

Notes diverses

Michel Billaud (michel.billaud@u-bordeaux.fr, michel.billaud@laposte.net)

6 novembre 2021

Table des matières

1	Licence	1
2	Polymorphisme en C	1
3	Génération des permutations	2
4	Vi, comment s'en sortir ? (et l'utiliser un peu)	2
5	Les tris	2
6	Réseau	3
7	Développer à partir des tests	3
8	Exercices “somme et moyenne d’un tableau”	3
9	Bases markdown	3

1 Licence



Cette collection de notes est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 2.0 France.

- Les notes sont publiées dans <https://www.mbillaud.fr/notes/>
- Sources dans <https://github.com/MichelBillaud/notes-diverses>

2 Polymorphisme en C

Novembre 2021. On montre sur un exemple détaillé comment réaliser du *polymorphisme* en C

Le code suivant :

```

Animal * animals[] = {
    (Animal *) new_Dog("Medor"),
    (Animal *) new_Fish("Yellow"),
    (Animal *) new_Dog("Rex")
};

for (int i = 0; i < 3; i++) {
    Animal_Talk(animals[i]);
    Animal_Feed(animals[i], "the kibbles"); // croquettes
}

```

exécute, lors des appels d'`Animal_Talk` et `Animal_Feed`, du code différent, qui dépend du type effectif (`Dog` ou `Fish`) des objets. C'est réalisé grâce à une “table de fonctions virtuelles” propres à chaque type.

Document : [polymorphisme-en-c.html](#)

3 Génération des permutations

Novembre 2021. Cette note explique en détail un algorithme classique pour passer d'une permutation, par exemple (3, 5, 2, 4, 1) à la suivante dans l'ordre lexicographique (3, 5, 4, 1, 2).

La technique présentée peut se généraliser à des énumérations d'autres types (note à venir).

Le code est fourni en Fortran 95 (ça m'amuse de l'apprendre, j'en étais resté à la version 77) et en C.

Document : [permutation-suivante.html](#)

4 Vi, comment s'en sortir ? (et l'utiliser un peu)

L'éditeur de textes `vi` est apparu en 1976, il est présent sur la majorité des systèmes Unix (`vim`, version améliorée). C'est parfois le seul éditeur installé au départ, et souvent l'éditeur de textes par défaut. On peut en préférer d'autres, mais il est important de savoir en sortir si on est tombé dedans, et de connaître quelques commandes de bases pour l'utiliser, même si on référerait autre chose.

Document : [vi-utilisation-basique.html](#)

5 Les tris

D'où viennent les idées sur les algorithmes de tri “naïfs” qu'on présente souvent aux débutants : tri par sélection, tri par insertion, tri à bulles ?

Document : [idees-tris-naifs.html](#)

6 Réseau

- Explication de la notion de trame, d'adresse (MAC), et de communication sur un réseau local.

Document : [reseau-local.html](#)

- Notions de routage dans un réseau (interconnexion de réseaux locaux). Adresses IP, préfixes et masques, configuration des interfaces, définition des routes.

Document : [routage.html](#)

7 Développer à partir des tests

Quand on programme, on se trompe souvent. Il est intéressant de prévoir des tests automatisés dès le début.

Illustration avec C.

Document : [dev-tests-c.html](#)

8 Exercices “somme et moyenne d’un tableau”

Exercices très simples qu’on donne souvent à écrire aux débutants. Le problème est qu’ils sont souvent mal posés, sur le plan pédagogique.

On présente une formulation de ces exercices (pour C), et on explique les problèmes courants avec les autres formulations.

Document : [exo-somme-tableau.html](#)

9 Bases markdown

Le strict minimum, pour les débutants.

Document : [bases-markdown.html](#)