Gebze Technical University
Computer Engineering
CSE 344
System Programming
Homework 5 Report

SEDA KANIK
151044068

June 8, 2020

In this homework, we are responsible for manage the shopping between clients and florists. The main thread will read including of the file and creates threads of florists respect to it. The file contains names of florists and their coordinates of x and y, speed and name of flowers, then, names of clients, coordinates of x and y and name of their flower.

My Solution:

There is 3 types of struct; for florists, clients, statistics. They have all the information about owners. Some pointers and values created as global. Here is the work: First of all all parameter test is done and setted some errno and printed screen then exited. Then all the threads are initialized. File: We can assume the file is not empty and that its contents have the proper expected format. So, I do not need to check the content of the file.

1 Mutexes, Condition Variables and Purposes:

pthread_cond_t cond_thread: main thread triggers to threads with this condition variable. Thread waits for this condition
pthread_mutex_t mutex: used to lock florists during the sale
pthread_mutex_t mutex_main: used to lock main thread during the sending condition to florists
pthread_mutex_t mutex_after: used to lock florist during the checking the rear of the queue

2 Created Threads:

2.1 Florists:

We do not know how many florist are there but we know that, each one has an endless supply of given flowers.

3 Function Definitions:

3.1 main:

In the main function, calculated number of florist and created florist threads in calculated number. Main is also thread. I gived an ID for every florist. Main calculates number of clients, number of flowers of florists and reads strings and converts them into string, float. Main thread locked own and sends signal cond_thread among the client number when all the file is read. Also, signal SIGINT is masked in some position, creating and waiting threads, making free the resources. Checks the usage errors at the same time.

3.2 pool:

The parameter is the ID of florists. Locks with mutex and waits condition signal cond_thread, then sells the flower to client which is in front of the queue. And client leaves.

3.3 handler:

Handler waits a singal as SIGINT. If the signal SIGINT is comes then check the if resources are freed or not freed. Frees if the resources are not freed respect to its checking.

3.4 pop:

Decrease number of clients in queue, sends the front client and sets the front as second client.

3.5 numberofFlowers:

Calculates the number of flowers of florists.

3.6 numberofClients:

Calculates the number of clients.

3.7 numberofFlorist:

Calculates the number of florists.

3.8 readString:

Parses strings from file with some characters and keeps them into a char pointer and assigns as name or float.

4 Sample Screenshots of Program:

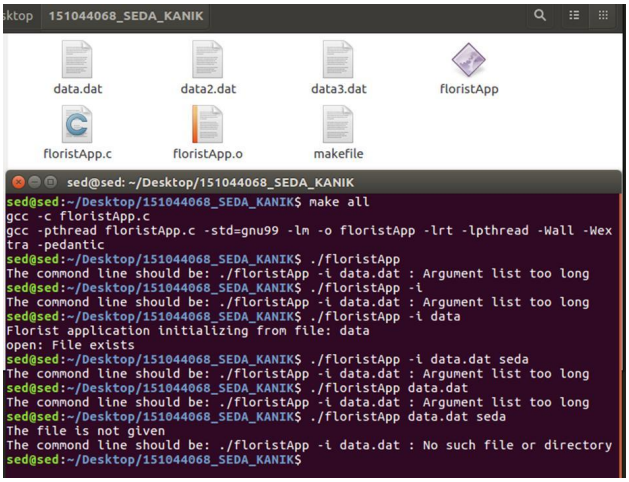


Figure 1: Tests for parameters

Test for memory leak:

