

Algorithmique et Programmation 1

IMAC 1ere année

TP 7

Paramètres de `main`, `string.h`, code modulaire, `makefile`, bibliothèque graphique MLV

Dans cette séance de travaux dirigés, les points suivants sont abordés :

- les paramètres de `main`;
 - les fonctions du `string.h`;
 - compilation de plusieurs fichiers source ;
 - automatisation de la compilation avec “make” ;
 - la bibliothèque graphique MLV.
-

Comme lors des séances précédentes, pensez à faire régulièrement des sauvegardes (dans des fichiers distincts) et à documenter votre code. Pensez à toujours tester votre code. À la fin de la séance, mettez les sources dans une archive (commande `tar -zcvf nom_archive fichiers_source`) et rendez-le suivant les instructions données sur e-learning.

Pensez à consulter les transparents de cours sur elearning régulièrement.

Exercice 1. (Paramètres de `main`)

Rappel : pour pouvoir traiter les arguments fournis à votre programme, utiliser le prototype de `main` suivant : `int main(int argc, char *argv[])`. Consulter les transparents du cours pour plus de détail.

1. Écrire un programme `repeter` qui affiche le nombre d’arguments qui lui ont été fournis et qui les répète, un par ligne.
2. Écrire un programme `multiplier` qui, si on lui fournit en argument deux nombres, affiche leur produit. `multiplier 12 4` affichera 48.

À savoir : tous les arguments sont stockés dans le tableau `argv` comme des chaînes de caractères. Pour les interpréter comme entiers, on peut utiliser la fonction `sscanf`. Elle fonctionne comme `scanf`, mais au lieu de travailler avec l’entrée du clavier, elle permet de lire n’importe quel chaîne de caractères (qui est passé en premier argument).

Alors `sscanf(monchaine, "%d", &var)`; tentera de lire un entier écrit dans la chaîne `monchaine` et stocker le résultat dans `var`.

Exercice 2. (`string.h`)

Familiarisez-vous avec les fonctions contenus dans `string.h`. Comme pour beaucoup de fonctions qui vous sont disponibles, leur documentation peut être trouvé dans la section 3 du manuel linux. (tester `man 3 string`, `man 3 strcmp`, ..., noter que `sscanf` y est documentée aussi)

1. Écrire un programme qui, si on lui fournit en argument deux chaînes de caractères, annonce, si le premier est contenu dans l'autre ou non. En cas d'autre nombre d'arguments, un message d'erreur est affiché.
2. (seulement s'il vous reste plus qu'une heure)
Écrire un programme **nb_occurences** qui, si on lui fournit en argument deux chaînes de caractères, calcule le nombre d'occurences de la première chaîne dans la deuxième. Si un autre nombre d'arguments est fourni, un message d'erreur est affiché.