

CSE 344 System Programming

HW3

Murat BEYAZ

151044004

Problem Definition

In this homework we are expected to simulate a bakery with using 8 processes, semaphores and shared memory. This bakery makes only g  lla   with 4 ingredients. These ingredients are:

- milk(M)
- flour(F)
- walnuts(W)
- sugar(S)

6 of these 8 processes are chef processes and the other two are wholesaler and pusher processes. Chef processes has 2 out of 4 ingredients infinitely and needs the other 2 ingredients for making g  lla  . For example first chef has milk and flour but lacks walnuts and sugar. Every chef waits for remaining two ingredients to arrive in order to go and prepare g  lla  . Once the g  lla   is ready they deliver it to wholesaler and continue to wait for more ingredients.

Wholesaler delivers 2 distinct ingredients out of 4 and lets chefs know that the ingredient have arrived via the pusher process and then waits for the dessert to be ready. Once the g  lla   is ready wholesaler takes the g  lla   to sell it. The wholesaler will read the ingredients to deliver from an input file containing two capital letters at each line, representing the ingredients to deliver.

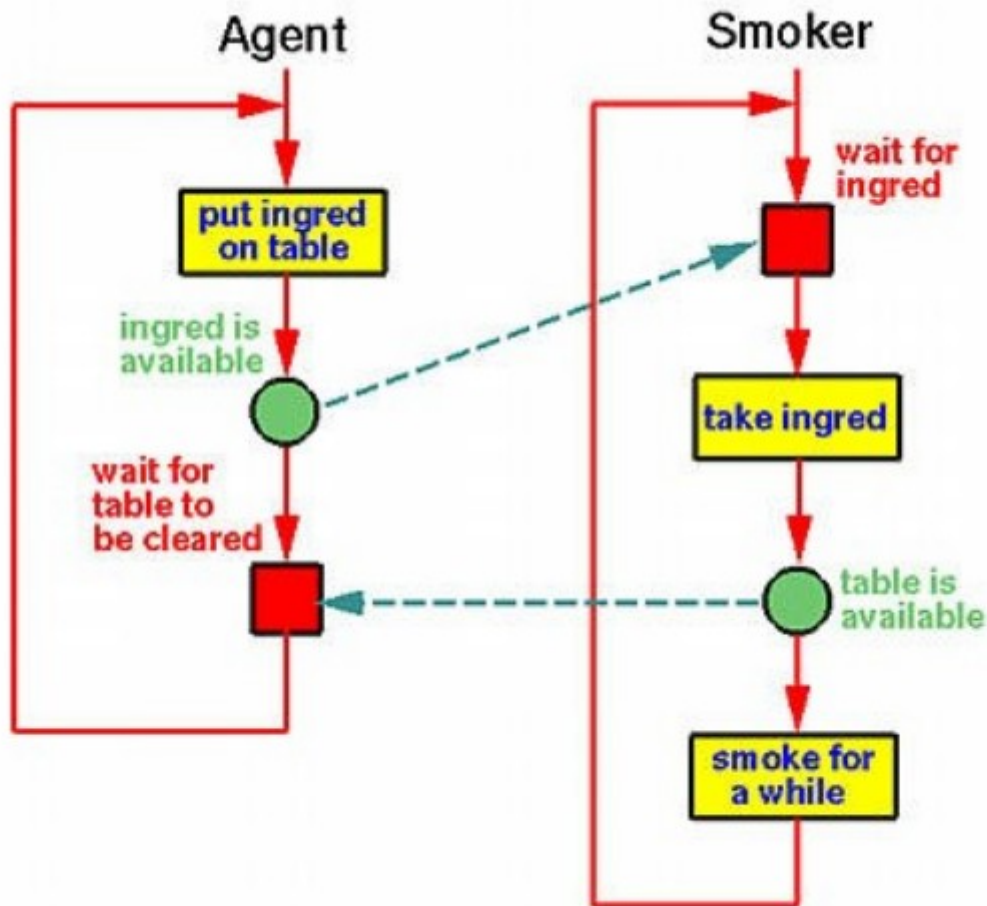
Wholesaler puts the ingredients to a table. This table in my implementation is a struct and this struct contains a character array to hold them. Also this struct is placed on shared memory to provide the accessibility of the ingredients. The struct is shown below.

```
typedef struct ingredients{
    char arr[2];
    int result;
    int checkChefVal;
}info;
```

