

TD 1

Goroutine & Channel

Mathieu Noizet

Mars 2025

1 Introduction

L'objectif de ce TD est de manipuler les *goroutines* et les *channels* afin de mettre en pratique leurs utilisations.

2 Votre premier channel

Créer une *goroutine* qui envoie un message sur un *channel*, puis le programme principal le reçoit et l'affiche.

Pour ce faire, vous réaliserez les instructions suivantes :

1. Déclarez un *channel* de type *string*.
2. Lancez une *goroutine* qui envoie "Bonjour, Goroutine !" sur ce *channel*.
3. Dans la fonction *main()*, recevez le message et affichez-le.

3 Échange linéaire

Créer une série de *goroutines* qui passent un entier de l'une à l'autre en utilisant un *channel*.

Pour ce faire, votre code réalisera les étapes suivantes :

- Créez N *goroutines* (par exemple, 10) qui s'envoient un entier en le modifiant.
- La première *goroutine* envoie 0.
- Chaque *goroutine* reçoit l'entier, l'incrémente et l'envoie à la suivante.
- La dernière *goroutine* affiche le résultat final.

4 Structure en étoile

Créer une *goroutine* "maître" qui communique avec M *goroutines* "ouvrières".

Pour ce faire, votre code réalisera les étapes suivantes :

- La *goroutine* *maître* envoie un nombre aléatoire aux *ouvrières* via un *channel*.
- Chaque *ouvrière* affiche le nombre reçu et renvoie son carré au *maître* via un autre *channel*.
- La *maître* affiche les résultats reçus.

5 Structure en anneau

Créer P *goroutines* qui forment un anneau et se passent un message.

Votre code devra réaliser les opérations suivantes :

- Chaque *goroutine* communiquera avec la *goroutine* précédente et suivante.
- La première envoie un message à la deuxième, qui l'envoie à la troisième, etc.
- La dernière envoie le message à la première pour compléter l'anneau.
- Faites circuler un message K fois.

5.1 Question 1

Réalisez ce code avec un seul *channel*.

5.2 Question 2

Réalisez ce code avec un *channel* par *goroutine*.

6 Le tri fusion parallèle

Implémenter le tri fusion en parallèle en utilisant des *goroutines* et des *channels*.

Pour ce faire, vous devez :

- Implémentez une fonction *mergeSort* qui divise un tableau en N parties.
- Lancez N *goroutines* pour trier chaque partie en parallèle.
- Utilisez un *channel* pour récupérer les résultats triés de chaque partie et les fusionner.
- Affichez le tableau trié à la fin de l'exécution.