CSE 344 SYSTEMS PROGRAMMING
MIDTERM PROJECT REPORT

Emire Korkmaz
141044043

In this project, we were supposed to synchronize multiple processes in order to achieve a food delivery.

Firstly, I created cook(**xN**) and student(**xM**) processes. In my main process, the supplier, reads the given file into an array and starts to deliver plates one by one. After the kitchen size becomes a positive number, cooks start to transfer operation. They check the counter state in order to make the best choice to avoid any possible deadlocks. The algorithm checks the number of plates in the counter and picks the plate index that is minimum. And transfers the plate from the kitchen to the counter to serve the students.

In student side, students begin to take their trays by checking the number of the plates on the counter. Since a tray contains three different plates, all three of them must be available on the counter. After this check, students decrement the numbers of the plates (taking three of them) and look for an empty table to eat their food. When they finished the food, they go to the counter to eat again this loop continues for **L** times.

When the loop ends, students print out a message and exit. After all students exit, cook processes, that were waiting for students to be finished, prints out a message and exit. Finally the supplier which were waiting for all children to be finished, prints out a message and exits.

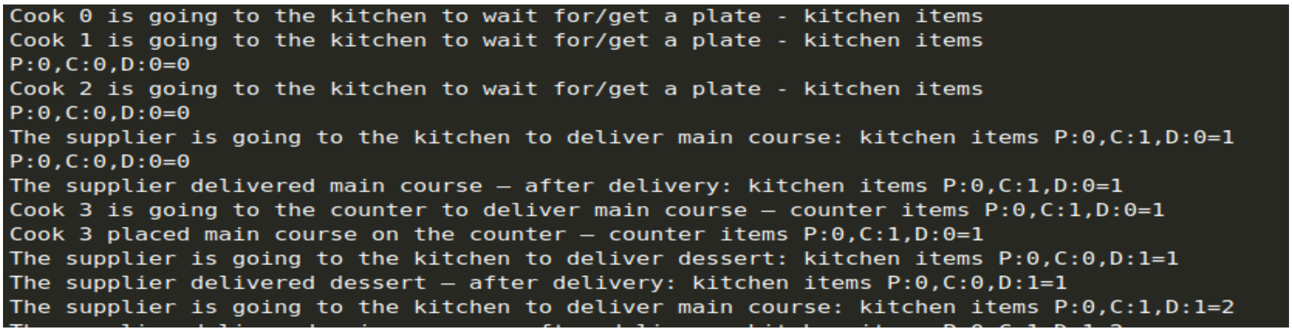
When **CTRL+C** comes, the handler catches the signal, terminates children processes and frees resources before exiting.

In order to use Graduate-Undergraduate mode, appropriate arguments should be given while running the program.

There are two modes first mode is with 13 arguments and the other one is with 15 arguments.

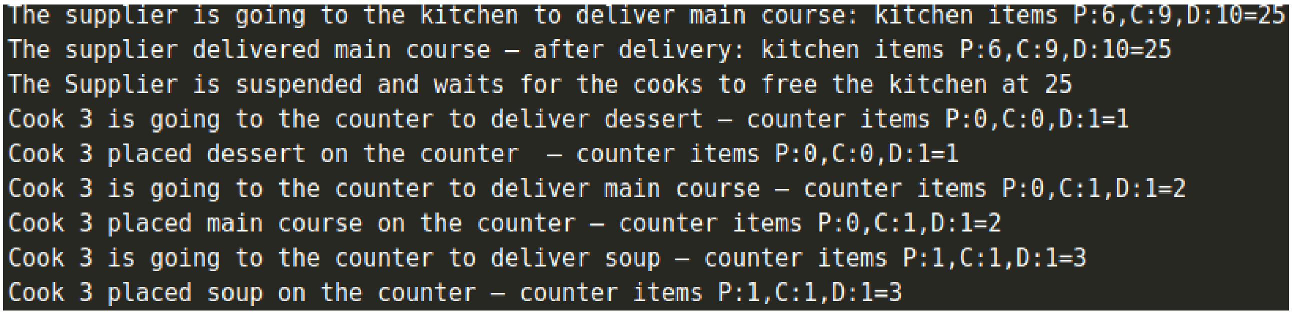
Example usages of the program are shown below.

