

# Exercices R405

R&T 2

2024

## Regexes

1. Écrire une regex pour matcher les dates *31/1/1984*.
2. Écrire une regex pour matcher les adresses IPv4.
3. Écrire une regex pour matcher les adresses mails.
4. Écrire une regex pour matcher les URLs.
5. Écrire une regex qui matche les md5 (nombres hexa à 32 chiffres).
6. Étant données des lignes de log au format suivant :

```
Sep  7 08:45:14 whirlwind sshd[28333]: Invalid user admin from 122.225.109.108
Sep  7 08:52:57 whirlwind sshd[28404]: pam_unix(sshd:auth): authentication failure; logname= uid=0 e
Sep  7 08:52:59 whirlwind sshd[28404]: Failed password for root from 116.10.191.169 port 42744 ssh2
Sep 15 09:30:59 whirlwind gnome-keyring-daemon[1143]: couldn't access control socket: /run/user/1000
Sep 15 09:30:59 whirlwind gnome-keyring-daemon[1143]: Gkm: using old keyring directory: /home/manu/
Sep 15 09:31:03 whirlwind sshd[593]: Received signal 15; terminating.
Sep 15 09:31:03 whirlwind sshd[1359]: Server listening on 0.0.0.0 port 22.
Sep 15 09:31:03 whirlwind sshd[1359]: Server listening on :: port 22.
Sep 15 09:47:57 whirlwind sudo: manu : TTY=pts/1 ; PWD=/home/manu ; USER=root ; COMMAND=/usr/bin/ta
```

- a. Écrire une regex qui ne donne que les lignes pour le 7 septembre.
  - b. Écrire une regex qui ne donne que les lignes pour le 7 septembre entre 9h30 et 10h.
  - c. Écrire une regex qui ne donne que les lignes pour le 7 septembre entre 9h30 et 10h concernant *sshd*.
7. Proposer une regex qui, dans un fichier html, matche les blocs div (d'une balise div ouvrante à la balise fermante correspondante).

## Pipes et filtres

Les commandes suivantes pourront être utilisées comme filtres : **grep** (trouver des motifs), **tr** (remplacer des caractères), **sed** (remplacer des motifs), **head** (sélectionner les premières lignes), **tail** (sélectionner les dernières lignes), **uniq** (supprimer les doublons), **wc** (compter les lignes, mots et caractères), **sort** (trier), **cut** (couper des colonnes), **comm** (lignes en commun entre deux fichiers), **join** (jointure). On pourra se référer à leur manuel pour plus d'informations.

1. Écrire une ligne de commande pour afficher tous les fichiers et dossiers qui se trouvent dans le dossier */usr/bin*.
2. Écrire une ligne de commande pour compter ces fichiers et dossiers.
3. Écrire une ligne de commande pour afficher les fichiers et dossiers et leurs inodes, triés par ordre décroissant d'inode.
4. Écrire une commande qui affiche les 10 plus grands inodes.
5. Écrire une ligne de commande qui compte les fichiers exécutables (ayant le droit **x**) dans le dossier */usr/bin*.
6. Écrire une commande qui affiche le PPID du processus ayant pour PID 12.

Pour les questions suivantes, nous allons reprendre un fichier de log (journal d'événements) ayant la structure vue dans l'exercice précédent.

7. Écrire une commande qui dans le fichier `/var/log/auth.log` extrait parmi les lignes pour le 10 septembre entre 9h30 et 10h00 concernant `sshd`, les IPs qui apparaissent.
8. Écrire une commande qui affiche les lignes du fichier `/var/log/auth.log` en remplaçant les IPs par `XXX.XXX.XXX.XXX` dans les lignes pour le 10 septembre entre 9h30 et 10h00 concernant `sshd`.

`/usr/share/dict/words` contient une liste de mots que connaît le système (pour faire des vérifications orthographiques) sous la forme d'un mot par ligne.

9. Écrire une commande pour afficher le nombre de mots dans ce fichier.
10. Ligne de commande pour afficher les mots de ce fichier contenant `truc`.
11. Ligne de commande pour afficher les mots contenant `truc` de plus de 8 lettres.
12. Ligne de commande pour faire dire par un pingouin le dernier mot de plus de 8 lettres contenant `truc` dans l'ordre alphabétique.
13. Étant donné un fichier `livre.txt` contenant un texte (avec des signes de ponctuations, des passages à la ligne, des majuscules). Proposer une ligne de commande donnant les 10 mots les plus utilisés. À défaut, un script donnant les 10 mots les plus utilisés...

