

RAPPORT DE PROJET

GESTION D'UN PARKING

Réalisé par : TRABOULSI Rawan, Dumez Ange

Responsable UV: DESCAMPS Philippe

UV: SY40



SOMMAIRE :

- I. Introduction
- II. Présentation de la solution
- III. Réseau de Petri
- IV. Les choix de conception
- V. Conclusion

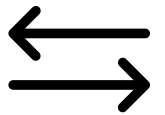
I. INTRODUCTION



Voiture abonnée



Voiture non abonnée



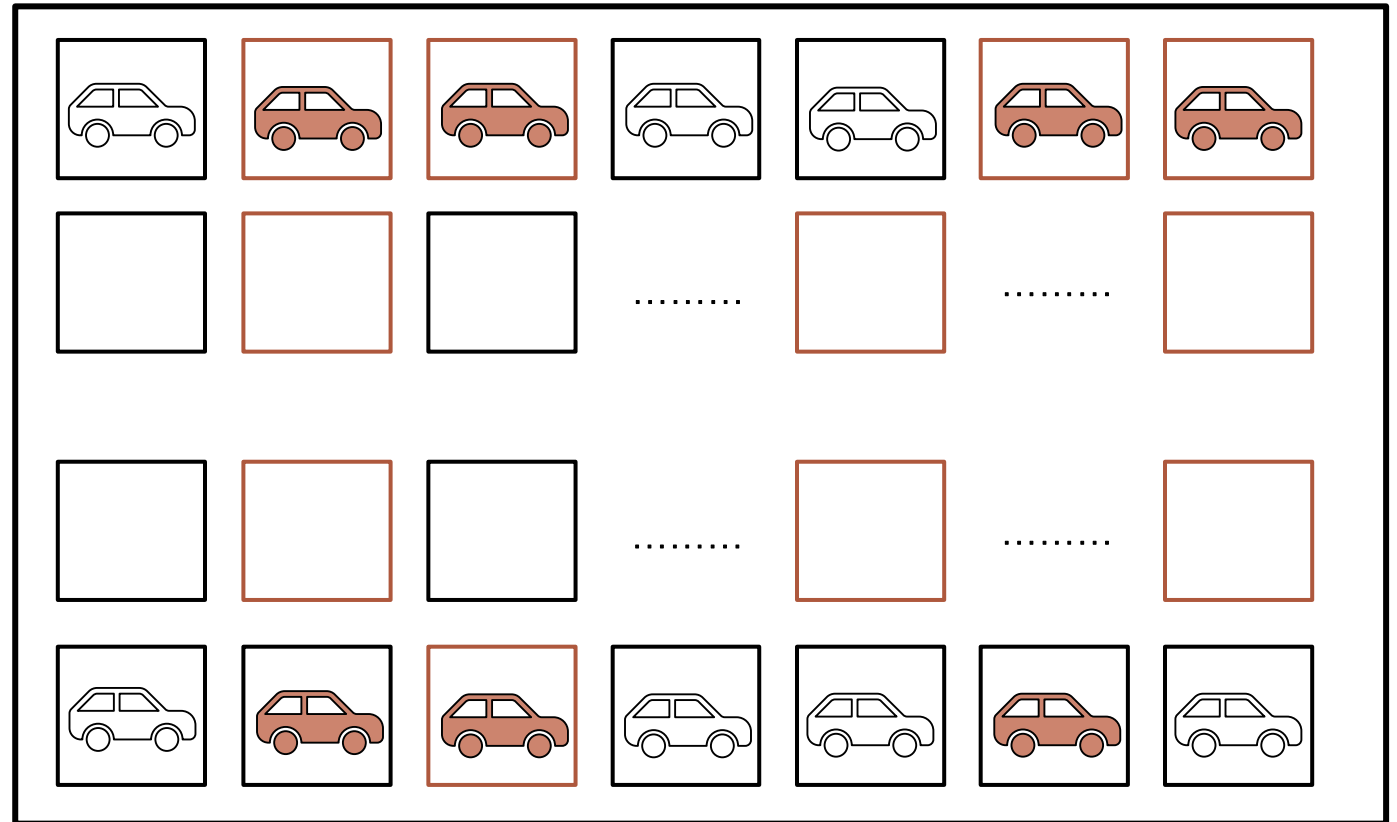
Entrée
sortie



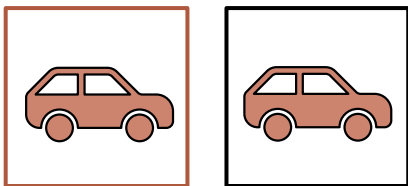
Place abonnée



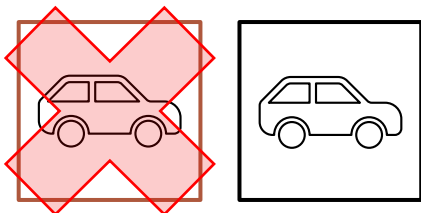
Place non abonnée



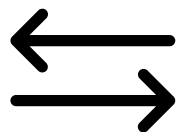
Priorité donnée aux voitures sortantes



Une voiture abonnée peut prendre une place abonnée ou non abonnée



Une voiture non abonnée ne peut prendre qu'une place non abonnée



Entrée
sortie

Dans le cas ou il y a une voiture sortante et une autre entrante dans le même temps la priorité est donnée à celle qui sort du parking

