: STATUS PREPARING
Order 28: STATUS READY
Order 5: STATUS PREPARING
Order 7: STATUS PREPARED
Order 31: STATUS PREPARED
Order 19: STATUS READY
Order 13: STATUS PREPARING
Order 20: STATUS PREPARED
Order 4: STATUS COOKED
Order 11: STATUS PREPARED
Order 10: STATUS PREPARED
Order 33: STATUS PREPARED
Order 29: STATUS PREPARED
Order 7: STATUS COOKED
Order 31: STATUS COOKED
Order 13: STATUS PREPARED
Order 20: STATUS COOKED
Order 4: STATUS READY
Order 33: STATUS COOKED
Order 11: STATUS COOKED
Order 33: STATUS READY
Order 29: STATUS COOKED
Order 29: STATUS READY
Order 13: STATUS COOKED
Order 13: STATUS READY
Order 10: STATUS COOKED
Order 5: STATUS PREPARED
Order 7: STATUS READY
Order 31: STATUS READY
```

All orders are completed..



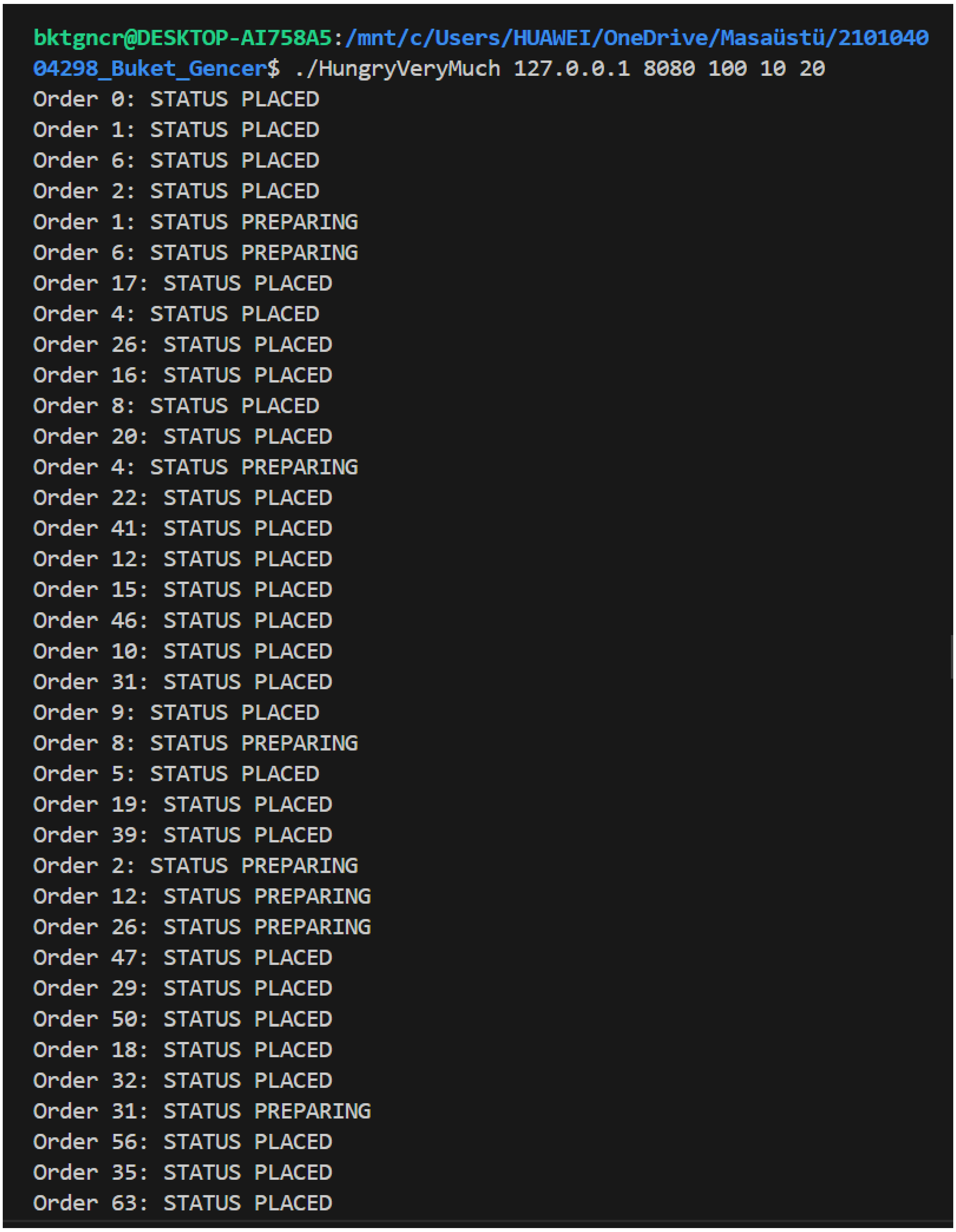
```
4298_Buket_Gencer$ ./PideShop 127.0.0.1 8080 10 8 2
 Cook 1 started.
