

Annexe : Makefile

1. Introduction
2. Principe de base : Les Règles
3. Règles de modèles
4. Définitions de variables
5. Variables prédéfinies

Introduction

Le flot de conception proposé par la chaîne de CAO Alliance se décompose en plusieurs étapes. Chaque étape est réalisée par un outil CAO, et les différents outils communiquent entre eux par des fichiers.

Les dépendances de données entre les différentes étapes sont donc matérialisées par des dépendances entre fichiers. L'outil **make** de UNIX permet de décrire ces dépendances, et donc d'automatiser l'enchaînement des étapes de conception.

Le fichier Makefile est un *script* décrivant les dépendances, dont l'exécution est lancée par la commande *make*.

Principe de base : Les Règles

Un Makefile est un fichier contenant une ou plusieurs règles traduisant les dépendances entre les actions et les fichiers.

Voici une règle type Makefile :

```
#Rq: chaque commande doit être précédée d'une tabulation
cible1 : dépendance1 dépendance2 ....
    commande_X
    commande_Y
```

Les dépendances et cibles représentent, en général, des fichiers. Seule la première règle du Makefile est examinée. Les règles suivantes sont ignorées si elles ne sont pas impliquées par la première. Si certaines dépendances d'une règle X sont elles-mêmes des règles dans le Makefile alors ces dernières seront examinées avant la règle X appelante. Pour chaque règle X examinée, si au moins une de ses dépendances est plus récente que sa cible alors les commandes de la règle X seront exécutées.

Remarque : les commandes servent généralement à produire la cible (i.e un nouveau fichier). Une cible peut ne pas représenter un fichier. Dans ce cas, les commandes de cette règle seront toujours exécutées.

Règles de modèles

Ces règles sont plus polyvalentes car vous pouvez spécifier des règles de dépendance plus complexes. Une règle de modèle ressemble à une règle normale, sauf qu'un symbole (%) apparaît dans le nom de la cible. Les dépendances emploient également (%) pour indiquer la relation entre les noms de dépendance et le nom de la cible. La règle de modèle suivante spécifie comment tous les fichiers vst sont formés à partir des vbe.

```
#exemple de règle pour la synthèse
%.vst : %.vbe
    boog $*
```

Définitions de variables

On peut définir des variables en n'importe quel endroit du fichier Makefile, mais une écriture lisible nous amène à les définir en début de fichier.

```
#définitions de variables
```

```
MY_COPY = cp -r
```

```
MY_NUM = 42
```

```
MY_STRING = "hello"
```

```
#Elles sont utilisables à n'importe quel endroit du Makefile. Elles doivent être précédées du ca
```

```
#utilisation d'une variable dans une règle
```

```
copie:
```

```
    ${MY_COPY} digicode.vbe tmp/
```

Variables prédéfinies

- **\$@** Nom complet de la cible.
- **\$*** Nom du fichier cible sans l'extension.
- **\$<** Nom du premier fichier dépendant.
- **\$+** Noms de tous les fichiers dépendants avec des dépendances doubles répertoriées dans leur ordre d'apparition.
- **\$^** Noms de tous les fichiers dépendants. Les doubles sont retirés.
- **\$?** Noms de tous les fichiers dépendants plus récents que la cible.
- **\$%** Nom de membre pour des cibles qui sont des archives (langage C). Si, par la cible est libDisp.a(image.o), **\$%** est image.o et **\$@** est libDisp.a.