The time program started the kitchen is not full yet.



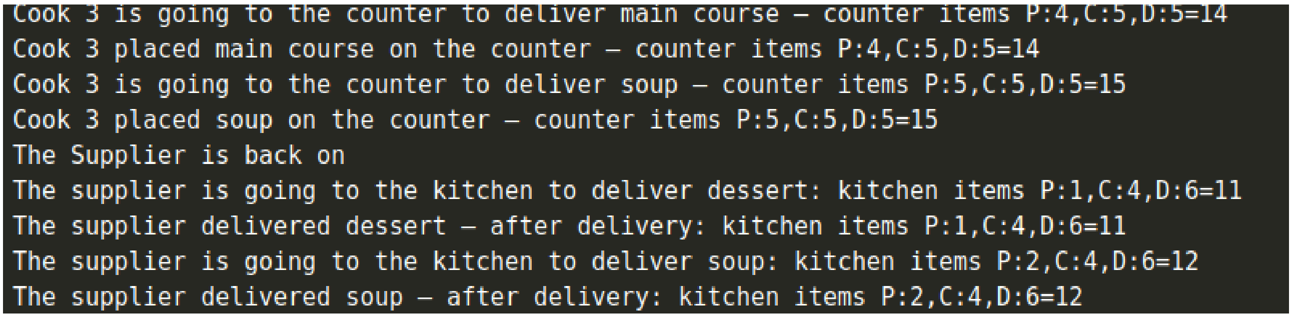
```
Cook 0 is going to the kitchen to wait for/get a plate - kitchen items
Cook 1 is going to the kitchen to wait for/get a plate - kitchen items
P:0,C:0,D:0=0
Cook 2 is going to the kitchen to wait for/get a plate - kitchen items
P:0,C:0,D:0=0
The supplier is going to the kitchen to deliver main course: kitchen items P:0,C:1,D:0=1
P:0,C:0,D:0=0
The supplier delivered main course – after delivery: kitchen items P:0,C:1,D:0=1
Cook 3 is going to the counter to deliver main course – counter items P:0,C:1,D:0=1
Cook 3 placed main course on the counter – counter items P:0,C:1,D:0=1
The supplier is going to the kitchen to deliver dessert: kitchen items P:0,C:0,D:1=1
The supplier delivered dessert – after delivery: kitchen items P:0,C:0,D:1=1
The supplier is going to the kitchen to deliver main course: kitchen items P:0,C:1,D:1=2
```

After kitchen is full and needed to be emptied. The Supplier is suspended in order to avoid busy waiting. After suspension, cooks take over and transfer the plates.



