

Programmation 1: Laboratoires

Éric Taillard

Filière informatique et sciences de la communication

eric(point)taillard(arrobase)heig-vd.ch

11 septembre 2024

Donnée

● Buts

- Prise en main de l'environnement informatique de la HEIG-VD
- Installation d'un compilateur, si nécessaire
- Maîtriser les bases de la console de commandes

● Travail à réaliser

- Saisir le premier programme « Hello world ! » à l'aide d'un éditeur de texte (Notepad, Notepad++, Wordpad, gedit, etc.) et le sauver sous : `votre_nom_labo1.cpp`
- Pour les utilisateurs du système d'exploitation Windows qui n'ont pas de compilateur g++ installé :
 - Voir page <https://winlibs.com/>
 - Ajouter le chemin ... `mingw64\bin` aux variables d'environnement du système (taper `env` dans la loupe)
 - Ouvrir une console (lancer `cmd.exe`)
- Se déplacer dans la console dans le bon sous-répertoire, compiler et exécuter le programme :
`g++ votre_nom_labo1.cpp -o votre_nom_labo1.exe`
`votre_nom_labo1.exe`

À rendre

- Le code assembleur de votre programme (obtenu avec `g++ -S votre_nom_labo1.cpp`)
- À déposer dans Cyberlearn

Trousse de secours des commandes de la console

| Commande | | Exemple | Effet |
|----------|------|------------------|-----------------------------------------------------|
| unix | DOS | | |
| cd | cd | cd xyz | Descendre dans le sous-répertoire xyz |
| | | cd .. | Remonter d'un répertoire |
| cp | copy | cp toto.txt xyz | Copier le fichier toto.txt dans le répertoire xyz |
| | | cp aa.txt bb.txt | Copier le fichier aa.txt dans le fichier bb.txt |
| ls | dir | ls xyz | Lister le contenu du sous-répertoire xyz |
| man | help | man cp | Manuel de la commande cp |
| mv | move | mv toto.txt xyz | Déplacer le fichier toto.txt dans le répertoire xyz |
| more | type | more toto.txt | Afficher le contenu du fichier toto.txt |

Laboratoire 2

- **But**

- Écrire par imitation un programme de transformation d'unité

- **Donnée**

- Compiler et tester le fonctionnement du programme de transformation d'une longueur exprimée en mètres vers des pieds et des pouces.
- S'inspirer de ce programme pour en écrire un qui transforme une durée, exprimée en secondes uniquement, vers des semaines, jours, heures et secondes
 - Exemple : si on introduit : 4075382 secondes, le programme doit répondre :
4075382 s = 6 semaines, 5 jours, 4 heures, 3 minutes et 2 secondes

- **Qualité du code**

- Comme pour tous les programmes à rendre, un commentaire au début du code doit indiquer quelle est la fonction du programme, qui en est l'auteur, la date et le numéro du laboratoire.
- Utiliser des constantes symboliques nommées plutôt que des nombres « magiques »
- Commenter le code pour en expliquer les principales étapes, sans le charger inutilement
- Choisir des noms d'identificateurs parlants
- Compiler avec les options `g++ -std=c++20 -O2 -Wall -Wextra -Wconversion -Wsign-conversion -Wvla -pedantic`. Ainsi, le compilateur aide à découvrir des erreurs sémantiques. (sous windows, on peut créer un fichier `gppw.bat` contenant `g++ -std=c++20 ... %1 %2 %3` qui mettra à disposition une nouvelle commande nommée `gppw` avec jusqu'à 3 paramètres)

- **Délai**

- Fin de la séance