Notes diverses

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

9 janvier 2022

Table des matières

1 Licence

- Les notes sont publiées dans https://www.mbillaud.fr/notes/

- Sources dans https://github.com/MichelBillaud/notes-diverses

2 Makefiles pour un projet en C avec un sous-projet

Quand un projet atteint une certaine taille, il arrive qu'on puisse en isoler une "bibliothèque" (sous-projet) qu'on espère réutiliser dans d'autres projets. Ici on montre comment organiser et compiler un projet (en C) qui contient un sous-projet. Et surtout, on explique les `Makefile`s qui vont avec.

Document : makefile-sous-projet.html

3 Sockets IPv4 et IPv6 en C

Janvier 2022. Les cours de programmation réseau/C utilisent obstinément une vieille API, en ignorant complètement IPv6, et les améliorations apportées au langage C depuis sa première normalisation en 1989.

On essaie de faire un peu mieux.

Document : sockets-ipv6.html

4 "Lambdas" en C

Novembre 2021. Comment simuler des "lambdas" en C dans un programme d'énumération des mots de Dyck (systèmes de parenthèses bien formés), qui se base sur un algorithme récursif, fondé sur la grammaire $S \rightarrow \epsilon \| aSbS$.

Document : enumeration-mots-dyck-en-c.html

5 "Factory" en C

Novembre 2021. On montre comment mettre en oeuvre en C le patron de conception "factory", dans lequel une fonction est utilisée pour fabriquer des objets de types distincts.

Document : factory-en-c.html

C'est une continuation de celui sur le polymorphisme en C

6 À propos des Makefiles

Novembre 2021. Récupération de quelques notes d'explication sur l'utilisation des Makefiles sur des projets de petite taille en C. Ces notes ont été écrite pour un cours C/Système en 2018.

Documents

- makefile-gen-dependances.html Un Makefile doit indiquer les dépendances entre les sources et les fichiers produits. La note montre comment l'option `-MMD` de `gcc` évite d'avoir à le faire manuellement.
- makefile-options.html Comment indiquer des options spécifiques pour *certaines* cibles d'un `Makefile`.
- makefile-projet.html Quand un projet contient de nombreux fichiers sources, il est intéressant de produire les fichiers intermédiaires et les exécutables dans des répertoires séparés des sources.

7 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

8 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'amusait de l'apprendre, j'en étais resté à la version 77) et en C.

Document : permutation-suivante.html

9 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

10 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

11 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

12 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

13 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

14 Bases markdown

Le strict minimum, pour les débutants.

Document : bases-markdown.html