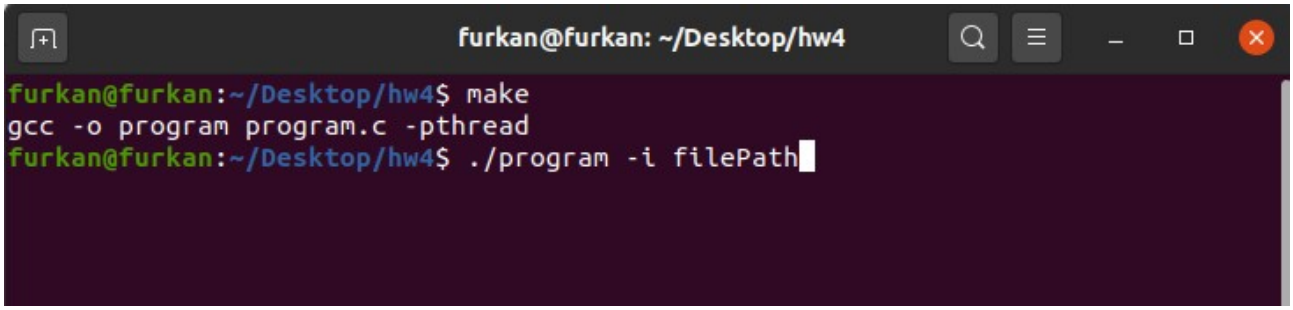
CSE344 – System Programming
REPORT - HW 4

Furkan ÖZEV
161044036

COMPILE & RUN:



MAIN IDEA:

→ There is an array.
→ The ingredients will be delivered this array.
→ This array located at the heap(because global), thus it shared among all involved threads.
→ There are 4 ingredients type.
→ When producing or using these ingredients, they are synchronized by a single semaphore.
→ Because, one of the chefs must have been supplied with the 2 ingredients to work.
→ 2 Ingredients must be supplied and took simultaneously.
→ So I preferred the sys-V semaphore.
→ It will create an temporay file for ftok (sys-V semaphore).
→ There is a simple semaphore for the wholesaler to wait for the dessert to be prepared and the chef to notify the dessert to be ready.
→ Since this semaphore will only wait and post for one product, I preferred posix semaphore.
→ The wholesaler waits for the preparation of the dessert with this semaphore, when Chef prepares the dessert, chef posts this semaphore.
→ Once the wholesaler is complete, it uses the notify () function to let the chefs know there will be no more content. It also uses the global flag (“flag”) for this process.
→ When there is an error, it releases the allocated memory and deletes the temporary file by using ExitFailure function.

WHOLESALER - MAIN IDEA:

→ The wholesaler reads 2 bytes from the file.
→ The wholesaler writes these bytes to the array data structure.
→ Then, using system-v semaphores, the value of the ingredients read is increased by 1.
→ Using the System-V semaphore, these 2 ingredients are increased simultaneously.
→ Thus, many synchronization problems are prevented.
→ After the wholesaler has done the necessary printing, wholesaler is waiting for the chef to prepare the dessert by using the posix semaphore.
→ When the dessert is prepared, it leaves without waiting and performs the necessary printing.
→ It continues the same operations as long as there are not enough ingredients in the file.
→ If there is no ingredient left, it changes the global variable named flag and notifies the chefs using the notify function.
→ So, if the chefs are on hold, they will come out and finish the process. If it is not on hold, it ends its process without going to wait again.

CHEFS - MAIN IDEA:

→ There are 4 ingredients in total, and each chef lack of 2 ingredients. So there are 6 combinations for chefs.
→ Each chef waits for 2 ingredients determined for her/himself using sys-V semaphores.
→ If there are, the related ingredients decreases their values by 1, otherwise it waits until it happens.
→ Using the System-V semaphore, these 2 ingredients are decreased simultaneously.
→ Thus, many synchronization problems are prevented.
→ Ingredients may have been increased by notify function. It understands this by checking flag.
→ If it is increased by notify, it means that the wholesaler is finished, so it will end in the chefs.
→ Otherwise, Chefs will do the necessary printing.
→ Then, chef can simulate dessert preparation by sleeping for a random number of seconds (1 to 5 inclusive).
→ When the dessert is ready, the chef wakes the wholesaler by posting the posix semaphore.
→ The same processes continue until the notify function works.

