



Laboratoire DID@VDI

Programmation en langage C sur PC clients



Pré requis

Matériel nécessaire :

- Laboratoire DIDAVDI+
- Station étudiante complète

Durée : 32 heures

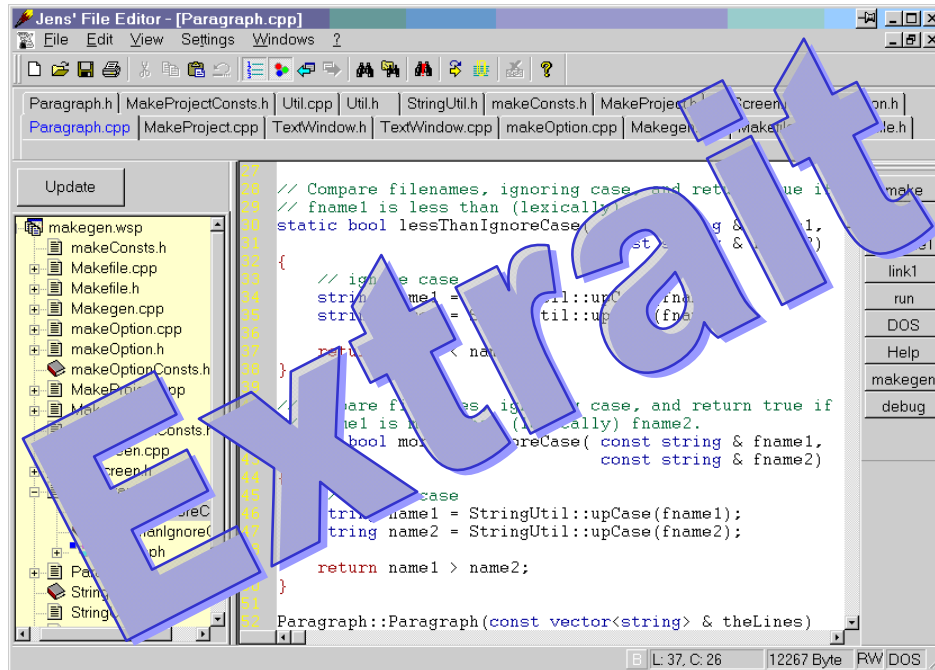
Sommaire

1	Compilateur Gcc AsheWare.....	5
1.1	Présentation	5
1.2	Prise en main :	5
1.2.1	Installation de GccAsheWare	5
1.2.2	Exemple du fonctionnement.....	6
1.3	Fonctionnement d'un compilateur C	7
1.3.1	Exécution.....	7
1.3.2	Les étapes de la compilation.....	8
1.3.3	Un exécutable est spécifique à un système.....	11
1.4	Application (TD)	12
2	Compilation de plusieurs fichiers C	13
3	Pour aller plus loin	15

2 Compilation de plusieurs fichiers C

De nombreux logiciels proposent de mettre en œuvre des projets comprenant plusieurs fichiers c et h.

Exemple :



La puissance de ce type de logiciel est qu'il suffit de faire un #include des fichiers h dans le fichier c principal (celui qui contient le main).

Ensuite l'utilisateur crée ses fichiers h de prototypes, et ses fonctions utiles dans différents fichiers c.

Enfin, il clique sur compile pour que le compilateur "en ressorte" un fichier exécutable.

Nous allons maintenant nous intéresser à la manière dont le compilateur arrive à compiler tous ces fichiers.

Pour illustrer notre mise en œuvre, nous allons prendre la problématique suivante :

Nous souhaitons déterminer les racines réelles d'un polynôme du second degré : $ax^2 + bx + c = 0$.

Pour cela il faut déterminer le discriminant (nommé aussi delta) : $\Delta = b^2 - 4ac$.

L'équation admet :

- deux solutions ; $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$ et $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$ si $\Delta > 0$
- une solution double : $x_1 = x_2 = -(b/2a)$ si $\Delta = 0$
- N'a pas de solution réelle (si delta négatif)

Nous imposons de créer 3 fichiers :

- equation.c : fonction de calcul de la racine carrée, et du résultat
- equation.h : prototypes des fonctions du fichier equ.c
- equation_main.c : fichier principal contenant le main, affichera l'équation, et résultat du delta, et la (les) solution(s).

Réalisez les fichiers avec les contraintes imposées.

Proposez une solution pour la compilation des fichiers.

Vérifier le fonctionnement avec la commande « ./Fichier_de_sortie »

Concluez.

Pour compiler notre programme, nous pouvons exécuter la ligne

« gcc -Wall equation.c equation_main.c -o Appli_equation »

Nota : l'option -Wall permet de voir les warning éventuels

Pour respecter les note d'application, la vraie méthode de compilation est la suivante :

Pour compiler notre programme simple, il faut d'abord compiler chaque source avec l'option -c (le fichier obtenu sera un fichier *objet*, avec l'extension *.o*), puis assembler les deux fichiers objets pour faire l'exécutable :

```
gcc -c equation.c
```

```
gcc -c equation_main.c
```

```
gcc -Wall equation.o equation_main.o -o Appli_equation
```

On se rend vite compte qu'une compilation de deux fichiers c implique une ligne de compilation relativement lourde. Par conséquent, plus nous avons de fichiers dans le même projet, plus la ligne s'allonge.