

**PROGRAMMATION  
CONCURRENTTE**



# I DISPOSITIONS & ORGANISATION DU COURS

Fabrice.Kordon@lip6.fr



## ● Programmation séquentielle

- Exécution «linéaire»
- Séquence d'instructions prévisible

## ● Programmation concurrente (il y a du parallélisme)

- Exécution non «linéaire»
  - Situations parfois délicates à reproduire
- Séquence d'instructions non prévisible
  - On parle d'«indéterminisme»
  - On parle de «causalité»

## ● Mais avant tout...

...la programmation concurrente, c'est d'abord de la programmation

## Remarque préliminaire

- Programmation  $\neq$  algorithmique
- Programmation & algorithmique sont liées

## Objectif 1: maîtriser les mécanismes de la concurrence

- On supposera acquise la programmation impérative
  - Même concepts qu'en C, syntaxe différente
  - Juste des rappels en début de cours!

Mémento  
C/Ada en ligne!

## Objectif 2: être opérationnel sur le travail de développement

- Programmer clairement
- Maîtriser les constructions d'un langage de programmation évolué
- Savoir lire du code et savoir insérer ses programmes dans un existant
- Savoir respecter les règles de «bonne programmation»
- Maîtriser un environnement de développement simple
  - Éditeurs, compilateurs, documents (sources et explication des sources)

## Objectif 3: Prendre l'habitude de respecter des contraintes

### Cahier des charges

- Il est précis, vous **devez** le respecter
- Si l'exécution n'est pas conforme à ce qui est attendu, ça ne passe pas
  - Des exercices en correction «automatique»

### Environnement de développement

- Soumission (delivery\_builder)
- Construction d'exécutables (Linux)

### Méthodes de travail

- Techniques d'élaboration d'un programme
  - reconstruction systématique (sans se poser de questions)
  - Lisibilité par des tiers
  - Documentation
- C'est aussi une «petite préparation» au Génie Logiciel