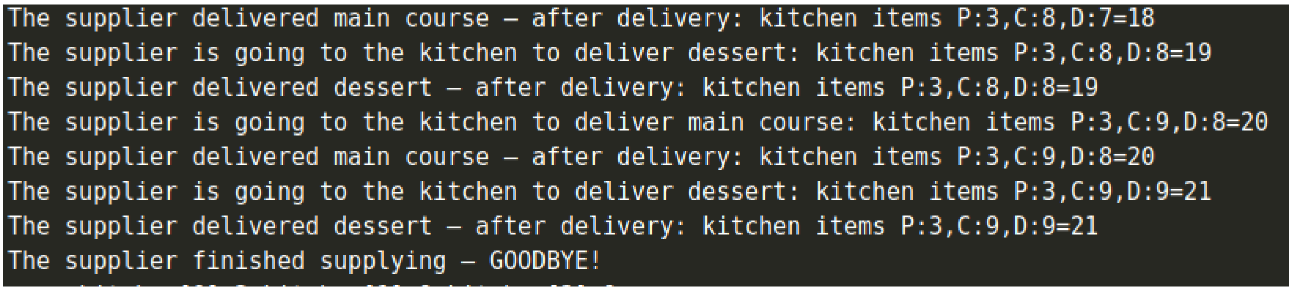
```
The supplier is going to the kitchen to deliver main course: kitchen items P:6,C:9,D:10=25
The supplier delivered main course – after delivery: kitchen items P:6,C:9,D:10=25
The Supplier is suspended and waits for the cooks to free the kitchen at 25
Cook 3 is going to the counter to deliver dessert – counter items P:0,C:0,D:1=1
Cook 3 placed dessert on the counter  – counter items P:0,C:0,D:1=1
Cook 3 is going to the counter to deliver main course – counter items P:0,C:1,D:1=2
Cook 3 placed main course on the counter – counter items P:0,C:1,D:1=2
Cook 3 is going to the counter to deliver soup – counter items P:1,C:1,D:1=3
Cook 3 placed soup on the counter – counter items P:1,C:1,D:1=3
```

After kitchen is emptied, a signal is sent to the supplier process to wake it up.



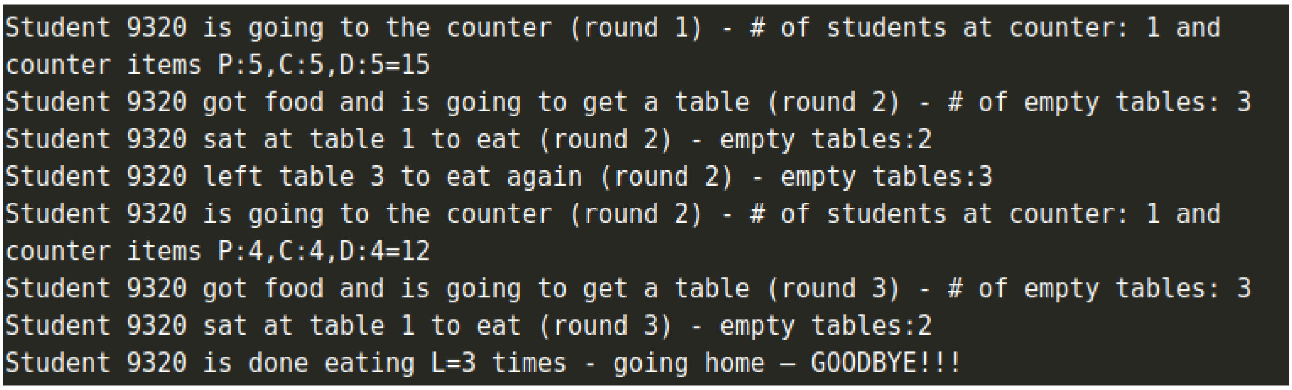
```
Cook 3 is going to the counter to deliver main course – counter items P:4,C:5,D:5=14
Cook 3 placed main course on the counter – counter items P:4,C:5,D:5=14
Cook 3 is going to the counter to deliver soup – counter items P:5,C:5,D:5=15
Cook 3 placed soup on the counter – counter items P:5,C:5,D:5=15
The Supplier is back on
The supplier is going to the kitchen to deliver dessert: kitchen items P:1,C:4,D:6=11
The supplier delivered dessert – after delivery: kitchen items P:1,C:4,D:6=11
The supplier is going to the kitchen to deliver soup: kitchen items P:2,C:4,D:6=12
The supplier delivered soup – after delivery: kitchen items P:2,C:4,D:6=12
```

After serving all the plates to the kitchen, the supplier prints out a message and waits the children processes (students **xM** and cooks **xN**).