The struct locating code sequence on to shared memory is show below:

```
int I_shm;
I_shm = shm_open("I_shrd", O_CREAT | O_RDWR, 0666);
ftruncate(I_shm, sizeof(info));
info* I_ptr = (info*)mmap(0, sizeof(info), PROT_WRITE, MAP_SHARED, I_shm, 0);
I_ptr->result=1;
```

For solving this problem I used the cigarette smokers problem solution. Chefs waiting for necessary ingredients so this prevents the busy waiting. When ingredients are ready child process that can use those ingredients wakes up and makes the g  lla  . Representation of the algorithm is shown below.(Taken from class materials)



The pusher process is like a bridge between the wholesaler and chefs. Wholesaler puts the ingredients on to table and wakes pusher process then pusher process wakes the chef to use it. Pusher process uses the functions shown below:

Named semaphore pusher:

```

void pusher(){
    sem_t smp = sem_open("smp",0);
    sem_t sem2 = sem_open("sem2",0);

    int shm_fd;
    shm_fd = shm_open("I_shrd", 0_RDWR, 0666);
    ing_table* ptr = (ing_table*)mmap(0, sizeof(ing_table), PROT_WRITE, MAP_SHARED, shm_fd, 0);

    while(ptr->final==1){
        sem_wait(&smp);
        sem_post(&sem2);
    }
    close(shm_fd);
}

```

Unnamed semaphore pusher:

```
void pusher(){
    int shm_fd;
    shm_fd = shm_open("I_shrd", O_RDWR, 0666);
    ing_table* ptr = (ing_table*)mmap(0, sizeof(ing_table), PROT_WRITE, MAP_SHARED, shm_fd, 0);

    while(ptr->final==1){
        sem_wait(sem1);
        sem_post(sem2);
    }
    close(shm_fd);
}
```

In this homework we are expected to implement two versions of the program. One is used unnamed semaphores for synchronization the other is used named semaphores for synchronization.

Unnamed Semaphore

When using unnamed semaphores for synchronization I placed the semaphores on to shared memory so that child processes(chefs) and parent process(wholesaler) can share the semaphore. Initializing and placing semaphores section is shown below:

```
sem1 = mmap(NULL, sizeof(sem1), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
sem2 = mmap(NULL, sizeof(sem2), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
sem3 = mmap(NULL, sizeof(sem3), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
semp = mmap(NULL, sizeof(semp), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
if ( sem_init(&sem1, 1, 0) < 0) {
    perror("sem_init");
    exit(EXIT_FAILURE);
}
if ( sem_init(&sem2, 1, 0) < 0) {
    perror("sem_init");
    exit(EXIT_FAILURE);
}
if ( sem_init(&sem3, 1, 0) < 0) {
    perror("sem_init");
    exit(EXIT_FAILURE);
}
if ( sem_init(&semp, 1, 0) < 0) {
    perror("sem_init");
    exit(EXIT_FAILURE);
}
```

After the execution and all ingredients sent to chefs and wholesaler sold all the desserts semaphores are destroyed.

```
sem_destroy(&sem1);
sem_destroy(&sem2);
sem_destroy(&sem3);
sem_destroy(&semp);
```

Named Semaphore

When using named semaphores for synchronization it is not needed to place semaphores on to shared memory because with using the name of semaphores child processes can reach them. Initialization of the named semaphores are shown below:

```
sem1 = sem_open(named sem, 0_CREAT, 0660, 0);  
sem2 = sem_open("sem2", 0_CREAT, 0660, 0);  
sem3 = sem_open("sem3", 0_CREAT, 0660, 0);  
semp = sem_open("semp", 0_CREAT, 0660, 0);
```

In chef processes first semaphores opened with `sem_open` with their names. First semaphore's name is taken from the user by `-n` option.

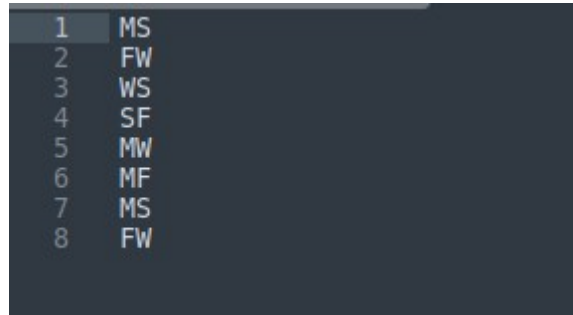
After the execution semaphores are unlinked:

```
sem_unlink(named sem);  
sem_unlink("sem2");  
sem_unlink("sem3");  
sem_unlink("sem4");
```

TEST

Named Version Test

Test file is shown below:



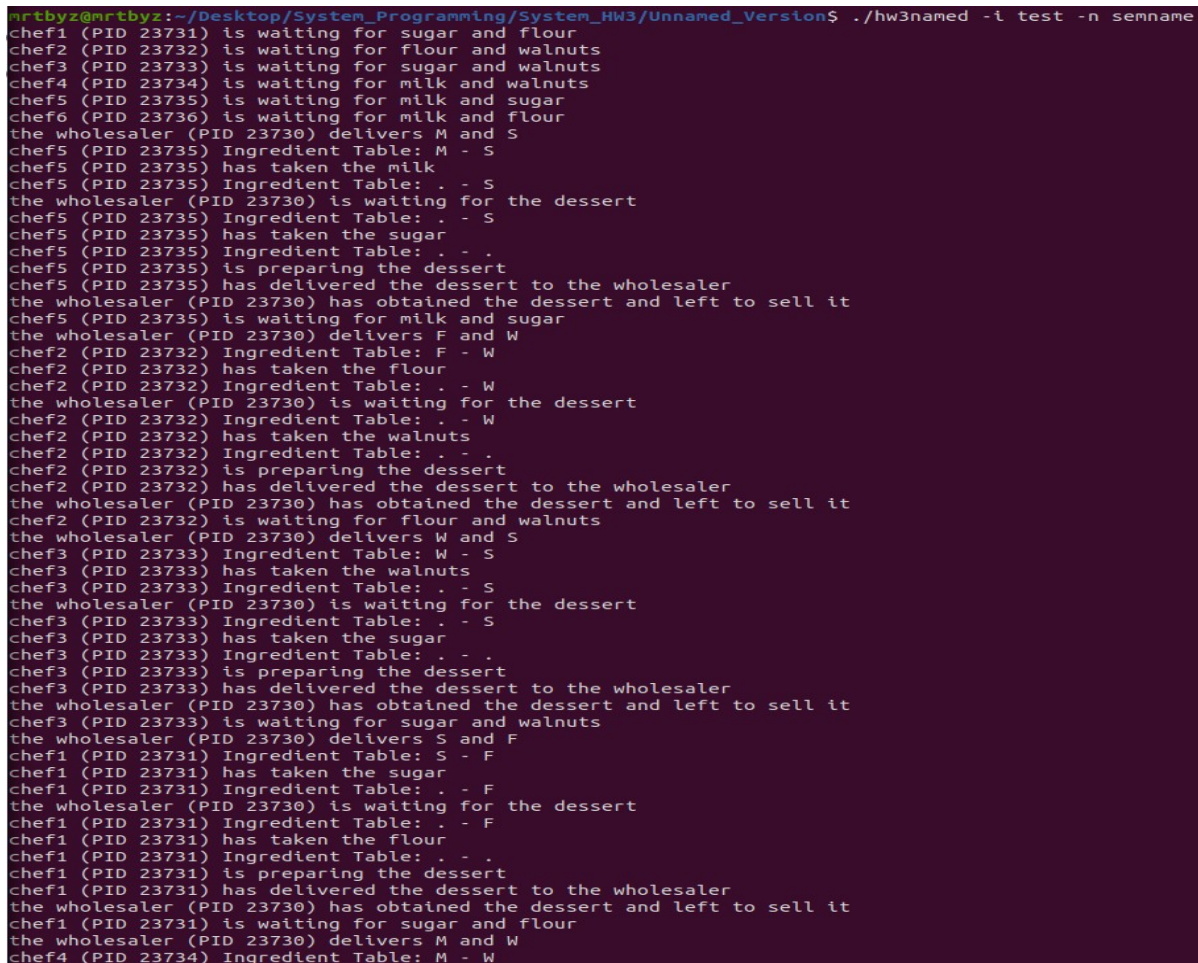
1	MS
2	FW
3	WS
4	SF
5	MW
6	MF
7	MS
8	FW

Execution command:



```
mrtbyz@mrtbyz:~/Desktop/System_Programming/System_HW3/Unnamed_Version$ ./hw3named -i test -n semname
```