## Scripts

1. Écrire un script qui récupère dans le fichier `/var/log/auth.log` les 10 dernières IPs différentes responsables de connexions ssh ayant échoué. Puis, pour chacune de ces IP l'ajoute à un fichier `indesirables` ainsi que son FQDN.
2. Écrire un script qui efface tous les fichiers dans `/tmp` appartenant à l'utilisateur qui lance le script.
3. Écrire un script qui teste si `firefox` est lancé et qui s'il l'est tente de l'arrêter en lui envoyant le signal `STOP`. Puis patiente 5 secondes et si `firefox` est toujours en cours (avec le même PID) envoie le signal `KILL`.
4. Écrire un script qui affiche un caractère au hasard parmi `'/'` et `'\'`, patiente 1 seconde et recommence. On pourra utiliser la variable `RANDOM` qui à chaque fois qu'on y accède contient un nouvel entier aléatoire entre 0 et  $2^{15} - 1$ .
5. Écrire un script pour dédupliquer des fichiers. Il prendra deux dossiers en arguments si deux fichiers ont le même contenu, il supprimera l'un des deux puis créera un lien physique pour économiser de l'espace disque. On pourra utiliser `md5sum` pour tester si deux fichiers sont identiques (s'ils ont le même md5, ils sont probablement identiques).

## Scripts d'administration

1. Créer des comptes utilisateurs pour chacun des membres d'une liste nom/prénom
  - Format de la liste (regex pour obtenir nom/prénom)
  - man adduser
  - traitement des homonymes
2. Envoyer à chaque utilisateur un mail de son utilisation disque
3. Backup daté
  - `./backup /home/truc`
  - fabriquer une archive (tar.xz ou zip) nommé 20150916
  - chiffrer cette archive
  - l'envoi de façon sécurisée sur un serveur distant
  - restore
  - supprimer backups vieux

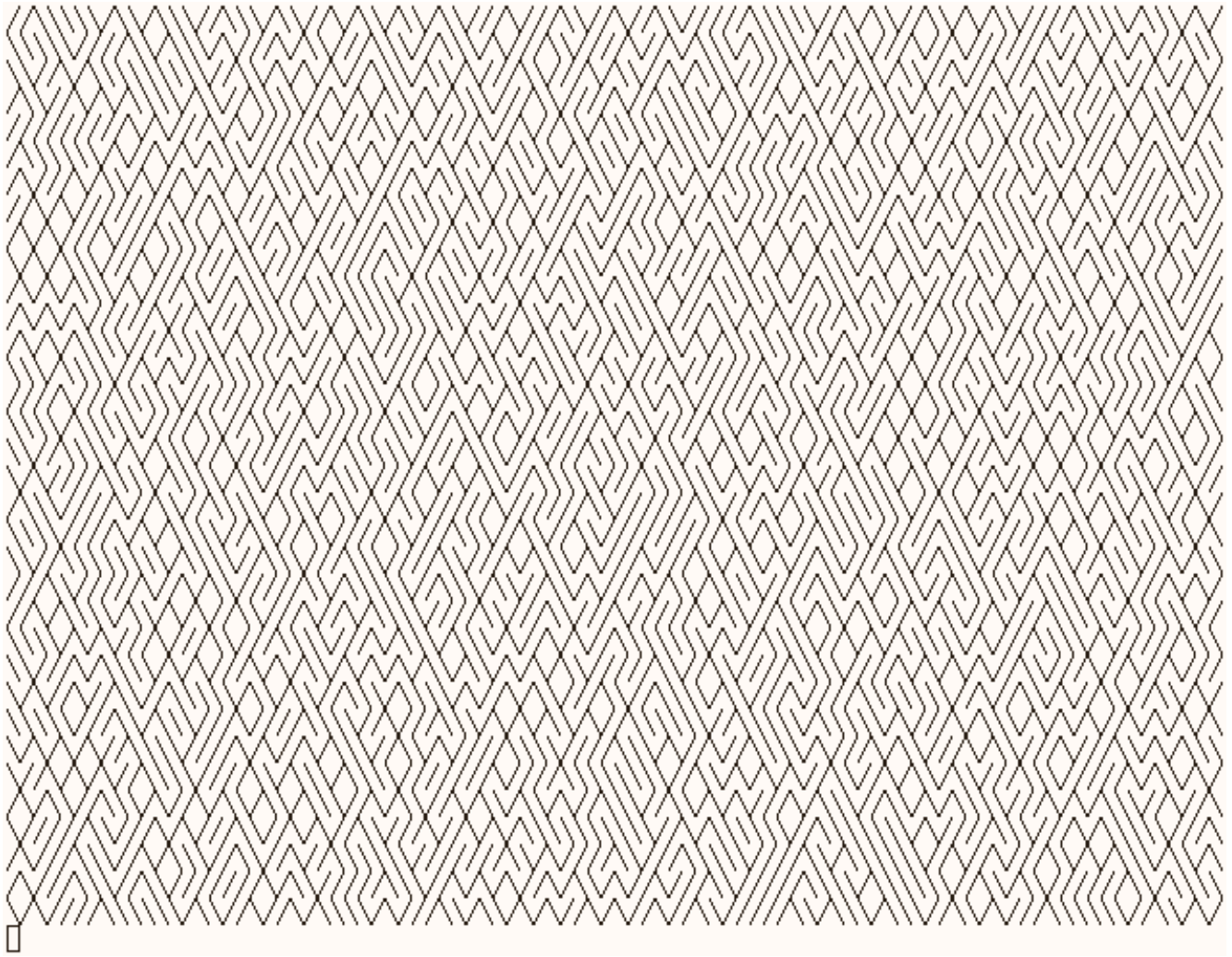


Figure 1: Labyrinthe