

**PROGRAMMATION  
CONCURRENTTE**



**V  
POINTEURS, GESTION DE LA  
MÉMOIRE ET TÂCHES  
DYNAMIQUES**

Fabrice.Kordon@lip6.fr



# 19 - LA NOTION DE POINTEUR

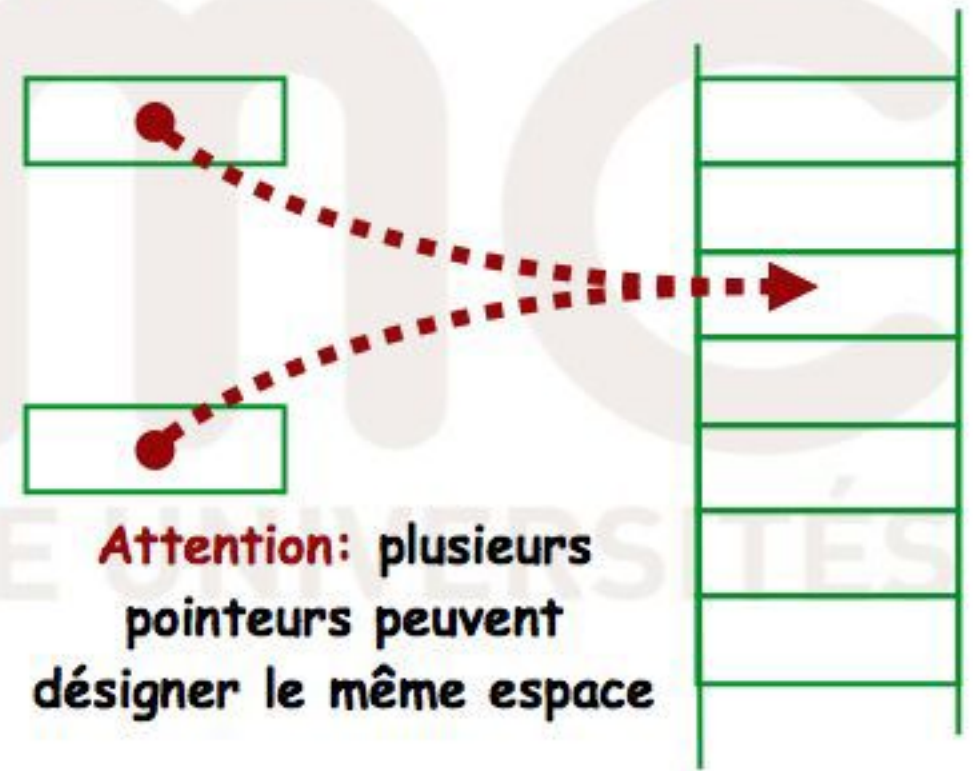
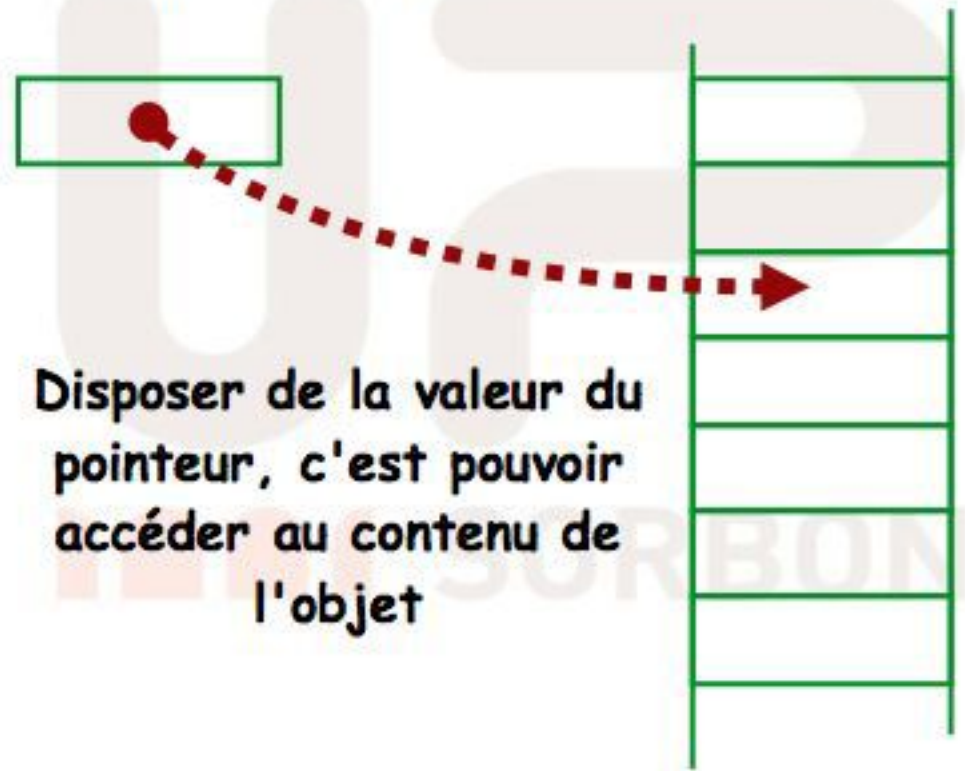
Programmation Concurrente - LI330  
Université P. & M. Curie - année scolaire 2013/2014

