



# Sujet 103

# Commandes GNU & Unix

# Commandes GNU & Unix

- 103.1 : Travail en ligne de commande (Val. 4)
- 103.2 : Traitement de flux de type texte avec des filtres (Val. 3)
- 103.3 : Gestion élémentaire des fichiers (Val. 4)
- 103.4 : Utilisation des flux, des tubes et des redirections (Val. 4)
- 103.5 : Création, contrôle et interruption des processus (Val. 4)
- 103.6 : Modification des priorités des processus (Val. 2)
- 103.7 : Recherche dans des fichiers texte avec les expressions rationnelles (Val. 2)
- 103.8 : Édition de fichiers texte avec vi (Val. 3)

# 103.7

## Recherche dans des fichiers texte avec les expressions rationnelles (Val. 2)

# Recherche dans des fichiers texte avec les expressions rationnelles

- Description : Les candidats doivent être en mesure de manipuler des fichiers et des données de type texte en utilisant des expressions rationnelles. Cet objectif inclut la création d'expressions rationnelles simples contenant différents caractères de notation. Cela inclut également l'utilisation des expressions rationnelles dans des commandes pour effectuer des recherches dans une arborescence ou dans le contenu d'un fichier.
- Termes, fichiers et utilitaires utilisés pour cet objectif :
  - grep
  - egrep
  - fgrep
  - sed
  - regex (7)

# Expression régulière

- Une **expression régulière** (regex : *Regular Expression*) ou **expression rationnelle** ou **motif**, est une chaîne de caractères, qui décrit, selon une syntaxe précise, un ensemble de chaînes de caractères possibles.
- Recherche dans des fichiers :
  - Extraire les lignes du fichiers qui répondent aux critères de l'expression régulière.

# grep

- Afficher les lignes correspondant à un motif donné.

**grep options motif fichier**

- Options

- **v** : effectue la recherche inverse, toutes les lignes ne correspondant pas au motif sont affichées.
- **c** : ne retourne que le nombre de lignes trouvées sans les afficher.
- **i** : ne différencie pas les majuscules et les minuscules.
- **n** : indique le numéro de ligne pour chaque ligne trouvée.
- **l** : dans le cas de fichiers multiples, indique dans quel fichier la ligne a été trouvée.

# egrep & fgrep

- **egrep**
  - Interprète le motif comme une expression régulière étendue.
  - Équivalente à `grep -E`
- **fgrep**
  - Interprète le motif comme une liste de caractères fixes.
  - Équivalente à `grep -F`

# motif

- Le motif peut contenir des caractères qui ont une signification dans le shell (caractères de substitution).
- Le motif **doit être interprété** par la commande grep et non pas par le shell.
- Il est donc recommandé de verrouiller tous les caractères du motif.

**grep options "motif" fichier**

# Caractères spéciaux

Caractère	Signification
<code>^</code>	Début de ligne.
<code>\$</code>	Fin de ligne.
<code>.</code>	Un caractère quelconque.
<code>\(...\)</code>	Groupement de caractères.
<code>[...]</code>	Un caractère cité dans la liste.
<code>[^...]</code>	Un caractère qui n'est pas cité dans la liste.
<code>\&lt; et \&gt;</code>	Début d'un mot et fin d'un mot. Les caractères d'un mot sont <code>[A-Za-z0-9_]</code>
<code>*</code>	Le caractère précédent apparaît 0 à n fois
<code>\+</code>	Le caractère précédent apparaît 1 à n fois.
<code>\?</code>	Le caractère précédent apparaît 0 ou 1 fois.
<code>\{n\}</code> - <code>\{n,\}</code> - <code>\{n,m\}</code>	Le caractère précédent apparaît exactement n fois – au moins n fois – entre n et m fois.

# Exemples

Expression	Signification
<code>Linux</code>	Ligne contenant "Linux".
<code>^Linux</code>	Ligne commençant par "Linux".
<code>Linux\$</code>	Ligne se terminant par "Linux".
<code>^\$</code>	Ligne vide (aucun caractère entre ^ et \$)
<code>^[A-Z][5-9].\$</code>	Ligne composée de 3 caractères. Le premier est une majuscule, le second est un chiffre entre 5 et 9 et le dernier un caractère quelconque.
<code>7space[0-79]A*</code>	Ligne contenant le chiffre 7, suivi d'un espace, suivi de n'importe quel chiffre sauf 8, suivi de la lettre A 0 à n fois.
<code>^[+-]?[0-9]+\$</code>	Ligne contenant uniquement un nombre entier d'au moins 1 chiffre précédé éventuellement du signe + ou -.
<code>[-a-zA-Z _]\{4,\}</code>	Ligne contenant au moins 4 caractères composées de lettres minuscules et/ou majuscules, de tiret, d'espace et de tiret bas.

# sed

- Éditeur de flux (*Stream Editor*) permettant de filtrer et de transformer du texte.

```
sed -e 'cmd' fichier
```

- Exemples

- Substitution (remplacement) d'une chaîne de caractères par une autre

```
sed -e 's/regex/chaîne/g' fichier
```