 Cook 3 started.
 Delivery person 2 started.
 Delivery person 5 started.
 Cook 2 started.
 Cook 9 started.
 Cook 10 started.
 Cook 5 started.
 Delivery person 4 started.
 Cook 7 started.
 Delivery person 7 started.
 Cook 6 started.
 PideShop active waitng for connection …
 Cook 4 started.
 Delivery person 1 started.
 Delivery person 3 started.
 Cook 8 started.
 Delivery person 6 started.
 Delivery person 8 started.
New connection accepted from 127.0.0.1:46684
New connection accepted from 127.0.0.1:46696
New connection accepted from 127.0.0.1:46710
New connection accepted from 127.0.0.1:46724
 NEW_ORDER 0 3 6
New connection accepted from 127.0.0.1:46728
New connection accepted from 127.0.0.1:46744
New connection accepted from 127.0.0.1:46748
 NEW_ORDER 1 7 15
 Order 0 received: PLACED
New connection accepted from 127.0.0.1:46760
New connection accepted from 127.0.0.1:46774
 NEW_ORDER 3 6 12
New connection accepted from 127.0.0.1:46780
New connection accepted from 127.0.0.1:46784
 Order 3 received: PLACED
New connection accepted from 127.0.0.1:46792
New connection accepted from 127.0.0.1:46794
New connection accepted from 127.0.0.1:46808
 NEW_ORDER 2 3 15
New connection accepted from 127.0.0.1:46814
New connection accepted from 127.0.0.1:46830
 NEW_ORDER 13 2 3
 Order 13 received: PLACED
 NEW_ORDER 5 2 7
 Order 5 received: PLACED
New connection accepted from 127.0.0.1:46838
```

Some parts of log file:

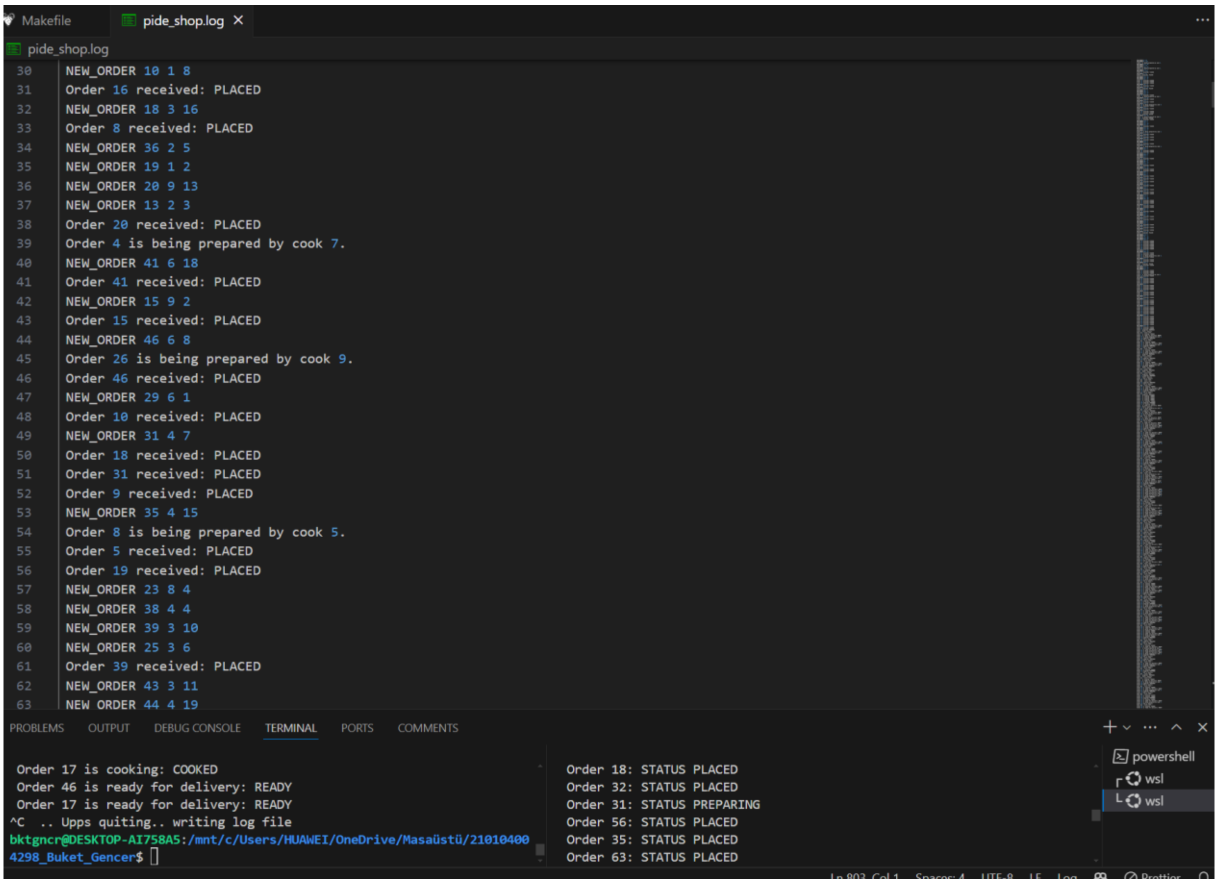


```
NEW_ORDER 9 2 16
Order 6 received: PLACED
Order 9 received: PLACED
NEW_ORDER 8 0 6
NEW_ORDER 12 2 10
NEW_ORDER 21 1 19
Order 21 received: PLACED
NEW_ORDER 7 3 6
NEW_ORDER 14 7 15
Order 7 received: PLACED
NEW_ORDER 11 7 9
NEW_ORDER 27 6 13
Order 1 received: PLACED
Order 27 received: PLACED
NEW_ORDER 18 3 16
Order 8 received: PLACED
NEW_ORDER 36 2 5
NEW_ORDER 39 3 10
NEW_ORDER 42 8 4
NEW_ORDER 22 4 17
Order 13 is being prepared by cook 3.
