

Makefile et L^AT_EX

TP n°4 - Hoerdet Mickaël/Quirin Arnaud - Mars 2004

Note : Dans les cas désespérés, de l'aide se trouve dans
<https://dpt-info.u-strasbg.fr/%7EQuirin/make.htm>

1 Exercices Makefile

- Récupérer les fichiers `global.c`, `valeur.c`, `mult.c`, `produit.c` et `defs.h` dans le dossier `/users/prof/hoerdet/GL/TP3/`. Ces fichiers constituent un programme permettant de faire du calcul sur des polynômes à coefficients entiers. L'édition des liens nécessite la librairie mathématique (`-lm`). La fonction `main()` se trouve dans `global.c`.

1.1 Exercice 1

- Ecrire un fichier `makefile` permettant de générer le programme exécutable ainsi que les cibles `valeur.o`, `mult.o` et `produit.o`.

1.2 Exercice 2

- Demander l'exécution d'une cible autre que le programme exécutable final. Exemple : `mult.o`
- Demander la génération du programme exécutable.
- Introduire une cible "clear" dans le fichier `makefile` pour détruire tous les `.o` (instruction `rm`). Exécuter `make` en appelant cette cible `clear`
- Supprimer l'écho de certaines instructions dans le fichier `makefile` (@).

1.3 Exercice 3

- Après avoir supprimé le programme exécutable et/ou des fichiers `.o`, faites appel à `make`
- en mode non exécutoire (option `-n`)
 - avec demande d'explications (option `-d`)
 - mode silencieux (option `-s`)

1.4 Exercice 4

- Modifier le fichier `defs.h` pour réduire la valeur de `N` à 5. L'exécution de `make` a-t-elle recompilé vos fichiers ? Si non, pourquoi ?

1.5 Exercice 5

- Comparer vos dépendances pour `produit.o`, `valeur.o` et `mult.o` par rapport à celles proposées à l'aide des options `-M`, `-MM` et `-H` du pré-processeur.

1.6 Exercice 6

- Modifier la fonction `saisie` pour que les coefficients soient demandés par ordre décroissant des puissances (et non plus par ordre croissant). Quelles compilations sont exécutées ?

1.7 Exercice 7

- Mettre le fichier `produit.c` dans un autre répertoire. Modifier la règle de génération de `produit.o` pour y inclure une instruction `cd`.
- Mettre `defs.h` dans un répertoire autre que `/usr/include` et le répertoire courant. Précisez le chemin d'accès à l'aide d'une macro.
- Utiliser des macros pour définir le compilateur, les options de compilation et les librairies. Modifier les règles pour faire référence à ces macros (par exemple : `CC`, `OPTS`, `LIBS`).
- Créer une macro `OBJS` qui comporte tous les fichiers `.o` nécessaires à la génération du programme exécutable. Faire référence à `OBJS` dans la liste des dépendances.

2 Exercices Makefile et L^AT_EX

Le traitement de texte par `LATEX` consiste :

- à saisir un texte source (`.tex`) comprenant le texte lui-même ainsi que des commandes de mise en forme, puis
- à compiler ce texte source par la commande "`latex nom_fichier.tex`" qui génère le fichier `nom_fichier.dvi`. Le fichier `nom_fichier.dvi` est un fichier image visualisable par la commande "`xdvi nom_fichier.dvi`".
- à générer un fichier `postscript` imprimable par la commande : "`dvips nom_fichier.dvi -o nom_fichier.ps`". Ce fichier peut être visualisé par la commande "`gv nom_fichier.ps`".

Copier dans votre répertoire le fichier `/users/prof/hoerdet/GL/TP3/sources.tar` puis exécuter la commande "`tar -xf sources.tar`". Un répertoire "sources" sera créé contenant les fichiers nécessaires au TP. Ces fichiers ont tous une extension `.gz` car ils ont été compressés par l'outil "gzip". Ces fichiers sont constitués de :

- `texte.tex.gz` : fichier texte source principal contenant des commandes d'inclusions des autres fichiers sources (`intro` et `chap1` à `chap4`)
- fichiers d'extension `.eps.gz` : fichiers d'images incluses dans le texte.

Ecrire un `makefile` dans le répertoire courant (parent du répertoire sources) permettant de :

1. décompresser le fichier d'extension `.tex.gz` et de les placer dans le répertoire courant. La commande de décompression de fichier est "`gzip -cd nom_fichier.gz > nom_fichier`". Indication : définir une règle implicite prenant en compte tous les fichiers.
2. compresser les fichiers `.tex` du répertoire courant et de les placer dans le répertoire sources. La commande de compression de fichier est "`gzip nom_fichier`".
3. compiler le fichier texte principal `texte.tex` et de générer le fichier de sortie `texte.dvi`. La compilation devra en priorité utiliser les fichiers `.tex` du répertoire courant et à défaut les fichiers compressés du répertoire sources.
4. générer le fichier imprimable `postscript`.
5. pour ces deux dernières cibles, de visualiser le fichier final à la demande de l'utilisateur (appel à `make` avec redéfinition d'une variable).
6. nettoyer le répertoire courant (effacement des fichiers `.aux`, `.log`, `.toc`, `.lof`, `.eps`).