PrC

- **Les variables d'un programme sont stockées en mémoire**
  - **Stockage dans un espace appelé «segment de données» (variable globales)**
    - **Topologie du segment de données (en adresse absolue)**
  - **Stockage sur la pile (variables locales)**
    - **Topologie de la pile (en déplacement par rapport au sommet)**
- **Le mécanisme de variable ne permet pas toute la souplesse voulue**
  - **Les structures ne sont pas forcément retatables**
  - **Difficile de stocker des structures de données récursives (listes, arbres etc.)**
- **Besoin d'un mécanisme de gestion dynamique de la mémoire**

La notion de **pointeur** est définie pour cela



- Objet contenant une référence (adresse) sur un autre objet
- L'objet ainsi désigné est accessible par «indirection» via le pointeur
- L'objet désigné doit être créé (alloué) puis détruit (désalloué)



## Allocation

-  Première opération à effectuer
-  Attribue un «espace mémoire» à l'objet désigné par le pointeur

## Accès

-  Effectué via la référence contenue dans le pointeur
-  Utilisation en lecture ou en écriture

## Désallocation

-  Libération de l'«espace mémoire» associé au pointeur

## Un «élément neutre» : null

-  Signification, aucun «espace mémoire» désigné



# MÉCANISME SOUS-JACENT

État initial de la mémoire

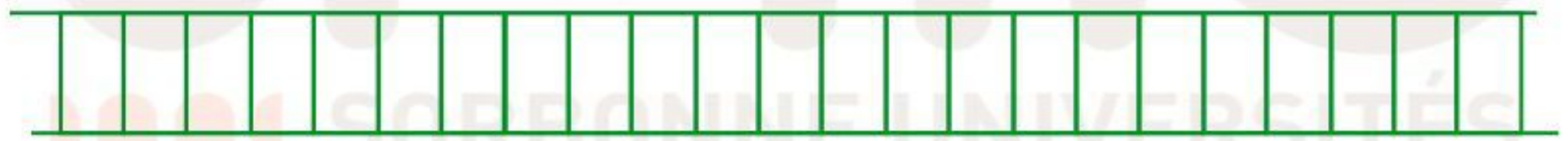
Pointeur 1

Pointeur 2

Pointeur 3

Pointeur 4

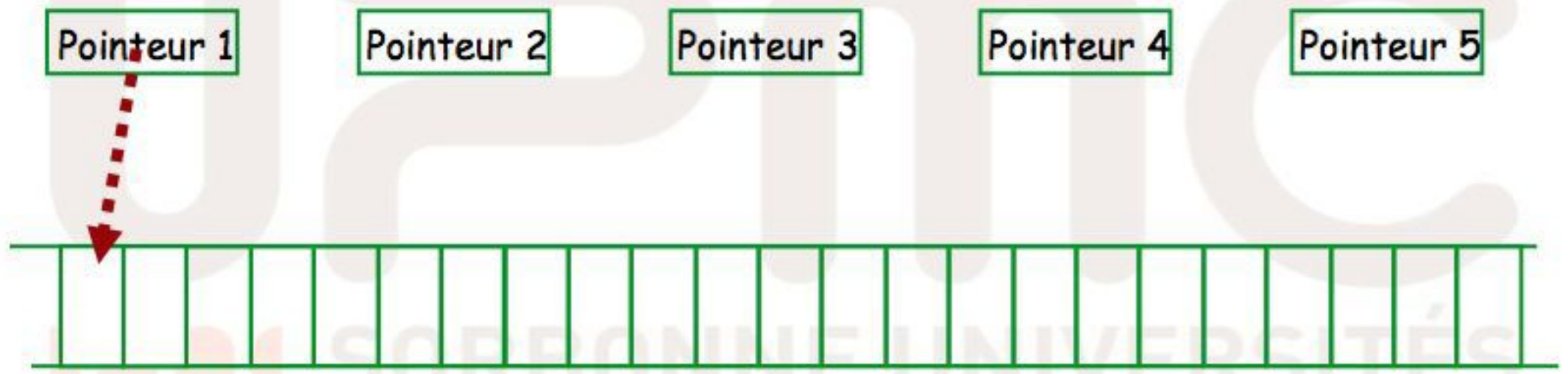
Pointeur 5





# MÉCANISME SOUS-JACENT

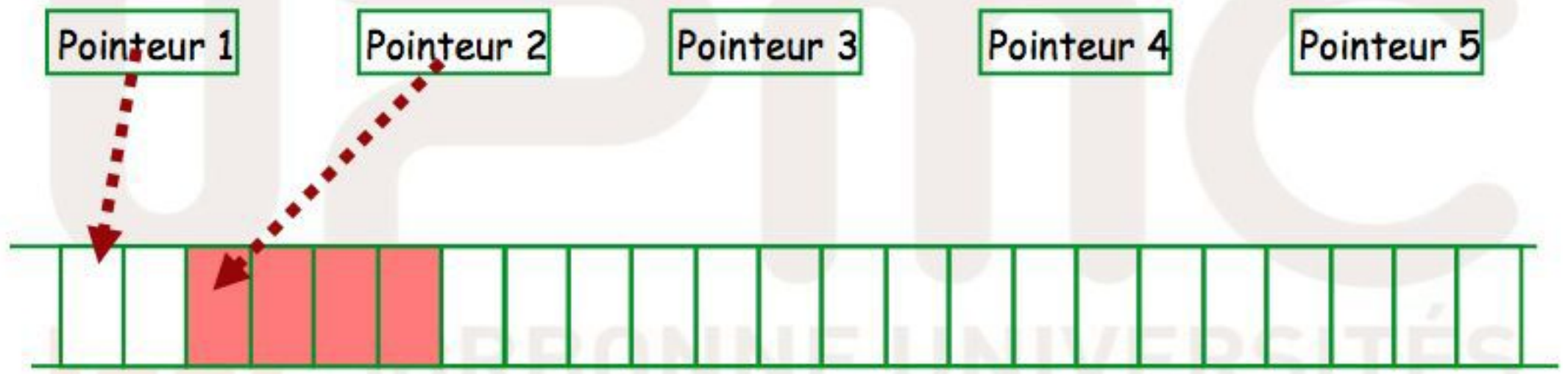
Allocation d'un entier (2 mots mémoire)





# MÉCANISME SOUS-JACENT

Allocation d'une chaîne de 4 caractères (4 mots mémoire)

