

Algorithmique et Structures de Données

L2 MPI-MI

Dr Ousmane DIALLO



Chap12: Les pointeurs

12.1. Notion d'adresse

- ❑ La mémoire centrale utilisée par les programmes est découpée en **octets**.
- ❑ Chacun de ces octets est identifié par un numéro séquentiel. Par convention, ce numéro est noté en **hexadécimal** et précédé par **0x**.

Exemple

0x3ffd10
0x3ffd11
0x3ffd12
0x3ffd13
0x3ffd14
0x3ffd15
0x3ffd16
0x3ffd17

...

Chap12: Les pointeurs

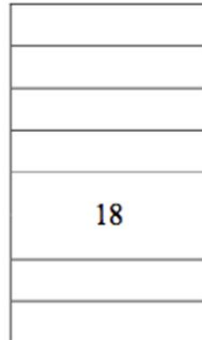
12.1. Notion d'adresse

- ❑ Déclarer une variable consiste à réserver une zone en mémoire occupant un certain nombre d'octets (sa taille) et à laquelle on donne un nom.
- ❑ Le numéro du premier octet de cette zone correspond à l'adresse de la variable.

Exemple

int n=18;

0x3ffd10
0x3ffd11
0x3ffd12
0x3ffd13
0x3ffd14
0x3ffd15
0x3ffd16
0x3ffd17
...



n

Dans cet exemple, la valeur de la variable n est 18 et son adresse est 0x3ffd14.

Chap12: Les pointeurs

12.1. Notion d'adresse

- ❑ Nous constatons que pour accéder à la valeur contenue dans une variable, on utilise tout simplement son **nom**.
- ❑ Toutefois, il est parfois très pratique de manipuler une variable par son **adresse**.
- ❑ En algorithmique, pour désigner l'adresse d'une variable, on utilise l'expression "**adresse de**" alors qu'en Pascal, on utilise l'**opérateur @**

Chap12: Les pointeurs

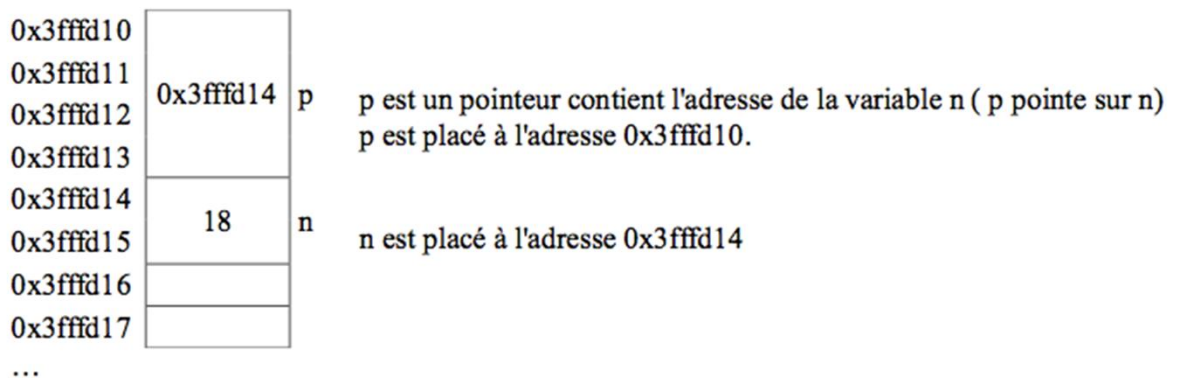
12.2. Notion de pointeur ou variable dynamique

- ❑ Un **pointeur** est une variable dont la valeur(le contenu) est l'adresse d'une autre variable.
 - *On dit que le pointeur pointe sur la variable dont il contient l'adresse.*
- ❑ Un pointeur est associé à un type de variable sur lequel il peut pointer.
 - *Par exemple, un pointeur sur entier ne peut pointer que sur des variables entières.*
- ❑ Un pointeur est lui-même une variable et à ce titre il possède une adresse.