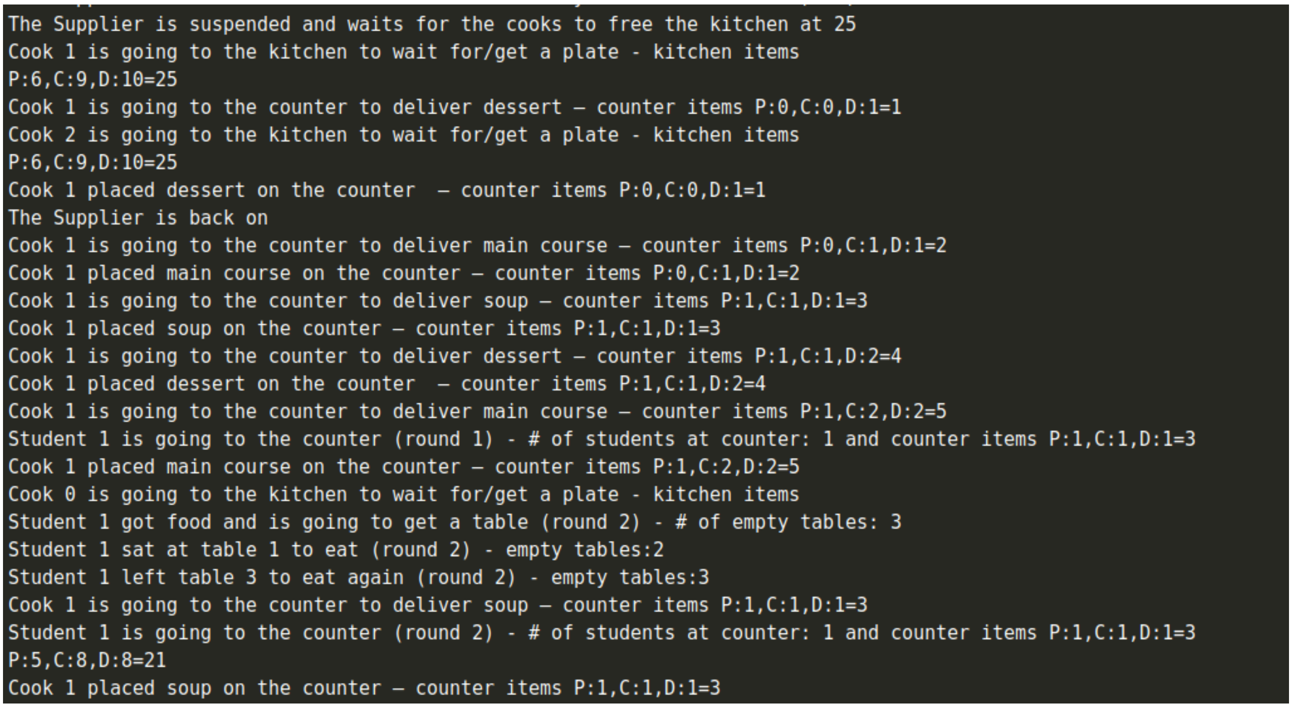
```
The supplier delivered main course – after delivery: kitchen items P:3,C:8,D:7=18
The supplier is going to the kitchen to deliver dessert: kitchen items P:3,C:8,D:8=19
The supplier delivered dessert – after delivery: kitchen items P:3,C:8,D:8=19
The supplier is going to the kitchen to deliver main course: kitchen items P:3,C:9,D:8=20
The supplier delivered main course – after delivery: kitchen items P:3,C:9,D:8=20
The supplier is going to the kitchen to deliver dessert: kitchen items P:3,C:9,D:9=21
The supplier delivered dessert – after delivery: kitchen items P:3,C:9,D:9=21
The supplier finished supplying – GOODBYE!
```

Cooks carry on serving the plates to the students and students take their trays for **xL** times.



