Aide-mémoire – Salles informatiques et guidances

Les bonnes adresses

- Problème technique ? Johnny Tsheke Shele (UB4.227 – extension 4906)
- Besoin de crédit papier ? Secrétariat dans le SMG, UA4.110 (pour vérifier le crédit : http://cs.ulb.ac.be/tools)
- Pas d'adresse @ulb.ac.be ? https://debby.ulb.ac.be/pam/pamlogin.php
- Nous contacter ? En guidance (Salle Socrate – UB4.130 de 12h30 à 13h30). Nous portons un tshirt ULB.

Terminal

Dans le menu Application/Accessoires, sélectionner Terminal

Commande	Signification
`ls`	List : Liste les fichiers du dossier courant
`cd` *dossier*	Change directory : Change le *dossier* courant
`mkdir` *nom*	Make directory : Crée un dossier avec le *nom* indiqué
`rm` *fichier*	Remove : Supprime le *fichier* spécifié
`rmdir` *dossier*	Remove Directory : Supprime le *dossier* spécifié.
`more` *fichier*	Affiche le contenu du fichier spécifié.

Exemples avec cd

`cd Pictures` : va dans le dossier Pictures
`cd ..` : va dans le dossier parent (si on est dans /toto/titi/tata, va dans le dossier /toto/titi)
`cd ~` : revient dans le « home directory »

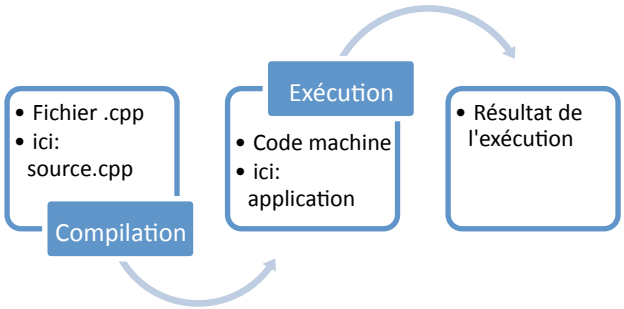
Editeur de texte

Dans un terminal, écrire `kate` et presser ↵ (Enter/return).

- Kate contient un terminal, mais il faut éventuellement tirer sur le séparateur pour l'afficher.
- Pour désactiver les points apparaissant dans le code, dans le menu Settings, sélectionner Configure Kate, Editing, et décocher la case Show Tabulator.
- Pour afficher les numéros de lignes, appuyer sur F11.

Compilation

- Compilation : **`g++ -Wall -o application source.cpp`**
- Exécution : **`./application`**
- Changer le fichier .cpp nécessite de recompiler, exécuter, pour que le résultat de l'exécution change.



Travailler de chez soi

- Code::Blocks : Pour écrire des fichiers C++ et les compiler chez vous. http://www.codeblocks.org/downloads/5

Squelette de programme

```cpp
#include <iostream>
using namespace std;

int main(int argc, char** argv)
{
	/* votre programme ici */
	cout << "Hello World" << endl;
	/* fin du programme */

	return 0;
}
```

Style

```cpp
/* Nom:               exp
 * Description:       Réalise l'exponentiation
 * Arguments:  x:     la base
 *             n:     l'exposant
 */
int exp(int x, int n)↵
{↵
→	/* maintient le résultat de l'opération */↵
→	int result = 1;↵
↵
→	for (int i = 0; i < n; i++)↵
→	{↵
→	→	result = result * x;↵
→	}↵
↵
→	return result;↵
}
```

Déclaration de variables

Déclaration	Signification
`int i;`	Déclare une variable entière, i
`double r;`	Déclare une variable réelle, r
`bool isRunning;`	Déclare une variable booléenne {true , false}, isRunning
`char digit;`	Déclare une variable capable de contenir un caractère de texte, digit
`int vector[30]`	Déclare un vecteur de 30 entiers
`char name[50];`	Déclare une chaine de caractère de longueur maximale 50
`int matrix[20][15];`	Déclare une matrice de 20 x 15 entiers

Structures de contrôle

```cpp
if ((a == 5) && (b == 0))
{
   /* execute ceci si a
     vaut 5 et b vaut 0 */
}
else if (b == 3)
{
   /* execute ceci si b vaut 3 */
}
else
{
   /* execute ceci dans
     les autres cas */
}
```

```cpp
while (c == 6)
{
    /* tant que c vaut 6 */
}

do
{
    /* tant que d vaut 3
     en sortie de boucle.
     Exécuté au moins une fois */
}
while (d == 3);
```

```cpp
for (int i = 0; i < 10; i++)
{
    /* effectue la boucle
    pour i allant de 0 à 9 */
}
```

Passage de paramètres

Par valeur

```cpp
int increment(int a)
{
	a = a + 1;
	return a;
}

int a = 5;
int b = increment(a);
/* affiche 5, 6 */
cout << a << ", " << b << endl;
```

Par référence

```cpp
int increment(int &a)
{
	a = a + 1;
	return a;
}

int a = 5;
int b = increment(a);
/* affiche 6, 6 */
cout << a << ", " << b << endl;
```

Interdits

Variables globales, goto, code pas/mal indenté, etc.