

TD5: Compilation et Makefile

13 novembre 2014

Exercice 1. Écrire un fichier Makefile

Créer un répertoire **Exo1**. Récupérer le fichier **exemple_decoupe.tar** sur e-campus et copier-le dans le répertoire **Exo1**. Décompresser l'archive à l'aide de la commande **tar xvf exemple_decoupe.tar**.

- a. Faites un schéma de dépendances entre les différents fichiers.
- b. Ecrivez dans un fichier **makefile** les commandes qui permettent de compiler chacun des fichiers séparément.
- c. Rajouter une règle qui permet de faire l'édition de liens entre les fichiers objets et de créer un fichier exécutable.

Exercice 2. Écrire un ensemble de fichiers et un fichier Makefile

On désire écrire un programme pour calculer la circonférence et la surface d'un cercle étant donné son rayon. Le programme doit être composé de plusieurs fichiers :

principal.c	contient la fonction main
circonference.c	contient la fonction circonference
surface.c	contient la fonction surface
pi.h	contient la définition de la valeur approchée de π

- a. Écrivez le contenu des différents fichiers décrits ci-dessus.
- b. Ecrivez le contenu des fichiers **conference.h** et **surface.h**.
- c. Faites un schéma de dépendances entre les différents fichiers.
- d. Écrivez ensuite le fichier **Makefile** permettant de compiler chacun des fichiers **.c** séparément et ensuite de faire l'édition de liens entre les différents fichiers objets.
- e. Ajoutez une cible **clean** dans le **Makefile** pour supprimer tous les fichiers intermédiaires ainsi que le fichier résultat.

Exercice 3. Avec du graphisme

Récupérez l'archive **graphisme.tar** qui contient un fichier **main.c**, l'ensemble des fichiers nécessaires à l'utilisation de la bibliothèque **graphics.c** et un **Makefile** qui permet de compiler cette bibliothèque.

- a. Complétez le **Makefile** pour que le programme puisse être compilé. Pour cela, la librairie **SDL** doit être incluse au moment de l'édition de lien avec la commande

```
'sdl-config --libs' -lSDL_ttf
```

- b. Ajouter une règle **run** qui permet de lancer l'exécution du programme. Celui-ci fait les choses suivantes (lire le code pour le constater) : à chaque clic dans la fenêtre graphique, un point est ajouté dans une liste et affiché relié aux points précédents jusqu'à ce que **NB_POINTS** soient affichés. A tout moment, en appuyant sur la touche **a**, on peut supprimer le dernier point ajouté.

- c. Découper le fichier **main.c** dans 3 fichiers :

- un fichier **liste.c** qui contient les opérations sur les listes, y compris l'affichage dans la fenêtre graphique ;

- un fichier `simul.c` qui contient la fonction qui génère la simulation ;
- un fichier `main.c` qui contient la fonction principale `int main()`.

Vous ajouterez les fichiers d'en-tête nécessaires. Les constantes (`#define`) et les définitions des structures seront également placées à des endroits adaptés.

d. Ajoutez les règles `all` et `clean` dans le `Makefile`.

e. Dans la fonction de simulation, vous ajouterez les fonctionnalités suivantes :

- quand la touche `z` est appuyée, le point *le plus ancien* qui a été ajouté est supprimé ;
- quand une flèche est appuyée, le dessin de la chaîne se décale d'un pixel dans la position correspondante ;
- quand la touche `t` est appuyée, les points sont triés par ordre d'abscisse dans la fenêtre graphique.