

# Algorithmique et structure de données

- Introduction
- Les algorithmes séquentiels simples
- Les structures conditionnelles
- Les boucles
- Les tableaux et les chaînes de caractères
- Les sous-programmes – Procédures et Fonctions
- Les types personnalisés
- Les fichiers
- Les listes chaînées
  - Introduction
  - Création d'une liste chaînée
  - Les listes doublement chaînées
  - **Les piles et les files**

Djelloul BOUCHIHA  
[bouchiha.dj@gmail.com](mailto:bouchiha.dj@gmail.com)  
2020-2021

# Les listes chaînées particulières

---

- **Propriétés :**

- ☞ En conditionnant l'accès aux listes, on obtient des listes chaînées particulières.
- ☞ Les listes chaînées particulières sont : les piles et les files.

# Les piles

---

- **Définition :**

- 👉 les éléments ne peuvent être ajoutés ou supprimés qu'à partir d'une extrémité appelée Sommet de la pile.

- **Condition d'accès :**

- 👉 dernier entré, premier sorti. En anglais, on dit : Last In, First Out (LIFO).

- **Exemple (vie quotidienne) :**

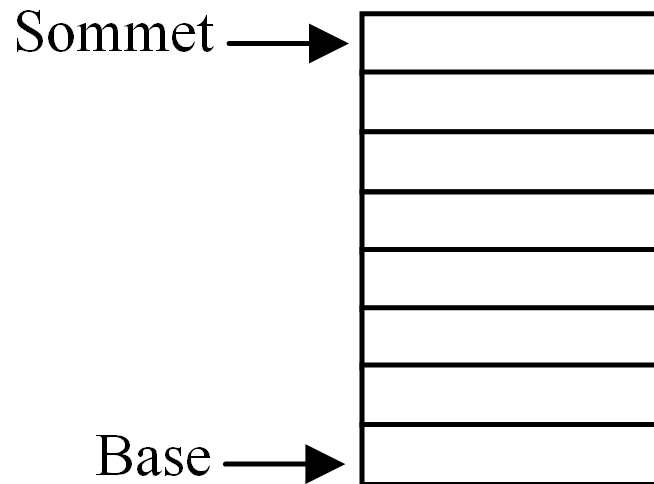
- 👉 la pile d'assiettes.