Output of the program:



```
mrtbyz@mrtbyz:~/Desktop/System_Programming/System_HW3/Unnamed_Version$ ./hw3named -i test -n semname
chef1 (PID 23731) is waiting for sugar and flour
chef2 (PID 23732) is waiting for flour and walnuts
chef3 (PID 23733) is waiting for sugar and walnuts
chef4 (PID 23734) is waiting for milk and walnuts
chef5 (PID 23735) is waiting for milk and sugar
chef6 (PID 23736) is waiting for milk and flour
the wholesaler (PID 23730) delivers M and S
chef5 (PID 23735) Ingredient Table: M - S
chef5 (PID 23735) has taken the milk
chef5 (PID 23735) Ingredient Table: . - S
the wholesaler (PID 23730) is waiting for the dessert
chef5 (PID 23735) Ingredient Table: . - S
chef5 (PID 23735) has taken the sugar
chef5 (PID 23735) Ingredient Table: . - .
chef5 (PID 23735) is preparing the dessert
chef5 (PID 23735) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef5 (PID 23735) is waiting for milk and sugar
the wholesaler (PID 23730) delivers F and W
chef2 (PID 23732) Ingredient Table: F - W
chef2 (PID 23732) has taken the flour
chef2 (PID 23732) Ingredient Table: . - W
the wholesaler (PID 23730) is waiting for the dessert
chef2 (PID 23732) Ingredient Table: . - W
chef2 (PID 23732) has taken the walnuts
chef2 (PID 23732) Ingredient Table: . - .
chef2 (PID 23732) is preparing the dessert
chef2 (PID 23732) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef2 (PID 23732) is waiting for flour and walnuts
the wholesaler (PID 23730) delivers W and S
chef3 (PID 23733) Ingredient Table: W - S
chef3 (PID 23733) has taken the walnuts
chef3 (PID 23733) Ingredient Table: . - S
the wholesaler (PID 23730) is waiting for the dessert
chef3 (PID 23733) Ingredient Table: . - S
chef3 (PID 23733) has taken the sugar
chef3 (PID 23733) Ingredient Table: . - .
chef3 (PID 23733) is preparing the dessert
chef3 (PID 23733) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef3 (PID 23733) is waiting for sugar and walnuts
the wholesaler (PID 23730) delivers S and F
chef1 (PID 23731) Ingredient Table: S - F
chef1 (PID 23731) has taken the sugar
chef1 (PID 23731) Ingredient Table: . - F
the wholesaler (PID 23730) is waiting for the dessert
chef1 (PID 23731) Ingredient Table: . - F
chef1 (PID 23731) has taken the flour
chef1 (PID 23731) Ingredient Table: . - .
chef1 (PID 23731) is preparing the dessert
chef1 (PID 23731) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef1 (PID 23731) is waiting for sugar and flour
the wholesaler (PID 23730) delivers M and W
chef4 (PID 23734) Ingredient Table: M - W
```



```

chef4 (PID 23734) has taken the milk
chef4 (PID 23734) Ingredient Table: . - W
the wholesaler (PID 23730) is waiting for the dessert
chef4 (PID 23734) Ingredient Table: . - W
chef4 (PID 23734) has taken the walnuts
chef4 (PID 23734) Ingredient Table: . - .
chef4 (PID 23734) is preparing the dessert
chef4 (PID 23734) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef4 (PID 23734) is waiting for milk and walnuts
the wholesaler (PID 23730) delivers M and F
chef6 (PID 23736) Ingredient Table: M - F
chef6 (PID 23736) has taken the milk
chef6 (PID 23736) Ingredient Table: . - F
the wholesaler (PID 23730) is waiting for the dessert
chef6 (PID 23736) Ingredient Table: . - F
chef6 (PID 23736) has taken the flour
chef6 (PID 23736) Ingredient Table: . - .
chef6 (PID 23736) is preparing the dessert
chef6 (PID 23736) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef6 (PID 23736) is waiting for milk and flour
the wholesaler (PID 23730) delivers M and S
chef5 (PID 23735) Ingredient Table: M - S
chef5 (PID 23735) has taken the milk
chef5 (PID 23735) Ingredient Table: . - S
the wholesaler (PID 23730) is waiting for the dessert
chef5 (PID 23735) Ingredient Table: . - S
chef5 (PID 23735) has taken the sugar
chef5 (PID 23735) Ingredient Table: . - .
chef5 (PID 23735) is preparing the dessert
chef5 (PID 23735) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef5 (PID 23735) is waiting for milk and sugar
the wholesaler (PID 23730) delivers F and W
chef2 (PID 23732) Ingredient Table: F - W
chef2 (PID 23732) has taken the flour
chef2 (PID 23732) Ingredient Table: . - W
the wholesaler (PID 23730) is waiting for the dessert
chef2 (PID 23732) Ingredient Table: . - W
chef2 (PID 23732) has taken the walnuts
chef2 (PID 23732) Ingredient Table: . - .
chef2 (PID 23732) is preparing the dessert
chef2 (PID 23732) has delivered the dessert to the wholesaler
the wholesaler (PID 23730) has obtained the dessert and left to sell it
chef2 (PID 23732) is waiting for flour and walnuts
chef5 (PID 23735) is exiting
chef6 (PID 23736) is exiting
chef1 (PID 23731) is exiting
chef4 (PID 23734) is exiting
chef3 (PID 23733) is exiting
chef2 (PID 23732) is exiting
the wholesaler (PID 23730) is done (total desserts: 8)
mrtbyz@mrtbyz:~/Desktop/System_Programming/System_HW3/Unnamed_Version$

```

As you can see ingredient table is printed before and after the chef gets the ingredients. At the end all chef processes exited and wholesaler exited via printing total desserts.

