GEBZE TECHNICAL UNIVERSITY

CSE344
SYSTEM PROGRAMMING COURSE

HOMEWORK 3
REPORT

BARAN SOLMAZ
1801042601

Problem Defination:

Implement a program as the chefs and the wholesaler in the form of 6+1 processes that print on screen their activities.

Implement two version of the program, one solving it with semaphores and another solving it with unnamed semaphores.

Problem Solution Approach:

Unnamed Semaphores: Defined in shared memory

```
};
struct Shared{
    char ingred[2];
    sem_t saler;
    sem_t sent;
    sem_t needs_sems[6];
} shared;
```

```
struct Shared temp;
for (int i = 0; i <6; i++)
    sem_init(&(temp.needs_sems[i]), 1,0);

sem_init(&(temp.saler), 1, 1);
sem_init(&(temp.sent), 1, 0);
memset(temp.ingred,'\0',2);
```

```
sem_post(&(shr_ptr->needs_sems[0]));
se if (('F'== shr_ptr->ingred[0] && 'W
sem_post(&(shr_ptr->needs_sems[1]));
se if (('F' == shr_ptr->ingred[0] && '
sem_post(&(shr_ptr->needs_sems[2]));
se if (('F' == shr_ptr->ingred[0] && '
sem_post(&(shr_ptr->needs_sems[3]));
se if (('M' == shr_ptr->ingred[0] && '
sem_post(&(shr_ptr->needs_sems[4]));
se
sem_post(&(shr_ptr->needs_sems[5]));
```

```
memory_read(int index,char needs[2]){
sem_wait(&(shr_ptr->needs_sems[index]));
if (sig_check2())
```

```
printf("The Wholesaler (pid
sem_wait(&(shr_ptr->sent));
printf("The Wholesaler(pid %
```

```
sem_wait(&(shr_ptr->saler));
```

Named Semaphores: Defined as global variables

```
struct Shared{
    char ingred[2];
} shared;

char *inputFile;
char *name;
char name1[6][50];
char name2[50];
sem_t *needs_sems[6];
sem_t *saler;
sem_t *sent;
```

```
struct Shared temp;
for (int i = 0; i < 6; i++)
    needs_sems[i] = sem_open(name1[i], O_CREAT | O_EXCL, 0777, 0);

saler = sem_open(name, O_CREAT | O_EXCL, 0777, 1);
sent = sem_open(name2, O_CREAT | O_EXCL, 0777, 0);
memset(temp.ingred, '\0', 2);
```

```
sem_wait(saler);
```

```
sem_wait(sent);
```

```
sem_wait(needs_sems[index]);
```

```
sem_post(needs_sems[0]);
se if (('F' == shr_ptr->ing
sem_post(needs_sems[1]);
se if (('F' == shr_ptr->ing
sem_post(needs_sems[2]);
se if (('F' == shr_ptr->ing
sem_post(needs_sems[3]);
se if (('M' == shr_ptr->ing
sem_post(needs_sems[4]);
se
sem_post(needs_sems[5]);
```

Unnamed/WholeSaler:

```c
void wholeSaler(){
    sleep(1);
    int fd = openFile(inputFile,O_RDONLY);
    while (1){
        char buffer[3];
        memset(buffer,'\0',3);
        sig_check();
        if (read(fd, buffer, 3) != 3){
            for (int i = 0; i < 6; i++){
                if (chefs[i].id!= -1)
                    kill(chefs[i].id,SIGINT);
            }
            totalDesserts=wait_child(6);
            printf("The Wholesaler (pid %d) is done. (Total Desserts : %d)\n",(int)getpid(),totalDesserts);
            break;
        }
        sig_check();
        memory_write(buffer);
    }
    closeFile(fd);
}
```