# Les piles

---

- **Représentation :**



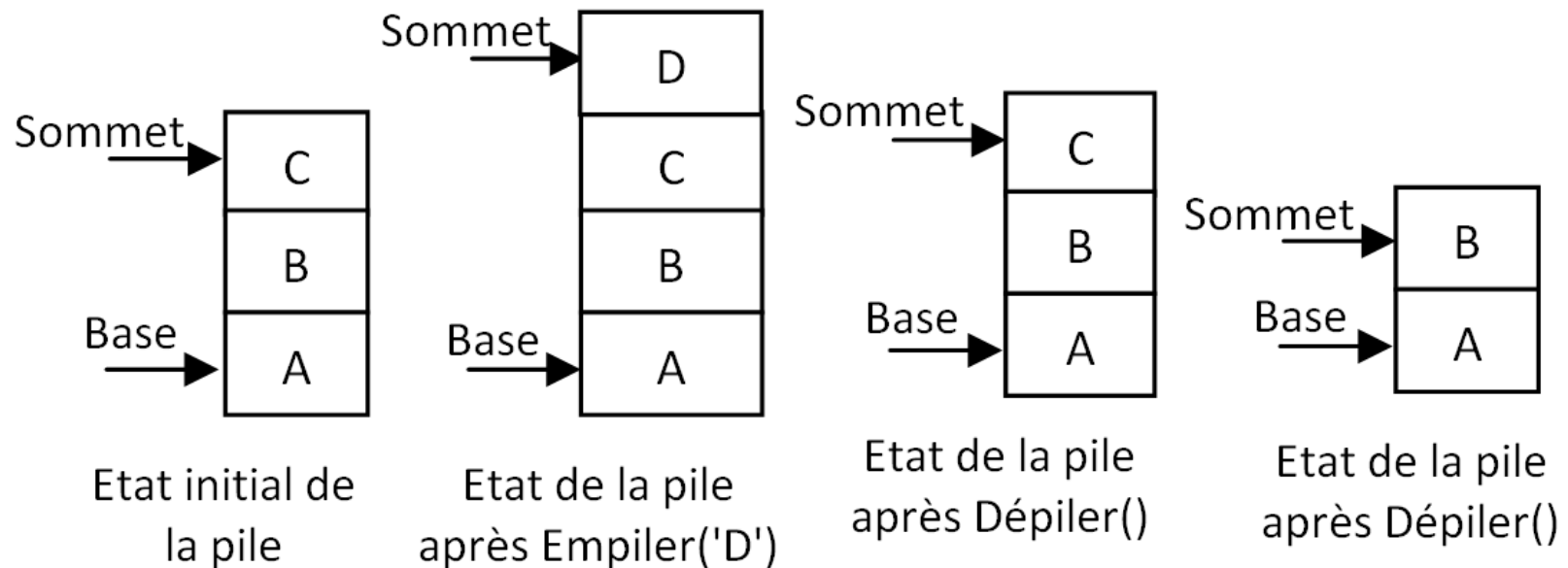
- **Primitives d'accès :**

- ☞ Empiler(E) : ajout d'un nouvel élément E au Sommet de la pile.
- ☞ Dépiler() : supprimer l'élément Sommet de la pile.

# Les piles

- **Exemple :**

- ☞ Soit une pile de caractères.
- ☞ La pile contient initialement les éléments : A, B et C.
- ☞ Empiler encore le caractère D.
- ☞ Ensuite, dépiler deux caractères.



# Les piles

- Implémenter une pile en tant que liste doublement chaînée :

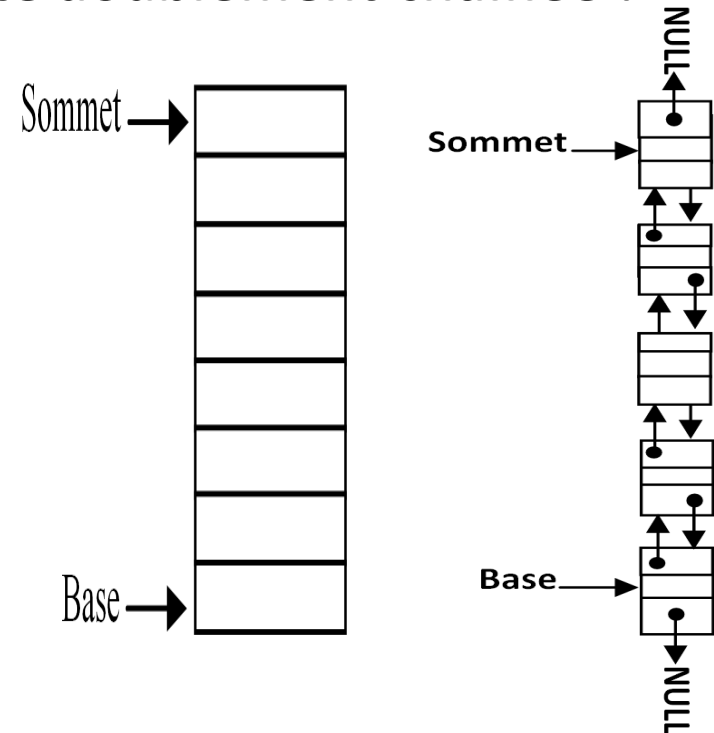
- ☞ Empilement : ajout en Tête.
- ☞ Dépilement : suppression de la Tête.
- ☞ Sommet et Base : Tête et Queue.