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique

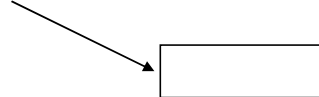
- ❑ Il convient donc de ne pas confondre l'adresse de la variable pointeur et sa valeur (qui est l'adresse de la variable pointée).



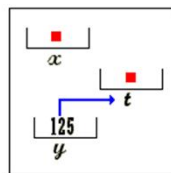
Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Représentation

pointeur



variable pointée



Mémoire Centrale

Nom	Adresse
x	5896
y	804
t	125

Table des symboles

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Déclaration d'un pointeur

- ❑ En langage Pascal, pour déclarer une variable pointeur vers un type de base, il faut :
- partir de la déclaration d'une variable ayant un type de base ;
 - ajouter le signe [^] avant le type.

(*En Algorithmique*)

Variable

Nom_pointeur: pointeur sur type_pointé;

Exemple

P: pointeur sur un entier;

...

(*En Pascal*)

var

Nom_pointeur: ^type;

Exemple

P: ^Integer;

...

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Déclaration d'un pointeur

- ❑ Autre déclaration via une définition de type

(*En Algorithme*)

```
...
Type
PointeurSurType = pointeur sur type pointé;

Variable
Nom_pointeur: PointeurSurType;
...
```

(*En Pascal*)

```
...
Type
PointeurSurType = ^Type;
Var
Nom_pointeur: PointeurSurType;
```

Exemple

```
Type
PointeurEntier = ^Integer;
Var
P1, P2: PointeurSurType;
```

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Manipulation d'un pointeur

- ❑ L'opérateur @ (**adresse de**) appliqué à une variable délivre l'adresse de celle-ci ; cette adresse pourra être affectée à une variable de type pointeur.
- ❑ On peut écrire par exemple

(*En Algorithm*)

```

Variable
n : entier;
P : pointeur sur un entier;
Debut
N ← 18;
P ← adresse de n;
...
Fin.

```

(*En Pascal*)

```

var
n : Integer; // est une variable entière
P : ^Integer; // P est 1 pointeur sur un entier
Begin
n := 18; // affecte une valeur à n
P := @n; // affecte l'adresse de n à P
...
End.

```

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Manipulation d'un pointeur

- ❑ Il est possible d'accéder à la zone mémoire pointée en utilisant l'opérateur $^$.
- ❑ Ainsi, $p^$ désigne la zone mémoire (la variable) pointée par le pointeur p .

(*En Algorithme*)

Variable

```
n : entier;
P : pointeur sur un entier;
Debut
N ← 33;
P ← adresse de n; // P pointe sur n
Ecrire (P^); // la valeur 33 est affichée à l'écran
P^ ← 34; // n vaut maintenant 34
...
Fin.
```

(*En Pascal*)

var

```
n : Integer; // est une variable entière
P : ^Integer; // P est 1 pointeur sur un entier
Begin
n := 33; // affecte une valeur à n
P := @n; // affecte l'adresse de n à P
Write(P^); // la valeur 33 est affichée à l'écran
P^ := 34; // n vaut maintenant 34
...
End.
```

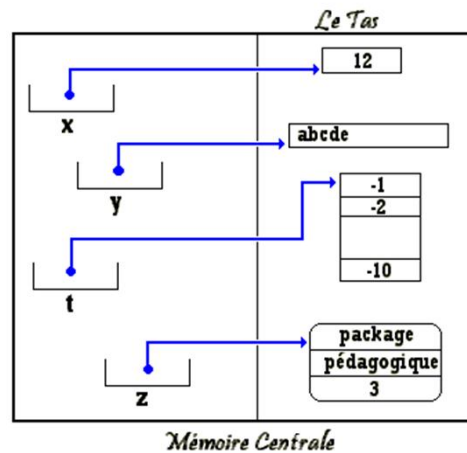
Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Manipulation d'un pointeur: Exemple

```

program LesPointeurs ;
Type
  Pentier = ^integer ;
  PString = ^string ;
  PTable = ^Table ;
  Table = Array[1..10] of integer ;
  PEnregistr = ^Enreg ;
  Enreg = record
    nom, prenom : string ;
    age : 1..100 ;
  end;
var
  x : Pentier ;
  y : PString ;
  t : PTable ;
  z : PEnregistr ;
begin
  x^ := 12 ;
  y^ := 'abcde' ;
  t^[1] := -1 ;
  t^[2] := -2 ;
  .....
  t^[10] := -10 ;
  z^.nom := 'package' ;
  z^.prenom := 'pédagogique' ;
  z^.age := 3 ;
end.

