

Chapitre 1

Les Pointeurs

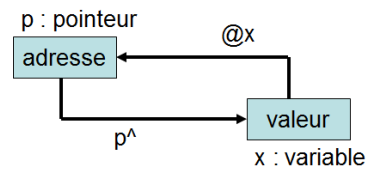
1-Définition d'un pointeur

- Concrètement, chaque langage a sa propre définition de pointeurs, mais en programmation, un pointeur est une variable destinée à contenir une adresse mémoire.
- Il peut être vu comme un « localisateur » : par son contenu, le pointeur désigne (**pointe sur**) l'emplacement physique d'une autre valeur

Cours ASD2 – ISET Charguia 2020/2021

2-Représentation graphique

- Si x désigne une variable, l'expression $@x$, qui fournit l'adresse de x , est **un pointeur** ;
ici $@x$ pointe vers x .



Cours ASD2 – ISET Charguia 2020/2021

3-Type du pointeur

- Un pointeur est typé : il est lié au type de l'entité vers laquelle il pointe.
- Le type de la variable pointée est appelé **le type du pointeur** ou **type de base**.

Cours ASD2 – ISET Charguia 2020/2021

4-Déclaration d'un pointeur

- Variable p : $\wedge T$

Explication:

Déclare une variable p qui pointe vers une valeur de type T

- Taille d'un type pointeur

Le nombre d'octets de l'adresse est indépendante du type de l'entité référencée.

- Taille(Pointeur Caractère) = Taille(Pointeur Réel) = 4 octets par exemple
- (Taille(Caractère) = 1) <> (Taille(Réel) = 8) par exemple

- La déclaration d'une variable pointeur réserve les octets nécessaires au codage d'une adresse mémoire **mais ne réserve aucune mémoire pour la donnée pointée.**

Cours ASD2 – ISET Charguia 2020/2021

5-La valeur Nil

- **La valeur Nil**

Not Identified Link, elle désigne la valeur associée au pointeur nul.

Elle peut être mémorisée dans toute variable pointeur, quel que soit son type.

- **Utilité de la valeur Nil**

Elle sert généralement à initialiser un pointeur quand on ne connaît pas l'adresse qu'il doit contenir, ou à signaler l'échec d'une opération ou l'apparition d'une erreur ...

Cours ASD2 – ISET Charguia 2020/2021

La valeur Nil

- **Représentation graphique**

La valeur Nil est souvent représentée par:



6-L'opérateur d'adresse

- Désigne l'adresse mémoire de l'entité (variable, constante).
- @x

7-L'opérateur d'indirection

- Il désigne le contenu de la valeur pointée par le pointeur.
- Opérateur d'indirection
 p^{\wedge} // avec p pointeur
- Explication
 Renvoie la valeur pointée par le pointeur p, p^{\wedge} est une variable du type de la zone pointée.

Cours ASD2 – ISET Charguia 2020/2021

L'opérateur d'indirection

- Lorsqu'un pointeur vaut Nil, il n'y a aucun sens de tenter d'accéder à la zone pointée.
- Ainsi l'instruction suivante produira un comportement imprévisible si p vaut Nil
 $x \leftarrow p^{\wedge}$????????

Cours ASD2 – ISET Charguia 2020/2021

Exemple

algorithme : Exemple_Opérateur_Adresse

Variables

x : Entier

p : ^Entier

Début

x ← 3

p ← @x // p pointe vers x

p^ ← p^ + 1 // incrémentation de x via p

Fin

- **Remarque:** à la fin de l'exécution, la variable x contiendra la valeur 4.

Cours ASD2 – ISET Charguia 2020/2021

8-Affectation d'un autre pointeur

- L'affectation entre pointeurs est permise si les deux pointeurs **ont le même type**.

- **Exemple**

Variables

p, q : ^T

Début

.....

p ← q // p pointe vers la même zone que q

Fin

Cours ASD2 – ISET Charguia 2020/2021

9-Initialisation de pointeurs

- Un pointeur est un type élémentaire. Par conséquent les éléments de type pointeur ne sont pas initialisés.
- L'initialisation pourra s'effectuer par l'affectation de :
 - L'adresse d'une entité @x.
 - Un autre pointeur du même type.
 - La valeur Nil du pointeur nul.

Cours ASD2 – ISET Charguia 2020/2021

10-Comparaisons entre pointeurs

- Les comparaisons concernant les pointeurs sont réduites. Ainsi :
 - Deux variables pointeurs peuvent être comparées entre elles si et seulement si elles sont du même type. Dans ce cas, seules les opérateurs de comparaison d'égalité = et d'inégalité <> ont un sens.
 - La valeur Nil peut être comparée à toutes les variables pointeurs quels que soient leurs types.

