

Unix (2)

1 Alias

Il est possible de définir des **alias** sous Unix permettant de personnaliser vos appels de commandes les plus utilisées. Les commandes à utiliser ici sont `alias` et `unalias`. Lancez `man alias`.

1.1 EXERCICE Affichez la liste des alias définis dans le shell courant.

1.2 EXERCICE Définissez un alias `date` sur la commande `ls`.

1.3 EXERCICE Détruisez l'alias précédent et rappelez la commande `date` afin d'en vérifier le bon fonctionnement.

1.4 EXERCICE Ouvrez le fichier `.bashrc` se trouvant dans votre répertoire d'accueil. Rajoutez les alias suivants à l'intérieur de ce fichier, à la suite des alias déjà existant :

```
alias rm='rm -i'
alias ll='ls -la'
```

Ajoutez également un alias permettant d'avoir le `man` en anglais directement.

Pour que ces modifications soient prises en compte par `bash`, tapez la commande `source .bashrc`

2 Protections

2.1 EXERCICE Donnez des exemples de commandes nécessitant un droit d'accès en lecture ou en écriture dans un répertoire.

2.2 EXERCICE Essayez `ls -l`; commentez l'affichage.

2.3 EXERCICE Regardez les différents droits d'accès de vos fichiers.

2.4 EXERCICE Créez avec `emacs` un fichier de nom `test-droits.txt` dans votre répertoire d'accueil. Ajoutez du texte. Enlevez l'accès en lecture pour l'utilisateur et essayez de lire ce fichier au moyen de `cat`. Un autre utilisateur peut-il encore le lire? Remettez l'accès en lecture. Tentez de lire le contenu du fichier au moyen de `cat`.

2.5 EXERCICE Enlevez l'accès en écriture pour l'utilisateur, puis lancez `emacs` et tentez de modifier le fichier : `emacs` a placé ce buffer en mode `read only` repérable par `-%-` en début de ligne de mode. Ce mode est désactivable au moyen de la commande `toggle-read-only` liée à la clé `C-x C-q`. Désactivez ce mode, modifiez le fichier et tentez de sauver. Stoppez `emacs`, redonnez l'accès en écriture et recommencez.

2.6 EXERCICE Essayez ces manipulations sur un fichier d'un autre utilisateur.

2.7 EXERCICE Expérimentez les droits d'accès des fichiers et des répertoires au moyen de `chmod`, `ls`, `cp` et `rm`. Vérifiez que l'on peut avoir le droit de détruire un fichier sans avoir le droit de le lire. Dans quels cas peut-on détruire un fichier qui ne nous appartient pas?

2.8 EXERCICE Pourquoi `ls -l` ne permet pas de lister le mode d'accès du répertoire courant?

3 Visite de l'arborescence du système Unix

3.1 EXERCICE Explorez quelques répertoires du système Unix.

- Allez à la racine du système
- Observez les droits d'accès des différents répertoires
- Visitez `/bin`. Que contient ce répertoire? Observez les droits d'accès des différents fichiers.
- Quel est le rôle des répertoires `/tmp`, `/dev`, `/dev/null`?
- Visitez `/usr/share/man`. Que contient ce répertoire?

3.2 EXERCICE Que fait la commande `whereis`? Essayez `whereis firefox`. Quelle est la différence avec la commande `which`?

3.3 EXERCICE Essayez la commande `find /lib -name "*.a" -print`.

3.4 EXERCICE Donnez la commande permettant de trouver tous les fichiers d'extension `.txt` dont le nom contient `python` sous le répertoire `/opt`.

3.5 EXERCICE Utilisez l'option `-type` de `find` pour lister les répertoires au dessous de `/dev`.

3.6 EXERCICE Donnez la commande permettant de trouver des fichiers chez vous modifiés depuis hier.

3.7 EXERCICE Donnez la commande permettant de trouver chez vous les fichiers modifiés il y a plus de deux jours mais moins de 10 jours.