# CSE 344 Systems Programming HW3 Report

Yakup Talha Yolcu
1801042609

UNNAMED IMPLEMENTATION

```
20  #define MAX_BLKSIZE 1024
21  #define NUMBER_OF_CHEF 6
22  #define ENDLESS_SUPPLY 2
23  #define TOTAL_INGREDIENTS 4
24  #define TOTAL_CHILD (NUMBER_OF_CHEF + TOTAL_INGREDIENTS)
25  #define MEMORY_NAME "memory_name"
26
27  typedef struct shared_memory {
28      char queue[2];
29      sem_t agentSem;
30      sem_t sugar;
31      sem_t flour;
32      sem_t milk;
33      sem_t wallnut;
34      sem_t m;
35      sem_t sf;
36      sem_t sm;
37      sem_t sw;
38      sem_t fm;
39      sem_t fw;
40      sem_t mw;
41      int ismilk;
42      int issugar;
43      int isflour;
44      int iswallnut;
45
46  }shared_memory;
```

I store the read ingredients in queue char array with length of 2
Then I have agentsem semaphore to ensure sync of chef and wholesaler. I will explain it briefly in coming lines.

I have sugar,flour,milk and wallnut semaphores as ingredients. This semaphores is waited in the pusher processes. They are posted in wholesaler process. Whenever wholesaler sees respective ingredient, it posts respective semaphore.

Semaphore m is mutex to lock shared memory area which is ismilk,issugar,isflour,iswallnut. I have 4 pushers. Each pusher waits on an ingredient. If it passes on that wait, then it gets the lock by waiting m and it checks is any previous pusher worked or not with this int flags. If passed, it posts the one of sf,sm,sw,fm,fw,mw semaphores which chef already has. I implemented the similar algorithm in smokers problem.

```
pusherA             pusherB             pusherC             pusherD
wait milk           wait flour          wait sugar          wait wallnut

is sugar            is sugar            is milk             is flour
    post fw             post mw             post fw             post ms
    issugar=false       issugar=false       ismilk=false        isflour=false

is flour            is milk             is flour            is milk
    post ws             post ws             post wm             post sf
    isflour=false       ismilk=false        isflour=false       ismilk=false

is wallnut          is wallnut          is wallnut          is sugar
    post fs             post ms             post mf             post fm
    iswallnut=false     iswallnut=false     iswallnut=false     issugar=false

else                else                else                else
    ismilk=true         isflour=true        issugar=true        iswallnut=true


chef0           chef1           chef2           chef3           chef4           chef5
wait mf         wait ms         wait mw         wait sw         wait sf         wait fw

wholesaler
post flour
post milk
```

Pseudocode of my algorithm

After wholesaler delivers ingredient to the pusher, it waits on agentsem semaphore to get the dessert. In the chef side, after chef passes on respective semaphore, it makes dessert ready and posts the agentsem semaphore. Then wholesaler takes that dessert and continute its execution.

I have total of 11 process.
1 process => main process
4 process => pusher processes
6 process => chef processes

After file read is done, I send SIGUSR1 signal to the all childs, then I wait all childs with waitpid. Then I collect the chefs return values.

```c
int wait_for_child(int childpid) {
    int status=0;
    if(waitpid(childpid,&status,0)==-1) {
        perror("Error waiting child");
        return -1;
    }
    int return_value = WEXITSTATUS(status);
    return return_value;
}
```

In pushers and chefs after every wait, I have signal check.

input.txt

```
MS
FM
WS
SM
FW
SF
MF
WM
WF
WS
```

```
the wholesaler (pid 4385) has obtained the dessert and left
chef0 (pid 7892) is waiting for wallnut and sugar => array:
chef0 (pid 7892) is exiting with desert 2 => array:
chef1 (pid 7893) is exiting with desert 2 => array:
chef2 (pid 7894) is exiting with desert 1 => array:
chef4 (pid 7896) is exiting with desert 1 => array:
chef5 (pid 7897) is exiting with desert 2 => array:
chef3 (pid 7895) is exiting with desert 2 => array:
the wholesaler (pid 7887) is done (total desserts: 10)
talha@Talha:~/Masaüstü/CSE344 System Prog/HW3/src/unnamed$
```

Sample input and output for unnamed

input.txt

```
MS
FM
WS
SM
```

```
chef5 (pid 9763) is waiting for milk and sugar => array:
chef0 (pid 9758) is exiting with desert 1 => array:
chef1 (pid 9759) is exiting with desert 0 => array:
chef4 (pid 9762) is exiting with desert 0 => array:
chef3 (pid 9761) is exiting with desert 1 => array:
chef5 (pid 9763) is exiting with desert 2 => array:
chef2 (pid 9760) is exiting with desert 0 => array:
the wholesaler (pid 9753) is done (total desserts: 4)
talha@Talha:~/Masaüstü/CSE344 System Prog/HW3/src/named$
```

Sample input and output for named
My failures: In unnamed one, I have this problem: When run it valgrind, it is blocked in one semaphore.

NAMED IMPLEMENTATION

```
sem_t* agentSem;
sem_t* sugar;
sem_t* flour;
sem_t* milk;
sem_t* wallnut;
sem_t* m;
sem_t* sf;
sem_t* sm;
sem_t* sw;
sem_t* fm;
sem_t* fw;
sem_t* mw;
int number_of_desserts;
int terminate_flag=0;
```

This time they are in global scale, but other all implementation is same