

# **PROGRAMMATION SYSTÈME**

24 Novembre 2025

---

Thomas Varin, Théotime Turmel, Mathias Loiseau

[https://github.com/zBlxst/TP\\_system/](https://github.com/zBlxst/TP_system/)

# CONTENTS

---

1. Introduction à la programmation système
2. Processus
3. Threads
4. Autres aspects de Programmation Système

# **INTRODUCTION À LA PROGRAMMATION SYSTÈME**

---

- Ensemble d'appels systèmes (syscalls)
- Interaction directe avec le noyau
- Gestion bas niveau : mémoire, processus, threads, fichiers

# PROCESSUS

---

# THREADS

---

- Moins coûteux que les processus
- Concurrence fine
- Meilleure performance en multicoeurs

## Norme POSIX

Fonctions de base :

- `pthread_create()`
- `pthread_join()`
- `pthread_exit()`

- Création via `int pthread_create()`
- Le thread est affecté à une fonction `void* fun(void* args)`

- `pthread_join(thread)` : attendre un thread
- `pthread_exit()` : terminer explicitement

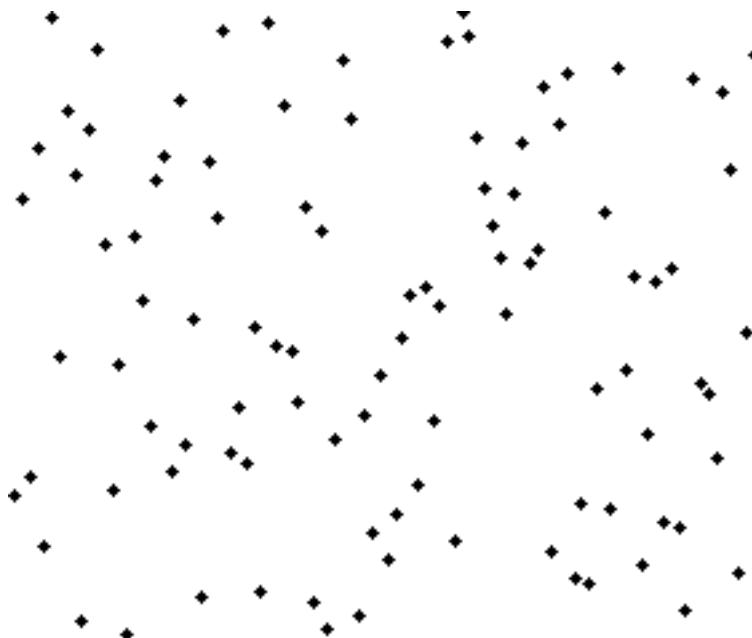
- Accès simultané à une même ressource
- Conditions de course
- Corruption mémoire

```
void* increment(void* arg) {  
    for (int i = 0; i < 1000; i++) counter++; // Section partagée  
    return NULL;  
}
```

- Verrou pour protéger une section critique
- `pthread_mutex_lock()`
- `pthread_mutex_unlock()`

## Exemple : tri bulles pair-impair

13 / 18



	Processus	Thread
Mémoire	Séparée	Partagée
Communication	Difficile	Directe
Isolation	Forte	Faible,
Identité	<b>PID</b>	<a href="#">pthread_t</a>

# AUTRES ASPECTS DE PROGRAMMATION SYSTÈME

---





