

INITIATION À L'ENVIRONNEMENT UNIX : TP SUR *FIND*
décembre 2023 — Pierre Rousselin

Exercice 1 : *find* dans */usr/include* (1)

À l'aide de la commande *find* et des critères *-name*, *-type* et *-size*, dire quelle commande permet de répondre aux questions suivantes.

Dans l'arborescence issue de */usr/include*,

1. combien y a-t-il de fichiers ?
2. combien y a-t-il fichiers normaux ?
3. combien y a-t-il de répertoire ?
4. combien y a-t-il de fichiers dont le nom de base commence par *std* ?
5. combien y a-t-il de fichiers normaux dont le nom de base commence par *std* ?
6. combien y a-t-il de fichiers normaux dont la taille dépasse 100Kio ?

--- * ---

Correction de l'exercice 1 :

1. `find /usr/include | wc -l`
2. `find /usr/include -type f | wc -l`
3. `find /usr/include -type d | wc -l`
4. `find /usr/include -name 'std*' | wc -l`
5. `find /usr/include -name 'std*' -type f | wc -l`
6. `find /usr/include -name 'std*' -size +100k | wc -l`

--- * ---

Exercice 2 : *find* dans */usr/include* (2)

En plus des critères précédents, on ajoute l'action *-exec NOM_COMM [ARGS1...] {} [ARGS2...]* où *NOM_COMM* est le nom de la commande à exécuter sur chaque fichier trouvé, *[ARGS1...]* sont les éventuels arguments qui précèdent le nom du fichier trouvé, *{}* sera remplacé par le nom du fichier trouvé, *[ARGS2...]* sont les éventuels arguments qui suivent le nom du fichier trouvé et *\;* termine la commande. On peut aussi dans certains cas terminer la commande par *+* (voir le CM correspondant).

Vous aurez notamment à utiliser les commandes *grep* et *du*.

Pour chacune des questions suivantes, donner une commande *find* permettant d'y répondre. Dans l'arborescence issue de */usr/include*,

1. quelle est la taille de chaque fichier normal dont le nom de base commence par *std* ?
2. dans le contenu de tous les fichiers normaux, combien de lignes en tout contiennent la chaîne *TODO* ?
3. combien de fichiers contiennent la chaîne la chaîne *TODO* ? la chaîne *FIXME* ?
4. Donner les 10 fichiers normaux les plus gros, triés par ordre décroissant de taille avec un affichage de la taille « lisible par un humain ».
5. Même question, mais seulement pour les fichiers qui contiennent la chaîne *TODO* (indice : Quelle est la valeur de vérité de *-exec* ? À quoi correspond le statut de sortie de *grep* ?)

--- * ---

Correction de l'exercice 2 :

1. `find /usr/include -type f -name '*.std' -exec du -h {} \;`
2. `find /usr/include -type f -exec grep 'TODO' + | wc -l`
3. `find /usr/include -type f -exec grep -l 'TODO' | wc -l`

4. `find /usr/include -type f -exec du -h {} + | sort -hr | sed 10q`
 5. `find /usr/include/ -type f -exec grep -q 'TODO' {} \; -exec du -sh {} \; | sort -hr | sed 10q`
- * ---

Exercice 3 : Script figr

Écrire le script (en shell standard) `figr` dont le message d'aide est le suivant :

```
$ ./figr
Usage: figr [REP] [EXTENSION] CHAINE
Afficher toutes les lignes contenant CHAINE, pour chaque fichier
dont l'extension est EXTENSION dans l'arborescence issue de
REP.
Si REP est omis, la recherche a lieu depuis le répertoire courant.
Si EXTENSION est omise, la recherche est faite dans tous les fichiers normaux.
Avec deux arguments, le premier argument est considéré comme une EXTENSION

Avec aucun argument ou plus de 3, ce message d'aide est affiché.
```

--- * ---

Correction de l'exercice 3 :

```
#!/bin/sh

HELP='Usage: figr [REP] [EXTENSION] CHAINE
Afficher toutes les lignes contenant CHAINE, pour chaque fichier
dont l'\''extension est EXTENSION dans l'\''arborescence issue de
REP.
Si REP est omis, la recherche a lieu depuis le répertoire courant.
Si EXTENSION est omise, la recherche est faite dans tous les fichiers normaux.
Avec deux arguments, le premier argument est considéré comme une EXTENSION'

case $# in
0)
    printf '%s\n' "$HELP"
    exit 1
    ;;
1)
    find . -exec grep -Hn "$1" {} \;
    ;;
2)
    find . -name ".*$1" -exec grep -Hn "$2" {} \;
    ;;
3)
    find "$1" -name ".*$2" -exec grep -Hn "$3" {} \;
    ;;
*)
    printf '%s\n' "$HELP"
    exit 1
esac
```

--- * ---