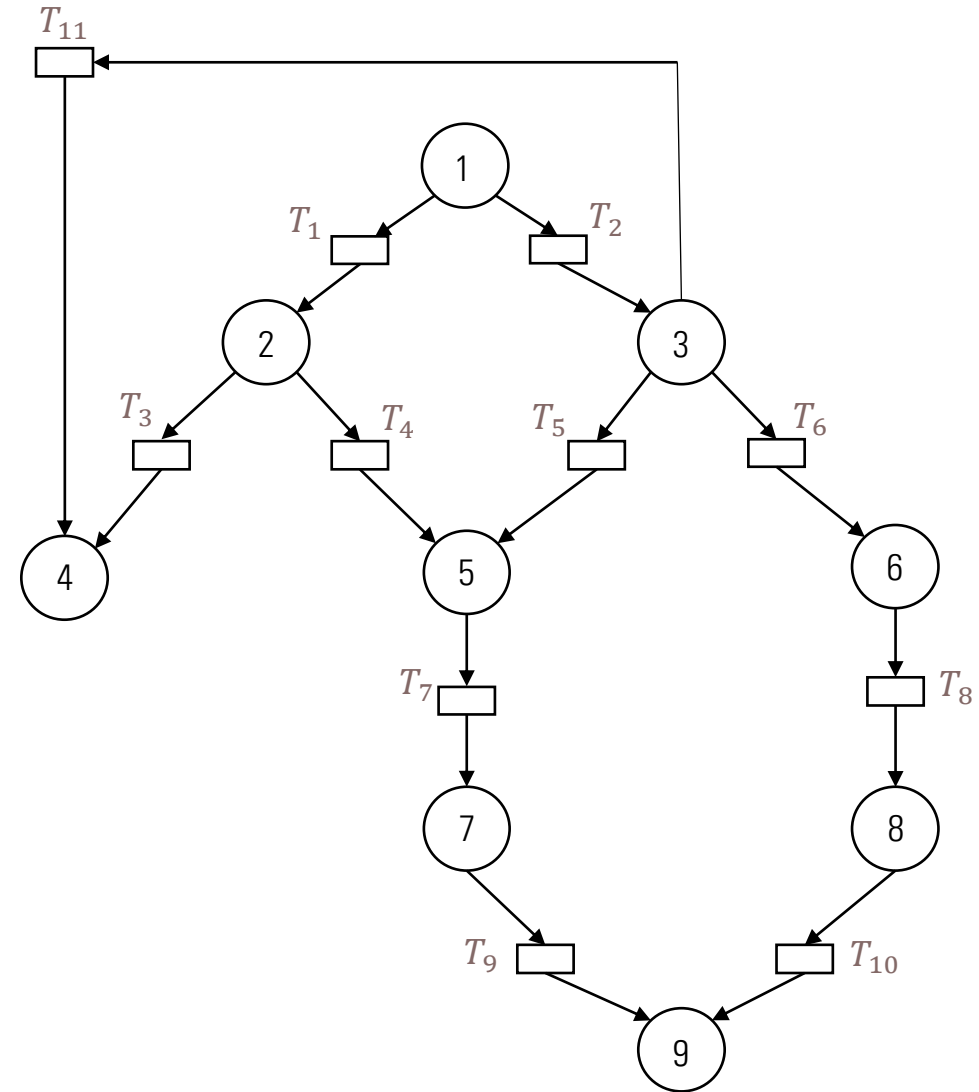
II. PRÉSENTATION DE LA SOLUTION

- La voiture arrive à l'entrée du parking. Si il n'y a pas de voitures en sorties le feu envoie un signal à la voiture pour entrer dans le parking, sinon la voiture est mise en file d'attente jusqu'à la fin du passage des voitures sortantes. Après cela, le feu reprend son activité normale c'est-à-dire envoyer des signaux vert à l'entrée.
- La voiture veut entrer dans le parking, on traite deux cas:
 - A: La voiture est abonnée: on teste s'il y a des places abonnées disponibles sinon on teste s'il y a des places non abonnées disponibles sinon la voiture quitte le parking.
 - B: La voiture n'est pas abonnée: on teste s'il y a des places non abonnées sinon la voiture quitte le parking.

- Voiture abonnée: la voiture entre dans le parking.
- Voiture non abonnée: la voiture prend un ticket et entre dans le parking.
- Une voiture abonnée peut prendre n'importe quelle place (abonnée ou non abonnée): S'il n'y a pas de places abonnées disponible la voiture cherche une place non abonnée, se gare et reste dans le parking pendant un temps aléatoire. A la fin la voiture veut sortir du parking.
- Une voiture non abonnée peut prendre seulement une place non abonnée. La voiture reste dans le parking pour un temps aléatoire. A la fin, la voiture veut sortir du parking.
- Priorité aux voitures sortante, le feu n'envoie plus de signal vert à l'entrée mais à la sortie et la voiture sort

III. RÉSEAU DE PÉTRI

1. Voiture en sortie ? :
 1. Oui : file d'attente
 2. Non: vérification Abonné ou non abonné
2. Places non abonnées disponibles ? :
