

# Systèmes d'exploitation

Stream EEditor

Yves Stadler

Codasystem, UPV-M

29 février 2012

# Qu'est-ce que SED

## SED

- SED = Stream Editor
- Utilitaire Unix
- Manipulation ligne par ligne du texte
- Utilise des expressions régulières

## Exemple typique

- Remplacement : `sed -e 's/Ancien/Nouveau/g' ne > ns`
- "Substitue 'Nouveau' à 'Ancien' en utilisant ne comme input, et ns en output"

## Ed

- Basé sur l'éditeur ed, toujours présent actuellement
- À la base, ed se voit adjoindre une fonction pour travailler son flux d'entrée
- "g/re/p" (global/regular expression/print)
- Donnera grep
- Puis sed

## Ligne de commande

- `sed -e 'actions' input`
- On préfère souvent rediriger la sortie

## Actions

- Remplacement
- Effacement
- Actions conditionnelles

# Syntaxe

## Syntaxe complète

```
sed [-options] [commande] [<fichier(s)>]  
sed [-n [-e commande] [-f script] [-i[.extension]]  
[l [cesure]] rsu] [<commande>] [<fichier(s)>]
```

## Commandes

- [addr[,addr]][!]cmd[args]
- [addr[,addr]]{  
cmd1  
cmd2  
cmd3
- [addr[,addr]]{cmd1;cmd2;cmd3}

## Affichage/Non affichage

- L'option `-n` force sed a ne rien afficher sur stdout
- On va pouvoir utiliser `p` pour forcer l'affichage dans des commandes sed

## Sélection de ligne

- Afficher la troisième ligne : `sed -n 3p fichier`
- Afficher un intervalle de ligne (de 7 à 10) :  
`sed -n '7,3 p' fichier`
- Afficher un groupe de ligne (toutes les 2 lignes à partir de la 3) :  
`sed -n 3~2p fichier`
- Afficher la dernière ligne d'un fichier : `sed -n '$ p' fichier`

## Ma regexp me manque !!

- Sélection (et affichage) de ligne par motif
- `sed -n '/motif/'` affiche toutes les lignes ou apparait le motif.
- `\#exp#` change le délimiteur pour `#`
- `/begin/,/end/ p` : du motif begin au motif end.
- `num,/motif/ /motif/,num` : de la ligne num au motif (et inverse).

# Options

## Paramètres

- -e cmd exécution d'une commande (peut-etre utilisé plusieurs fois)
- -n silencieux
- -i [Suffixe] sauvegarde le fichier dans fichier.Suffixe et écrase le précédent
- -r Utilisation de RE étendue.



# Fichiers de scripts

## Commentaires

- `#Commentaire`
- `#n` au début : force l'option `-n`

## Commandes

- `q` : quitter
- `d` : delete
- `p` : print
- `n` : next line
- Accolades : groupement de commandes
- `s` : substitution

# Fichiers de scripts

## Exemples

- Supprimer les lignes contenant "Vol pour Madrid"
- `/Vol pour Madrid/d`
- Afficher la ligne suivant la 4e.
- `4 n;p`
- Afficher la quatrième ligne et quitter
- `4 n;q`

# La substitution

## Substitution simple

- `s/motif/motif2/`
- motif est une 'basic regular expression' (backslash de |)
- On peut mettre des backreference dans motif2 (backslash parenthèse)
- `&` représente le motif en correspondance.

## Substitution composée

- `s/motif/motif2/drapeaux`

# La substitution

## Les drapeaux

- g : global (toutes les occurrences sur la ligne)
- N (nombre) : la Nième occurrence
- p : print (affiche la ligne remplacée)
- w : write (enregistre la ligne dans un fichier)
- e : evaluate (fait interpréter une commande par le shell)
- l : case insensitive
- M : compliqué (pour le traitement de plusieurs lignes)

# La substitution

## Exemples

- Substituer les lignes contenant "Vol pour Rio" par Annulé
- `s/Vol pour Rio/Annulé/`
- Ecrire l'heure à la place du nom du vol
- `s/Vol pour Rio/date/e`
- Trouver toutes les instances de Chapitre, quelque soit sa casse, et remplacer par CH
- `s/Chapitre/CH/gpi`

# Substitution

y!

- y/AEIOU/aeiou/
- Remplace un caractère de la première chaîne, par celui correspondant dans la seconde.

## Autres commandes

- a  
text : ajout de texte (après la mise en correspondance)
- i  
text : ajout de texte (avant la mise en correspondance)
- c  
text : échange la/les lignes(s) en correspondance
- r fichier : lit un fichier
- w fichier : écrit la ligne dans le fichier.

# Substitution

## Exemples

- Ajouter Vol annulé a la suite du nom de vol
- /Vol pour Rio/a Vol annulé
- Après la 5e ligne
- 5a Vol Annulée