If can't read , sends signal to chefs to finish

```c
void memory_write(char ing[3]){
    strncpy(shr_ptr->ingred, ing, 2);
    sem_wait(&(shr_ptr->saler));
    printf("The Wholesaler (pid %d) delivers %s and %s\n",(int)getpid(),ingrediants(ing[0]),ingrediants(ing[1]));
    pusher1();
    printf("The Wholesaler (pid %d) is waiting for the dessert\n", (int)getpid());
    sem_wait(&(shr_ptr->sent));
    printf("The Wholesaler(pid %d) has obtained the dessert and left\n", (int)getpid());
}
```

Posts the pair

```c
void pusher1(){
    if (('W' == shr_ptr->ingred[0] && 'S' == shr_ptr->ingred[1]) || ('W' == shr_ptr->ingred[1] && 'S' == shr_ptr->ingred[0])){
        sem_post(&(shr_ptr->needs_sems[0]));
    }else if (('F'== shr_ptr->ingred[0] && 'W' == shr_ptr->ingred[1]) || ('F'== shr_ptr->ingred[1] && 'W' == shr_ptr->ingred[0])){
        sem_post(&(shr_ptr->needs_sems[1]));
    }else if (('F' == shr_ptr->ingred[0] && 'S' == shr_ptr->ingred[1]) || ('F' == shr_ptr->ingred[1] && 'S' == shr_ptr->ingred[0])){
        sem_post(&(shr_ptr->needs_sems[2]));
    }else if (('F' == shr_ptr->ingred[0] && 'M' == shr_ptr->ingred[1]) || ('F' == shr_ptr->ingred[1] && 'M' == shr_ptr->ingred[0])){
        sem_post(&(shr_ptr->needs_sems[3]));
    }else if (('M' == shr_ptr->ingred[0] && 'W' == shr_ptr->ingred[1]) || ('M' == shr_ptr->ingred[1] && 'W' == shr_ptr->ingred[0])){
        sem_post(&(shr_ptr->needs_sems[4]));
    }else
        sem_post(&(shr_ptr->needs_sems[5]));
}
```

shr_ptr->needs_sems[0] for Walnuts and Sugar pairs
shr_ptr->needs_sems[1] for Flour and Walnuts pairs
shr_ptr->needs_sems[2] for Flour and Sugar pairs
shr_ptr->needs_sems[3] for Flour and Milk pairs
shr_ptr->needs_sems[4] for Milk and Walnuts pairs
shr_ptr->needs_sems[5] for Sugar and Milk pairs

Unnamed/Chefs:

Every chef waits for their missing ingredient pairs.

```c
void chef(int chefIndex, char needs[2]){
    int counter=0;
    printf("Chef%d (pid %d) is waiting for %s and %s\n", chefIndex,
        (int)getpid(), ingrediants(needs[0]), ingrediants(needs[1]));
    sleep(1);
    while (1){
        if (sig_check2())
            break;
        int i=memory_read(chefIndex,needs);

        if (i != 0){
            counter++;
            printf("Chef%d (pid %d) has delivered the dessert\t Ingredients: %c%c\n",
                chefIndex, (int)getpid(), shr_ptr->ingred[0], shr_ptr->ingred[1]);
        }
        if (sig_check2())
            break;
        sem_post(&(shr_ptr->saler));
        if (sig_check2())
            break;
    }
    printf("Chef%d (pid %d) is exiting\t Prepared Desserts : %d\n", chefIndex, (int)getpid(),counter);
    //sleep(1);
    _exit(counter);
}
```

```c
int memory_read(int index,char needs[2]){
    sem_wait(&(shr_ptr->needs_sems[index]));
    if (sig_check2())
        return 0;
    printf("Chef%d (pid %d) has taken the %s\t Ingredients: %c%c\n",
        index, (int)getpid(), ingrediants(shr_ptr->ingred[0]), shr_ptr->ingred[0], shr_ptr->ingred[1]);
    printf("Chef%d (pid %d) has taken the %s\t Ingredients: %c%c\n",
        index, (int)getpid(), ingrediants(shr_ptr->ingred[1]), shr_ptr->ingred[0], shr_ptr->ingred[1]);
    memset(shr_ptr->ingred, '\0', 2);
    printf("Chef%d (pid %d) is preparing the dessert\t Ingredients: %c%c\n",
        index, (int)getpid(), shr_ptr->ingred[0], shr_ptr->ingred[1]);
    if (sig_check2())
        return 0;
    sem_post(&(shr_ptr->sent));
    if (sig_check2())
        return 0;
    return 1;
}
```

