

# Architecture des systèmes informatiques

## TP 8 : Mini projet (deux séances...)

Félicitations, vous êtes devenu tellement bon en système que vous allez pouvoir remplacer les enseignants. Et, dans ce métier, il y a un moment où il faut faire les corrections.

Pour ce mini projet, des étudiants fictifs (et ils sont encore plus nombreux qu'en L1 à Marne-La-Vallée...), vous ont rendu un fichier par personne. Ils devaient réaliser un programme C relativement simple. Voici l'énoncé de leur exercice :

### EXERCICE 53 :

Écrire un programme C à paramètres qui prend deux entiers en argument, disons 3 et 4, et qui affiche la phrase suivante sur la sortie standard :  
La somme de 3 et 4 vaut 7  
(attention, n'oubliez pas le retour à la ligne à la fin de l'affichage)

Vous devrez rendre les sources de votre programme en un seul fichier nommé `prenom_nom.c` où `prenom` et `nom` sont votre prénom et votre nom. Votre fichier devra être compilable avec les options `-Wall` et `-ansi`.

Le secrétariat pédagogique vient de vous envoyer un mail en disant : “Je vous prie de me rendre les notes des étudiants pour hier !”. Bon courage, mais comme les étudiants sont très nombreux et que vous comptez dormir la nuit, il va falloir automatiser un peu tout cela...

Tous les coups sont permis (du C, du Python, du bash...). Toutefois, il est très vivement conseillé d'utiliser la bibliothèque **subprocess** de Python3. Subprocess permet de lancer d'autres programmes dans un programme python. Sans vendre trop fortement la mèche, on peut, avec subprocess:

- Appeler **ls** et récupérer sa sortie standard pour en faire une liste python de tous les rendus.
- Avec une boucle `for` contenant `subprocess` dedans, appeler **gcc** sur chaque rendu.
- Lancer des tests en exécutant les fichiers fabriqués par **gcc** s'ils existent...
- Compter les lignes de documentation dans chaque rendu.

Le secrétariat attend un fichier CSV qui peut facilement s'ouvrir avec Excel. Chaque ligne du fichier CSV devra contenir les informations relatives à un étudiant. Voici par exemple une ligne à venir du fichier CSV :

```
Ivan,Rochard,1,2,0,0,2,"2,00"
```

Chaque chiffre a un sens dans cette ligne... Après le prénom et le nom, il y a 0 ou 1 : 0 si la compilation ne produit rien et 1 si ça produit un fichier exécutable.

Le chiffre suivant donne le nombre de warning, ici le code source d'Ivan compile avec 2 warnings (levé par `-Wall` ou `-ansi`).

Ensuite 7 tests ont été lancés :

```
./programme 0 0  
./programme 1 0  
./programme 0 1  
./programme 1 1  
./programme 12 12  
./programme 12 -43  
./programme -1 -52
```

Chacune des exécutions a produit un affichage qui a été comparé avec ce qui était attendu en fonctionnement normal. Le chiffre suivant après celui des warnings donne le nombre de tests réussis. Le code d'Ivan ne passe aucun test correctement.

Le chiffre suivant, après les tests, donne le nombre de lignes de documentation dans le code source d'Ivan. Ici, Ivan n'a pas du tout commenté son code.

Les deux derniers chiffres sont des notes et n'ont pas été construits par Python. Une fois le fichier CSV établi avec les premiers chiffres, on a ouvert ce fichier dans un tableur (Excel, openoffice Calc, gnumeric...).

Dans le tableur, on commence par calculer une note de compilation pour chaque élève. Cette note vaut 3 si ça compile et 0 si ça ne compile pas. De plus, chaque warning enlève 0,5 points. Finalement, veillez à ce que cette note reste positive (compilé avec 7 warnings donne 0 et pas -0,5...).

Ivan a ainsi 2 à sa note de compilation car 3 points moins 2 fois 0,5 donnent 2.

La note finale (calculée deux chiffres après la virgule...) est la note de compilation sur 3 plus la note des tests sur 5 (le nombre de tests corrects multiplié par 5/7) plus finalement une note de qualité sur 2. Chaque commentaire rapporte 2/3 de points et avec 3 commentaires, on obtient la note de 2 sur 2 en qualité de code.

Tout cela est fortement fictif mais vous avez normalement les connaissances en système pour faire un tel assemblage. Bon courage.

Pour ce TP 8, sous forme de mini projet et à faire sur deux semaines, vous devrez rendre un compte-rendu décrivant précisément les actions à entreprendre pour construire votre CSV. C'est surtout les programmes que vous allez faire/utiliser qui vont intéresser votre chargé de TP. LA DÉMARCHE EST TOUJOURS PLUS IMPORTANTE QUE LE PRODUIT FINI !!! En particulier ici car le but est de générer des notes fictives. Dans votre rapport, il faudra expliquer surtout comment procéder. Ne décrivez pas ce que vous obtenez, décrivez comment l'obtenir.