

TP makefile

1. Documentation de référence

Les docs en ligne :

<http://bat710.univ-lyon1.fr/~jciehl/Public/educ/Makefile.html>

Make est outil très général permettant, entre autre, d'automatiser la compilation d'un projet.

2. Compilation d'un projet

Supposons que le projet soit constitué des sources C suivants : main.c et file.c

2.1) Compiler en une seule commande :

```
gcc -o prog -Wall main.c file.c
```

2.2) une commande pour chaque fichier source et une autre pour créer l'exécutable :

```
gcc -c -Wall main.c
gcc -c -Wall file.c
```

```
puis :
gcc -o prog main.o file.o
```

2.3) Utilisation d'un make :

Son rôle est de produire automatiquement la séquence de commandes permettant de construire un projet.

Il faut d'abord définir les dépendances (ou les relations) entre ces fichiers :

File.c produit file.o

main.c produit main.o

et

main.o et file.o permettent de construire le projet prog.

Il suffit de décrire ces relations à Make, ainsi que les commandes associées pour qu'il puisse produire la séquence de commandes correcte. Ces relations sont décrites dans un fichier texte, nommé Makefile ou makefile.

Une relation s'écrit de la manière suivante dans le fichier makefile :

```
produit : source
        commande
```

Le fichier makefile s'écrit de la manière suivante :

```

prog: main.o file.o
    gcc -o prog main.o file.o

main.o : main.c
    gcc -c -Wall main.c

file.o : file.c
    gcc -c -Wall file.c

```

Lors d'une modification d'un fichier, make se base sur la date de modification des fichiers pour déterminer les mises à jours à effectuer.

2.4) Utilisation d'un make avancé:

L'utilisation des variables permet l'écriture simplifier d'un makefile.

Pour définir une variable, il suffit de lui donner un nom et une valeur.

Pour substituer ensuite une variable à sa valeur, il suffit d'écrire \$nomvariable.

Exemple :

```

##### makefile
CC= gcc      # compilateur
CFLAGS= -Wall # options de compilation pour les sources C
NAME=prog    # nom du fichier binaire final

# liste des fichiers .c nécessaires au projet
sources= main.c file.c

# nommage automatique des fichiers objets d'après les noms des sources C
objets= $(sources:.c=.o)

$(NAME) : $(objets)
    $(CC) -o $@ $^ -lm

#Cette règle indique que chaque fichier .o nécessaire à la création du projet est obtenu à partir
#du fichier .c correspondant et que la commande $(CC) -o $@ -c $< permet de créer le
#fichier objet à partir du fichier source
%.o: %.c
    $(CC) $(CFLAGS) -o $@ -c $<
##### fin makefile

```

Signification :

\$@ : produit (ou but) de la règle

\$< : nom de la première dépendance (ou source)

\$? : toutes les dépendances plus récentes que le but

\$^ : toutes les dépendances

Si on veut rajouter un fichier .c dans le projet, il suffit de le rajouter dans la liste des sources.