Référence rapide AWK

Motifs, champs, tableaux, fonctions, traitement de texte

Bases

Lancer AWK

```
awk '{ print }' file.txt          # afficher chaque ligne
awk '{ print $1 }' file.txt       # afficher le premier champ
awk -F: '{ print $1 }' /etc/passwd  # délimiteur personnalisé
awk -f script.awk file.txt        # exécuter depuis un fichier
cmd | awk '{ print $2 }'          # entrée via pipe
```

Structure du programme

`awk 'pattern { action }'`	Forme de base — l'action s'exécute quand le motif correspond
`BEGIN { ... }`	Exécuté une fois avant le traitement
`END { ... }`	Exécuté une fois après tout le traitement
`Sans motif`	L'action s'exécute pour chaque ligne
`Sans action`	Action par défaut : `{ print }`

Motifs et actions

Types de motifs

```
awk '/error/' file.txt            # correspondance regex
awk '$3 > 100' file.txt           # comparaison
awk 'NR >= 5 && NR <= 10' file.txt  # plage de lignes
awk '/start/,/end/' file.txt      # motif de plage
```

Référence des motifs

`/regex/`	Faire correspondre la ligne à une regex
`$1 ~ /pat/`	Le champ correspond à la regex
`$1 !~ /pat/`	Le champ ne correspond pas à la regex
`expr1, expr2`	Plage : de la première à la deuxième correspondance
`expr1 && expr2`	ET logique
`expr1 || expr2`	OU logique
`!expr`	NON logique

Variables

Variables intégrées

`NR`	Numéro d'enregistrement (ligne) courant
`NF`	Nombre de champs dans l'enregistrement courant
`FS`	Séparateur de champ en entrée (défaut : espace)
`OFS`	Séparateur de champ en sortie (défaut : espace)
`RS`	Séparateur d'enregistrement en entrée (défaut : saut de ligne)
`ORS`	Séparateur d'enregistrement en sortie (défaut : saut de ligne)
`FILENAME`	Nom du fichier d'entrée courant
`FNR`	Numéro d'enregistrement dans le fichier courant

Variables utilisateur

```
awk '{ total += $1 } END { print total }' file.txt
awk -v threshold=50 '$1 > threshold' file.txt
awk 'BEGIN { count = 0 } /pat/ { count++ }
     END { print count }' file.txt
```

Champs

Accès aux champs

`$0`	Ligne entière courante
`$1, $2, ...`	Premier, deuxième, ... champ
`$NF`	Dernier champ
`$(NF-1)`	Avant-dernier champ

Séparateurs de champs

```
awk -F, '{ print $2 }' data.csv     # virgule
awk -F'\t' '{ print $1 }' data.tsv  # tabulation
awk 'BEGIN { FS = "[,:]" } { print $1 }' f  # multi-car
awk 'BEGIN { OFS = "," } { print $1, $3 }' f  # sep sortie
```

Flux de contrôle

Conditions et boucles

```
awk '{ if ($1 > 50) print "high"; else print "low" }' f
awk '{ for (i = 1; i <= NF; i++) print $i }' f
awk '{ i = 1; while (i <= NF) { print $i; i++ } }' f
awk '/skip/ { next } { print }' f  # ignorer les lignes
correspondantes
```

Instructions de contrôle

`if (cond) { ... } else { ... }`	Condition
`for (i = 0; i < n; i++) { ... }`	Boucle for style C
`for (key in array) { ... }`	Itérer sur les clés d'un tableau
`while (cond) { ... }`	Boucle while
`do { ... } while (cond)`	Boucle do-while
`next`	Passer à l'enregistrement suivant
`exit`	Arrêter le traitement, exécuter le bloc END

Fonctions

Fonctions définies par l'utilisateur

```
awk 'function max(a, b) {
    return (a > b) ? a : b
}
{ print max($1, $2) }' file.txt
```

Fonctions numériques

`int(x)`	Tronquer vers l'entier
`sqrt(x)`	Racine carrée
`sin(x), cos(x)`	Fonctions trigonométriques
`log(x), exp(x)`	Logarithme naturel et exponentielle
`rand()`	Flottant aléatoire entre 0 et 1
`srand(seed)`	Initialiser le générateur de nombres aléatoires

Tableaux

Tableaux associatifs

```
awk '{ count[$1]++ }
     END { for (k in count) print k, count[k] }' f
awk '{ arr[NR] = $0 }
     END { for (i = NR; i >= 1; i--) print arr[i] }' f
```

Opérations sur les tableaux

`arr[key] = val`	Définir un élément
`arr[key]`	Lire un élément (créé automatiquement à l'accès)
`key in arr`	Vérifier si la clé existe
`delete arr[key]`	Supprimer un élément
`delete arr`	Supprimer tout le tableau
`for (k in arr)`	Itérer sur les clés (ordre non défini)
`length(arr)`	Nombre d'éléments (gawk)

Fonctions de chaînes

Référence des fonctions de chaînes

`length(s)`	Longueur de la chaîne
`substr(s, start, len)`	Sous-chaîne (indexée à 1)
`index(s, target)`	Position de target dans s (0 si absent)
`split(s, arr, sep)`	Découper une chaîne en tableau
`sub(pat, repl, s)`	Remplacer la première occurrence
`gsub(pat, repl, s)`	Remplacer toutes les occurrences
`match(s, pat)`	Position de la correspondance regex (définit RSTART, RLENGTH)
`tolower(s) / toupper(s)`	Conversion de casse
`sprintf(fmt, ...)`	Formater une chaîne (comme printf en C)

Exemples de chaînes

```
awk '{ gsub(/old/, "new"); print }' f    # remplacement style sed
awk '{ print toupper($0) }' f             # tout en majuscules
awk '{ print substr($0, 1, 40) }' f       # tronquer à 40 caractères
```

E/S

Sortie

```
awk '{ print $1, $2 }' f              # séparé par des espaces
awk '{ printf "%s,%d\n", $1, $2 }' f  # sortie formatée
awk '{ print $1 > "out.txt" }' f      # rediriger vers un fichier
awk '{ print $1 >> "out.txt" }' f     # ajouter à un fichier
```

Référence E/S

`print`	Afficher avec ORS (saut de ligne par défaut)
`printf fmt, ...`	Affichage formaté (sans saut de ligne final)
`print > file`	Rediriger la sortie vers un fichier
`print >> file`	Ajouter la sortie à un fichier
`print | cmd`	Envoyer la sortie vers une commande
`getline < file`	Lire une ligne depuis un fichier
`cmd | getline var`	Lire la sortie d'une commande dans une variable
`close(file)`	Fermer un fichier ou un pipe

Patterns courants

Raccourcis

```
awk '{ sum += $1 } END { print sum }' f   # sommer une colonne
awk 'END { print NR }' f                   # compter les lignes
awk '!seen[$0]++' f                        # supprimer les doublons
awk 'NF' f                                 # supprimer les lignes
vides
awk '{ print NF }' f                       # champs par ligne
```

Recettes

`CSV vers TSV`	`awk -F, 'BEGIN{OFS="\t"} {$1=$1; print}'`
`Somme colonne 2`	`awk '{ s += $2 } END { print s }'`
`N premières lignes`	`awk 'NR <= 10'` (comme head)
`Comptage de fréquences`	`awk '{ c[$1]++ } END { for (k in c) print k, c[k] }'`
`Entre des marqueurs`	`awk '/BEGIN/,/END/'`
`Afficher le N-ième champ`	`awk '{ print $N }'` (remplacer N)