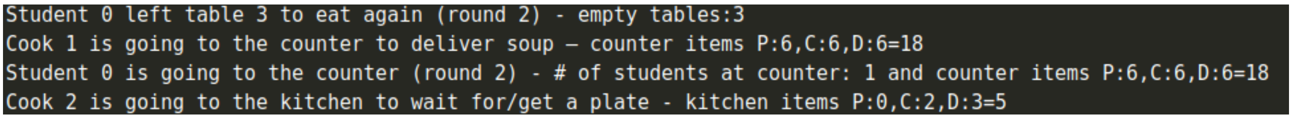
```
Student 9320 is going to the counter (round 1) - # of students at counter: 1 and
counter items P:5,C:5,D:5=15
Student 9320 got food and is going to get a table (round 2) - # of empty tables: 3
Student 9320 sat at table 1 to eat (round 2) - empty tables:2
Student 9320 left table 3 to eat again (round 2) - empty tables:3
Student 9320 is going to the counter (round 2) - # of students at counter: 1 and
counter items P:4,C:4,D:4=12
Student 9320 got food and is going to get a table (round 3) - # of empty tables: 3
Student 9320 sat at table 1 to eat (round 3) - empty tables:2
Student 9320 is done eating L=3 times - going home – GOODBYE!!!
```

An example of how the program works.

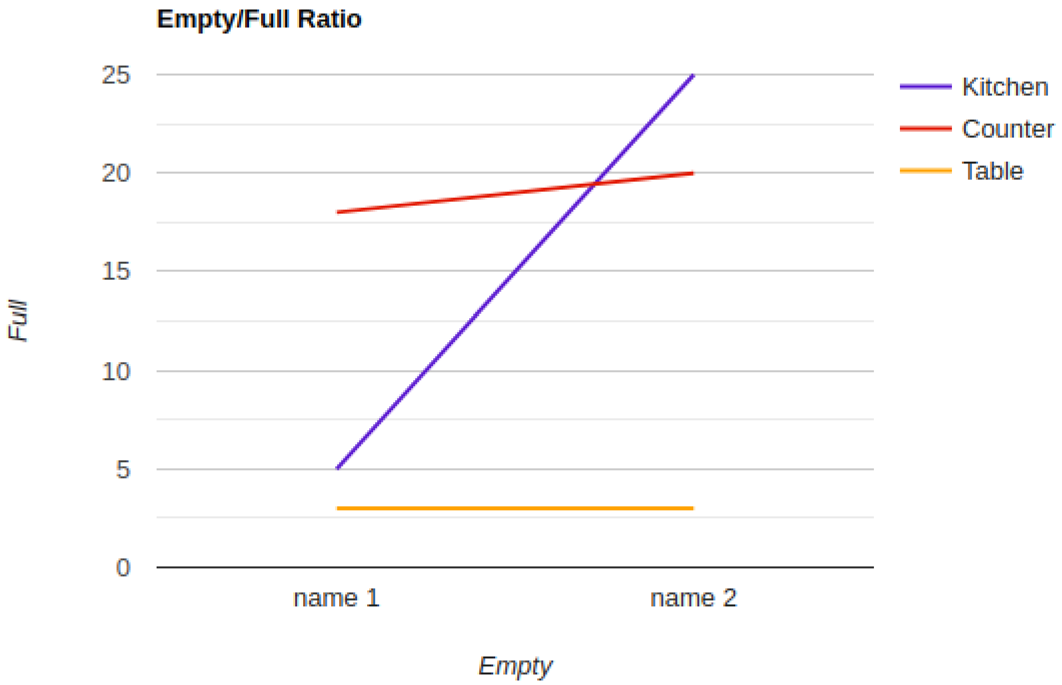


```
The Supplier is suspended and waits for the cooks to free the kitchen at 25
Cook 1 is going to the kitchen to wait for/get a plate - kitchen items
P:6,C:9,D:10=25
Cook 1 is going to the counter to deliver dessert – counter items P:0,C:0,D:1=1
Cook 2 is going to the kitchen to wait for/get a plate - kitchen items
P:6,C:9,D:10=25
Cook 1 placed dessert on the counter  – counter items P:0,C:0,D:1=1
The Supplier is back on
Cook 1 is going to the counter to deliver main course – counter items P:0,C:1,D:1=2
Cook 1 placed main course on the counter – counter items P:0,C:1,D:1=2
Cook 1 is going to the counter to deliver soup – counter items P:1,C:1,D:1=3
Cook 1 placed soup on the counter – counter items P:1,C:1,D:1=3
Cook 1 is going to the counter to deliver dessert – counter items P:1,C:1,D:2=4
Cook 1 placed dessert on the counter  – counter items P:1,C:1,D:2=4
Cook 1 is going to the counter to deliver main course – counter items P:1,C:2,D:2=5
Student 1 is going to the counter (round 1) - # of students at counter: 1 and counter items P:1,C:1,D:1=3
Cook 1 placed main course on the counter – counter items P:1,C:2,D:2=5
Cook 0 is going to the kitchen to wait for/get a plate - kitchen items
Student 1 got food and is going to get a table (round 2) - # of empty tables: 3
Student 1 sat at table 1 to eat (round 2) - empty tables:2
Student 1 left table 3 to eat again (round 2) - empty tables:3
Cook 1 is going to the counter to deliver soup – counter items P:1,C:1,D:1=3
Student 1 is going to the counter (round 2) - # of students at counter: 1 and counter items P:1,C:1,D:1=3
P:5,C:8,D:8=21
Cook 1 placed soup on the counter – counter items P:1,C:1,D:1=3
```