- Structure :

```
typedef struct pile
{
    char Car ;
    struct pile *Precedent, *Suivant;
} Pile ;
```

- Déclaration :

```
Pile *Base, *Sommet ;
```



# Les files

---

- **Définition :**

- 👉 les éléments sont ajoutés à une extrémité appelée Queue, et retirés de l'autre appelée Tête.

- **Condition d'accès :**

- 👉 premier entré, premier sorti. En anglais, on dit : First In, First Out (FIFO).

- **Exemple (vie quotidienne) :**

- 👉 la file d'attente à un guichet.



# Les files

---

- **Représentation :**



- **Primitives d'accès :**

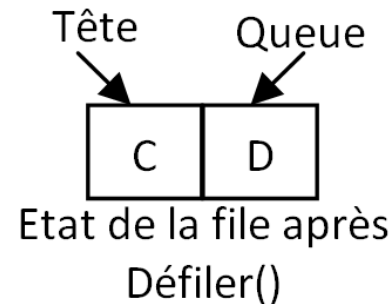
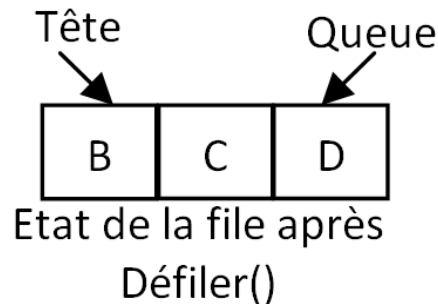
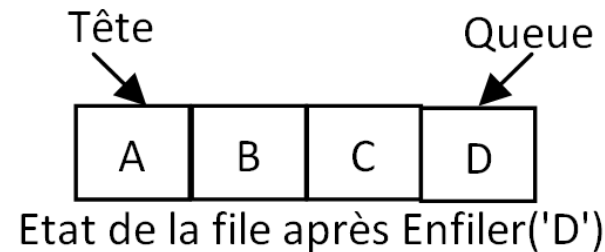
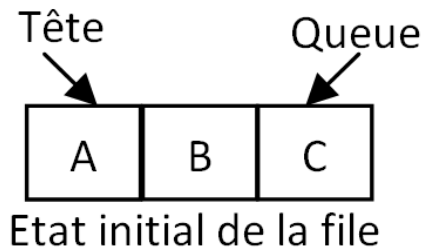
- 👉 **Enfiler(E)** : ajouter l'élément E à la Queue de la file.
- 👉 **Défiler()** : supprimer l'élément Tête de la file.



# Les files

- **Exemple :**

- ☞ Soit une file de caractères.
- ☞ La file contient initialement les éléments : A, B et C.
- ☞ Enfiler encore le caractère D.
- ☞ Ensuite, défiler deux caractères.



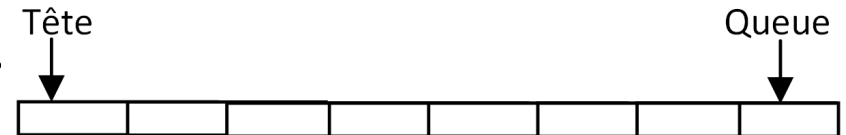
# Les files

- Implémenter une file en tant que liste doublement chaînée :

- 👉 Enfilement : ajout en Queue.

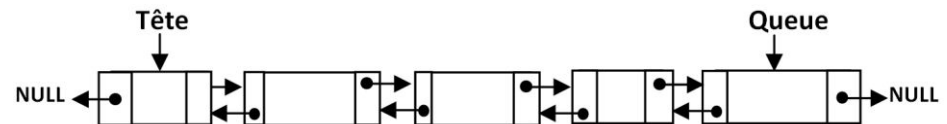
- 👉 Défilement : suppression de la Tête.

- 👉 Tête et Queue : Tête et Queue.



- Structure :

```
typedef struct file
{
    char Car ;
    struct file *Precedent, *Suivant ;
} File ;
```



- Déclaration :

```
File *Tete, *Queue ;
```

---

# Merci