

CSE 344 SYSTEM PROGRAMMING

HOMEWORK 05 - REPORT

GÖKHAN HAS - 161044067

1 Problem Solutions

In the homework, I got the number of florists by reading the file at first. Then I created my arrays. I kept it all as a global pointer. Actually, I had struct at first. But especially when the ctrlc signal came in, sometimes it wasn't completely free in the memory leak test. In order to prevent this, I went to such a solution.

Then I read the file one by one and initialized everything. I created as many threads as the number of florists. I also created the condition variable and mutex as much as the number of florists and initialized them.

The wish list is created one by one. The distance is calculated before it is created. Here Chebyshev distance is used. Then, the smallest distance is taken by comparing all the distances. This means that it is the closest florist that sells the same flower. This is also required from us in homework.

A problem arises here. Sometimes, no seller can sell the flower that the customer wants. I also checked this. If florists do not sell the flower they want, for example, it cannot be calculated. Since this distance is not calculated, the index does not return in the florist array. So the thread cannot be awakened. Because no florist will be able to serve that customer. Because they don't sell.

Then the calculated florist thread is awakened. This is of course done using the cwait function. One two-dimensional integer array was used before being awakened. This array serves as client queue. All of its elements are initially filled with -1. Then the client in the sequence added to the queue is changed to 1. This variable has been defined globally. A separate mutex has been defined since both threads and the main function can access and change. Mutex lock is used when the client queue variable is interfered with. Thus, anyone can interfere with this variable without creating any trouble.

All resources received must be returned. This process is done at the end of the mains by destroying mutex and condition variables and using the free function. In case of any signal (ctrlc) these operations should be done and done again.

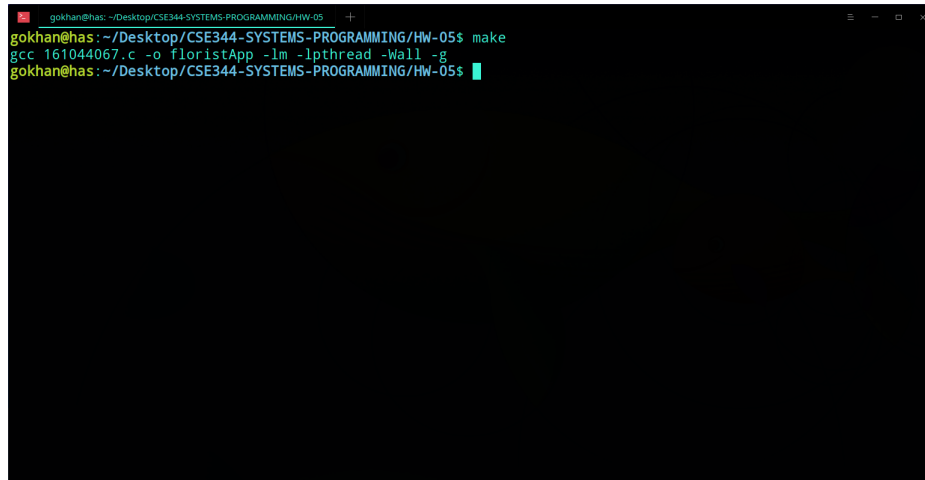
The thread function takes an index as a parameter and returns a struct that contains the information of that thread, that is, the products that the florist sells. In other words, the total sales figure and the total time are returned from the thread. Some control has been done in this function. When the CTRLC signal is received, the thread quits immediately, leaving the requests they are making. If everything goes normally, it makes his requests and leaves.

A global flag is marked in the signal function, the signal handler. This flag is checked continuously in threads. As the CTRLC signal (SIGINT) arrives at any time, the threads terminate because the value of the flags will change. After the threads are terminated, the program returns everything it uses. So there is no memory leak process. The screenshots of the tests related to this are available below. There is no memory leak when there is both normal and signal termination.