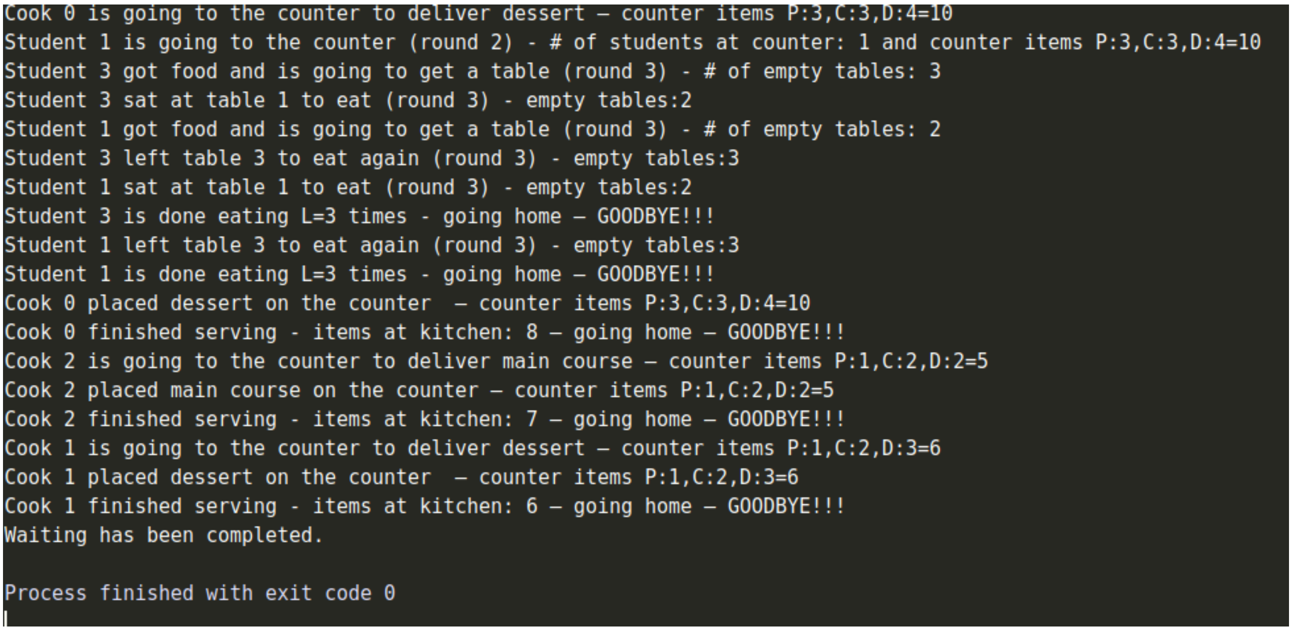
A graph plot is shown with the values below. I picked a random time to show empty space/full size ratio of the kitchen, counter and tables.



```
Student 0 left table 3 to eat again (round 2) - empty tables:3
Cook 1 is going to the counter to deliver soup – counter items P:6,C:6,D:6=18
Student 0 is going to the counter (round 2) - # of students at counter: 1 and counter items P:6,C:6,D:6=18
Cook 2 is going to the kitchen to wait for/get a plate - kitchen items P:0,C:2,D:3=5
```



