

Contrôle des options de script

F. Lassabe

Contents

1	Exercice 1	2
2	Exercice 2	2
3	Exercice 3	2

Introduction

Ce TD se base sur la commande **getopt** qui permet d'analyser et de réordonner des arguments de script. Un exemple d'utilisation de **getopt** est disponible dans le fichier `/usr/share/doc/util-linux`. Le fonctionnement de la commande est également défini par ses pages de manuel, lisez-les.

1 Exercice 1

En vous basant sur les exemples et la documentation, écrivez un script qui prend passe ses arguments à la commande **getopt** et qui affiche les options activées ainsi que leurs paramètres éventuels (obligatoires ou optionnels). Les arguments à analyser sont les suivants :

- **h** sans paramètre
- **o** avec paramètre optionnel
- **s** avec paramètre obligatoire

Le script doit afficher les options activées ainsi que les éventuels paramètres qui y sont rattachés. Puis, il affiche le reste des arguments du script.

2 Exercice 2

Dans ce script, nous allons ajouter des options longues (plus d'un caractère d'option) :

- *source* avec paramètre obligatoire
- *destination* avec paramètre obligatoire
- *dry-run* sans paramètre
- *log-file* avec paramètre optionnel

Comme dans le premier exercice, le programme doit afficher les options et leurs éventuels paramètres, puis le reste des arguments du script.

3 Exercice 3

Reprendre le script de sauvegarde de l'exercice 4 du TD4, et le rendre paramétrable avec **getopt**. Les options à gérer sont les suivantes :

-h sans paramètre qui permet d'afficher un texte d'aide pour utiliser le script, avec le rappel de son fonctionnement et la liste des options supportées. L'appel à cette option arrête le script après avoir affiché l'aide.

-v sans paramètre qui permet d'afficher ce que fait le script au cours de son exécution (création des répertoires et des fichiers, renommage des fichiers de la sauvegarde modifiés)

source avec paramètre obligatoire pour spécifier le dossier source (à sauvegarder)

destination avec paramètre obligatoire pour spécifier le dossier cible (le chemin de la sauvegarde)

dry-run sans paramètre pour tester le script et afficher les opérations qui auront lieu sans les effectuer (comme **-v**).

log-file avec paramètre optionnel pour journaliser ce qu'a fait le script (comme **-v**) dans un fichier. Si aucun paramètre n'est donné, le fichier s'appelle *journal.log*, sinon il prend le nom passé en paramètre.