2 Used Functions

- void printUsage (): Prints the program's proper usage if it is misused.
- void errorExit (char * error): The program should give an error message and terminate cleanly.
- int getFloristCount (FILE * fptr, int * countMaxFLower): The number of flowers is taken from the file.
- void initializeArrays (int count, int countClient, int maxFlowerCount): Arrays are initialized. Each of them takes place in malloc.
- void freeArrays (int count, int clientCount, int maxFlowerCount): When the program is finished or terminated with ctrl + c, the received places are returned by this function.
- int getClientCount (FILE * fptr): The number of customers is returned.
- void readFile (FILE * fptr, int floristCount, int clientCount): The file is read and the import is done to the arrays. It should not be forgotten that the order of requests will be created one by one again.
- int getChCount (char * line, char ch): If there is a few of the value entered as a parameter in the string, it is returned.
- void * florist (void * index): The function where threads will run.

- `void endSignals (int floristCount)`: If the request list is short or sometimes the request part is made in the main, threads may not work by not entering an interrupt. To prevent this, the condition variable in the threads must be awakened again ...
- `double getChebyShevDistance (double x1, double y1, double x2, double y2)`: Chebyshev distance is calculated, the process is made with the shortest distance in the machine.
- `int isQueueHasElement (int index, int clientCount)`: It is checked if there is any element in the request list, if it is not, thread should be put into action.
- `void printInfo (int count, FloristEndInfo * infoArr)`: The last informative messages are printed.
- `void signal_catcher (int sigNo)`: Signal capture function, SIGINT signal is captured.

A terminal window with a black background and green text. The window title is 'gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05'. The command 'make' has been executed, resulting in the compilation of '161044067.c' into 'floristApp' with various compiler flags: '-lm -lpthread -Wall -g'.

```
gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05  
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$ make  
gcc 161044067.c -o floristApp -lm -lpthread -Wall -g  
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$
```

Figure 1: Compile

```

gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$ make
gcc 161044067.c -o floristApp -lm -lpthread -Wall -g
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$ ./floristApp -i
./floristApp: option requires an argument -- 'i'

##### USAGE #####
# -i : Data file parameter to floristApp #
#####

Aborted
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$ ./floristApp -i yokdosya.dat
ERROR ! Program filePath (-i) : No such file or directory
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$

```

Figure 2: Wrong Input Example

```

gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$
Florist Murat has delivered a orchid to client11 in 61ms
Florist Fatma has delivered a clove to client16 in 247ms
Florist Ayse has delivered a rose to client19 in 85ms
Florist Murat has delivered a violet to client12 in 170ms
Florist Ayse has delivered a rose to client22 in 22ms
Florist Fatma has delivered a daffodil to client17 in 45ms
Florist Fatma has delivered a daffodil to client18 in 53ms
Florist Ayse has delivered a rose to client23 in 233ms
Florist Murat has delivered a orchid to client15 in 170ms
Florist Ayse has delivered a orchid to client24 in 221ms
Florist Fatma has delivered a daffodil to client21 in 215ms
Florist Murat has delivered a orchid to client20 in 34ms
All requests processed.
Ayse closing shop.
Fatma closing shop.
Murat closing shop.
Sale statistics for today:
-----
Florist    # of sales    Total Time
-----
Ayse       10           1651ms
Fatma      7           902ms
Murat      7           971ms
-----
==18232==
==18232== HEAP SUMMARY:
==18232==   in use at exit: 0 bytes in 0 blocks
==18232== total heap usage: 91 allocs, 91 frees, 10,628 bytes allocated
==18232==
==18232== All heap blocks were freed -- no leaks are possible
==18232==
==18232== For counts of detected and suppressed errors, rerun with: -v
==18232== ERROR SUMMARY: 90 errors from 2 contexts (suppressed: 0 from 0)
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$