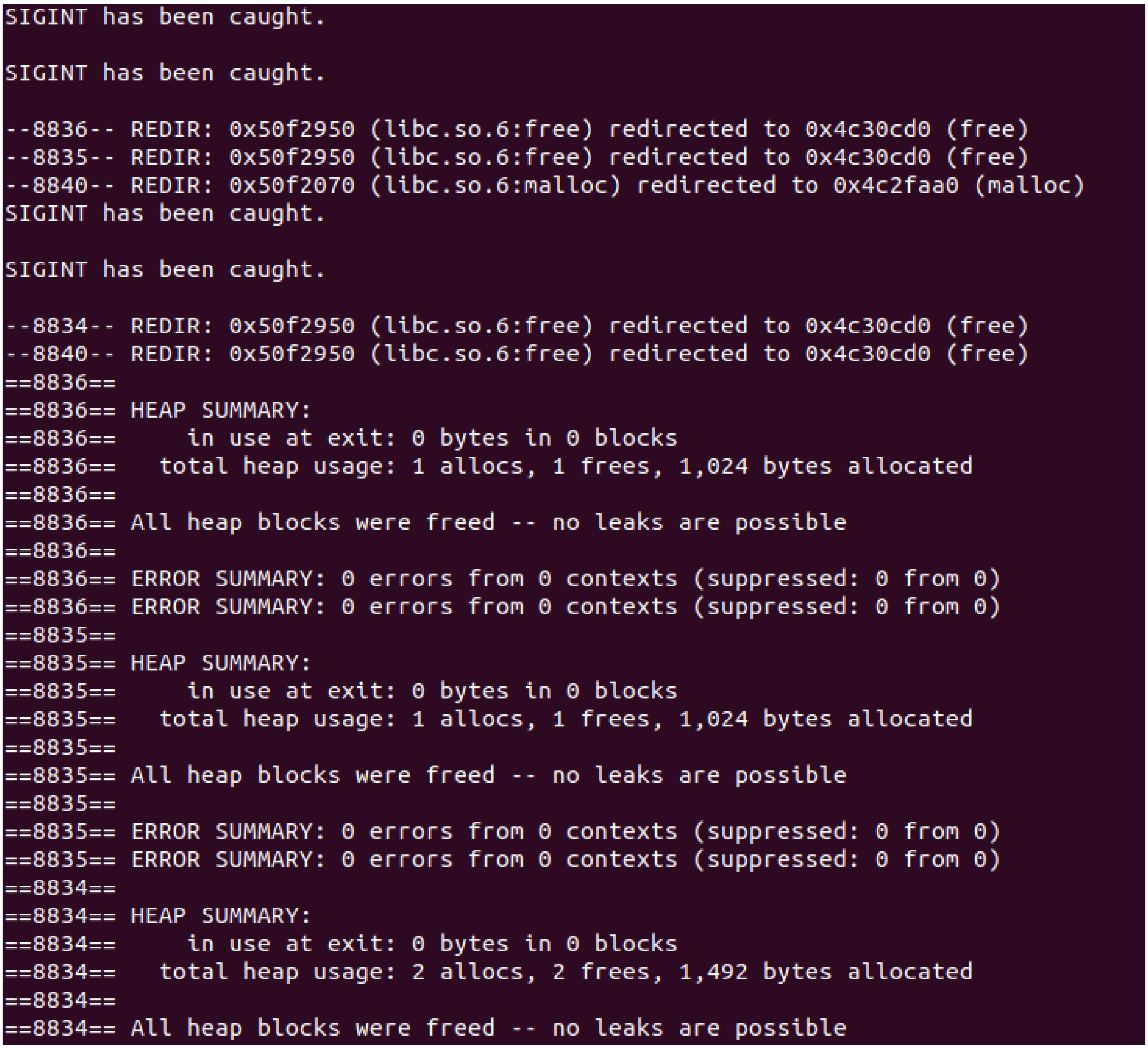
After all processes are finished, program ends like this.



```
Cook 0 is going to the counter to deliver dessert – counter items P:3,C:3,D:4=10
Student 1 is going to the counter (round 2) - # of students at counter: 1 and counter items P:3,C:3,D:4=10
Student 3 got food and is going to get a table (round 3) - # of empty tables: 3
Student 3 sat at table 1 to eat (round 3) - empty tables:2
Student 1 got food and is going to get a table (round 3) - # of empty tables: 2
Student 3 left table 3 to eat again (round 3) - empty tables:3
Student 1 sat at table 1 to eat (round 3) - empty tables:2
Student 3 is done eating L=3 times - going home – GOODBYE!!!
Student 1 left table 3 to eat again (round 3) - empty tables:3
Student 1 is done eating L=3 times - going home – GOODBYE!!!
Cook 0 placed dessert on the counter  – counter items P:3,C:3,D:4=10
Cook 0 finished serving - items at kitchen: 8 – going home – GOODBYE!!!
Cook 2 is going to the counter to deliver main course – counter items P:1,C:2,D:2=5
Cook 2 placed main course on the counter – counter items P:1,C:2,D:2=5
Cook 2 finished serving - items at kitchen: 7 – going home – GOODBYE!!!
Cook 1 is going to the counter to deliver dessert – counter items P:1,C:2,D:3=6
Cook 1 placed dessert on the counter  – counter items P:1,C:2,D:3=6
Cook 1 finished serving - items at kitchen: 6 – going home – GOODBYE!!!
Waiting has been completed.

Process finished with exit code 0
```