Waits for missing Ingredients

Inital Values:

Saler : 1 -> Wholesaler starts first,

Sent : 0 -> To wait chef for dessert, if 1 ,wholesaler gets dessert,

All ingredients pairs : 0 -> In the beginning there is no ingredient for chefs from wholesaler, everytime wholesaler brings ingredients, its pair becomes 1.

Tests:

Inputs:

```
1  MS
2  FM
3  WS
4  SM
5
```

PS: input file needs an extra line because I read 3 characters every time.

Outputs:

```
Chef0 (pid 155260) is waiting for WALNUTS and SUGAR
Chef1 (pid 155261) is waiting for FLOUR and WALNUTS
Chef3 (pid 155263) is waiting for MILK and FLOUR
Chef4 (pid 155264) is waiting for MILK and WALNUTS
Chef5 (pid 155265) is waiting for SUGAR and MILK
Chef2 (pid 155262) is waiting for SUGAR and FLOUR
The Wholesaler (pid 155259) delivers MILK and SUGAR
The Wholesaler (pid 155259) is waiting for the dessert
Chef5 (pid 155265) has taken the MILK    Ingredients: MS
Chef5 (pid 155265) has taken the SUGAR   Ingredients: MS
Chef5 (pid 155265) is preparing the dessert      Ingredients:
Chef5 (pid 155265) has delivered the dessert     Ingredients:
The Wholesaler(pid 155259) has obtained the dessert and left
The Wholesaler (pid 155259) delivers FLOUR and MILK
The Wholesaler (pid 155259) is waiting for the dessert
Chef3 (pid 155263) has taken the FLOUR   Ingredients: FM
Chef3 (pid 155263) has taken the MILK    Ingredients: FM
Chef3 (pid 155263) is preparing the dessert      Ingredients:
Chef3 (pid 155263) has delivered the dessert     Ingredients:
The Wholesaler(pid 155259) has obtained the dessert and left
```

```
The Wholesaler (pid 155259) delivers WALNUTS and SUGAR
The Wholesaler (pid 155259) is waiting for the dessert
Chef0 (pid 155260) has taken the WALNUTS         Ingredients: WS
Chef0 (pid 155260) has taken the SUGAR   Ingredients: WS
Chef0 (pid 155260) is preparing the dessert      Ingredients:
Chef0 (pid 155260) has delivered the dessert     Ingredients:
The Wholesaler(pid 155259) has obtained the dessert and left
The Wholesaler (pid 155259) delivers SUGAR and MILK
The Wholesaler (pid 155259) is waiting for the dessert
Chef5 (pid 155265) has taken the SUGAR   Ingredients: SM
Chef5 (pid 155265) has taken the MILK    Ingredients: SM
Chef5 (pid 155265) is preparing the dessert      Ingredients:
Chef5 (pid 155265) has delivered the dessert     Ingredients:
The Wholesaler(pid 155259) has obtained the dessert and left
Chef0 (pid 155260) is exiting    Prepared Desserts : 1
Chef1 (pid 155261) is exiting    Prepared Desserts : 0
Chef5 (pid 155265) is exiting    Prepared Desserts : 2
Chef3 (pid 155263) is exiting    Prepared Desserts : 1
Chef4 (pid 155264) is exiting    Prepared Desserts : 0
Chef2 (pid 155262) is exiting    Prepared Desserts : 0
The Wholesaler (pid 155259) is done. (Total Desserts : 4)
```