

Portfolio de Buzelin Léo

Présentation du portfolio

Commençons par expliquer que j'ai commencé ce semestre avec un niveau en C proche de zéro. Par proche de zéro, j'entends ne jamais avoir fait de C avant.

Au cours de ce portfolio, je présenterais les TPs que j'ai effectué, pourquoi je les ai effectué ainsi que les facilités et les complexités que les différents TPs auront présenté. Il en va sans dire qu'il est possible d'observer au cours des différents TPs joint à ce portfolio que le niveau et la compréhension du C ainsi que de la manière de programmer à grandement évolué.

Étant donné que j'étais nouveau pour la programmation en C, j'ai décidé de suivre les TPs du fil rouge.

TP et compétences

Pour chacun des TPs, les tableaux de compétences attendus au cours du TP seront écrit.

Projet	Projets	I/O	Type	Programme	Module	Compil	Récur	Tableaux	Pointeurs	Structures	Allocation	Fichier	Bit à bit	Fct Pointer	Bibli

TP1 :

Ayant déjà fait de la programmation système et beaucoup de Linux antérieurement à notre formation et aussi au cours du premier semestre de formation à l'Esipe, j'ai décidé de passer ce TP.

TP2 :

TP 2 (exos)	6	1	1	1		1	1					1		
-------------	---	---	---	---	--	---	---	--	--	--	--	---	--	--

Ce TP est le premier TP que j'aie décidé de faire au cours de cette année.

Ayant déjà fait de la programmation, principalement du python et aussi un peu de Java), j'ai pris beaucoup de temps à comprendre comment bien faire le TP et le faire surtout proprement.

Ce TP m'as permis de comprendre les bases de C vu en cours précédemment, même s'il s'agissait d'un des premiers TP de l'année. Durant ce tp, j'ai aussi revu la base pour refaire des Makefiles propres afin de faciliter la compilation et l'efficacité globale.

Les I/Os, ainsi que les types ont été les compétences principales étudiées au cours de ce TP. Pour ce qui est de la compilation, le makefile de ce projet est le premier vrai makefile que j'ai réalisé afin de compiler un programme. Les autres compétences sont des compétences qui ont été facilement acquises bien qu'au début je ne comprenais pas ce qu'était la récursion et que j'ai compris ce dont il s'agissait dans un TP ultérieur.

TP3 :

TP 3 (exos)	6	1	1	1		1	1	1						
-------------	---	---	---	---	--	---	---	---	--	--	--	--	--	--

Ce TP est le premier TP avec des programmes assez complexes à première vu.

Au cours de ce TP, j'ai compris comment manipuler des tableaux, bien que le début, j'ai compris le principe fondamental des tableaux (Plusieurs données stockés à la chaine), pour le reste, il ne s'agissait que de connaissances déjà aborder dans le premier TP.

Une des difficultés que j'ai eues lors de ce TP a été la manipulation de chaine de caractères car, c'est un principe très différent des tableaux de nombres. Par exemple, pour la taille, un tableau de char fait la taille du mot désiré plus un, ce qui

pour moi au début était incompréhensible mais qui est devenu plus clair au bout deux ou trois tests pour calculer la taille des chaînes de caractères.

TP4 :

Je n'ai pas eu le temps de finir proprement ce TP.

TP 4 (pile d'entiers)	6	1	1	1	1	1									1
------------------------------	---	---	---	---	---	---	--	--	--	--	--	--	--	--	---

TP5 :

TP 5 (tableaux fin -1)	7	1	1	1		1	1	1	1						
-------------------------------	---	---	---	---	--	---	---	---	---	--	--	--	--	--	--

Après avoir fait le TP 3, j'ai bien compris comment utiliser les tableaux ce qui m'a grandement facilité la tâche pour ce TP. Il n'y a pas eu de vrai problème au cours de ce TP. I

TP6 :

TP 6 (backtracking)	10	1	1	1	1	1	2	1	1			1			
----------------------------	----	---	---	---	---	---	---	---	---	--	--	---	--	--	--

Pour ce TP, il a d'abord fallu que je comprenne le principe.

C'est-à-dire que je ne comprenais pas le principe de backtracking au début, grâce à d'autres élèves de ma formation, j'ai réussi à comprendre le principe et donc à réussir à conclure ce TP ainsi que réussir à afficher la grille grâce à LIBMLV.

TP7 :

TP 7 (sudoku graphique)	10	1	1	1	1	2		1		1		1			1
--------------------------------	----	---	---	---	---	---	--	---	--	---	--	---	--	--	---

Ce TP m'a posé beaucoup de problème, c'est-à-dire que je n'ai pas réussi à le finir pour ce rendu.

Concrètement, la seule chose qui m'a posé problème est la lib mlv, plusieurs fois pour faire de l'affichage, je ne comprenais pas comment cela fonctionnait. Pour l'affichage de différentes grilles à plusieurs endroits de l'écran. Malgré que je comprenne comment faire l'affichage des différentes grilles, j'ai compris comment fonctionnait le TP et afficher la première grille grâce au TP 6. J'ai donc réussi à avoir la grille afficher depuis les fichiers externes après l'avoir utilisé dans le programme.

Review :

Étant donné mon niveau assez faible au début de l'année, je n'ai pas pu efficacement faire de review pour des élèves de ma promotion, néanmoins, j'ai eu la chance d'avoir une review au sujet de mon TP5 :

Concision et propreté des rendus

Les productions transmises et leurs éléments sont corrects et bien identifiés.

Un seul fichier de source complet regroupant les productions de tous les exos.

Installation et compilation des sources transmises

Un makefile propre, lisible et complet permet la compilation de toutes les sources sans aucun warning

et avec les drapeaux -Wall et -ansi (voire -pedantic). Le makefile gère correctement les dépendances

et intègre une règle pour le nettoyage des produits de compilation.

Les warnings (qui sont des warnings mineurs) viennent du fait que pedantic demande de déclarer l'intégralité des variables locales avant la première ligne de code. Sans pedantic, il y aurait aucun warning avec -Wall -ansi.

Fonctionnalités implantées dans les rendus

Les programmes contiennent quelques preuves de leurs bons fonctionnements.

Complet fonctionnel, main qui fait tourner toutes les fonctionnalités avec une démonstration aléatoires du tri final. Le lecteur finit par vérifier visuellement que le tri fonctionne

Utilisations et sécurité des programmes produits

Tous les cas d'utilisations normales sont traités correctement par les programmes.

Il faut produire des entrées erronées pour casser le programme (une taille nulle pour le tableau à entrée à la main par exemple), sinon c'est toujours fonctionnel.

Nommage des éléments dans les productions rendues

Les identifiants suivent des règles de nommages sur tout le code de manière cohérente.

Une documentation légère et propre dont la syntaxe suit la logique de doxygen...

Efficacité et performance des productions rendues

Les programmes sont basés sur des algorithmes attendues.

Le TP est comme ça, on peut rien inventer de foufou dedans. C'est les algo attendus.

Commentaires dans les sources des productions rendues

Des commentaires précisent les parties critiques du code, levant les ambiguïtés.

derniers commentaires :

C'est pas au programme du premier semestre mais la perspective d'amélioration est d'utiliser l'utilitaire valgrind (sudo apt-get install valgrind) pour chasser les fuites mémoires.

C'est pas l'objectif de ce TP5 mais ce serait en excellent exercice lors des premiers cours de prog C2 sur l'allocation dynamique. Bon travail, rendu efficace.