

Automation and Make

Ce tutoriel permet de comprendre et d'apprendre l'utilisation d'un Make, dans le cadre d'un projet informatique (ici, un projet python consistant à lire le nombre d'occurrence de lettres dans un livre et de vérifier la loi de Zipf). Pour analyser ce tutoriel et voir ce que je pouvais en tirer comme connaissance, j'ai décidé d'analyser ici chapitre par chapitre l'apport et les critiques de ce tutoriel.

Introduction :

Ce chapitre permet de comprendre pourquoi une utilisation d'un make est important. Nous rencontrons rapidement la problématique suivante "Comment utiliser un programme plusieurs fois sans avoir à taper toutes les commandes". J'ai trouvé cela intéressant que le tutoriel soit amené ainsi pour nous comprenons l'utilité d'un make.

Les auteurs expliquent qu'un make permet une meilleure explication du pipeline, d'éviter la reproductibilité des erreurs et une meilleure reproductibilité des données. J'ai trouvé que ce dernier point était assez mal expliqué, et il manquerait une comparaison avec d'autres logiciels comme le jupyter notebook.

De plus, les auteurs montrent que le désavantage d'un script shell est de recompiler tous les fichiers qui sont déjà créés (plus tard dans le tutoriel, les auteurs expliquent comment éviter cela), et donc parlent de temps de calcul supplémentaires. Ils oublient aussi de mentionner le .pyc, qui permet une compilation plus rapide.

Makefile :

Ce chapitre permet d'apprendre rapidement à faire un makefile. Il est bien construit, et les auteurs expliquent bien l'importance des dépendances et comment s'en servir. De plus, leur makefile est clair, chaque étape est amené par une problématique (exemple : créer un dossier "clean" et voir qu'un make clean ne fonctionne plus correctement).

Variables automatiques :

Ce chapitre permet d'intégrer les raccourcis pour rendre un makefile plus lisible. On gagne beaucoup de temps à écrire, et cela permet de changer les variables comme nous le souhaitons sans avoir à tout réécrire. Ce chapitre aurait pu être regroupé avec les autres chapitres qui parlent de variables.

Dépendances :

Ce chapitre permet de comprendre que les dépendances sont transitive : si A dépend de B qui dépend de C, un changement sur C va apporter une mise à jour sur A.

Personnellement, ce chapitre ne m'a pas apporté beaucoup d'informations. En informatique, cette notion de transitivité des fichiers est déjà connu (notamment avec l'utilisation

d'un fichier ou d'une classe dans un autre fichier). Ce tutoriel est clairement adressé à un public qui a déjà codé. Cette notion de transitivité semble naturelle, et même pour un Makefile.

Pattern rules :

Ce chapitre est identique à celui des variables automatiques, puisque finalement c'est également de nouvelles variables qui sont apportés.

Variables :

Ce chapitre permet l'utilisation de variables bash dans un Makefile. J'ai trouvé ce chapitre intéressant car il rappelle le fait qu'on puisse utiliser des commandes bash dans un Makefile (ce qui n'est pas évident lorsque l'on a jamais fait de Make).

Fonctions :

Ce chapitre montre l'utilité de certaines fonctions pour augmenter la lisibilité de ce code. Ce chapitre n'est pas très bien expliqué, et on commence rapidement à se perdre avec un deuxième makefile. De plus, à force d'ajouter des variables et des fonctions, nous perdons plus de temps à comprendre quelle variable fait quoi.

Documentation des Makefiles :

Ce chapitre est très intéressant, et cela permet de faire un make help rapidement. Mais il est très court et il se fait très rapidement (par rapport au temps qui est estimé pour faire ce chapitre). On pourrait mettre ce chapitre avec celui de variables, puisque la documentation se fait avec des commandes bash.

Conclusion :

Ce chapitre est regroupé en plusieurs exercices, qui sont tous intéressants. Mais la question du chapitre qui est posé est "Quels sont les avantages et inconvénients d'un Makefile", et finalement, le seul désavantage que l'on peut voir c'est qu'un dossier n'est pas recompilé si on ajoute de nouveaux fichiers à l'intérieur. Finalement, ce désavantage peut être résolu en ajoutant des vérifications nécessaires.

Ce tutoriel est intéressant. Il explique bien les notions de base d'un Makefile et son utilité. Je trouve cependant que ce tutoriel manque de recul : au fur et à mesure des chapitres, nous créons un nombre considérable de variables, qui sont finalement utilisées qu'une seule fois dans notre script. Je pense que l'auteur pourrait prendre le temps d'expliquer à quel moment une variable est nécessaire (quand on l'utilise plus de 2 fois dans notre makefile), et quand est-ce qu'elle ne l'est pas.

De plus, les chapitres ne sont pas égaux en terme de temps, le temps estimé n'est pas bon, que ce soit pour un débutant ou un expert. Je pense que plusieurs chapitres pourraient être regroupés pour éviter les répétitions.