NOTIFY - MAIN IDEA:

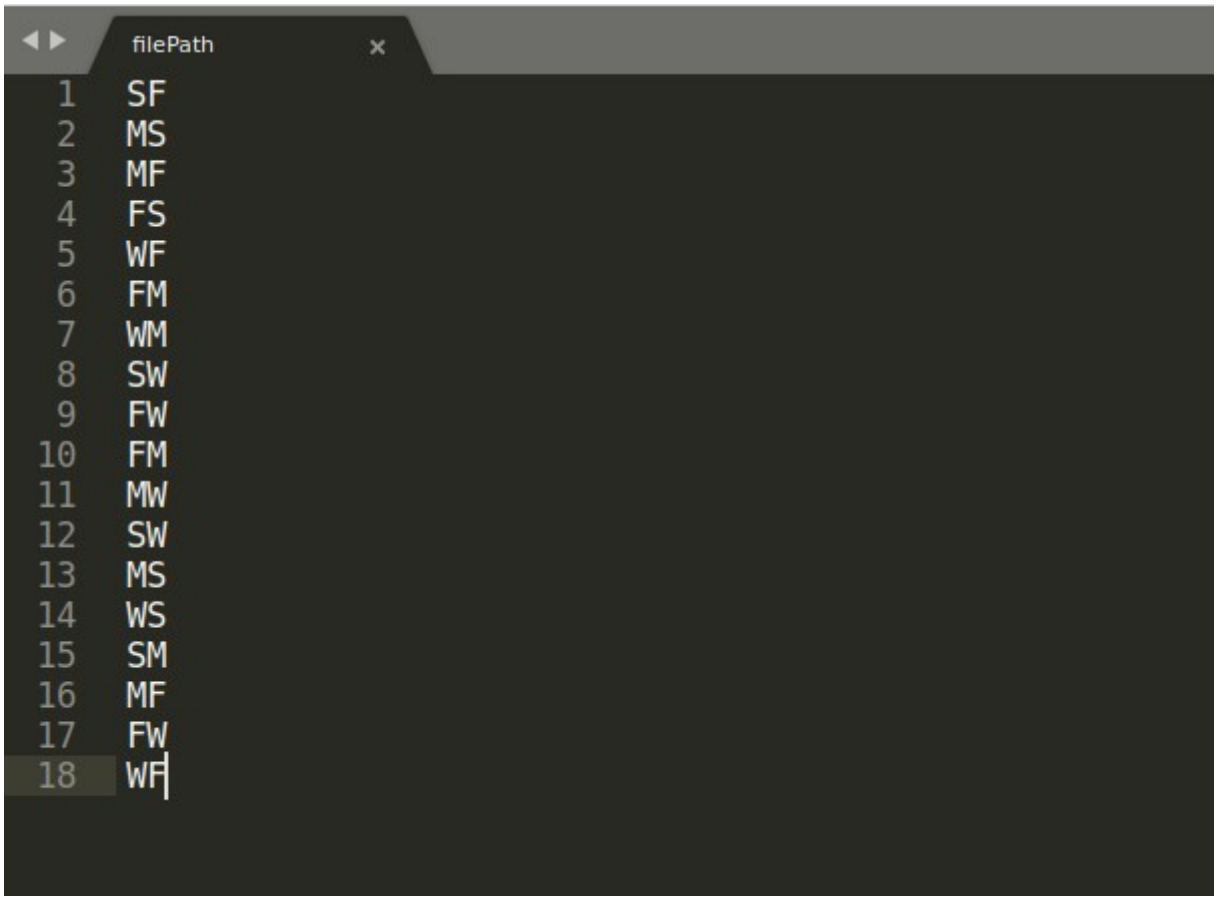
→ Once the wholesaler is done, she/he'll need a way to notify the chefs that there won't be any more ingredients.
→ It ensures that all pending threads are awakened.
→ By increasing each ingredients by 3 each, all kinds of threads are released without waiting.
→ These threads will check the flag variable and understand that it is finished.

EXIT - MAIN IDEA:

→ When there is an error, it releases the allocated memory and deletes the temporary file.

RUN EXAMPLE :

INPUT FILE: (18 ingredients couple)