Unnamed Version

Test file is shown below:

```
1 MS
2 FW
3 WS
4 SF
5 MW
6 MF
7 MS
8 FW
9 MS
10 FW
11 WS
12 SF
13 MW
14 MF
15 MS
16 FW
17 MS
18 FW
19 WS
20 SF
21 MW
22 MF
23 MS
24 FW
25 MS
26 FW
27 WS
28 SF
29 MW
30 MF
31 MS
32 FW
33 MS
34 FW
35 WS
36 SF
37 MW
38 MF
39 MS
40 FW
```

Execution command:

```
mrtybz@mrtybz:~/Desktop/System_Programming/System_HW3/Realy_Unnamed$ ./hw3unnamed -i test
```


Output of the program:

```
chef4 (PID 17704) Ingredient Table: M - W
chef4 (PID 17704) has taken the milk
chef4 (PID 17704) Ingredient Table: . - W
the wholesaler (PID 17700) is waiting for the dessert
chef4 (PID 17704) Ingredient Table: . - W
chef4 (PID 17704) has taken the walnuts
chef4 (PID 17704) Ingredient Table: . - .
chef4 (PID 17704) is preparing the dessert
chef4 (PID 17704) has delivered the dessert to the wholesaler
the wholesaler (PID 17700) has obtained the dessert and left to sell it
chef4 (PID 17704) is waiting for milk and walnuts
the wholesaler (PID 17700) delivers M and F
chef6 (PID 17706) Ingredient Table: M - F
chef6 (PID 17706) has taken the milk
chef6 (PID 17706) Ingredient Table: . - F
the wholesaler (PID 17700) is waiting for the dessert
chef6 (PID 17706) Ingredient Table: . - F
chef6 (PID 17706) has taken the flour
chef6 (PID 17706) Ingredient Table: . - .
chef6 (PID 17706) is preparing the dessert
chef6 (PID 17706) has delivered the dessert to the wholesaler
the wholesaler (PID 17700) has obtained the dessert and left to sell it
chef6 (PID 17706) is waiting for milk and flour
the wholesaler (PID 17700) delivers M and S
chef5 (PID 17705) Ingredient Table: M - S
chef5 (PID 17705) has taken the milk
chef5 (PID 17705) Ingredient Table: . - S
the wholesaler (PID 17700) is waiting for the dessert
chef5 (PID 17705) Ingredient Table: . - S
chef5 (PID 17705) has taken the sugar
chef5 (PID 17705) Ingredient Table: . - .
chef5 (PID 17705) is preparing the dessert
chef5 (PID 17705) has delivered the dessert to the wholesaler
the wholesaler (PID 17700) has obtained the dessert and left to sell it
chef5 (PID 17705) is waiting for milk and sugar
the wholesaler (PID 17700) delivers F and W
chef2 (PID 17702) Ingredient Table: F - W
chef2 (PID 17702) has taken the flour
chef2 (PID 17702) Ingredient Table: . - W
the wholesaler (PID 17700) is waiting for the dessert
chef2 (PID 17702) Ingredient Table: . - W
chef2 (PID 17702) has taken the walnuts
chef2 (PID 17702) Ingredient Table: . - .
chef2 (PID 17702) is preparing the dessert
chef2 (PID 17702) has delivered the dessert to the wholesaler
the wholesaler (PID 17700) has obtained the dessert and left to sell it
chef2 (PID 17702) is waiting for flour and walnuts
chef5 (PID 17705) is exiting
chef1 (PID 17701) is exiting
chef4 (PID 17704) is exiting
chef6 (PID 17706) is exiting
chef3 (PID 17703) is exiting
chef2 (PID 17702) is exiting
the wholesaler (PID 17700) is done (total desserts: 40)
mrtbyz@mrtbyz:~/Desktop/System_Programming/System_HW3/Realy_Unnamed$
```

As you can see ingredient table is printed before and after the chef gets the ingredients. At the end all chef processes exited and wholesaler exited via printing total desserts.