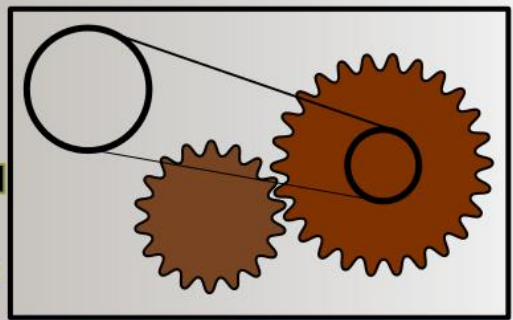


MAKE

Make accélère le développement



Les bases de GNU Make

`make` est un élément central de la construction de paquets composés de code compilé ou de documentation sur un système LINUX. Quand `make` s'exécute, il lit un ou plusieurs fichiers texte contenant des cibles et des dépendances. Les cibles ne sont exécutées que si un fichier n'est pas présent ou s'il est plus récent que le fichier de sortie. La syntaxe autorise également des dépendances multiples, de telle sorte que la construction de quelques fichiers se fasse obligatoirement avant d'autres. Commençons avec un premier exemple :

```
# Crée newFile.txt s'il n'est pas présent
newFile.txt:
    touch newFile.txt
```

À l'instar des langages de script, les commentaires débutent par un caractère dièse (#). Il n'y a pas d'espace ni de tabulation devant la cible `newFile.txt` et elle est suivie par un caractère deux-points. Des dépendances pour la cible peuvent être ajoutées après le deux-points. Les actions sont placées sur des lignes après le nom de la cible et doivent être préfixées par un caractère de tabulation. Si des espaces blancs sont utilisés, `make` signalera une erreur.

Par défaut, `make` recherche un fichier appelé `Makefile`. Donc, utilisons `nano` (décrit dans l'article sur le C du numéro 3) pour créer un fichier nommé `Makefile` et contenant l'exemple ci-dessus. Ensuite, tapez `make`

La première fois que la commande `make` est invoquée, elle exécutera la cible, qui utilise la commande `touch` pour créer un fichier vide avec le nom spécifié. La seconde fois que `make` est appelé, il se rend compte que le fichier existe déjà, et aucune action ne sera nécessaire.

Un petit programme en C peut être utilisé pour démontrer les dépendances. Dans un nouveau répertoire, créez trois fichiers :

`main.c`

```
#include "printString.h"
int main() {
    printString();
    return 0;
}
```

`printString.c`

```
#include <stdio.h>
void printString() {
    printf("Construit avec make !\n");
}
```

et `printString.h`

```
void printString();
```