```

Figure 3: Normal Leak Test

```

gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05
==18310== at 0x4839D2A: strcmp (vg_replace_strmem.c:848)
==18310== by 0x401806: main (161044067.c:219)
==18310== Uninitialised value was created by a heap allocation
==18310== at 0x483577F: malloc (vg_replace_malloc.c:299)
==18310== by 0x4022B8: initializeArrays (161044067.c:413)
==18310== by 0x40154D: main (161044067.c:171)
==18310== Conditional jump or move depends on uninitialised value(s)
==18310== at 0x401809: main (161044067.c:219)
==18310== Uninitialised value was created by a heap allocation
==18310== at 0x483577F: malloc (vg_replace_malloc.c:299)
==18310== by 0x4022B8: initializeArrays (161044067.c:413)
==18310== by 0x40154D: main (161044067.c:171)
==18310==
Florist Murat has delivered a orchid to client1 in 84ms
Florist Ayse has delivered a orchid to client4 in 211ms
Florist Murat has delivered a daffodil to client3 in 160ms
Florist Ayse has delivered a orchid to client6 in 77ms
Florist Fatma has delivered a clove to client2 in 220ms
Florist Ayse has delivered a rose to client7 in 221ms
Florist Fatma has delivered a rose to client9 in 45ms
^CFlorist Murat has delivered a violet to client5 in 217ms
SIGINT WILL BE CAUGHT
PROGRAM WILL BE FINISHED, PLEASE BE PATIENT ...
==18310==
==18310== HEAP SUMMARY:
==18310== in use at exit: 0 bytes in 0 blocks
==18310== total heap usage: 91 allocs, 91 frees, 10,628 bytes allocated
==18310==
==18310== All heap blocks were freed -- no leaks are possible
==18310==
==18310== For counts of detected and suppressed errors, rerun with: -v
==18310== ERROR SUMMARY: 90 errors from 2 contexts (suppressed: 0 from 0)
gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$

```

Figure 4: Signal Leak Test

```

gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05
Florist Fatma has delivered a clove to client13 in 20ms
Florist Murat has delivered a orchid to client1 in 29ms
Florist Ayse has delivered a orchid to client6 in 193ms
Florist Murat has delivered a daffodil to client3 in 87ms
Florist Ayse has delivered a rose to client7 in 119ms
Florist Fatma has delivered a clove to client16 in 149ms
Florist Ayse has delivered a rose to client8 in 87ms
Florist Fatma has delivered a daffodil to client17 in 155ms
Florist Murat has delivered a violet to client5 in 204ms
Florist Ayse has delivered a violet to client10 in 228ms
Florist Ayse has delivered a orchid to client14 in 107ms
Florist Fatma has delivered a daffodil to client18 in 241ms
Florist Murat has delivered a orchid to client11 in 239ms
Florist Murat has delivered a violet to client12 in 43ms
Florist Ayse has delivered a rose to client19 in 90ms
Florist Fatma has delivered a daffodil to client21 in 148ms
Florist Murat has delivered a orchid to client15 in 93ms
Florist Murat has delivered a orchid to client20 in 44ms
Florist Ayse has delivered a rose to client22 in 116ms
Florist Ayse has delivered a rose to client23 in 71ms
Florist Ayse has delivered a orchid to client24 in 15ms
All requests processed.
Ayse closing shop.
Fatma closing shop.
Murat closing shop.
Sale statistics for today:
-----
Florist # of sales Total Time
-----
Ayse 10 1206ms
Fatma 7 860ms
Murat 7 739ms
-----
gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$

```

Figure 5: Normal Test

```
gokhan@has: ~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$ ./floristApp -i data.dat
Florist application initializing from file: data.dat
3 florists have been created.
Processing request
Florist  Ayse has delivered a  orchid to  client4 in  64ms
Florist  Fatma has delivered a  clove to  client2 in 144ms
Florist  Murat has delivered a  orchid to  client1 in  21ms
Florist  Fatma has delivered a  rose to   client9 in  88ms
Florist  Ayse has delivered a  orchid to  client6 in 140ms
Florist  Fatma has delivered a  clove to  client13 in 73ms
Florist  Murat has delivered a  daffodil to client3 in 187ms
Florist  Fatma has delivered a  clove to  client16 in 120ms
^C SIGINT WILL BE CAUGHT
PROGRAM WILL BE FINISHED, PLEASE BE PATIENT ...
gokhan@has:~/Desktop/CSE344-SYSTEMS-PROGRAMMING/HW-05$
```

Figure 6: Normal Signal Test