

Guide `ed/red/sed` Pour l'Impatient(e)

“`ed/red` - text editor”, /usr/bin/man 2004.

“sed is a stream editor”, /usr/bin/man 2004.

“Le Guide `ed/red/sed` Pour l'Impatient(e) présente un éditeur antipathique sympathique.”, nj 2005.

Point de départ : un fichier à écrire ou à modifier.

Objectif : écrire un ensemble de commandes visant à modifier ce fichier et afficher le résultat.

1 Édition d'un fichier avec `ed`

`ed fic1` édite le fichier `fic1` ligne à ligne s'il existe. L'ensemble des modifications est réalisé sur une image mémoire du fichier `fic1` et devient effectif par le biais d'une commande explicite. Édité ligne à ligne, les opérations sont réalisées à chaque instant sur la ligne courante. `ed` peut fonctionner selon deux modes :

- le mode commande (actif par défaut) ;
- le mode insertion.

En cas de séquence de commandes inattendue, le caractère `?` apparait en début de ligne. La commande `h` demande le développement du message d'erreur associé au dernier caractère `?` affiché. La commande `H` implique l'affichage de tous les messages d'erreur pour tous les caractères `?`. En mode commande, `! cmd` permet d'exécuter la commande `cmd` sans sortir de l'éditeur. `red` est une restriction de `ed` limitée au répertoire courant et ne pouvant pas exécuter de commande avec `!` `cmd`.

- `f` affiche le nom du fichier courant
- `.` affiche le numero de ligne courante
- `f fic3` fixe le nom du fichier courant à `fic3`
- `w` enregistre les modifications dans le fichier courant
- `w fic3` enregistre les modifications dans le fichier `fic3`
- `1,3 w fic5` sauvegarde les lignes 1 à 3 dans le fichier `fic5`
- `q` permet de quitter l'éditeur, en laissant une chance de sauvegarde avec un contrôle de sauvegarde actif ; La seconde tentative de sortie est toujours effective et les modifications non sauvegardées sont perdues

1.1 Affichage

- `1` déplace la ligne courante sur la ligne 1 et l'affiche
- `<ENTREE>` déplace la ligne courante sur la ligne suivante et l'affiche
- `$` déplace la ligne courante sur la dernière ligne et l'affiche
- `1,3 1` affiche les lignes 1 à 3 (caractères non imprimables)
- `1,$ p` affiche toutes les lignes du fichier courant (caractères imprimables)
- `.,+25 n` affiche les 25 prochaines lignes avec leur numéro de ligne (caractères imprimables)

1.2 Mode insertion

- `a` passe en mode insertion après la ligne courante
- `i` passe en mode insertion avant la ligne courante
- `c` passe en mode insertion en substituant la ligne courante
- `.` permet de revenir au mode commande

1.3 Insertion, suppression, déplacement

- `r fic4` insère le contenu du fichier `fic4` après la ligne courante
- `3 r fic4` insère le contenu du fichier `fic4` après la ligne 3
- `d` supprime la ligne en cours
- `1 d` supprime la première ligne
- `1,3 d` supprime les lignes 1 à 3
- `1,$ j` supprime les caractères de fin de ligne pour toutes les lignes
- `1,$ m 0` déplace toutes les lignes après la ligne 0
- `2,4 t 3` recopie les lignes 2 à 4 après la ligne 3

1.4 Motifs et substitution

- `/mot/` affiche la première ligne contenant le motif 'mot'
- `g/mot/` affiche toutes les lignes contenant le motif 'mot'
- `s/mot/met/g` remplace 'mot' par 'met' dans toutes les lignes
- `1,$s/^~/` insère un espace au début de chaque ligne
- `g/^/s// /p` insère un espace au début de chaque ligne et affiche le résultat
- `s/abc/&d&e/` remplace 'abc' par 'abcdabce' sur la ligne courante

2 Éditeur d'un fichier avec sed

Le principe de sed est de lire un fichier `fic1` ligne à ligne en appliquant des commandes et rediriger les lignes résultantes sur la sortie standard; Le fichier de commandes `t1.sed` est appliqué avec `sed -f t1.sed fic1`; Autorisé à l'exécution, il doit contenir en première ligne `#!/bin/sed -f`. `fic1` n'est pas modifié. Dans un script sed, les commentaires sont précédés du caractère #.

- `1,3 s/abc/def/` remplace la première occurrence de 'abc' par 'def' pour les lignes 1 à 3 du fichier courant
- `1,3 s/abc/def/g` remplace 'abc' par 'def' pour les lignes 1 à 3 du fichier courant
- `1,3 s/abc/def/p` remplace et affiche la première occurrence de 'abc' par 'def' pour les lignes 1 à 3 du fichier courant
- `1,3 s/abc/def/gw fic3` remplace 'abc' par 'def' pour les lignes 1 à 3 du fichier courant et écrit le résultat dans le fichier `fic3`
- `6,10 d` supprime les lignes 6 à 10
- `4a\ <texte>` ajoute du texte après la ligne 4
- `4i\ <texte>` insère du texte avant la ligne 4
- `w fic3` écrit dans le fichier `fic3`
- `=` écrit les numéro de ligne

3 Exemple

```
#!/bin/sed -f
# c2tex - c vers tex
s/-/-/g
s/[ ]*$/ /
s/\\//backslash/g
s/[{}#%&_]/\&/g
s/\\backslash/{&}/g
s/~//~/g
s/~/~/g
s/e :/e{ }/g
s/$/ \\ mbox{} \\\ /
s/^\\mbox{/ /
1i\
\\begin{flushleft}\\hspace{1cm}\\begin{tt}\\begin{tabular}{|l}
$s$/\\end{tabular}\\end{tt}\\end{flushleft}/
```