

## Linux : *shell-scripts*

---

Info0302, "Stage Unix - *scripting*", traite des bases de Linux et des *shell-scripts*.

Ce support, qui suppose la connaissance des bases de Linux, est consacré à l'étude des shell-scripts.

Nous utiliserons le document mis au point par Pascal Mignot donc ce document ne contient pas de cours à proprement parler ; il présente essentiellement une liste d'exercices.

---

Ch Jaillet, département MMI, UFR Sciences, Université de Reims Champagne-Ardenne

**Remerciements :** - à P Mignot, qui nous met à disposition un support très pointu rédigé pour un cours similaire il y a quelques années

- à S Rampacek et H Baala, qui ont rédigé de nombreux énoncés d'exercices et m'ont donné le droit de les réutiliser librement.

# 1 Document de référence

Sur la page Web de cette matière, vous trouverez un lien vers le support de Pascal Mignot "Les shell-scripts sous Unix".

## 2 Liste d'exercices

---

### Exercice 1 (Les variables)

Exécutez et commentez les exemples sur les variables de la page 3 du polycopié "*les shell-scripts sous Unix*".

---

### Exercice 2 (Les alias)

Les alias permettent de faire des raccourcis pour les commandes longues. Ainsi, au lieu de taper `ls -al`, on peut simplement taper `ll`. Pour faire cela, il suffit de taper `alias ll='ls -al'` dans un shell ou de le rajouter dans le fichier `~/.bashrc`.

---

### Exercice 3 (Shellscripts)

1°) Ecrivez un shellscript récupérant les informations disponibles sur un fichier.

```
./info tp4.tex
Nom du fichier :  tp2.tex
Type :           LaTeX 2e document text
Repertoire :     /home/chj/enseignement/Info0302/support
Taille :         1761 octets
Proprietaire :   chj
Droits :         -rw-r--r--
Cree le :        sep 3 16:28
```

2°) Ecrivez une version de ce script acceptant des noms de fichiers génériques (du type `info *.tex`). La commande s'exécutera alors successivement sur tous les fichiers correspondants.

---

### Exercice 4 (Opérations arithmétiques)

- 1°) Exécutez et commentez les exemples sur les opérations arithmétiques de la page 4.
  - 2°) Construisez une fonction itérative qui calcule la factorielle dans un shellscript.
  - 3°) Ecrivez maintenant la version récursive de cette fonction.
-

**Exercice 5 (Entrées/Sorties)**

- 1°) Ecrivez un shellscript qui prend comme paramètres le nom du fichier d'entrée ainsi que le nombre de lignes de ce fichier et qui génère un fichier dont la  $i$ -ème ligne contient le nombre  $i$  puis la  $i$ -ème ligne du fichier d'origine.
  - 2°) Ecrivez un autre script, utilisant le premier, qui réalise le même traitement mais sans avoir à passer en paramètre le nombre de lignes du fichier d'entrée.
  - 3°) Ecrivez un shellscript qui lit un fichier texte en entrée (nommé par exemple `toto`) et qui place les lignes paires et impaires de ce fichier dans deux nouveaux fichiers différents (par exemple `toto_pair` et `toto_impair`).
- 

**Exercice 6 (Traitement de chaînes de caractères)**

- 1°) Ecrivez un shellscript qui insère des espaces entre tous les caractères d'une chaîne en entrée.
  - 2°) Modifiez ce shellscript afin qu'il compte également le nombre de minuscules, majuscules et chiffres dans la chaîne passée.
  - 3°) Ecrivez un shellscript permettant de renommer tous les fichiers `*.dat` en `*.data`.
- 

**Exercice 7 (Exercice sur un fichier *words*)**

- 1°) (a) Ecrivez une commande qui donne le nombre de mots dans le fichier *words*.
  - 2°) Récupérez dans un fichier *s\_words* l'ensemble des mots qui commencent par la lettre *s*. Comment obtenir le même résultat en utilisant `head` et `tail`?
  - 3°) Ecrivez une commande qui renvoie l'ensemble des palindromes de cinq lettres.
  - 4°) Ecrivez une commande qui renvoie l'ensemble des mots de plus de 18 lettres présents dans le fichier *words*.
  - 5°) Ecrivez une commande qui calcule la longueur moyenne d'un mot. Puis même chose pour les mots commençant par *s*.
  - 6°) Ecrivez une commande qui calcule le nombre de noms propres et de noms communs présents dans *words*.
- 

**Exercice 8 (Problème (donné en examen))**

L'option `-s` de la fonction `ls` permet d'afficher en première colonne le nombre de blocs occupés par un fichier (en général, 1 bloc = 1024 octets : comparer les sorties de `ls -l` et `ls -lh`). On souhaite écrire une nouvelle commande qui permet d'obtenir la place réelle du fichier sur le disque (multiple du nombre de blocs), ainsi que la place perdue. La sortie de cette commande est la suivante :

Size	Real	Lost	Files
56148	57344	1196	tp5.ps
3420	4096	676	tp5.dvi
3130	4096	966	tp5.tex
96	1024	928	makefile
62794	66560	3766	TOTAL

- 1°) Ecrivez une commande utilisant exclusivement des filtres, des tubes et la fonction `awk` qui permet d'obtenir ce résultat.
  - 2°) Ecrivez un shellscript donnant le même résultat. Attention, vous ne devrez passer ni par un fichier intermédiaire, ni utiliser la fonction `awk`.
- 

### Exercice 9 (page Web locale)

On dispose sur notre compte personnel d'un certain nombre de fichiers d'image au format JPG.

- 1°) Ecrivez un script permettant de disposer de la liste des fichiers en question avec leur chemin absolu.
  - 2°) Transformez le script de telle sorte que la sortie soit un fichier html donnant les chemins puis les liens correspondants (testez!!)
  - 3°) Pour chaque image rencontrée vous ajouterez dans le répertoire qui la contient une miniature de l'image (commande `convert`) et ferez en sorte que la base du lien soit cette miniature.
- 

### Exercice 10 (Algorithmique)

- 1°) Ecrivez un script permettant de calculer la première puissance de deux au moins égale à une certaine passée en paramètre de la commande.
- 2°) Ecrivez une commande calculant le nombre de nombres premiers dans un intervalle (les deux bornes sont passées en paramètre ; elles se sont pas forcément données par ordre croissant).  
NB : écrivez les fonctions intermédiaires nécessaires.
- 3°) ...

---

### Et maintenant ?

... <https://www.cours-gratuit.com/unix-linux>  
(exercices sur les scripts) ...