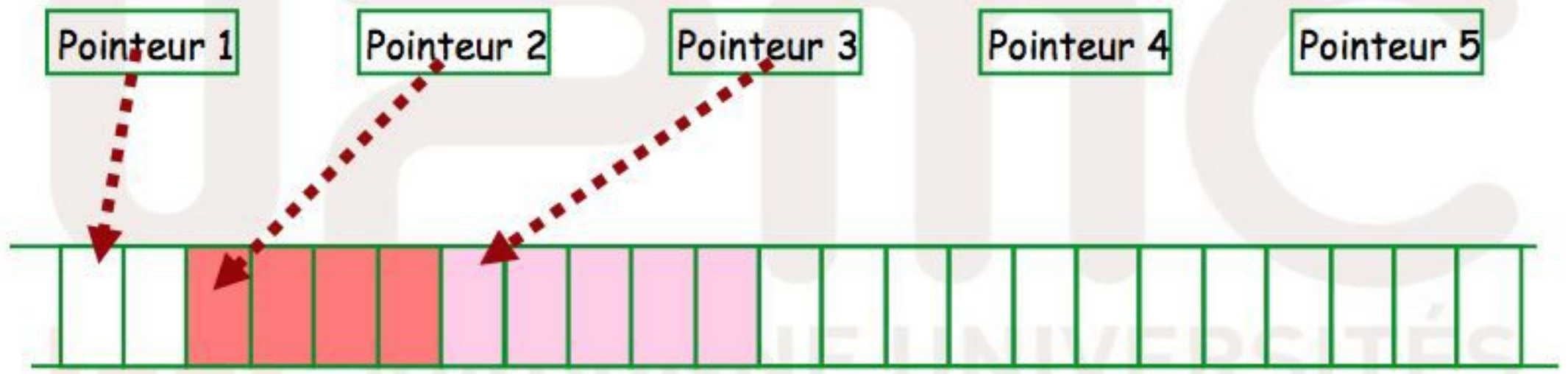






# MÉCANISME SOUS-JACENT

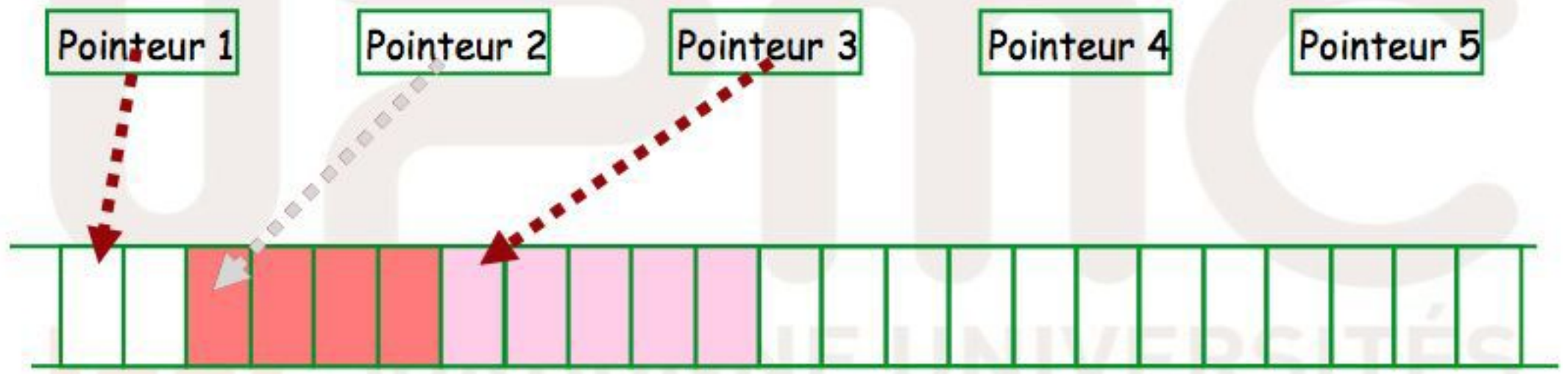
Allocation tableau de 5 booléens (5 mots mémoire)