The screen shot below shows the Valgrind result.



```
SIGINT has been caught.

SIGINT has been caught.

--8836-- REDIR: 0x50f2950 (libc.so.6:free) redirected to 0x4c30cd0 (free)
--8835-- REDIR: 0x50f2950 (libc.so.6:free) redirected to 0x4c30cd0 (free)
--8840-- REDIR: 0x50f2070 (libc.so.6:malloc) redirected to 0x4c2faa0 (malloc)
SIGINT has been caught.

SIGINT has been caught.

--8834-- REDIR: 0x50f2950 (libc.so.6:free) redirected to 0x4c30cd0 (free)
--8840-- REDIR: 0x50f2950 (libc.so.6:free) redirected to 0x4c30cd0 (free)
==8836==
==8836== HEAP SUMMARY:
==8836==     in use at exit: 0 bytes in 0 blocks
==8836==   total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==8836==
==8836== All heap blocks were freed -- no leaks are possible
==8836==
==8836== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==8836== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==8835==
==8835== HEAP SUMMARY:
==8835==     in use at exit: 0 bytes in 0 blocks
==8835==   total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==8835==
==8835== All heap blocks were freed -- no leaks are possible
==8835==
==8835== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==8835== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==8834==
==8834== HEAP SUMMARY:
==8834==     in use at exit: 0 bytes in 0 blocks
==8834==   total heap usage: 2 allocs, 2 frees, 1,492 bytes allocated
==8834==
==8834== All heap blocks were freed -- no leaks are possible
```