NEW_ORDER 15 9 2
NEW_ORDER 44 4 19
NEW_ORDER 25 3 6
Order 44 received: PLACED
NEW_ORDER 16 2 18
Order 11 received: PLACED
NEW_ORDER 28 2 10
NEW_ORDER 30 5 5
Order 30 received: PLACED
Order 9 is being prepared by cook 10.
NEW_ORDER 33 6 9
Order 18 received: PLACED
NEW_ORDER 34 3 17
NEW_ORDER 20 9 13
Order 34 received: PLACED
Order 8 is being prepared by cook 2.
Order 10 is being prepared by cook 8.
NEW_ORDER 40 7 8
Order 39 received: PLACED
Order 42 received: PLACED
Order 4 received: PLACED
```

With 100 and 200 customer:



```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 100 10 20
Order 0: STATUS PLACED
Order 1: STATUS PLACED
Order 6: STATUS PLACED
Order 2: STATUS PLACED
Order 1: STATUS PREPARING
Order 6: STATUS PREPARING
Order 17: STATUS PLACED
Order 4: STATUS PLACED
Order 26: STATUS PLACED
Order 16: STATUS PLACED
Order 8: STATUS PLACED
Order 20: STATUS PLACED
Order 4: STATUS PREPARING
Order 22: STATUS PLACED
Order 41: STATUS PLACED
Order 12: STATUS PLACED
Order 15: STATUS PLACED
Order 46: STATUS PLACED
Order 10: STATUS PLACED
Order 31: STATUS PLACED
Order 9: STATUS PLACED
Order 8: STATUS PREPARING
Order 5: STATUS PLACED
Order 19: STATUS PLACED
Order 39: STATUS PLACED
Order 2: STATUS PREPARING
Order 12: STATUS PREPARING
Order 26: STATUS PREPARING
Order 47: STATUS PLACED
Order 29: STATUS PLACED
Order 50: STATUS PLACED
