



**ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR**

Membre de  
**HONORIS UNITED UNIVERSITIES**

3IIR



# Systeme d'exploitation (UNIX)

8<sup>ème</sup> partie

Professeur: IOUNOUSSE Jawad

# Exercice 12

- Écrire un script « **bigps.sh** » qui affiche le processus qui consomme le plus de mémoire à l'instant, ainsi que la taille de l'espace mémoire utilisé par ce processus. Vous pouvez vous servir des commandes **ps** et **free** en plus des filtres de votre choix.

# Exercice 12 (corrigé)

bigps.sh

```
#!/bin/bash

echo -e "les infos du processus le plus gourmand en Mem:\n"
ps -eF | head -n 1
ps -eF | sort -nr +5 -6 | head -n 1

bps_info=$(ps -eF | sort -nr +5 -6 | head -n 1)
mem_info=$(free | grep Mem)

tab1=($bps_info)
tab2=($mem_info)

mem_used=${tab2[2]}
bps_mem=${tab1[5]}
bps_name=${tab1[10]}

usage=$((bps_mem * 100 / mem_used))

echo -e "\n$bps_name occupe actuellement $usage% de la mémoire utilisée"
```

# Les filtres étendus

# Le filtre sed

## ❖ Introduction:

- **sed** est un éditeur ligne par ligne non interactif, il lit les lignes d'un **fichier** une à une (ou provenant de l'**entrée** standard) leur applique un certain nombre de commandes d'édition et renvoie les lignes résultantes sur la **sortie** standard.
- c'est une évolution de l'éditeur **ed** lui même précurseur de **vi**, la syntaxe n'est pas très conviviale, mais il permet de réaliser des commandes complexes sur de gros fichiers.

## ❖ Usage et comportement:

**Syntaxe :** `$ sed -e 'liste_d_instructions' fichier_à_traiter`

`$ sed -f fichier_script fichier_à_traiter`

- Si aucun fichier à traiter n'est indiqué, **sed** attend les données sur son entrée standard.
- Lorsqu'on fournit directement les commandes sur la ligne, grâce à l'option **-e**, il est préférable de les inclure entre **apostrophes simples**, en raison de l'usage fréquent des caractères **\$**, **\***, **?**, etc., susceptibles d'être interprétés par le shell.
- **Sed** est un outil très puissant à la main des administrateurs système puisqu'il permet de traiter de grandes quantités de données en un temps record et avec un minimum de ressources.
- option **-n** : mode silencieux, envoi vers la sortie seulement si demandé

# Le filtre sed

## ❖ Fonction de substitution s:

▪ **Utilisation:** `sed 's/regexp/remplacement/options'` exemples:

`sed 's/toto/TATA/' file` : remplace la 1<sup>ère</sup> occurrence de **toto** par **TATA**.

`sed 's/toto/TATA/3' file` : remplace la 3<sup>ème</sup> occurrence de **toto** par **TATA**.

`sed 's/toto/TATA/g' file` : remplace toutes les occurrences de **toto** par **TATA**.

`sed -n 's/toto/TATA/p' file` : Si remplacement, la ligne concernée est affichée sur la sortie.

`sed 's/toto/TATA/w resultat.txt' file` : Si substitution, la ligne en entrée est inscrite dans un fichier résultat.

`sed -e 's/[Cc]haise/CHAISE/g' file` : substitue toutes (**Chaise** ou **chaise**) par **CHAISE**.

`sed -e 's/^#//' file` : décommente tous les commentaires situés en début de ligne.

`sed -e 's/^[a-zA-Z ]*$/#&/g' file` : commente toute ligne qui contient seulement des lettres et des espaces.

`sed -e 's/\([0-9][0-9]*\) \([a-z]\)/__\1__\2/g'`

`sed -E 's/([0-9]+)([a-z])/__\1__\2/g'` : entoure tous les nombres avec des `__` s'ils sont suivis par une lettre.

▪ **Note:** `&` permet de faire une référence arrière vers la chaîne filtrée.

- `\1`, `\2` .... permettent de faire une référence arrière à une sous chaîne filtrée, les sous chaînes sont délimitées par des parenthèses.

# Le filtre sed

## ❖ Fonction de suppression d:

- La fonction **d** de **sed** supprime les lignes correspondantes au critère donné :

**sed '20,30d' file** : supprimer les lignes de **20** à **30** du fichier **file**.

**sed '/toto/d' file** : supprime les lignes contenant la chaîne **toto**.

**sed '/toto/!d' file** : supprime les lignes ne contenant pas la chaîne **toto**.

- REMARQUE** : les changements apportés par **sed** n'affectent pas le fichier en entrée. Elles sont juste envoyées vers la sortie, sauf si l'option **-i** est spécifiée!

## ❖ Fonction d'insertion i:

**sed '1i Nom:Prenom:Age:Adresse' file** : insère la ligne spécifiée au début de **file**.

## ❖ Fonction d'ajout a:

**sed '2a Hello world' file** : ajoute la ligne spécifiée après la 2<sup>ème</sup> ligne de **file**.

## ❖ Autres fonctions w,p,q,=:

- p (print)** : filtre les lignes sélectionnées vers la sortie (même si **-n** est activée)
- w (write)** : écrire les lignes filtrées dans un fichier
- q (quit)** : quitter **sed** dès que l'expression est réalisée (ne pas traiter les lignes suivantes)
- =** : afficher le numéro de la ligne

# Exercice 13

1. Ecrire un script qui modifie le contenu d'un fichier template et l'enregistre dans un fichier spécifié en utilisant les variables paramètres.

template

Nom: <n>  
Prénom: <p>  
Age: <a>

```
bash form.sh Tazi Hiba 20 result.txt
```

result.txt

Nom: Tazi  
Prénom: Hiba  
Age: 20

2. Supprimer tous les commentaires commençant par `//` dans un fichier php.
3. Mettre en gras le mot **Linux** ou **linux** dans un fichier html.
4. Sachant qu'on a un fichier qui contient des dates au format **YYYY-MM-DD**, écrire un script qui permet de les convertir au format **DD/MM/YYYY**