OUTPUT:

```
furkan@furkan:~/Desktop/hw4$ make
gcc -o program program.c -pthread
furkan@furkan:~/Desktop/hw4$ ./program -i filePath
chef1 is waiting for milk and flour
the wholesaler delivers flour and sugar
the wholesaler is waiting for the dessert
chef4 is waiting for flour and walnuts
chef5 is waiting for flour and sugar
chef5 has taken the sugar
chef5 has taken the flour
chef5 is preparing the dessert
chef6 is waiting for walnuts and sugar
chef2 is waiting for milk and walnuts
chef3 is waiting for milk and sugar
chef5 has delivered the dessert to the wholesaler
chef5 is waiting for flour and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and sugar
the wholesaler is waiting for the dessert
chef3 has taken the milk
chef3 has taken the sugar
chef3 is preparing the dessert
chef3 has delivered the dessert to the wholesaler
chef3 is waiting for milk and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and flour
the wholesaler is waiting for the dessert
chef1 has taken the milk
chef1 has taken the flour
chef1 is preparing the dessert
chef1 has delivered the dessert to the wholesaler
chef1 is waiting for milk and flour
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers flour and sugar
the wholesaler is waiting for the dessert
chef5 has taken the flour
chef5 has taken the sugar
chef5 is preparing the dessert
chef5 has delivered the dessert to the wholesaler
chef5 is waiting for flour and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers flour and walnuts
the wholesaler is waiting for the dessert
chef4 has taken the walnuts
chef4 has taken the flour
chef4 is preparing the dessert
chef4 has delivered the dessert to the wholesaler
chef4 is waiting for flour and walnuts
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and flour
the wholesaler is waiting for the dessert
chef1 has taken the flour
chef1 has taken the milk
chef1 is preparing the dessert
chef1 has delivered the dessert to the wholesaler
chef1 is waiting for milk and flour
```

```
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and walnuts
the wholesaler is waiting for the dessert
chef2 has taken the walnuts
chef2 has taken the milk
chef2 is preparing the dessert
chef2 has delivered the dessert to the wholesaler
chef2 is waiting for milk and walnuts
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers walnuts and sugar
the wholesaler is waiting for the dessert
chef6 has taken the sugar
chef6 has taken the walnuts
chef6 is preparing the dessert
chef6 has delivered the dessert to the wholesaler
chef6 is waiting for walnuts and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers flour and walnuts
the wholesaler is waiting for the dessert
chef4 has taken the flour
chef4 has taken the walnuts
chef4 is preparing the dessert
chef4 has delivered the dessert to the wholesaler
chef4 is waiting for flour and walnuts
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and flour
the wholesaler is waiting for the dessert
chef1 has taken the flour
chef1 has taken the milk
chef1 is preparing the dessert
chef1 has delivered the dessert to the wholesaler
chef1 is waiting for milk and flour
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and walnuts
the wholesaler is waiting for the dessert
chef2 has taken the milk
chef2 has taken the walnuts
chef2 is preparing the dessert
chef2 has delivered the dessert to the wholesaler
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers walnuts and sugar
the wholesaler is waiting for the dessert
chef2 is waiting for milk and walnuts
chef6 has taken the sugar
chef6 has taken the walnuts
chef6 is preparing the dessert
chef6 has delivered the dessert to the wholesaler
chef6 is waiting for walnuts and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and sugar
the wholesaler is waiting for the dessert
chef3 has taken the milk
chef3 has taken the sugar
chef3 is preparing the dessert
chef3 has delivered the dessert to the wholesaler
chef3 is waiting for milk and sugar
```

```
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers walnuts and sugar
the wholesaler is waiting for the dessert
chef6 has taken the walnuts
chef6 has taken the sugar
chef6 is preparing the dessert
chef6 has delivered the dessert to the wholesaler
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and sugar
the wholesaler is waiting for the dessert
chef3 has taken the sugar
chef3 has taken the milk
chef3 is preparing the dessert
chef6 is waiting for walnuts and sugar
chef3 has delivered the dessert to the wholesaler
chef3 is waiting for milk and sugar
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers milk and flour
the wholesaler is waiting for the dessert
chef1 has taken the milk
chef1 has taken the flour
chef1 is preparing the dessert
chef1 has delivered the dessert to the wholesaler
chef1 is waiting for milk and flour
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers flour and walnuts
the wholesaler is waiting for the dessert
chef4 has taken the flour
chef4 has taken the walnuts
chef4 is preparing the dessert
chef4 has delivered the dessert to the wholesaler
chef4 is waiting for flour and walnuts
the wholesaler has obtained the dessert and left to sell it
the wholesaler delivers flour and walnuts
the wholesaler is waiting for the dessert
chef4 has taken the walnuts
chef4 has taken the flour
chef4 is preparing the dessert
chef4 has delivered the dessert to the wholesaler
chef4 is waiting for flour and walnuts
the wholesaler has obtained the dessert and left to sell it
wholesaler finished supplying ingredients.
cook 5 finished preparing desserts.
cook 6 finished preparing desserts.
cook 2 finished preparing desserts.
cook 1 finished preparing desserts.
cook 3 finished preparing desserts.
cook 4 finished preparing desserts.

----- Program was successfully completed. -----

furkan@furkan:~/Desktop/hw4$
```

NOTICED:

1- I used the getopt() library method for parsing command line arguments.

2- All chef threads execute the same function. (function name: “thread_function”).

3- Every chef use the same semaphores for synchronization. (sem_sync: sys-V, sem_mut: POSIX)

4- Wholesaler is unaware of how many threads/chefs are waiting.

5- Once the wholesaler is complete, it uses the notify () function to let the chefs know there will be no more content.

6- In case of an error, I exit program by printing to stderr a nicely formatted informative errno based message.

7- If the command line arguments are missing/invalid my program will print usage information and exit

8- Each thread remove allocated resources explicitly.

9- I prevent the zombie process with using pthred_joind and semaphores.

10- Compilation is warning-free