

Corrigé TD11. Concurrency

Exercice 4. Le barbier endormi

1.

```
1 int nb_places = N #places disponibles en salle d attente
2
3 barbier () :
4     while true :
5         si (nb_places < N) : #client present
6             nb_places = nb_places + 1
7             couper les cheveux
8
9 client () :
10     si nb_places > 0 :
11         nb_places = nb_places - 1
12         s asseoir en salle d attente
13         se faire couper les cheveux
```

2. Il faut protéger les accès concurrents à `nb_places` à l'aide d'un mutex :

```
1 int nb_places = N
2 mutex m
3
4 barbier () :
5     while true :
6         lock(m)
7         si (nb_places < N) : #client present
8             nb_places = nb_places + 1
9             unlock(m)
10            couper les cheveux
11        sinon :
12            unlock(m)
13 client () :
14     lock(m)
15     si nb_places > 0 :
16         nb_places = nb_places - 1
17         unlock(m)
18         s asseoir en salle d attente
19         se faire couper les cheveux
20     sinon :
21         unlock(m)
```

3. On introduit un sémaphore *c* qui est bloquant tant qu'un client n'est pas prêt.

```
1 int nb_places = N
2 mutex m
3 c = sem(0)
4
5 barbier () :
6     while true :
7         acquire(c) #attend un client prêt
8         lock(m)
9         si (nb_places < N) : #client present
10             nb_places = nb_places + 1
11             unlock(m)
12             couper les cheveux
13         sinon :
14             unlock(m)
15 client () :
16     lock(m)
17     si nb_places > 0 :
18         nb_places = nb_places - 1
19         unlock(m)
20         s asseoir en salle d attente
21         release(c) #indique qu il est prêt
22         se faire couper les cheveux
23     sinon :
24         unlock(m)
```

4. On introduit un nouveau sémaphore, *b*, qui est bloquant tant que le siège du barbier n'est pas disponible.

```
1 int nb_places = N ; mutex m ; c = sem(0) ; b = sem(0)
2
3 barbier () :
4     while true :
5         acquire(c) #attend un client prêt
6         lock(m)
7         si (nb_places < N) : #client present
8             nb_places = nb_places + 1
9             unlock(m)
10             release(b) # le barbier est prêt
11             couper les cheveux
12         sinon :
13             unlock(m)
14 client () :
15     lock(m)
16     si nb_places > 0 :
17         nb_places = nb_places - 1
18         unlock(m)
19         s asseoir en salle d attente
20         release(c) #indique qu il est prêt
21         acquire(b) #attend le barbier
22         se faire couper les cheveux
23     sinon :
24         unlock(m)
```