

TP4 - Système d'exploitation

Exercice 1 (fonction)

Certains programmes écrivent des traces (logs) que l'on souhaite conserver dans un fichier texte (fichier de log) afin d'analyser par la suite le comportement du programme.

Lorsque le fichier de log est trop volumineux, il est nécessaire d'archiver son contenu et ensuite de le nettoyer. La technique utilisée en général est la **rotation des logs** qui permet de conserver dans le temps différentes versions d'un même fichier. Avant d'être vidé, le fichier de log est copié et archivé dans un autre fichier avec un suffixe .0 (par exemple un fichier de log nommé « message.log » s'archivera en « message.0.tar.gz »). Le fichier archivé « message.0.tar.gz » devient à la prochaine rotation de log : « message.1.tar.gz » et ainsi de suite. De cette manière, on peut garder tout l'historique d'un fichier de log.

On se propose d'implémenter ce mécanisme dans un script bash en utilisant les fonctions.

1. Ecrire la fonction **is_integer(i)** qui vérifie que l'argument *i* donné en entrée de la fonction est un entier.
2. Ecrire la fonction **get_number_i(nom)** qui renvoie le numéro *i* de *nom* en vérifiant que *nom* est de la forme *nomf.i.tar.gz* (i.e. lorsque vous passerez le paramètre message.123.tar.gz la fonction retournera 123).
3. Ecrire la fonction **get_max_number_i(logFile)** qui renvoie le *i* le plus grand des fichiers compressé existants sous la forme *log_file.i.tar.gz* (en vérifiant bien à chaque fois que *i* est un entier)
4. Ecrire la fonction **rotation_log(logFile)** qui fait la rotation de log pour le fichier *logFile*. Concrètement, la fonction incrémente le numéro de tous les fichiers existants dont le nom est sous la forme *log_file.i.tar.gz*, *i* est un entier (pour ce faire elle

recherchera d'abord le i le plus grand) ; puis elle archive *logFile* avec la commande `tar` en le renommant *log_file.0.gz* ; enfin, elle crée un nouveau fichier *logFile* vide.