Order 18: STATUS PLACED
Order 32: STATUS PLACED
Order 31: STATUS PREPARING
Order 56: STATUS PLACED
Order 35: STATUS PLACED
Order 63: STATUS PLACED
```

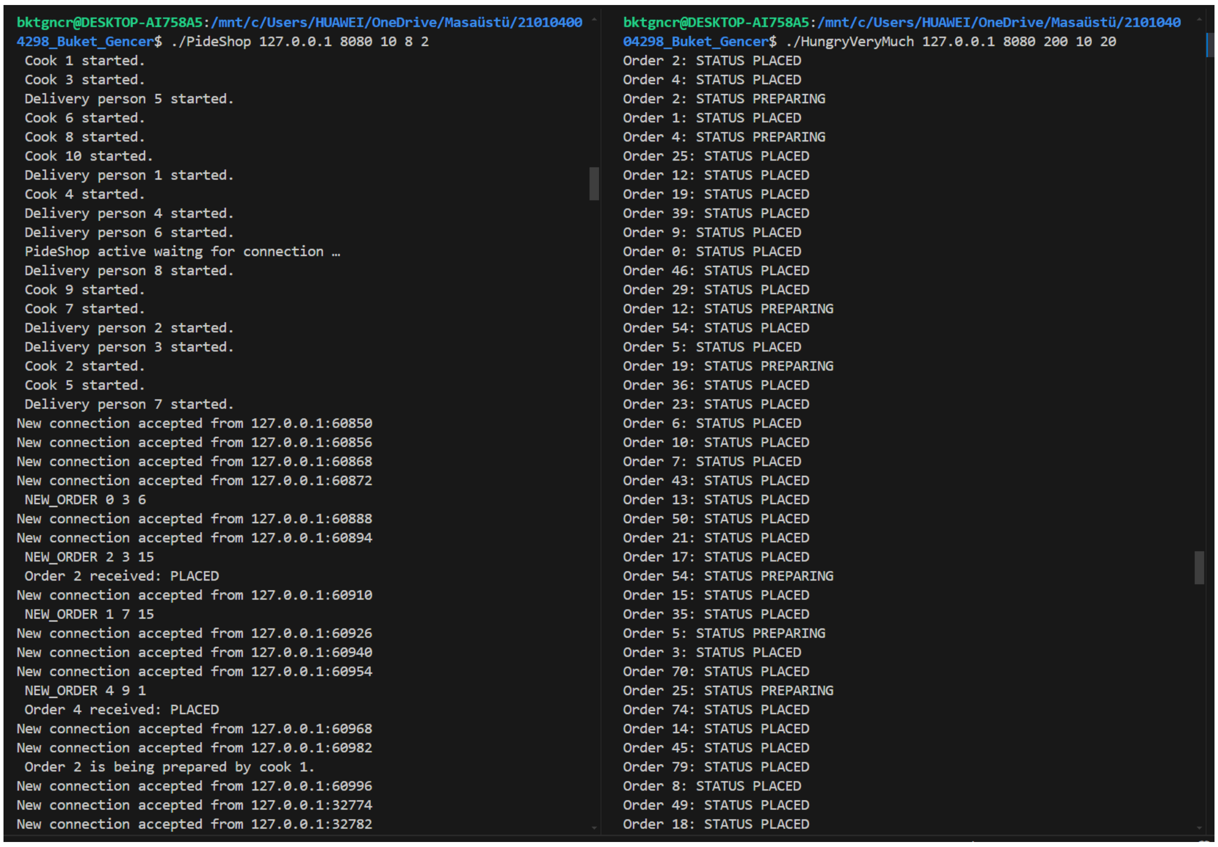


```
NEW_ORDER 10 1 8
Order 16 received: PLACED
NEW_ORDER 18 3 16
Order 8 received: PLACED
NEW_ORDER 36 2 5
NEW_ORDER 19 1 2
NEW_ORDER 20 9 13
NEW_ORDER 13 2 3
Order 20 received: PLACED
Order 4 is being prepared by cook 7.
NEW_ORDER 41 6 18
Order 41 received: PLACED
NEW_ORDER 15 9 2
Order 15 received: PLACED
NEW_ORDER 46 6 8
Order 26 is being prepared by cook 9.
Order 46 received: PLACED
NEW_ORDER 29 6 1
Order 10 received: PLACED
NEW_ORDER 31 4 7
Order 18 received: PLACED
Order 31 received: PLACED
Order 9 received: PLACED
NEW_ORDER 35 4 15
Order 8 is being prepared by cook 5.
Order 5 received: PLACED
Order 19 received: PLACED
NEW_ORDER 23 8 4
NEW_ORDER 38 4 4
NEW_ORDER 39 3 10
NEW_ORDER 25 3 6
Order 39 received: PLACED
NEW_ORDER 43 3 11
NEW_ORDER 44 4 19
```

```
 Order 17 is cooking: COOKED
 Order 46 is ready for delivery: READY
 Order 17 is ready for delivery: READY
^C  .. Upps quiting.. writing log file
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4298_Buket_Gencer$ 
```

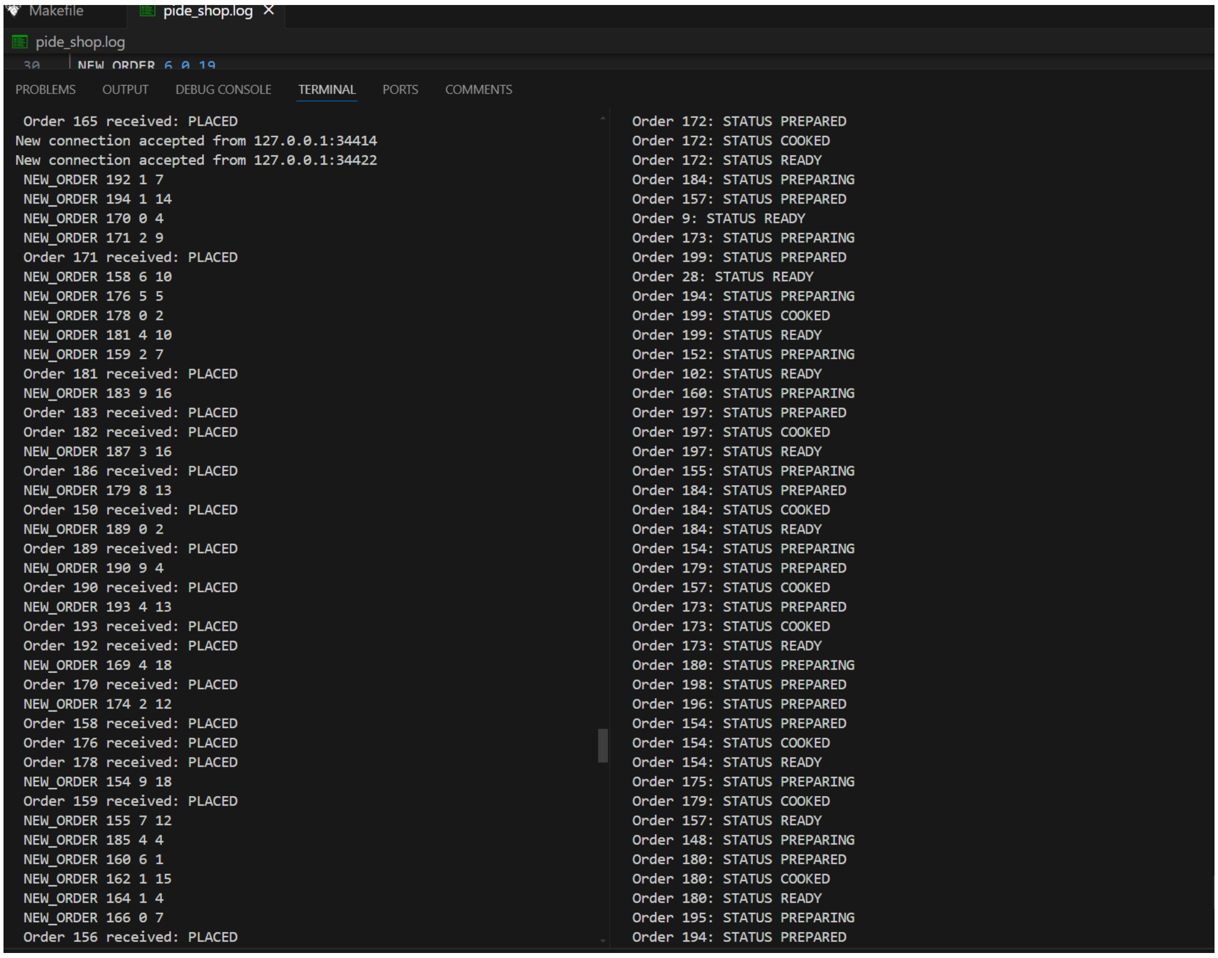
```
Order 18: STATUS PLACED
Order 32: STATUS PLACED
Order 31: STATUS PREPARING
Order 56: STATUS PLACED