 1. Oui : T4 validé
 2. Non : T3 validé
3. Places abonnées ou non abonnées disponible ? :
 1. Oui : se garer
 2. Non : sortie du parking
4. Après un temps aléatoire libération d'une place
5. Sortie du parking



IV. CHOIX DE CONCEPTION

- Synchronisation moniteur et threads :
 - On a pris la décision d'utiliser les moniteurs pour gérer la synchronisation des différents processus du projet. On a donc 3 types d'entités processus(**threads**) dans ce projet:
 - Les voitures se présentent à la barrière du parking et attendent (**pthread_cond_wait**) un signal (**pthread_cond_signal**) de la barrière afin de poursuivre leur exécution.
 - La barrière observe chaque seconde la file d'attente de sortie des voitures, si celle-ci est non vide(**test_fileSortir() == false**) on envoie un signal à la sortie sinon à l'entrée.
 - L'horloge est simplement la représentation du temps dans la simulation. A partir d'une certaine heure elle permet également de restreindre le nombre de places du parking et de les relâcher peu à peu afin de gérer la congestion.
- Signaux
 - Les voitures peuvent être abonnées ou non au parking. Afin de vérifier si une voiture est abonnée on envoie un signal SIGUSR1(numsignal=10) qui permet de signal à la barrière qu'il s'agit d'un abonné.

V. CONCLUSION

- Ce projet nous a permis de nous familiariser avec la façon dont les systèmes d'exploitations gère les processus et les signaux.
- Il nous a également montré l'importance du multitâche et de la synchronisation des threads.
- Enfin, il nous a permis de mettre à profit les connaissances théoriques acquises en cours.