```



Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique

Remarques

- ❑ Attention à toujours initialiser un pointeur. Un pointeur qui n'est pas initialisé s'appelle un **pointeur pendant**.
- ❑ *Un pointeur pendant ne pointe pas nulle part mais n'importe où.*
- ❑ Si l'on déréférence ce pointeur et qu'on affecte une nouvelle valeur, on va écraser un emplacement mémoire quelconque
 - et on risque de faire planter le programme.

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique

Remarques

- ❑ Si on veut que le pointeur pointe nulle part, il faut l'initialiser à **NIL**. C'est l'équivalent de 0 pour les pointeurs.

(*En Algorithmes*)

Variable

P : pointeur sur un entier;

Debut

P ← NIL; // P pointe nulle part

...

Fin.

(*En Pascal*)

var

P : ^Integer; // P est 1 pointeur sur un entier

Begin

P := NIL; // P pointe nulle part

...

End.

Chap12: Les pointeurs

12.2. Notion de pointeur ou variable dynamique Double indirection

❑ Le fait qu'un pointeur pointe sur une variable s'appelle **indirection** (comme accès **indirect**). Quand un pointeur pointe sur un autre pointeur, il y a double indirection.

Exemple:

...	0xfffd10	P2	
...			
0x3ffd10			P2 pointe sur P1 qui pointe sur n
0x3ffd11	0xfffd14	P1	
0x3ffd12			On peut accéder à n par P2 en utilisant deux fois l'opérateur ^
0x3ffd13			
0x3ffd14	18	n	P2^^ est équivalent à P1^, c'est à dire n
0x3ffd15			
...			
...			

Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire

- ❑ Au lieu d'affecter à un pointeur l'adresse d'une variable existante, on peut réserver de manière dynamique une nouvelle zone en mémoire (on réserve la place pour chaque variable en cours d'exécution du programme) et affecter au pointeur l'adresse de cette zone.

➤ *Allocation dynamique de mémoire*

- Le pointeur va alors pointer sur cette zone.

Allocation dynamique en Algorithme

- ❑ L'allocation dynamique se fait par l'instruction :

$identificateur_pointeur \leftarrow allouer(unType)$

Où

$identificateur_pointeur$ est l'identificateur du pointeur $unType$ est le type de l'objet pointé.

Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire Allocation dynamique en Algorithme

Exemple

Variable

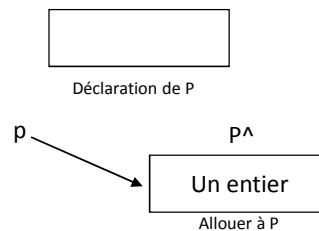
P: pointeur sur entier

Début

allouer(P)

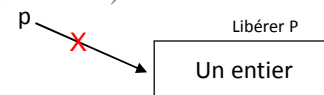
...

Fin



- Enfin, lorsque l'on n'a plus besoin de l'espace mémoire alloué dynamiquement (c'est-à-dire quand on n'utilise plus le pointeur identificateur_pointeur), il faut **libérer** cette place en mémoire. Cela se fait par l'instruction :

désallouer(identificateur_pointeur)



Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire

Allocation dynamique en Pascal

❑ En Pascal, l'allocation dynamique de mémoire se fait grâce à l'opérateur **new**

var

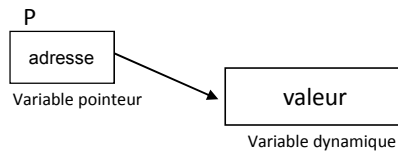
P: ^Integer;

Begin

New(P);

...

