

# Systemes d'exploitation 420-W12-SF

## La ligne de commandes Linux Bash

Source : ROHAUT, Sébastien (2020), Linux Maîtrisez l'administration du système (6<sup>e</sup> édition) . Édition Eni ISBN : 9782409025716

Jean-Pierre Duchesneau, Automne 2021

## Les cours

---

1. Intro au infrastructure informatique et les composantes interne du PC
2. Les composantes interne du PC et réseau
3. Système d'Exploitation
4. Virtualisation de clients et de serveurs
5. Disque dur, partition et système de fichier
6. **La ligne de commandes (Shell) et les scripts**
  1. CMD
  2. Linux
  3. Bash
  4. PowerShell
7. Git, le contrôle de version
8. WAMP

# Le shell bash

L'interpréteur de commandes, ou interprète, permet d'exécuter des instructions que vous saisissez au clavier ou au sein d'un script et vous en retourne les résultats.

Cet interpréteur est un programme appelé shell.

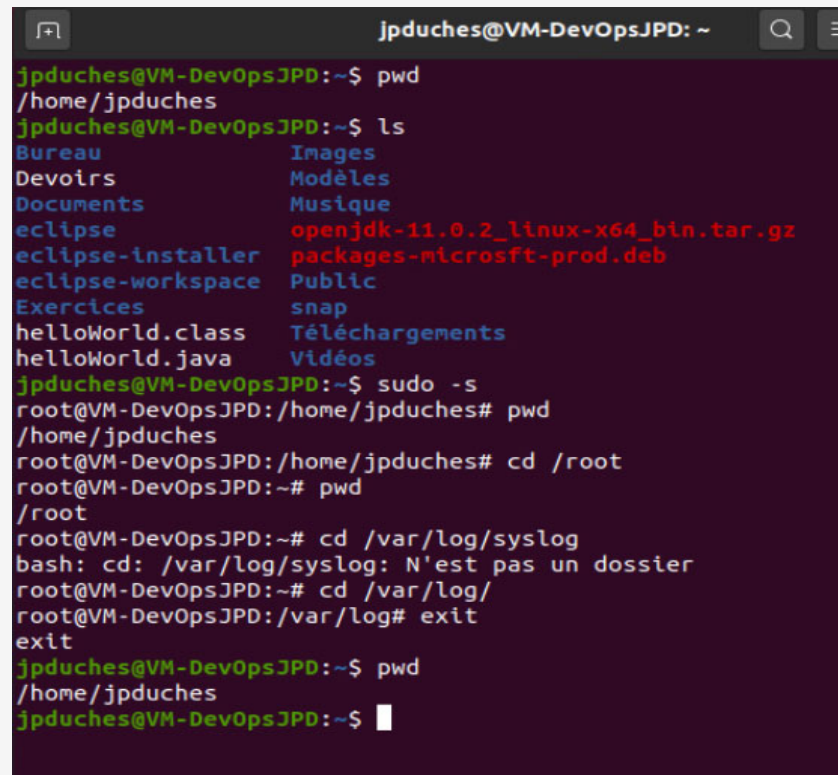
C'est à rapprocher du mot **kernel** : le kernel, signifiant noyau. Shell signifiant coquille, c'est donc ce qui « entoure » le **noyau Linux** : le moyen de l'utiliser à l'aide de commandes. C'est donc une interface fonctionnant en mode texte entre le noyau Linux et les utilisateurs (avancés), voire les applications.

Il existe plusieurs shells, chacun disposant de spécificités propres. Le Bourne Shell (sh) est le shell le plus connu et le plus courant sur les Unix.

Le shell de référence sous Linux se nomme le Bourne Again Shell (bash) et dérivé du sh.

Le shell fonctionne au sein d'un terminal. Il attend des saisies au clavier dans la console ou la fenêtre, et affiche ses résultats au même endroit.

L'invite (prompt) fournit des informations sur le terminal et votre position dans le système de fichiers.



```
jpduches@VM-DevOpsJPD: ~  
jpduches@VM-DevOpsJPD:~$ pwd  
/home/jpduches  
jpduches@VM-DevOpsJPD:~$ ls  
Bureau          Images  
Devoirs         Modèles  
Documents       Musique  
eclipse         openjdk-11.0.2_linux-x64_bin.tar.gz  
eclipse-installer packages-microsoft-prod.deb  
eclipse-workspace Public  
Exercices       snap  
helloWorld.class Téléchargements  
helloWorld.java Vidéos  
jpduches@VM-DevOpsJPD:~$ sudo -s  
root@VM-DevOpsJPD:/home/jpduches# pwd  
/home/jpduches  
root@VM-DevOpsJPD:/home/jpduches# cd /root  
root@VM-DevOpsJPD:~# pwd  
/root  
root@VM-DevOpsJPD:~# cd /var/log/syslog  
bash: cd: /var/log/syslog: N'est pas un dossier  
root@VM-DevOpsJPD:~# cd /var/log/  
root@VM-DevOpsJPD:/var/log# exit  
exit  
jpduches@VM-DevOpsJPD:~$ pwd  
/home/jpduches  
jpduches@VM-DevOpsJPD:~$
```

# Syntaxe générale des commandes

Les commandes ont très souvent une syntaxe reprenant la même structure :

## Commande **[paramètre][arguments]**

Une commande peut avoir ni paramètres, ni arguments. Dans ce cas elle exécute l'action par défaut pour laquelle elle est programmée, ou affiche un message d'erreur si ceux-ci sont nécessaires.