Order 35: STATUS PLACED
Order 63: STATUS PLACED
```

ı added some part of output because output is so long...



```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4298_Buket_Gencer$ ./PideShop 127.0.0.1 8080 10 8 2
 Cook 1 started.
 Cook 3 started.
 Delivery person 5 started.
 Cook 6 started.
 Cook 8 started.
 Cook 10 started.
 Delivery person 1 started.
 Cook 4 started.
 Delivery person 4 started.
 Delivery person 6 started.
 PideShop active waitng for connection …
 Delivery person 8 started.
 Cook 9 started.
 Cook 7 started.
 Delivery person 2 started.
 Delivery person 3 started.
 Cook 2 started.
 Cook 5 started.
 Delivery person 7 started.
New connection accepted from 127.0.0.1:60850
New connection accepted from 127.0.0.1:60856
New connection accepted from 127.0.0.1:60868
New connection accepted from 127.0.0.1:60872
 NEW_ORDER 0 3 6
New connection accepted from 127.0.0.1:60888
New connection accepted from 127.0.0.1:60894
 NEW_ORDER 2 3 15
 Order 2 received: PLACED
New connection accepted from 127.0.0.1:60910
 NEW_ORDER 1 7 15
New connection accepted from 127.0.0.1:60926
New connection accepted from 127.0.0.1:60940
New connection accepted from 127.0.0.1:60954
 NEW_ORDER 4 9 1
 Order 4 received: PLACED
New connection accepted from 127.0.0.1:60968
New connection accepted from 127.0.0.1:60982
 Order 2 is being prepared by cook 1.
