

SCR1



"Cause a little ~~awk~~ awkward
goes a long way."

awk : expressive language

- Exécute les « commandes » définies par l'utilisateur sur les données entrantes
- Orienté ligne, découpe les « colonnes » selon un délimiteur
- Le résultat sort sur la sortie standard (stdout)
- Aucune modification des données entrée (pas d'altération du fichier d'entrée)

awk : principe

- Le programme contient une suite de

```
motif { action }
```

- Si le motif est absent, l'action est exécutée pour **chaque ligne** de l'entrée
- Si l'action est absente, seules les lignes correspondantes au motif **sont affichées**
- **Tout** le programme est traité pour chaque ligne de l'entrée

awk : motif

- Expression régulière écrite entre deux slash

```
/\w+\.txt/
```

```
!/^-/
```

- Expression de comparaison

```
$5 == "Jun"
```

```
$5 ~ /May|Jun/
```

- Appel de fonction

```
is_link($0)
```


awk : champs

- Chaque ligne est automatiquement découpée en **champs**, la ligne complète dans la variable **\$0**, les champs dans **\$1**, **\$2**, etc. **NF** contient le nombre de champs.

- Définir le séparateur de champs

```
BEGIN { FS = ";" }
```

- L'entrée est découpée en enregistrement par le séparateur défini par **RS**. **NR** contient le nombre d'enregistrement traités

awk : action

- Variables auto-déclarée
- 3 types: **String**, **Numeric**, **Array**
- Structures de contrôles à la C
(if, while, do-while, for, switch)
- Arguments du scripts:
ARGC: Nombre d'arguments
ARGV: Tableau d'arguments

awk : exemples

- Supprime les lignes 10 à 20

```
awk 'NR < 10 || NR > 20'
```

- Plagiat: Convertit tous les Torvalds par Gates dans les commentaires

```
awk '/\s*#/ {gsub(/Torvalds/, "Gates"); print} !/\s*#/'
```