Cours ASD2 – ISET Charguia 2020/2021

11-Conclusion

- Le test entre variables pointeurs ne permet de savoir que deux choses :
 - Si deux variables pointeurs pointent sur la même variable pointée.
 - Si une variable pointeur mémorise Nil.

Cours ASD2 – ISET Charguia 2020/2021

II-Arithmétique des pointeurs

- La valeur d'un pointeur étant un entier. On peut lui appliquer un certain nombre d'opérateurs arithmétiques classiques.
- Les opérations arithmétiques valides sur les pointeurs sont:
 - **l'addition** d'un entier à un pointeur.
 - **la soustraction** d'un entier à un pointeur.
 - **la différence de deux pointeurs** p et q pointant tous les deux vers des objets de **même type**.
- Le résultat est un entier dont la valeur est égale à $(p-q)/\text{Taille}(\text{type})$
- **Remarque importante:** La somme de deux pointeurs n'est pas autorisée.

Cours ASD2 – ISET Charguia 2020/2021

Arithmétique des pointeurs

- Si k est un entier et p est un pointeur sur un objet de type T .
- L'expression " $p+k$ " désigne un pointeur sur un objet de type T dont la valeur est égale à la valeur de p incrémentée de $k \cdot \text{Taille}(T)$.
- Il en va de même pour la soustraction d'un entier à un pointeur

Cours ASD2 – ISET Charguia 2020/2021

Arithmétique des pointeurs

Exemple 1:

Algorithme Exemple1

Variables

i : Entier

$p1, p2$: $^{\wedge}$ Entier

Début

$i \leftarrow 3$

$p1 \leftarrow @i$

$p2 \leftarrow p1$

Ecrire("p1 =", $p1^{\wedge}$, "et p2 =", $p2^{\wedge}$)

Fin

Cours ASD2 – ISET Charguia 2020/2021

Arithmétique des pointeurs

Exemple 2:

Algorithme Exemple2

Variables

pi : ^Entier

pr : ^Réel

pc : ^Caractère

Début

pi \leftarrow 4

(pi+1) \leftarrow 5

pr \leftarrow 45.7

pr \leftarrow pr+1

pc \leftarrow 'j'

Fin

Cours ASD2 – ISET Charguia 2020/2021

III-Allocation Statique & Dynamique

- **1-Définition:** L'allocation est dite **statique** car décidée au moment de l'écriture de l'algorithme et non évolutive lors de l'exécution.

- **2-Conséquence :**

Cela a de nombreux inconvénients car il n'est pas possible de contrôler au mieux l'espace mémoire disponible.

Cours ASD2 – ISET Charguia 2020/2021

3-Inconvénients: Allocation Statique

- Fréquemment aussi, dans le cas d'un vecteur (tableau), on ne peut pas connaître, a priori lors de l'écriture de l'algorithme, le nombre d'éléments qu'il contiendra. Cette taille n'est parfois calculable ou donnée que pendant l'exécution du programme.
- La solution consistait à déclarer un vecteur d'une taille très importante, qui occupait beaucoup d'espace mémoire parfois inutilement.
- **Il est donc indispensable de disposer d'un mécanisme permettant de créer et de supprimer des variables de différents types pendant l'exécution.**
- C'est ce qu'on appelle la **gestion dynamique** (pendant l'exécution) de la mémoire.

Cours ASD2 – ISET Charguia 2020/2021

4-Allocation dynamique

- L'allocation dynamique est une allocation de mémoire **effectuée lors de l'exécution** mais non prévue lors de la compilation
- Cette allocation est à la charge du programmeur, il lui faut donc :
 - une fonction permettant de réserver une zone mémoire: **allouer()**
 - une fonction permettant de libérer une zone mémoire: **liberer()**
 - une variable (et donc un type) permettant de référencer cette zone mémoire allouée : **le pointeur**

Cours ASD2 – ISET Charguia 2020/2021

Allocation dynamique

- La fonction "**Allouer**".
- Sa syntaxe est la suivante :

$$\text{Allouer}(\text{ptr}, \text{nbr_objets})$$
- Si $\text{nbr_objets} = 1$ alors la syntaxe est la suivante :

$$\text{Allouer}(\text{ptr})$$

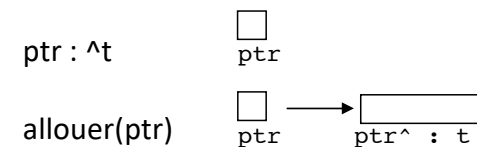
Cours ASD2 – ISET Charguia 2020/2021

Allocation dynamique

Allouer(ptr)

- réserve un emplacement mémoire de la taille correspondant au type **t**,
- met dans la variable ptr l'adresse de la zone mémoire qui a été réservée.

