

	<b>Module : Système d'Exploitation et Programmation Système</b> <i>1ère Année Cycle Ingénieur</i> Mohamed BAKHOUYA, Abdelhak KHARBOUCH <b>Projets</b>	<b>Année Universitaire</b> 2019/2020
---	---	---

---

## Projet 5

### Un réseau de Petri

---



**Fait par : EL HANAFI Maha**

**Objectif :** Écrire le processus baigneur en ajoutant les synchronisations à l'aide de sémaphores. Nous pouvons assimiler les baigneurs à des processus concurrents; les cabines et les paniers sont les ressources partagées.

### Programmes et données :

\* Soit le programme C suivant : `Projet5_CodeSource.c`

```

1  #include <stdio.h>
2  #include <stdio.h>
3  #include <sys/types.h>
4  #include <sys/ipc.h>
5  #include <sys/sem.h>
6  #include <unistd.h>
7  #include <stdlib.h>
8  #include <semaphore.h>
9  #define SKEY (key_t) IPC_PRIVATE
10 #define IFLAGS (SEMPERM | IPC_CREAT)
11 #define SEMPERM 0600
12
13
14 int NbPaniersRestant;
15 int NbCabinesRestant;
16
17 int j;
18 int semid;
19
20 struct sembuf sem_oper_P ;
21 struct sembuf sem_oper_V ;
22
23 int initsem(key_t semkey)
24 {
25
26     int status = 0;
27     int semid_init;
28     // Quand plusieurs sémaphores
29     union semun {
30         int val;
31         struct semid_ds *stat;
32         unsigned short * array;
33     } ctl_arg;
34
35     //creation des commentaires
36     if ((semid_init = semget(semkey, 2, IFLAGS)) > 0) {
37
38         unsigned short array[2] = {2, 3}; // Initialisation du nombre de cabines et du nombre de paniers
39         ctl_arg.array = array;
40         status = semctl(semid_init, 0, SETALL, ctl_arg); // initialisation en utilisant la structure ctl_arg
41     }
42
43     if (semid_init == -1 || status == -1) {
44         perror("Erreur initsem");
45         return (-1);
46     } else return (semid_init);
47 }
48
49 int P(int semnum){
50
51     union semun {
52         int val;
53         struct semid_ds *buf;
54         ushort * array;
55     } arg;
56
57     sem_oper_P.sem_num = semnum;
58     sem_oper_P.sem_op = -1;
59     sem_oper_P.sem_flg = 0 ;
60     semop(semid, &sem_oper_P , 1);
61
62
63     if (semnum == 1) // RAPPORT AU NOMBRE DE PANIERS
64     {
65         NbPaniersRestant = semctl(semid, semnum, GETVAL, arg);
66         // printf("NbPaniersRestant %d\n", NbPaniersRestant);

```

```

63     if (semnum == 1)    // RAPPORT AU NOMBRE DE PANIERS
64     {
65         NbPaniersRestant = semctl(semid, semnum, GETVAL, arg);
66         // printf("NbPaniersRestant %d\n", NbPaniersRestant);
67         if (NbPaniersRestant < 0)
68         {
69             printf("Plus de paniers pour les baigneurs\n");
70         }
71         else{
72             // printf("NbPaniers : %d \n", NbPaniersRestant);
73         }
74     }
75
76     if (semnum == 0)    // RAPPORT AU NOMBRE DE CABINES
77     {
78         NbCabinesRestant = semctl(semid, semnum, GETVAL, arg);
79         if (NbCabinesRestant < 0)
80         {
81             printf("Plus de cabines pour les baigneurs\n");
82         }
83         // printf("NbCabines : %d \n", NbCabinesRestant);
84     }
85
86     return 0;
87 }
88
89 /* OPERATION V */
90
91 int V(int semnum){
92
93     union semun {
94         int val;
95
96 int V(int semnum){
97
98     union semun {
99         int val;
100         struct semid_ds *buf;
101         ushort * array;
102     } arg;
103
104     sem_oper_P.sem_num = semnum;
105     sem_oper_P.sem_op = +1;
106     sem_oper_P.sem_flg = 0 ;
107     semop(semid, &sem_oper_P , 1);
108
109     if (semnum == 1)    // RAPPORT AU NOMBRE DE PANIERS
110     {
111         NbPaniersRestant = semctl(semid, semnum, GETVAL, arg);
112         if (NbPaniersRestant < 0)
113         {
114             printf("Plus de paniers pour les baigneurs\n");
115         }
116         else{
117             // printf("NbPaniers : %d \n", NbPaniersRestant);
118         }
119     }
120
121     if (semnum == 0)    // RAPPORT AU NOMBRE DE CABINES
122     {
123         NbCabinesRestant = semctl(semid, semnum, GETVAL, arg);
124         if (NbCabinesRestant < 0)
125         {
126             printf("Plus de cabines pour les baigneurs\n");
127         }
128         else{
129             // printf("NbCabines : %d \n", NbCabinesRestant);
130         }
131     }
132
133     return 0;
134 }