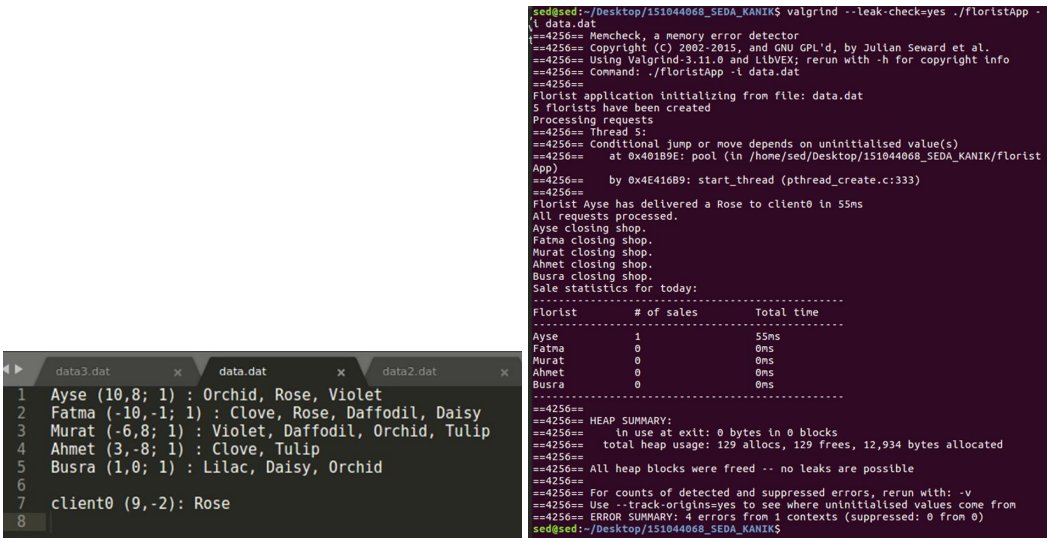


Figure 2: Input and Output Files

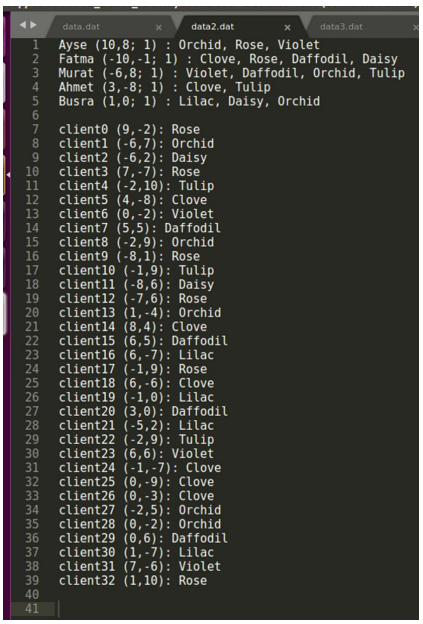


Figure 3: Input File

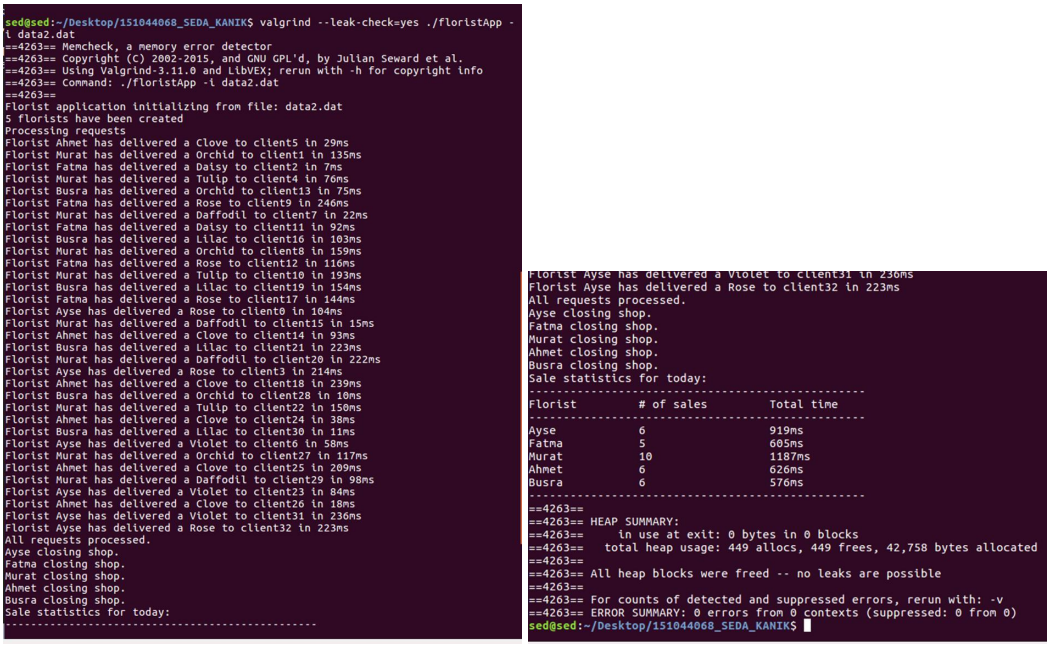
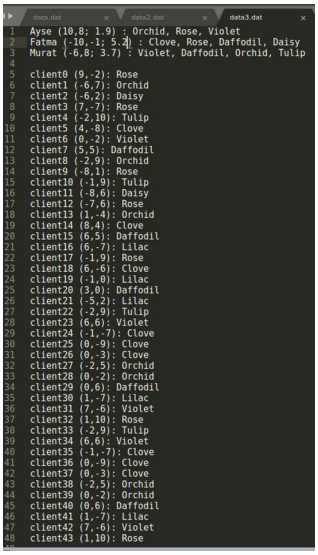