New connection accepted from 127.0.0.1:60996
New connection accepted from 127.0.0.1:32774
New connection accepted from 127.0.0.1:32782
```

```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 200 10 20
Order 2: STATUS PLACED
Order 4: STATUS PLACED
Order 2: STATUS PREPARING
Order 1: STATUS PLACED
Order 4: STATUS PREPARING
Order 25: STATUS PLACED
Order 12: STATUS PLACED
Order 19: STATUS PLACED
Order 39: STATUS PLACED
Order 9: STATUS PLACED
Order 0: STATUS PLACED
Order 46: STATUS PLACED
Order 29: STATUS PLACED
Order 12: STATUS PREPARING
Order 54: STATUS PLACED
Order 5: STATUS PLACED
Order 19: STATUS PREPARING
Order 36: STATUS PLACED
Order 23: STATUS PLACED
Order 6: STATUS PLACED
Order 10: STATUS PLACED
Order 7: STATUS PLACED
Order 43: STATUS PLACED
Order 13: STATUS PLACED
Order 50: STATUS PLACED
Order 21: STATUS PLACED
Order 17: STATUS PLACED
Order 54: STATUS PREPARING
Order 15: STATUS PLACED
Order 35: STATUS PLACED
Order 5: STATUS PREPARING
Order 3: STATUS PLACED
Order 70: STATUS PLACED
Order 25: STATUS PREPARING
Order 74: STATUS PLACED
Order 14: STATUS PLACED
Order 45: STATUS PLACED
Order 79: STATUS PLACED
Order 8: STATUS PLACED
Order 49: STATUS PLACED
Order 18: STATUS PLACED
```

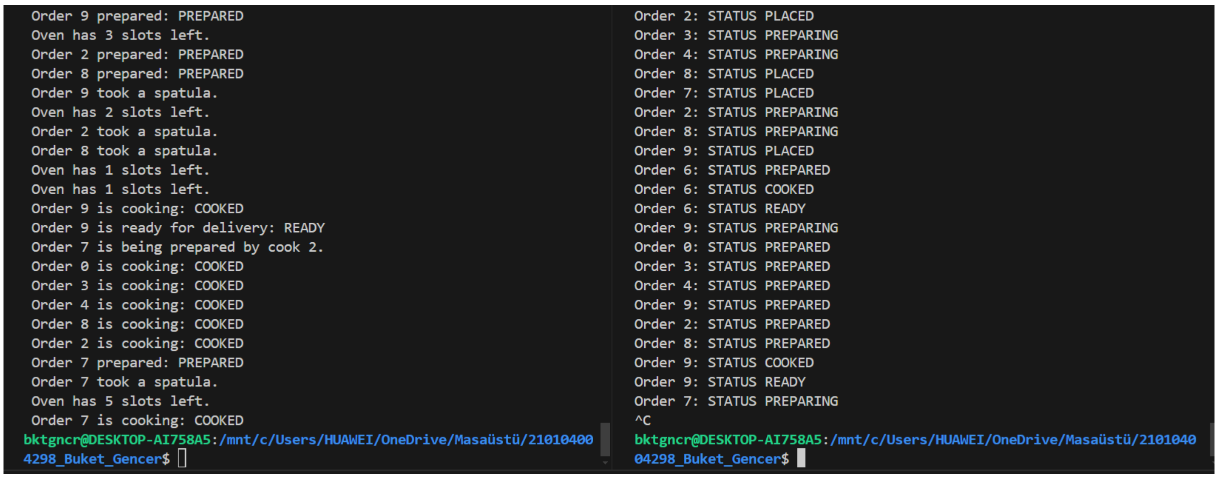


```
 Order 165 received: PLACED
New connection accepted from 127.0.0.1:34414
New connection accepted from 127.0.0.1:34422
 NEW_ORDER 192 1 7
 NEW_ORDER 194 1 14
 NEW_ORDER 170 0 4
 NEW_ORDER 171 2 9
 Order 171 received: PLACED
 NEW_ORDER 158 6 10
 NEW_ORDER 176 5 5
 NEW_ORDER 178 0 2
 NEW_ORDER 181 4 10
 NEW_ORDER 159 2 7
 Order 181 received: PLACED
 NEW_ORDER 183 9 16
 Order 183 received: PLACED
 Order 182 received: PLACED
 NEW_ORDER 187 3 16
 Order 186 received: PLACED
 NEW_ORDER 179 8 13
 Order 150 received: PLACED
 NEW_ORDER 189 0 2
 Order 189 received: PLACED
 NEW_ORDER 190 9 4
 Order 190 received: PLACED
 NEW_ORDER 193 4 13
 Order 193 received: PLACED
 Order 192 received: PLACED
 NEW_ORDER 169 4 18
 Order 170 received: PLACED
 NEW_ORDER 174 2 12
 Order 158 received: PLACED
 Order 176 received: PLACED
 Order 178 received: PLACED
 NEW_ORDER 154 9 18
 Order 159 received: PLACED
 NEW_ORDER 155 7 12
 NEW_ORDER 185 4 4
 NEW_ORDER 160 6 1
 NEW_ORDER 162 1 15
 NEW_ORDER 164 1 4
 NEW_ORDER 166 0 7
 Order 156 received: PLACED
```

```
Order 172: STATUS PREPARED
Order 172: STATUS COOKED
Order 172: STATUS READY
Order 184: STATUS PREPARING
Order 157: STATUS PREPARED
Order 9: STATUS READY
Order 173: STATUS PREPARING
