

TP 1 Bridge Problem

Problem

Un pont supporte une charge maximale de 15 tonnes. Ce pont est traversé par des camions dont le poids est de 15 tonnes ainsi que par des voitures dont le poids est de 5 tonnes. On vous demande de gérer l'accès au pont de sorte que la charge maximale du pont soit respectée.

Code

```
1  int random_spin(double max) {
2      int j = (int)(max * rand() / (RAND_MAX + 1.0));
3      if (j < 1)
4          return 1;
5      return j;
6  }
7
8  void waiting(double max) {
9      struct timespec delay;
10     delay.tv_sec = random_spin(max);
11     delay.tv_nsec = 0;
12     nanosleep(&delay, NULL);
13 }
```

```

1 void acces_to_bridge(int tons, int id) {
2     pthread_mutex_lock(&sc);
3     if (limit + tons) {
4         limit += tons;
5         state[id] = TRAVERSE;
6         sem_post(&sempriv[id]);
7     }
8     else {
9         state[id] = WAIT;
10        if (tons == 15)
11            number_of_blocked_trucks++;
12    }
13    pthread_mutex_unlock(&sc);
14    sem_wait(&sempriv[id]);
15 }
16
17 void liberate_bridge(int tonnes, int pid) {
18     int i;
19     pthread_mutex_lock(&sc);
20     state[pid] = IDLE;
21     limit -= tonnes;
22
23     for (i = 0; i < NB_TRUCKS; i++) {
24         if ((state[pid] == WAIT) && (limit == 0)) {
25             sem_post(&sempriv[i]);
26             limit = 15;
27             number_of_blocked_trucks--;
28         }
29     }
30     for (i = NB_TRUCKS; i < NB_VEHICULES; i++) {
31         if ((limit < 15) && (number_of_blocked_trucks == 0) && (state[i] == WAIT)) {
32             limit += 15;
33             sem_post(&sempriv[i]);
34         }
35     }
36     pthread_mutex_unlock(&sc);
37 }

```



```
1 void *truck(void *args) {
2     int pid = *((int *)args);
3     waiting(5.0);
4     acces_to_bridge(15, pid);
5     printf("Truck %d traversing the bridge \n", pid);
6     waiting(5.0);
7     printf("Truck %d leaving the bridge\n", pid);
8     liberate_bridge(15, pid);
9     pthread_exit(NULL);
10 }
11
12 void *car(void *args) {
13     int pid = *((int *)args);
14     waiting(5.0);
15     acces_to_bridge(5, pid);
16     printf("Car %d traversing the bridge \n", pid);
17     waiting(5.0);
18     printf("Car %d leaving the bridge\n", pid);
19     liberate_bridge(5, pid);
20     pthread_exit(NULL);
21 }
22
23 int main() {
24     int i;
25     pthread_t id;
26     for (i = 0; i < NB_VEHICULES; i++) {
27         state[i] = IDLE;
28         sem_init(&sempriv[i], 0, 0);
29     }
30     pthread_mutex_init(&sc, 0);
31     for (i = 0; i < NB_VEHICULES; i++) {
32         int *j = (int *)malloc(sizeof(int));
33         *j = i;
34         if (i < NB_TRUCKS)
35             pthread_create(&id, NULL, truck, j);
36         else
37             pthread_create(&id, NULL, car, j);
38     }
39     pthread_exit(NULL);
40     return 0;
41 }
```

Execution de code

```
wa101@wa101-latitude5490:~/Projects/INSAT/TP-STR-1 > gcc main.c -o main -pthread
~/Projects/INSAT/TP-STR-1 > ./main
Truck 1 traversing the bridge
Car 5 traversing the bridge
Car 6 traversing the bridge
Car 7 traversing the bridge
Car 6 leaving the bridge
Car 9 traversing the bridge
Truck 2 traversing the bridge
Truck 3 traversing the bridge
Car 8 traversing the bridge
Truck 1 leaving the bridge
Car 7 leaving the bridge
Truck 0 traversing the bridge
Truck 4 traversing the bridge
Car 5 leaving the bridge
Truck 0 leaving the bridge
Truck 3 leaving the bridge
Car 8 leaving the bridge
Car 9 leaving the bridge
Truck 2 leaving the bridge
Truck 4 leaving the bridge
~/Pr/INSAT/TP-STR-1 > 7s
```