# MÉCANISME SOUS-JACENT

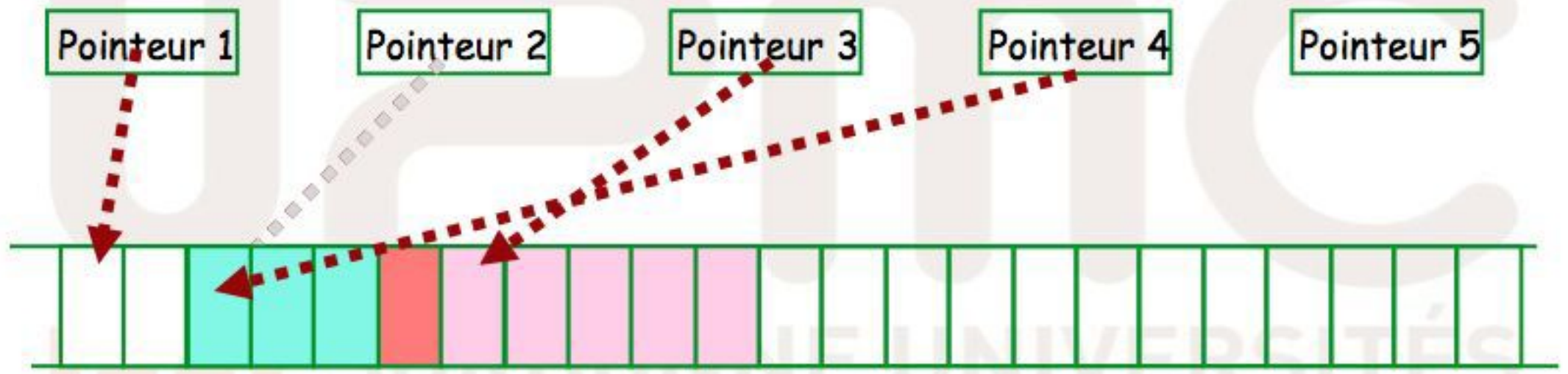
Libérer l'espace associé au pointeur 2





# MÉCANISME SOUS-JACENT

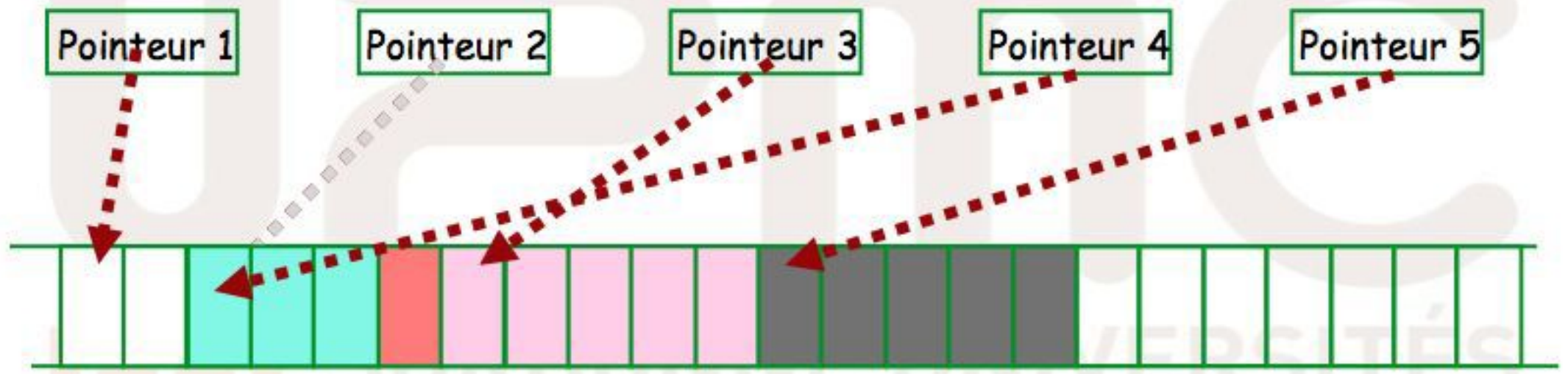
Allouer un tableau de 3 caractères (3 mots mémoire)





