

# Commandes "évoluées"

---

## 8. Personnaliser son environnement

- 8.1 - Personnaliser l'apparence de votre invite de commande (syntaxe, couleurs) en modifiant la variable PS1 : par exemple, afficher le nom de la machine en jaune.
- 8.2 - Ajouter la personnalisation du PS1 à votre `.bashrc` et propagez ces changements sur vos shells ouverts.
- 8.3 - Ajouter aussi un message de bienvenue comme "May the force be with you!" qui s'affichera à chaque ouverture d'un shell (rappel : `echo` peut être utilisé pour afficher un tel message).
- 8.4 - Changer le `.bashrc` de root pour que son invite de commande soit en rouge !
- 8.5 - S'assurer que vous disposez de l'alias `ll` (pour `ls -l`), et que `--color=auto` est activé implicitement lorsque vous utilisez `ls`.
- 8.6 - Créer un alias `amiconnected` qui exécute la commande `ping` sur l'IP 8.8.8.8 avec l'option `-q` et l'option `-c 1`
- 8.7 - Créer un alias `r2d2` qui permet d'ouvrir un shell en tant que `r2d2` avec `sudo` et `su`.
- 8.8 - En utilisant `echo`, comment faire pour faire en sorte que la commande `ls` retourne systématiquement 'Bof, G pas envie' au lieu de son comportement normal ?
- 8.9 - (Avancé) Se renseigner sur `LS_COLORS` et personnaliser cette variable.

## 9 partie 1 - redirections et assemblages

---

- 9.1 - Créer un fichier à l'aide de `echo` et d'une redirection. Par exemple, un fichier `hello.txt` qui contient "Hello, world!"
- 9.2 - Dans un tty, regarder le résultat de `ls /usr/bin` n'est pas pratique, car la sortie est trop longue et il n'est pas possible de scroller vers le haut. Pour contourner le problème, enregistrez la sortie dans un fichier puis lisez-là avec `less`.
- 9.3 - En **une seule ligne de commande** (qui pourra comporter plusieurs sous-commandes séparées par `;`) : créer un dossier `formationlinux` dans votre répertoire personnel, puis créer un sous-dossier `bash`, et dedans un fichier `intro` qui contient "Je fais du bash, c'est stylé !"
- 9.4 - `bc` est un utilitaire permettant de faire de petits calculs. Testez `bc` en mode interactif pour faire quelques additions (Ctrl+D pour quitter). Mettez maintenant une suite de calcul dans un fichier, que vous injecterez directement dans `bc`. Enfin, faites des calculs similaires mais en injectant directement une chaîne dans `bc`.
- 9.5 - `curl` est un utilitaire qui permet de télécharger et le code source d'une page (par exemple, `fr.wikipedia.org`) dans la console.
  - Comment pouvez-vous faire pour sauvegarder cette page dans un fichier ?
  - Retentez l'expérience avec une adresse qui n'existe pas, et modifiez votre commande pour supprimer les erreurs mais afficher tout de même "Ca n'a pas marché" dans le cas où vous tentez de télécharger une page qui n'existe pas...
- 9.6 - (Avancé) Créez un fichier `/tmp/chat` dans lequel padawan et r2d2 peuvent tous les deux écrire. Renseignez-vous sur l'option `-f` de la commande `tail` puis :
  - Lancez `tail -f /tmp/chat` en tâche de fond dans deux terminaux (l'un de padawan, l'autre de r2d2) ;
  - À l'aide de redirections, envoyez des messages dans le fichier `/tmp/chat` et observez les deux terminaux ;
  - Améliorez le système en créant un alias `say` qui affiche un message préfixé de votre nom d'utilisateur (ex: `[r2d2] beep boop`).

## 9 partie 2 - pipes et boîte à outils

---

- 9.7 - Si ce n'est pas déjà le cas, ajoutez un alias pour activer automatiquement `--color=auto` chaque fois que la commande `grep` est utilisée. Essayez quelques manipulations avec `grep` pour confirmer que les occurrences trouvées sont bien mises en valeur.
- 9.8 - Lister les lignes de `/etc/passwd` qui correspondent aux utilisateurs ayant `/bin/bash` comme shell
- 9.9 - Même chose, mais cette fois en affichant uniquement le nom des utilisateurs
- 9.10 - Lister les utilisateurs (uniquement leur nom !) qui ont comme shell `nologin`
- 9.11 - Lister les utilisateurs (uniquement leur nom) qui ont un mot de passe non vide (rappel : historiquement les passwords se trouvaient dans `/etc/passwd`, mais ce n'est plus le cas de nos jours !)
- 9.12 - Écrivez un alias `estcequecestleweekend` qui vérifie si on est Samedi ou Dimanche en utilisant `date`
- 9.13 - Sachant que pour `grep`, `^` et `$` désignent un début et une fin de ligne, pouvez-vous afficher le contenu de `/etc/login.defs`
  - sans les commentaires (ce sont les lignes commençant par `#`)
  - puis sans les commentaires ni les lignes vides ? (Indice : une ligne vide est une ligne qui se commence puis se termine tout de suite)
- 9.14 - Écrire **une seule ligne de commande** qui affichera (uniquement) "Oui" ou "Non" suivant si par exemple le paquet `vim` est installé. Pour tester si un paquet est installé, on pourra se baser sur `dpkg --get-selections | grep -q vim` ou bien sur `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` `<nom_du_paquet>` (Testez et validez le comportement avec d'autres paquets)
- 9.15 - À l'aide des pages de man de `grep`, trouvez un moyen de lister toutes les occurrences du mot `daemon` dans tous les fichiers à l'intérieur de `/etc/` (récursivement)
- 9.16 - Écrivez une ligne de commande qui affiche "Oui" ou "Non" si `r2d2` est actuellement connecté. On pourra pour ce faire utiliser (entre autre chose) le fait que `grep` a un code de retour, et éventuellement l'option `-q`.
- 9.17 - À l'aide de `ps`, `sort` et `uniq` générer un bilan du nombre de processus actuellement en cours par utilisateur
- 9.18 - À l'aide de `sort` et `uniq`, analysez le fichier `loginattempts.log` (demander au formateur comment l'obtenir), et produisez un résumé du nombre de tentative de connections par ip
- 9.19 - (Avancé) Construisez une ligne de commande qui récupère les adresses des images présentes dans le code du site `yo1oswag.team`. Vous aurez possiblement besoin de `curl`, `grep`, `tr`, `awk` et `sed`.