End



Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire Allocation dynamique en Pascal : Exemple

Type Ville = record

 nom: **String**;

 CodePostal: **Integer**;

End;

PVille = ^Ville;

var

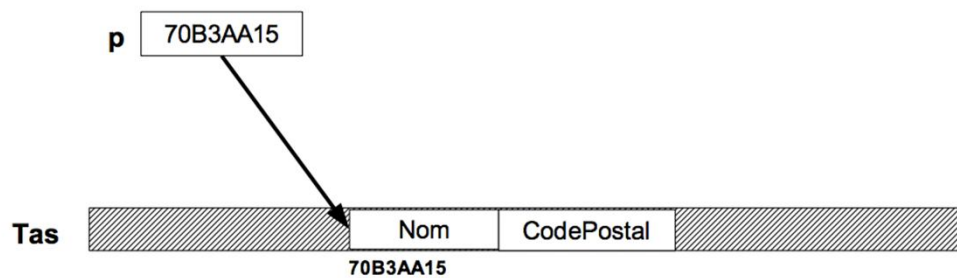
p: Pville;

Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire Allocation dynamique en Pascal : Illustration

Allocation dynamique de mémoire : new

Effet de **new** (p) :



Création d'une variable sans nom pointé par p

Chap12: Les pointeurs

12.3. Allocation dynamique de mémoire

Allocation dynamique en Pascal

Remarque

- ❑ La libération d'une zone mémoire allouée de manière dynamique n'est pas automatique en Pascal.
- ❑ Une telle zone continue à occuper de l'espace mémoire jusqu'à la fin de l'exécution du programme si on ne la libère pas de manière explicite.
- ❑ Pour ce faire, il faut utiliser la fonction **Dispose**:

Dispose(identificateur_pointeur);

Exemple:

- `Dispose(p);` // permet de libérer la zone mémoire pointée par p.

Chap12: Les pointeurs

12.4. Pointeurs et enregistrements

- ❑ Il est très courant d'utiliser un pointeur pour mémoriser l'adresse d'une variable enregistrement.
- ❑ Si p est un pointeur sur une enregistrement, p^{\wedge} désigne l'enregistrement pointé.
- ❑ On peut donc accéder à un champ de l'enregistrement pointée par l'expression : $p^{\wedge}.champ$

Chap12: Les pointeurs

12.4. Pointeurs et enregistrements

Exemple

Type Ville = record

 nom: **String**;

 CodePostal: **Integer**;

End;

PVille = ^Ville;

var

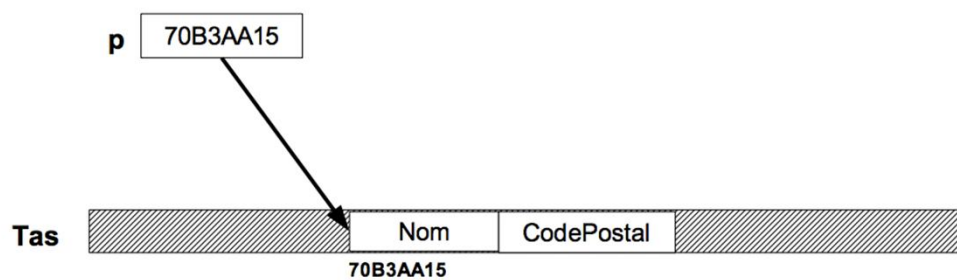
p: Pville;

Chap12: Les pointeurs

12.4. Pointeurs et enregistrements Exemple

Allocation dynamique de mémoire : new

Effet de **new** (p) :