# MÉCANISME SOUS-JACENT

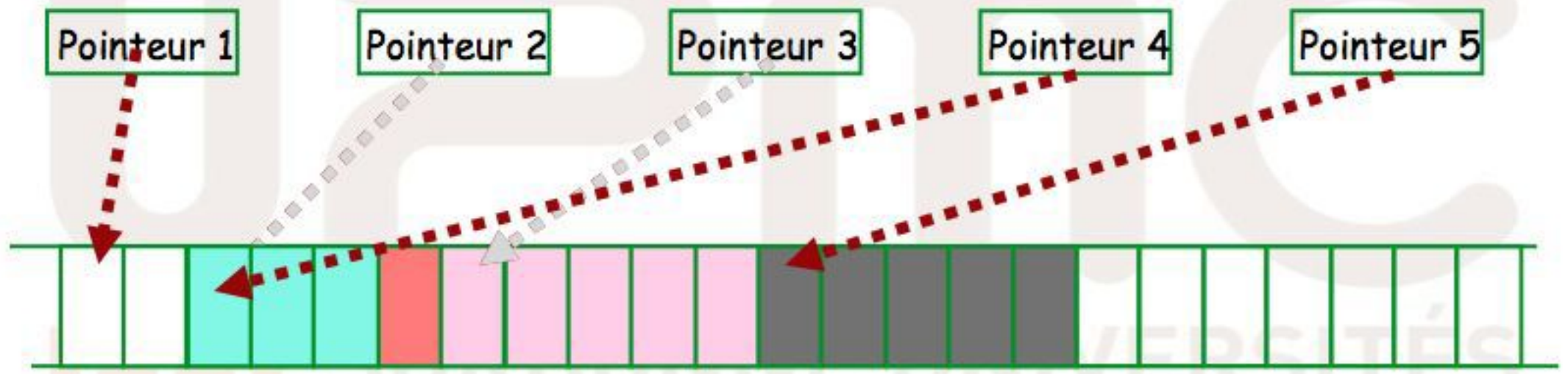
Allouer une structure contenant un pointeur ( 5 mots mémoire)





# MÉCANISME SOUS-JACENT

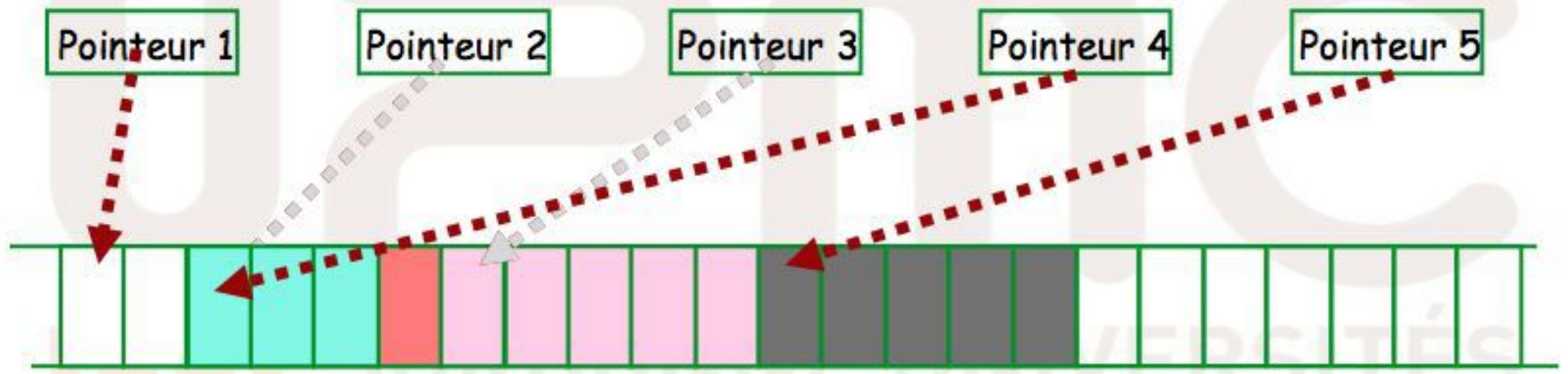
Désallocation de l'espace mémoire associé au pointeur 3





# MÉCANISME SOUS-JACENT

Allocation d'un entier dans le pointeur contenu dans l'espace désigné par pointeur 5



①

Les pointeurs permettent de manipuler des espaces de mémoire dynamiquement en fonction des besoins d'une application

②

C'est l'exécutif (runtime) du langage qui gère la mémoire en fonction des demandes

③

L'utilisateur n'a aucun moyen de contrôler les choix d'adressage

④

Sous certaines conditions, un mécanisme «ramasse-miette» (garbage collector) peut être mis en œuvre dans l'exécutif

⑤

Ada et C ne sont pas dotés d'un mécanisme «ramasse-miette»



**Il faut considérer trois choses extrêmement importantes**

①

**La durée de vie de l'espace mémoire alloué**

②

**L'existence d'un accès à l'espace mémoire en question**

③

**Les pointeurs sont des variables obéissant aux règles de vie des variables**