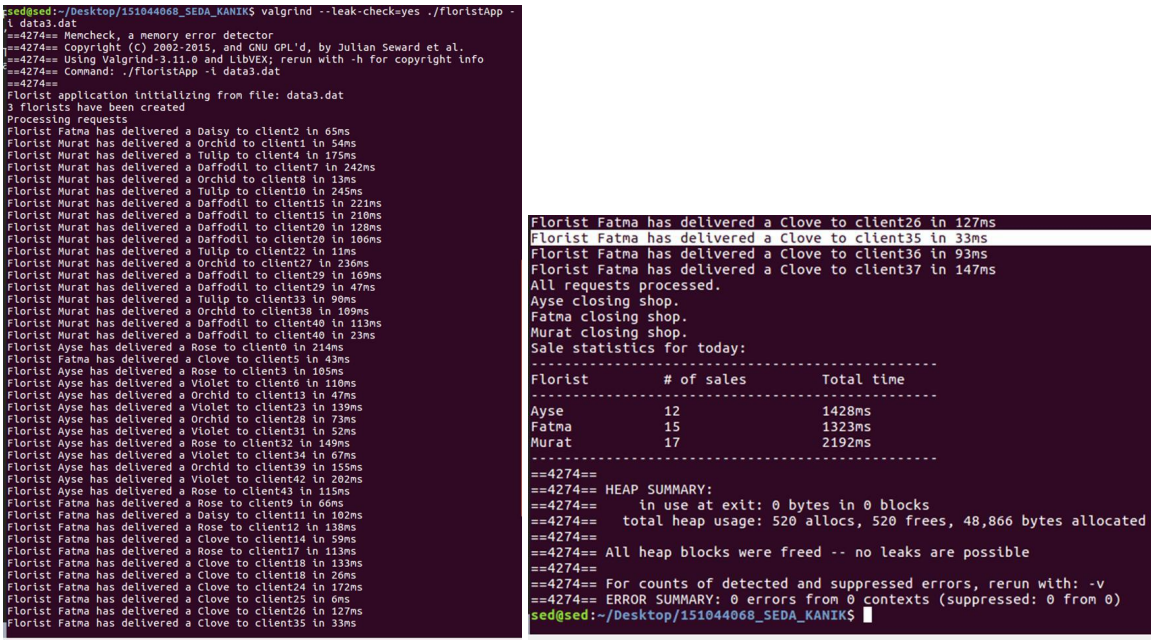
Figure 4: Output Files



```
Ayse (10,8; 1.9) : Orchid, Rose, Violet
Fatma (-10,-1; 5.2) : Clove, Rose, Daffodil, Daisy
Murat (-6,8; 3.7) : Violet, Daffodil, Orchid, Tulip

client0 (9,-2): Rose
client1 (-6,7): Orchid
client2 (-6,2): Daisy
client3 (7,-7): Rose
client4 (-2,10): Tulip
client5 (4,-8): Clove
client6 (0,-2): Violet
client7 (5,5): Daffodil
client8 (-2,9): Orchid
client9 (-8,1): Rose
client10 (-1,9): Tulip
client11 (-8,6): Daisy
client12 (-7,6): Rose
client13 (1,-4): Orchid
client14 (8,4): Clove
client15 (6,5): Daffodil
client16 (6,-7): Lilac
client17 (-1,9): Rose
client18 (6,-6): Clove
client19 (-1,0): Lilac
client20 (3,0): Daffodil
client21 (-5,2): Lilac
client22 (-2,9): Tulip
client23 (6,6): Violet
client24 (-1,-7): Clove
client25 (0,-9): Clove
client26 (0,-3): Clove
client27 (-2,5): Orchid
client28 (0,-2): Orchid
client29 (0,6): Daffodil
client30 (1,-7): Lilac
client31 (7,-6): Violet
client32 (1,10): Rose
client33 (-2,9): Tulip
client34 (6,6): Violet
client35 (-1,-7): Clove
client36 (0,-9): Clove
client37 (0,-3): Clove
client38 (-2,5): Orchid
client39 (0,-2): Orchid
client40 (0,6): Daffodil
client41 (1,-7): Lilac
client42 (7,-6): Violet
client43 (1,10): Rose
```