Création d'une variable sans nom pointé par p

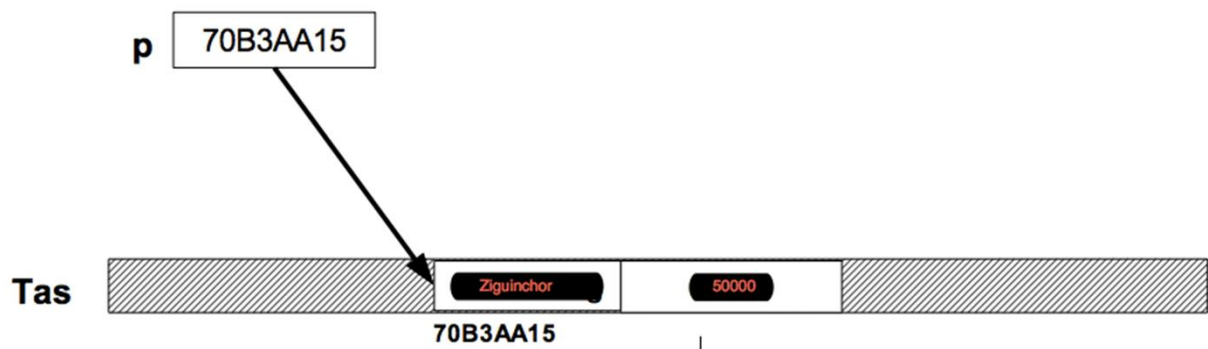
Chap12: Les pointeurs

12.4. Pointeurs et enregistrements

Exemple

Accès à la variable pointée: P^{\wedge}

```
P^.nom := 'Ziguinchor';  
P^.CodePostal := 50000;
```



Chap12: Les pointeurs

12.5. Pointeurs et tableaux

❑ Il est possible de combiner les enregistrements, les tableaux et les pointeurs.

```

Program TabPoinEnr ;
Const Max = 10 ;
Type Personne = Record
    nom, prenom : String ;
    matricule : Integer ;
End ;
Tableau = Array[1..Max] Of Personne ;
PTableau = ^Tableau ;
Var
    Tab : PTableau ;
    i : Integer ;

BEGIN
New(Tab);
With Tab^[1] Do
    Begin
        nom := 'Diallo' ;
        prenom := 'Mohameth' ;
        matricule := 1256 ;
    End ;
For i:=1 To Max Do WriteLn(Tab^[i].nom) ;
Dispose(Tab);
END.

```

(*Autres exemples*)

```

Type TabP = Array[1..100] Of ^Integer ;
Var Tab : TabP ; //Tableau de pointeurs pointant vers des entiers. Tab[i] est un
// pointeur et Tab[i]^ est un entier.

Type Tab = Array[1..100] Of Integer ;
PTab = ^Tab ;
Var Tab : PTab ; // Pointeur pointant vers un tableau d'entiers. Tab^[i] est un
// entier et Tab est un pointeur.

```

(*Autres exemples*)

```

Const Max = 20 ;
Type Station = Record
    nom : String ;
    liaisons : Array[1..10] Of Station ;
End ;
TabStation = Array[1..Max] Of Station ;
PTabStation = ^TabStation ;
Var Senegal : PTabStation ; // Senegal est un pointeur pointant vers un
// tableau d'enregistrement dont l'un des champs
//est un tableau et l'autre un enregistrement (récurif)

```

Chap12: Les pointeurs

12.5. Conclusion

- ❑ La notion de pointeurs est très commode dans les langages de programmation, car elle permet de représenter simplement des structures de données dynamiques comme les piles, les files, les listes, les arbres etc...
- ❑ Toutefois le revers de la médaille se situe dans la trop grande proximité d'un pointeur avec le bas niveau, celui de la machine, ce qui nuit à l'abstraction du programme !
- ❑ C'est pourquoi si les pointeurs sont présents dans le pascal, le C, le C++, ils ont été éliminés dans Java et remplacés par des références qui sont des encapsulations de pointeurs (Delphi travaille essentiellement avec des références mais autorise l'utilisation de pointeurs).

FIN CHAP12

Les pointeurs

QUESTIONS ??

Dr Ousmane DIALLO