Order 199: STATUS PREPARED
Order 28: STATUS READY
Order 194: STATUS PREPARING
Order 199: STATUS COOKED
Order 199: STATUS READY
Order 152: STATUS PREPARING
Order 102: STATUS READY
Order 160: STATUS PREPARING
Order 197: STATUS PREPARED
Order 197: STATUS COOKED
Order 197: STATUS READY
Order 155: STATUS PREPARING
Order 184: STATUS PREPARED
Order 184: STATUS COOKED
Order 184: STATUS READY
Order 154: STATUS PREPARING
Order 179: STATUS PREPARED
Order 157: STATUS COOKED
Order 173: STATUS PREPARED
Order 173: STATUS COOKED
Order 173: STATUS READY
Order 180: STATUS PREPARING
Order 198: STATUS PREPARED
Order 196: STATUS PREPARED
Order 154: STATUS PREPARED
Order 154: STATUS COOKED
Order 154: STATUS READY
Order 175: STATUS PREPARING
Order 179: STATUS COOKED
Order 157: STATUS READY
Order 148: STATUS PREPARING
Order 180: STATUS PREPARED
Order 180: STATUS COOKED
Order 180: STATUS READY
Order 195: STATUS PREPARING
Order 194: STATUS PREPARED
```

c. Test order cancellation

when the client cancelled an order server is shot down.:

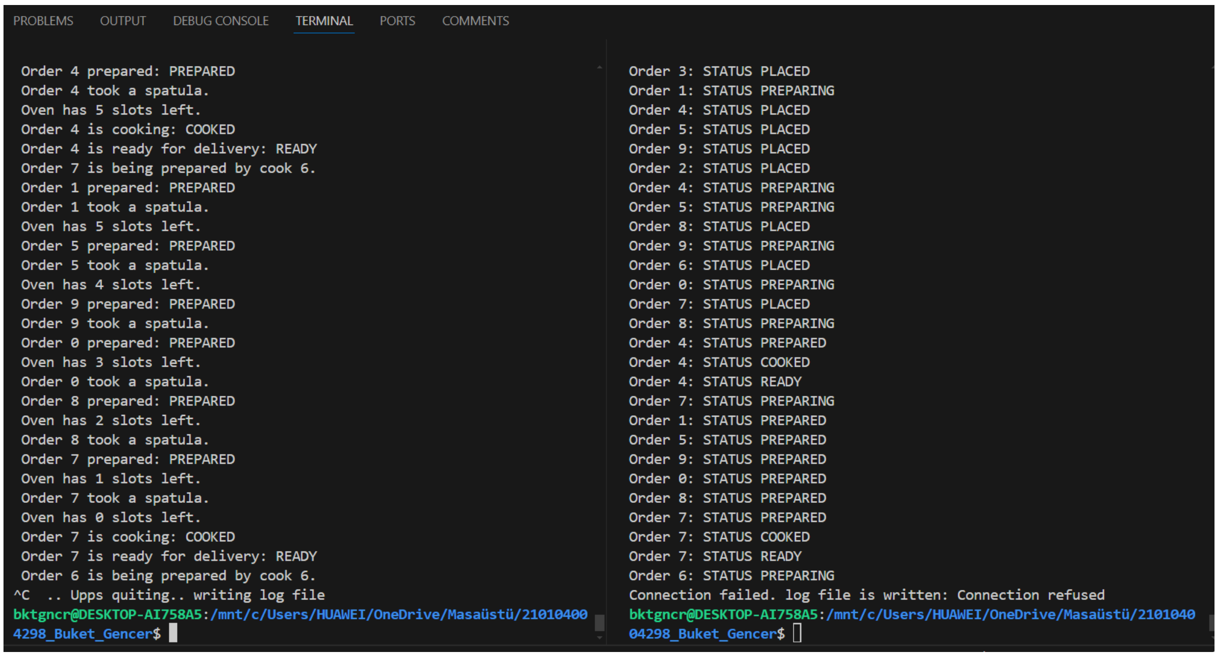


```
 Order 9 prepared: PREPARED
 Oven has 3 slots left.
 Order 2 prepared: PREPARED
 Order 8 prepared: PREPARED
 Order 9 took a spatula.
 Oven has 2 slots left.
 Order 2 took a spatula.
 Order 8 took a spatula.
 Oven has 1 slots left.
 Oven has 1 slots left.
 Order 9 is cooking: COOKED
 Order 9 is ready for delivery: READY
 Order 7 is being prepared by cook 2.
 Order 0 is cooking: COOKED
 Order 3 is cooking: COOKED
 Order 4 is cooking: COOKED
 Order 8 is cooking: COOKED
 Order 2 is cooking: COOKED
 Order 7 prepared: PREPARED
 Order 7 took a spatula.
 Oven has 5 slots left.
 Order 7 is cooking: COOKED
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4298_Buket_Gencer$ 
```

```
Order 2: STATUS PLACED
Order 3: STATUS PREPARING
Order 4: STATUS PREPARING
Order 8: STATUS PLACED
Order 7: STATUS PLACED
Order 2: STATUS PREPARING
Order 8: STATUS PREPARING
Order 9: STATUS PLACED
Order 6: STATUS PREPARED
Order 6: STATUS COOKED
Order 6: STATUS READY
Order 9: STATUS PREPARING
Order 0: STATUS PREPARED
Order 3: STATUS PREPARED
Order 4: STATUS PREPARED
Order 9: STATUS PREPARED
Order 2: STATUS PREPARED
Order 8: STATUS PREPARED
Order 9: STATUS COOKED
Order 9: STATUS READY
Order 7: STATUS PREPARING
^C
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ 
```

d. Test Signal Handling

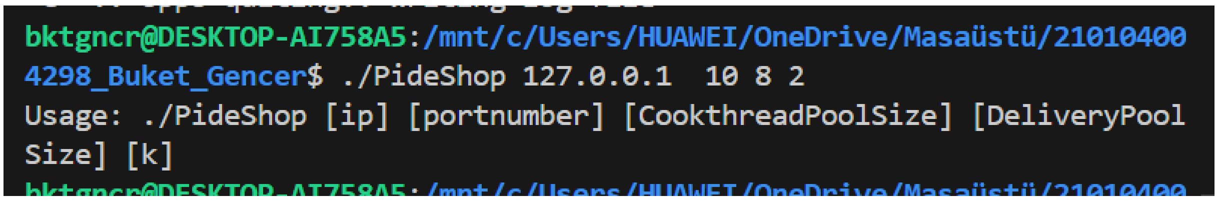
when the user press the ctrl c log file is written and programs terminate.