Figure 5: Input File



```
sed@sed:~/Desktop/151044068_SEDA_KANIK$ valgrind --leak-check=yes ./floristApp -
i data3.dat
==4274== Memcheck, a memory error detector
==4274== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==4274== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==4274== Command: ./floristApp -i data3.dat
==4274==
Florist application initializing from file: data3.dat
3 florists have been created
Processing requests
Florist Fatma has delivered a Daisy to client2 in 65ms
Florist Murat has delivered a Orchid to client1 in 54ms
Florist Murat has delivered a Tulip to client4 in 175ms
Florist Murat has delivered a Daffodil to client7 in 242ms
Florist Murat has delivered a Orchid to client8 in 13ms
Florist Murat has delivered a Tulip to client10 in 245ms
Florist Murat has delivered a Daffodil to client15 in 221ms
Florist Murat has delivered a Daffodil to client15 in 210ms
Florist Murat has delivered a Daffodil to client20 in 128ms
Florist Murat has delivered a Daffodil to client20 in 106ms
Florist Murat has delivered a Tulip to client22 in 11ms
Florist Murat has delivered a Orchid to client27 in 236ms
Florist Murat has delivered a Daffodil to client29 in 169ms
Florist Murat has delivered a Daffodil to client29 in 47ms
Florist Murat has delivered a Tulip to client33 in 90ms
Florist Murat has delivered a Orchid to client38 in 109ms
Florist Murat has delivered a Daffodil to client40 in 113ms
Florist Murat has delivered a Daffodil to client40 in 23ms
Florist Ayse has delivered a Rose to client0 in 214ms
Florist Fatma has delivered a Clove to client5 in 43ms
Florist Ayse has delivered a Rose to client3 in 105ms
Florist Ayse has delivered a Violet to client6 in 110ms
Florist Ayse has delivered a Orchid to client13 in 47ms
Florist Ayse has delivered a Violet to client23 in 139ms
Florist Ayse has delivered a Orchid to client28 in 73ms
Florist Ayse has delivered a Violet to client31 in 52ms
Florist Ayse has delivered a Rose to client32 in 149ms
Florist Ayse has delivered a Violet to client34 in 67ms
Florist Ayse has delivered a Orchid to client39 in 155ms
Florist Ayse has delivered a Violet to client42 in 202ms
Florist Ayse has delivered a Rose to client43 in 115ms
Florist Fatma has delivered a Rose to client9 in 66ms
Florist Fatma has delivered a Daisy to client11 in 102ms
Florist Fatma has delivered a Rose to client12 in 138ms
Florist Fatma has delivered a Clove to client14 in 59ms
Florist Fatma has delivered a Rose to client17 in 113ms
Florist Fatma has delivered a Clove to client18 in 133ms
Florist Fatma has delivered a Clove to client18 in 26ms
Florist Fatma has delivered a Clove to client24 in 172ms
Florist Fatma has delivered a Clove to client25 in 6ms
Florist Fatma has delivered a Clove to client26 in 127ms
Florist Fatma has delivered a Clove to client35 in 33ms
Florist Fatma has delivered a Clove to client36 in 93ms
Florist Fatma has delivered a Clove to client37 in 147ms
All requests processed.
Ayse closing shop.
Fatma closing shop.
Murat closing shop.
Sale statistics for today:
-------------------------------------------------
Florist         # of sales         Total time
-------------------------------------------------
Ayse            12                 1428ms
Fatma           15                 1323ms
Murat           17                 2192ms
-------------------------------------------------
==4274==
==4274== HEAP SUMMARY:
==4274==     in use at exit: 0 bytes in 0 blocks
==4274==   total heap usage: 520 allocs, 520 frees, 48,866 bytes allocated
==4274==
==4274== All heap blocks were freed -- no leaks are possible
==4274==
==4274== For counts of detected and suppressed errors, rerun with: -v
==4274== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
sed@sed:~/Desktop/151044068_SEDA_KANIK$
```

