

La compilation pour les nuls

Assistants: A. Casier & C. Philemotte & F. Servais & B. Verhaegen
Etudiants-assistants: F. Picalausa & B. Poulain & A. Sidis & K. Yagoubi

1. Editer du code

1. Lancer Terminal :
 1. Applications/Accessoires/Terminal
2. Taper `kate` et attendre qu'il se lance (cela peut prendre quelques minute et ne pas fermer le terminal). Attention !! `kate` s'écrit avec une minuscule.
3. Cliquer sur « Open Session »
4. Cliquer sur l'onglet Terminal se trouvant au bas de Kate
5. Agrandir la taille du Terminal en tirant sur le séparateur
6. Si lors de l'édition du code des points gênants apparaissent, il faut désactiver les tabulations :
 1. Aller dans le menu Settings, 2. Configure Kate , 3. Editing,
 4. Décocher la case Show Tabulator
7. Editer votre code et sauvegarder sous un nom de votre choix avec comme extension, ce qui suit le point = « **cpp** » .
Par exemple : `moncode.cpp`
8. Conseil : Eviter les espaces et les caractères spéciaux dans les noms des fichiers.

2. Points importants

1. Ne pas oublier d'inclure les librairies standards
 1. `#include <iostream>` : permet d'accéder aux commande `cin`,
`cout`, ...
2. Utilisation du namespace standard pour pouvoir taper : `cin` au lieu de `std::cin`
 1. `using namespace std;`
3. La fonction main :
 1. `int main(){}`
 2. On peut également utiliser la fonction main suivante : `int main(int argc, char ** argv){}`
`argc` représente le nombre de paramètres envoyés par la ligne de commande et `argv` représente ces paramètres.

3. La Compilation

1. La compilation est nécessaire pour transformer le code source en code exécutable par la machine.
2. Aller dans le Terminal comme par exemple celui se trouvant au bas de Kate.
3. Commande utiles :
 1. `ls` : affiche le contenu du dossier
 2. `cd <nomdurepertoire>` : permet d'entrer dans le répertoire portant le nom `<nomrepertoire>`. (Change Directory)
 3. `cd ..` : permet de sortir du répertoire courant et d'aller dans le répertoire

parent. Remarque : il y a un espace entre cd et ..

4. cd ~ : pour retourner dans son dossier home
 5. rm <nomdufichier> : permet de supprimer un fichier (Remove)
 6. mkdir <nomdurepertoire> : permet de créer un répertoire (Make Dir.)
 7. rmdir <nomdurepertoire> : permet de supprimer le répertoire (si il est vide) (Remove Directory)
4. Pour compiler :
`g++ -Wall -o <nomdufichierdesortie> <nomdefichiercode.cpp>`
 5. -Wall : Active les warnings
 6. -o : Permet de spécifier le nom du fichier de sortie

7. Exécution

1. Aller dans le Terminal et taper :
`./<nomdufichierdesortie>`
2. Lorsque le fichier .cpp est modifié, il faut absolument le recompiler (point 4) avant de l'exécuter, pour voir les modifications lors de l'exécution.

8. Exemple

1. Mon code est enregistré sous "tutorial.cpp"
2. Pour compiler, la commande est :
`g++ -Wall -o tutorialcompiled tutorial.cpp`
3. Pour exécuter, la commande est : `./tutorialcompiled`

9. Connexion aux PCs des salles informatiques à distance

1. La connexion se réalise au travers un client ssh ou sftp.
2. 3 adresses où X est le numéro du PC compris entre 0 et 30 à prendre au hasard.
 1. Socrate ==> roxanex.ulb.ac.be
 2. Aristote ==> romeoX.ulb.ac.be
 3. Descartes ==> sca-xtX.ulb.ac.be
3. Login et mot de passe ceux de la boîte mail (remarque votre login ne contient pas de @ulb.ac.be)
4. Pour linux et Mac OS : > ssh login@romeo13.ulb.ac.be
5. S'il y a un problème avec les PC contacter M. Johnny Tsheke Shele (UA4.221/ Ext : 4096 / Mail : johnny.tsheke.shele@ulb.ac.be)

10. Sitewebs utiles :

1. <http://www.magellan.fpms.ac.be/linux/introduction-au-c>
2. <http://cs.ulb.ac.be/public/teaching/infoh100>

11. Logiciels utiles

1. Kate : éditeur sous Linux
2. DevCpp : éditeur de code et compilateur sous Windows
3. WinScp : permet de récupérer des fichiers de la salle info ou d'en envoyer à partir d'internet sous Windows
4. FileZilla : même chose que WinScp mais il faut qu'il soit en mode SFTP (Win/Mac)
5. putty : client SSH sous windows pour se connecter à distance aux salles