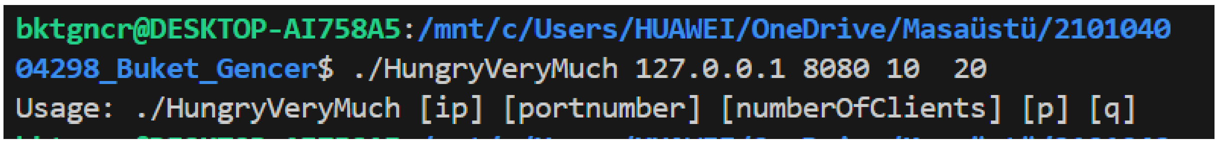
```
Order 4 prepared: PREPARED
Order 4 took a spatula.
Oven has 5 slots left.
Order 4 is cooking: COOKED
Order 4 is ready for delivery: READY
Order 7 is being prepared by cook 6.
Order 1 prepared: PREPARED
Order 1 took a spatula.
Oven has 5 slots left.
Order 5 prepared: PREPARED
Order 5 took a spatula.
Oven has 4 slots left.
Order 9 prepared: PREPARED
Order 9 took a spatula.
Order 0 prepared: PREPARED
Oven has 3 slots left.
Order 0 took a spatula.
Order 8 prepared: PREPARED
Oven has 2 slots left.
Order 8 took a spatula.
Order 7 prepared: PREPARED
Oven has 1 slots left.
Order 7 took a spatula.
Oven has 0 slots left.
Order 7 is cooking: COOKED
Order 7 is ready for delivery: READY
Order 6 is being prepared by cook 6.
^C  .. Upps quiting.. writing log file
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4298_Buket_Gencer$
```

```
Order 3: STATUS PLACED
Order 1: STATUS PREPARING
Order 4: STATUS PLACED
Order 5: STATUS PLACED
Order 9: STATUS PLACED
Order 2: STATUS PLACED
Order 4: STATUS PREPARING
Order 5: STATUS PREPARING
Order 8: STATUS PLACED
Order 9: STATUS PREPARING
Order 6: STATUS PLACED
Order 0: STATUS PREPARING
Order 7: STATUS PLACED
Order 8: STATUS PREPARING
Order 4: STATUS PREPARED
Order 4: STATUS COOKED
Order 4: STATUS READY
Order 7: STATUS PREPARING
Order 1: STATUS PREPARED
Order 5: STATUS PREPARED
Order 9: STATUS PREPARED
Order 0: STATUS PREPARED
Order 8: STATUS PREPARED
Order 7: STATUS PREPARED
Order 7: STATUS COOKED
Order 7: STATUS READY
Order 6: STATUS PREPARING
Connection failed. log file is written: Connection refused
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$
```

e. Trying enter with wrong input:

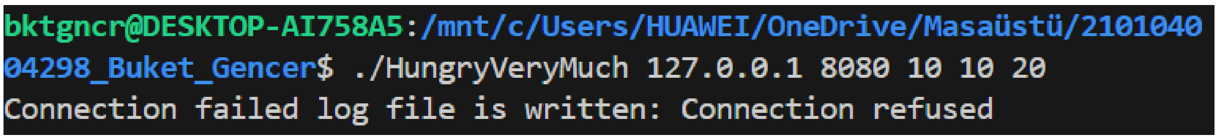


```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/21010400
4298_Buket_Gencer$ ./PideShop 127.0.0.1  10 8 2
Usage: ./PideShop [ip] [portnumber] [CookthreadPoolSize] [DeliveryPool
Size] [k]
```



```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 10  20
Usage: ./HungryVeryMuch [ip] [portnumber] [numberOfClients] [p] [q]
```

f. If the client tries to connect while the server is not running



```
bktgncr@DESKTOP-AI758A5:/mnt/c/Users/HUAWEI/OneDrive/Masaüstü/2101040
04298_Buket_Gencer$ ./HungryVeryMuch 127.0.0.1 8080 10 10 20
Connection failed log file is written: Connection refused
```

6. Conclusion

The Pide Shop simulation effectively demonstrates the use of multi-threading and IPC mechanisms to handle concurrent tasks in a server-client architecture. The implementation successfully manages orders from placement to delivery, with proper synchronization and resource management.