Figure 6: Output Files

SIGINT:

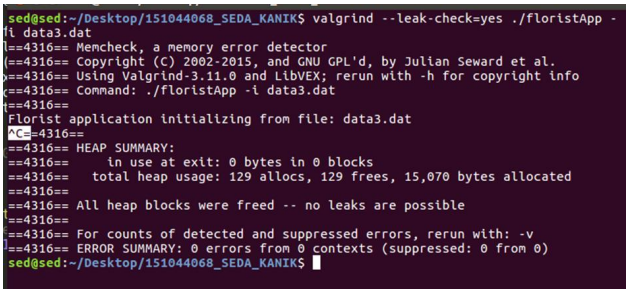


Figure 7: SIGINT before the creating of threads

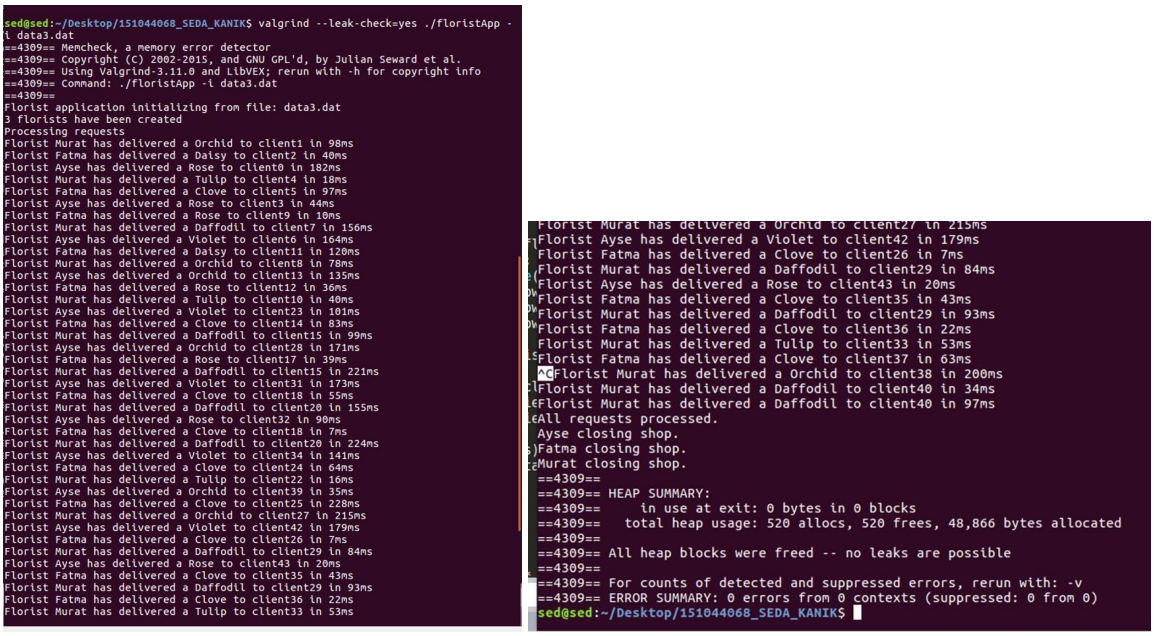


Figure 8: SIGINT in while execution of threads