```

```

120     if (NbCabinesRestant < 0)
121     {
122         printf("Plus de cabines pour les baigneurs\n");
123     }
124     // printf("NbCabines : %d \n", NbCabinesRestant);
125 }
126
127 return 0;
128 }
129
130 void baigneur(int j){
131
132     /* LES BAIGNEURS ARRIVE */
133     printf("Le baigneur %d ARRIVE \n\n", j);
134
135     /* UN BAIGNEUR PREND UN PANIER */
136     printf("Le baigneur %d PREND un PANIER \n\n", j);
137     P(1);
138
139     /* UN BAIGNEUR PREND UNE CABINE */
140     printf("Le baigneur %d PREND une CABINE \n\n", j);
141     P(0);
142
143     /* SE CHANGER */
144     printf("Le baigneur %d SE CHANGE \n\n", j);
145     usleep(rand()%500);
146
147     /* UN BAIGNEUR SE BAIGNE */
148     printf("Le baigneur %d se baigne \n\n", j);
149     usleep(rand()%5000000);
150
151     /* UN BAIGNEUR REND LA CABINE */
152     printf("Le baigneur %d REND sa CABINE \n\n", j);
153
154     /* UN BAIGNEUR PREND UNE CABINE */
155     printf("Le baigneur %d PREND une CABINE \n\n", j);
156     P(0);
157
158     /* SE CHANGER */
159     printf("Le baigneur %d SE CHANGE \n\n", j);
160     usleep(rand()%500);
161
162     /* UN BAIGNEUR SE BAIGNE */
163     printf("Le baigneur %d se baigne \n\n", j);
164     usleep(rand()%5000000);
165
166     /* UN BAIGNEUR REND LA CABINE */
167     printf("Le baigneur %d REND sa CABINE \n\n", j);
168     V(0);
169
170     /* UN BAIGNEUR REND LE PANIER */
171     printf("Le baigneur %d REND SON PANIER \n\n", j);
172     V(1);
173
174     /* LES BAIGNEURS S'EN VONT */
175     printf("Le baigneur %d PART \n\n", j);
176
177     exit(0);
178 }
179
180 }

```

L'affichage de l'exécution du Projet5\_CodeSource.c

```
uir_student@ubuntu:~/Desktop/projetmaha/tests$ gedit projet5.c
uir_student@ubuntu:~/Desktop/projetmaha/tests$ gcc -o projet5 projet5.c
uir_student@ubuntu:~/Desktop/projetmaha/tests$ ./projet5
Le baigneur 1 ARRIVE
Le baigneur 0 ARRIVE
Le baigneur 1 PREND un PANIER
Le baigneur 0 PREND un PANIER
Le baigneur 1 PREND une CABINE
Le baigneur 0 PREND une CABINE
Le baigneur 1 SE CHANGE
Le baigneur 0 SE CHANGE
Le baigneur 2 ARRIVE
Le baigneur 2 REND sa CABINE
Le baigneur 2 REND SON PANIER
Le baigneur 2 PART
Le baigneur 3 SE CHANGE
Le baigneur 4 REND sa CABINE
Le baigneur 4 REND SON PANIER
Le baigneur 4 PART
Le baigneur 3 se baigne
Le baigneur 3 REND sa CABINE
Le baigneur 3 REND SON PANIER
Le baigneur 3 PART
uir_student@ubuntu:~/Desktop/projetmaha/tests$
```

**Sources :**

<http://www.singaomara.com/repCC++/ipcSystemV.html>