

# Rendu Séance 21/04

## Producteur

prod.c

```
1 #include <stdio.h>
2 #include <sys/ipc.h>
3 #include <sys/shm.h>
4 #include <stdlib.h>
5
6 int main() {
7     // Générer la même clé que le producteur
8     key_t cle = ftok("test", 65);
9     if (cle == -1) {
10         perror("Erreur ftok");
11         exit(EXIT_FAILURE);
12     }
13
14     // Récupérer le segment existant
15     int shm_id = shmget(cle, 1024, 0666); // Pas de IPC_CREAT ici
16     if (shm_id < 0) {
17         perror("Erreur shmget (segment introuvable)");
18         exit(EXIT_FAILURE);
19     }
20     printf("Consommateur : mémoire partagée ID = %d\n", shm_id);
21
22     // Attachement
23     int *p = (int *) shmat(shm_id, NULL, 0);
24     if (p == (int *) -1) {
25         perror("Erreur shmat");
26         exit(EXIT_FAILURE);
27     }
28
29     // Lecture
30     for (int i = 0; i < 10; i++) {
31         printf("Consommateur a lu : %d\n", p[i]);
32     }
33
34     // Détachement
35     if (shmdt(p) == -1) {
36         perror("Erreur shmdt");
37         exit(EXIT_FAILURE);
38     }
39
40     return 0;
41 }
```

```

mahdi@mahdi-virtual-machine:~/Desktop/SE2/21_04$ gcc prod.c -o prod
mahdi@mahdi-virtual-machine:~/Desktop/SE2/21_04$ ./prod
Segment de mémoire partagée créé avec ID : 4
Écrit : 0
Écrit : 10
Écrit : 20
Écrit : 30
Écrit : 40
Écrit : 50
Écrit : 60
Écrit : 70
Écrit : 80
Écrit : 90

```

## Consommateur

```

1 #include <stdio.h>
2 #include <sys/ipc.h>
3 #include <sys/shm.h>
4 #include <stdlib.h>
5
6 int main() {
7     // Générer la même clé que le producteur
8     key_t cle = ftok("test", 65);
9     if (cle == -1) {
10         perror("Erreur ftok");
11         exit(EXIT_FAILURE);
12     }
13
14     // Récupérer le segment existant
15     int shm_id = shmget(cle, 1024, 0666); // Pas de IPC_CREAT ici
16     if (shm_id < 0) {
17         perror("Erreur shmget (segment introuvable)");
18         exit(EXIT_FAILURE);
19     }
20     printf("Consommateur : mémoire partagée ID = %d\n", shm_id);
21
22     // Attachement
23     int *p = (int *) shmat(shm_id, NULL, 0);
24     if (p == (int *) -1) {
25         perror("Erreur shmat");
26         exit(EXIT_FAILURE);
27     }
28
29     // Lecture
30     for (int i = 0; i < 10; i++) {
31         printf("Consommateur a lu : %d\n", p[i]);
32     }
33
34     // Détachement
35     if (shmdt(p) == -1) {
36         perror("Erreur shmdt");
37         exit(EXIT_FAILURE);
38     }
39
40     return 0;
41 }

```

```
mahdi@mahdi-virtual-machine: ~/Desktop/SE2/21_04
mahdi@mahdi-virtual-machine:~/Desktop/SE2/21_04$ gcc consom.c -o consom
mahdi@mahdi-virtual-machine:~/Desktop/SE2/21_04$ ./consom
Consommateur : mémoire partagée ID = 4
Consommateur a lu : 0
Consommateur a lu : 10
Consommateur a lu : 20
Consommateur a lu : 30
Consommateur a lu : 40
Consommateur a lu : 50
Consommateur a lu : 60
Consommateur a lu : 70
Consommateur a lu : 80
Consommateur a lu : 90
mahdi@mahdi-virtual-machine:~/Desktop/SE2/21_04$
```

## Exemple 2

```
Open  prod.c  Save
1 #include <stdio.h>
2 #include <sys/ipc.h>
3 #include <sys/shm.h>
4 #include <sys/types.h>
5 #include <stdlib.h>
6 #include <unistd.h>
7 #include <sys/wait.h>
8
9 int main() {
10     // Créer le segment de mémoire partagée
11     int shid = shmget(IPC_PRIVATE, sizeof(int), IPC_CREAT | 0666);
12     if (shid < 0) {
13         perror("Erreur lors de la création du segment de mémoire partagée");
14         exit(EXIT_FAILURE);
15     }
16
17     // Attacher le segment
18     int *p = (int *) shmat(shid, NULL, 0);
19     if (p == (int *) -1) {
20         perror("Erreur lors de l'attachement");
21         exit(EXIT_FAILURE);
22     }
23
24     // Initialiser la valeur
25     *p = 0;
26
27     // Créer 10 processus fils
28     for (int i = 0; i < 10; i++) {
29         pid_t pid = fork();
30         if (pid < 0) {
31             perror("Erreur lors du fork");
32             exit(EXIT_FAILURE);
33         }
34         if (pid == 0) { // Code du fils
35             for (int j = 0; j < 10; j++) {
36                 int temp = *p;
37                 temp++;
38                 *p = temp;
39             }
40             shmdt(p); // Le fils se détache
41             exit(0);
42         }
43     }
44
45     // Attendre tous les processus fils

```

```
$ gedit prod.c
$ gcc prod.c -o prod
$ ./prod
```

La valeur finale est : 100