L'emplacement pointé par ptr sera accessible par ptr^{\wedge} .



Cours ASD2 – ISET Charguia 2020/2021

Exemple 1

Algorithme Exemple1

Variables

pi, pj : ^Entier

pr : ^Réal

pc : ^Caractère

Début

//réserver 10 octets mémoire, soit la place pour 10 caractères

Allouer(pc, 10)

//réserver un espace pour 4 entiers

Allouer(pi,4)

//réserver un espace pour 6 réels

Allouer(pr,6)

//réserver un espace pour 1 entier

Allouer(pj) // Par défaut le nombre d'objets est 1

Fin

Cours ASD2 – ISET Charguia 2020/2021

5-Libération dynamique

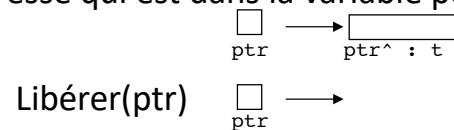
- La fonction "**Libérer**"
- Lorsque l'on n'a plus besoin de l'espace mémoire alloué dynamiquement c'est-à-dire quand on n'utilise plus le pointeur ptr, il faut libérer cette place en mémoire.
- Ceci se fait à l'aide de l'instruction "**Libérer**" qui a pour syntaxe :
Libérer(ptr)

Cours ASD2 – ISET Charguia 2020/2021

Libération dynamique

Libérer(ptr)

- Libère la place de la zone mémoire dont l'adresse est dans ptr (et la rend disponible pour l'allocation d'autres variables)
- Laisse la valeur du pointeur en l'état (n'efface pas l'adresse qui est dans la variable pointeur).



Cours ASD2 – ISET Charguia 2020/2021

Exemple2

Algorithme Exemple2

Variables

pi : ^Entier

pr : ^Réal

Début

//réserver un espace pour 4 entiers

Allouer(pi, 4)

//réserver un espace pour 6 réels

Allouer(pr, 6)

...

//libérer la place précédemment réservée pour pi

Libérer(pi)

//libérer la place précédemment réservée pour pr

Libérer(pr)

Fin

Cours ASD2 – ISET Charguia 2020/2021

Libération dynamique

Remarque :

Attention: Si on fait appel au pointeur désalloué (Libéré), il renvoie une information qui n'a aucun sens.

Cours ASD2 – ISET Charguia 2020/2021

IV-Pointeurs et tableaux

- **Tout tableau est un pointeur constant.**

- Dans les déclarations :

tab : Tableau de [1..10] d'Entier

p : ^Entier

$p \leftarrow \text{tab} \iff p \leftarrow @\text{tab}[1]$

car tab prends l'adresse du premier élément du tableau

Cours ASD2 – ISET Charguia 2020/2021

Pointeurs et tableaux (suite)

- Les écritures suivantes sont équivalentes :

$$\text{tab}[3] \quad \Leftrightarrow \quad (p+3)^\wedge$$
- tab est un pointeur constant non modifiable dont la valeur est l'adresse du premier élément du tableau.
- Autrement dit tab a pour valeur @tab[1].
- **On peut donc utiliser un pointeur initialisé à tab pour parcourir les éléments du tableau.**

Cours ASD2 – ISET Charguia 2020/2021

Exemple

```

tab : Tableau de [1..5] d'Entier
p : ^Entier
i : Entier

...
p ← tab
Pour i de 1 à 5 faire
    Ecrire(p^i)
    p ← p+1
Fin Pour
  
```

Cours ASD2 – ISET Charguia 2020/2021

Exemple

Ce qui est aussi équivalent à:

tab : Tableau de [1..5] d'Entier

p : ^Entier

...

p ← tab

Pour p de tab à tab+4 faire

 Ecrire(p^)

Fin Pour

Cours ASD2 – ISET Charguia 2020/2021

V-Pointeur et enregistrements

- Il est bien sûr possible de déclarer un pointeur sur un type structuré.
- Ainsi l'accès au champ se fera de la façon suivante :
(nom_pointeur^).champ_de_l_enregistrement

Cours ASD2 – ISET Charguia 2020/2021

Exemple

Type Pays = Enregistrement

nom : chaîne de caractères

nb_hab : Entier

FinEnregistrement

Algorithme Pointeur_Enregistrement

Variables

ppays : ^Pays

Début

Allouer(ppays)

Ecrire("Saisissez le nom du pays et sa population")

Lire(ppays^.nom)

Lire(ppays^.nb_hab)

Ecrire("Vous avez saisi que le ", ppays^.nom, " a ", ppays^.nb_hab, "habitant(s)")

Libérer (ppays)

Fin