Un **paramètre** est une option de la commande. Les deux mots sont ici synonymes. C'est souvent une simple lettre ou un simple chiffre précédé d'un tiret : -l, -p, -s, etc.

Les **arguments** sont les entités sur lesquelles la commande doit exécuter son action. Leur type dépend de la commande. Ce peut être un fichier, du texte, des nombres, etc.

La commande **cal** admet deux arguments optionnels (les seuls dans la norme POSIX). Si un seul est précisé, il s'agit de l'année, et l'intégralité du calendrier de cette année est affichée. Si deux arguments sont précisés, le premier est le mois, le second l'année.

```
jpduches@VM-De
jpduches@VM-DevOpsJPD:~$ cal
      Juin 2020
di lu ma me je ve sa
  1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

jpduches@VM-DevOpsJPD:~$ cal 12 1976
      Décembre 1976
di lu ma me je ve sa
          1  2  3  4
  5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

jpduches@VM-DevOpsJPD:~$
```

## Le Shell et ses principales commande

---

- **ls** : liste le contenu d'un répertoire
- **pwd** : afficher le répertoire courant
- **cd** : change de répertoire
- **cd ..** : répertoire parent
- **cd ~** : allez au répertoire de l'utilisateur
- **mkdir** : créer un répertoire
- **rmdir** : supprime un répertoire
- **cp** : copie de fichier
- **mv** : déplacement de fichier.
- **rm** : supprime le fichier (rm -rf /home/toto/temp
- **passwd** : change le mot de passe de l'utilisateur
- **cat**: affiche le contenu du fichier
- **date** : date et l'heure actuelles.

## Rappel de l'historique

La flèche du haut remonte dans l'historique. La flèche du bas navigue dans l'autre sens, jusqu'à l'invite d'origine. Si vous appuyez sur la touche [Entrée] vous lancez de nouveau la commande.

Plus vous tapez des commandes, plus l'historique s'agrandit. Le shell conserve ainsi un grand nombre d'entrées dans l'historique (le nombre de lignes conservées peut être modifié). Cet historique est conservé dans un fichier caché de votre répertoire personnel appelé `.bash_history`. Vous pouvez voir le contenu de l'historique avec la commande **history**.

```
jpduches@VM-DevOpsJPD: ~  
421 ls -al  
422 git status  
423 git log  
424 mkdir JPDProf  
425 cd JPDProf/  
426 touch JPDtext.txt  
427 gedit JPDtext.txt  
428 git status  
429 git add .  
430 git commit -m 'Mon premier commit sur un dépôt distant'  
431 git status  
432 git push  
433 pwd  
434 cd  
435 pwd  
436 cat .ssh/id_ed25519.pub  
437 cd Bureau/infraubr4361  
438 ls  
439 git status  
440 git remote -v  
441 git fetch  
442 ls  
443 ls -al  
444 git pull  
  
532 type pwd  
533 type cat  
534 clear  
535 historey  
536 clear  
537 history  
jpduches@VM-DevOpsJPD:~$
```

## Rappel de l'historique

La commande **fc** effectue presque la même chose lorsqu'on utilise le paramètre **-l**. Par défaut elle se limite aux quinze dernières commandes. Aussi vous pouvez lui passer le nombre des dernières commandes, comme

Vous pouvez rappeler une commande avec **fc** et le paramètre **-s** suivi du numéro de la commande. Elle sera alors automatiquement lancée.

D'autres raccourcis-clavier sont pratiques :

- **[Ctrl] a** : aller au début de la ligne.
- **[Ctrl] e** : aller en fin de ligne.
- **[Ctrl] l** : effacer le contenu du terminal, et afficher l'invite en haut de celui-ci.
- **[Ctrl] u** : effacer la ligne jusqu'au début.
- **[Ctrl] k** : effacer la ligne jusqu'à la fin.

```
528 clear
529 cal
530 cal 12 1976
531 type date
532 type pwd
533 type cat
534 clear
535 historey
536 clear
537 history
jpduches@VM-DevOpsJPD:~$ fc -s 530
cal 12 1976
    Décembre 1976
di lu ma me je ve sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

jpduches@VM-DevOpsJPD:~$
```

# Travail à faire

---

## Exercices

- Dernière période pour l'installation de Linux
- Exercice 9 Visite guidée de Ubuntu (**attention au nouveau compte Prof**)
- Exercice 10 Commandes de bases du Shell Linux
- Exercice 11 Gestion des usagers Linux