**MAKEfile**

Les Makefiles sont des fichiers, généralement appelés makefile ou Makefile, utilisés par le programme make pour exécuter un ensemble d'actions, comme la compilation d'un projet, l'archivage de document, la mise à jour de site, etc.

Un Makefile est un fichier constitué de plsrs règles de la forme :

Cible : dependance

Commandes

**MAKEFILE MINIMAL**

Une image contenant texte, plante

Description générée automatiquement

Dans cet exemple, Nous cherchons à créer le fichier exécutable hello, la première dépendance est la cible d'une des règles de notre Makefile, nous évaluons donc cette règle. Comme aucune dépendance de hello.o n'est une règle, aucune autre règle n'est à évaluer pour compléter celle-ci.  
Deux cas se présentent ici : soit le fichier hello.c est plus récent que le fichier hello.o, la commande est alors exécutée et hello.o est construit, soit hello.o est plus récent que hello.c est la commande n'est pas exécutée. L'évaluation de la règle hello.o est terminée.  
Les autres dépendances de hello sont examinées de la même manière puis, si nécessaire, la commande de la règle hello est exécutée et hello est construit.

**MAKEFILE ENRICHI**

L’exemple du dessus ne permet pas de générer plsrs éxécutables distincts, les fichiers intermediaires restent sur le disque dur même lors de la mise en production et il n’est pas possible de forcer la régénération intégrale du projet.

🡪 il faut ajouter des règles complémentaires :

-all : généralement la première du fichier, elle regroupe dans ces dépendances l'ensemble des exécutables à produire

-clean : elle permet de supprimer tous les fichiers intermédiaires.

- mrproper : elle supprime tout ce qui peut être régénéré et permet une reconstruction complète du projet.

Une image contenant texte

Description générée automatiquement

**LES VARIABLES**

Il est possible de définir des variables dans un Makefile, ce qui rend les évolutions bien plus simples et plus rapides, en effet plus besoin de changer l'ensemble des règles si le compilateur change, seule la variable correspondante est à modifier.  
Une variable se déclare sous la forme NOM=VALEUR et se voir utiliser via $(NOM).  
Nous allons donc définir quatre variables dans notre Makefile :

* Une désignant le compilateur utilisé nommée CC (une telle variable est typiquement nommée CC pour un compilateur C, CXX pour un compilateur C++).
* CFLAGS regroupant les options de compilation (généralement cette variable est nommée CFLAGS pour une compilation en C, CXXFLAGS pour le C++).
* LDFLAGS regroupant les options de l'édition de liens.
* EXEC contenant le nom des exécutables à générer.

Une image contenant texte

Description générée automatiquement

**VARIABLES INTERNES**

**Une image contenant table

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

Options de compilateur :

-c : indique de ne pas linker

-v : affichera tout les details

-I : specifie le ou les repertoires des headers

-Ox : indique le degré d’optimisation (x varie de 1 à 6)

-w : supprime les avertissements

-Wall : GCC est très exigeant

-Werror : tt warning devient erreur : pour compiler parfaitement

-g : mode Debugger

(man gcc : affichera les tous)

https://gl.developpez.com/tutoriel/outil/makefile/