
grep

Affiche les lignes qui contiennent une correspondance pour un motif

Options - contrôle de correspondance

- e PATTERN, -regexp=PATTERN** Utilise PATTERN comme motif. Si cette option est utilisée plusieurs fois ou combinée avec -f, recherche tous les motifs donnés.
- f FILE, -file=FILE** Obtient les patterns depuis le fichier, un par ligne. Si cette option est utilisée plusieurs fois ou combinée avec -e, recherche pour tous les patterns donnés. Un fichier vide ne contient aucun pattern, et ne matche rien.
- i, -ignore-case** Ignore la casse.
- v, -invert-match** Inverse le sens de match, pour sélectionner des lignes non-matchant.
- w, -word-regexp** Sélectionne uniquement les lignes contenant les matchs qui forment un mot entier. Le test est que la sous-chaîne matchant doit être soit au début de la ligne, ou précédé par un caractère non-word. Similairement, il est doit être soit à la fin de la ligne ou suivi par un caractère word. N'a pas d'effet avec -x
- x, -line-regexp** Sélectionne seulement les matchs qui correspondent exactement à la ligne entière.

Options - Contrôles de sortie généraux

- c, -count** Supprime la sortie normale ; Affiche un compteur de lignes correspondantes pour chaque fichier d'entrée. Avec -v, compte les lignes qui ne matchent pas.
- color [=WHEN], -colour [=WHEN]** Mode couleur. Les couleurs sont définies par la variable d'environnement GREP_COLORS (défaut : ms=01 ;31 :mc=01 ;31 :sl= :cx= :fn=35 :ln=32 :bn=32 :se=36)
- L, -files-without-match** Supprime la sortie normale ; affiche le nom de chaque fichier d'entrée dans lequel aucune sortie ne serait affichée.
- l, -files-with-matches** Supprime la sortie normale ; affiche le nom de chaque fichier d'entrée dans lequel une sortie aurait été affichée.
- m NUM, -max-count=NUM** Stop la lecture d'un fichier après NUM lignes matchant.
- o, -only-matching** Affiche seulement les parties matchants des lignes matchant.
- q, -quiet, -silent** N'affiche rien sur stdout. Quitte avec un status 0 si un match est trouvé, même si une erreur a été détectée.
- s, -no-messages** Supprime les messages d'erreur sur des fichiers non-existants ou illisibles.

Options - contrôle de préfixe de ligne

- b, -byte-offset** Affiche l'offset dans le fichier d'entrée avant de sortir chaque ligne. si -o est spécifié, affiche l'offset de la partie matchant elle-même.
- H, -with-filename** Affiche le nom de fichier pour chaque match. C'est le défaut quand il y a plus d'un fichier à rechercher.
- h, -no-filename** Supprime l'ajoute du nom de fichier dans la sortie. C'est le défaut quand il n'y a qu'un fichier ou seulement l'entrée standard à rechercher
- label=LABEL** Affiche l'entrée venant de stdin comme entrant depuis le fichier LABEL. Utile pour implémenter des outils comme zgrep (ex : gzip -cd foo.gz | grep -label=foo -H something)
- n, -line-number** Préfixe chaque ligne de sortie avec le numéro de ligne dans son fichier d'entrée

-
- T**, **-initial-tab** S'assure que le premier caractère du contenu de la ligne courante s'appuie sur une tabulation, pour que l'alignement des tabulations semblent normales. Utile avec les options qui préfixent leur sortie avec le contenu actuel.
 - u**, **-unix-byte-offsets** Reporte les offsets dans le style Unix.
 - Z**, **-null** Affiche un ASCII NUL au lieu du caractère qui suit normalement un nom de fichier.

Options - Contrôle de ligne de contexte

Sans regarder comment ces options sont définies, `grep` n'affiche jamais une ligne donnée plus d'une fois. Si `-o` est spécifié, ces options n'ont pas d'effet et une alerte est donnée.

- A** NUM, **-after-context=NUM** Affiche NUM lignes supplémentaires après chaque ligne matchant
- B** NUM, **-before-context=NUM** Affiche NUM lignes supplémentaires avant chaque ligne matchant
- C** NUM, **-NUM**, **-context=NUM** Affiche NUM lignes avant et après chaque ligne matchant
- group-separator=STRING** Avec `-A`, `-B` ou `-C`, affiche un séparateur entre les groupes de ligne
- no-group-separator** Avec `-A`, `-B` ou `-C`, n'affiche pas de séparateur entre les groupes de ligne

Notes sur la manière dont `grep` choisit le séparateur pour afficher les préfixes :

- Les lignes matchant utilisent normalement ' : ' comme séparateur entre les préfixes et le contenu de la ligne courante
- Les lignes non-matchant utilisent - à la place
- Quand le contexte n'est pas spécifié, les lignes adjacentes dans l'entrée forment un groupe et sont affichées une après l'autre, alors qu'un séparateur apparaît entre les groupes non-adjacents
- Le séparateur par défaut est -
- Chaque groupe peut contenir de nombreuses lignes matchant quand elles sont suffisamment proches et peuvent être fusionnées en une simple zone contiguë.

Options - Sélection de fichier et de répertoire

- a**, **-text** Traite un fichier binaire comme si c'était du texte.
- binary-files=TYPE** Trait le fichier contenant des données binaires en assumant que le fichier est du type spécifié : `binary`, `without-match`, `text`.
- D** ACTION, **-devices=ACTION** Si un fichier d'entrée est un périphérique, FIFO ou socket, utilise ACTION pour le traiter. Si ACTION est 'read', tous les périphériques sont lus comme si c'étaient des fichiers ordinaires. Si ACTION est 'skip', les périphériques, FIFO, et sockets sont ignorés silencieusement.
- d** ACTION, **-directories=ACTION** Si un fichier d'entrée est un répertoire, utilise l'action pour le traiter. par défaut, ACTION est 'read', qui signifie que les répertoires sont lus comme si c'était des fichiers ordinaires. Si ACTION est 'skip', les répertoires sont ignorés silencieusement. 'recurse' lis tous les fichiers sous chaque répertoire, récursivement.
- exclude=GLOB** Saute les fichiers de la ligne de commande avec un suffixe de nom qui matche le glob.
- exclude-from=FILE** Saute les fichiers dont les noms matchent un des patterns lus depuis le fichier spécifié.
- exclude-dir=GLOB** Saute les répertoires de la ligne de commande avec un suffixe qui matche le glob.
- I** Traite un fichier binaire comme s'il ne contenait pas de données correspondantes. (équivalent à `-binary-files=without-match`)
- include=GLOB** Ne recherche que les fichiers dont les noms matchent le glob.
- r**, **-recursive** Pour chaque opérande répertoire, lit et traite tous les fichiers dans ce répertoire, récursivement. identique à `-directories=recurse`
- R**, **-dereference-recursive** Pour chaque opérande répertoire, lit et traite tous les fichiers dans ce répertoire, en suivant tous les liens symboliques.

Autres options

-line-buffered Utilise le tampon de ligne sur la sortie. Peut réduire les performances

-U, -binary Traite les fichiers en binaire.

-z, -null-data Traite les données d'entrée et de sortie comme séquence de lignes, chacune terminée par un ASCII NUL au lieu d'un newline.

Variables d'environnement

LC_ALL Spécifie la locale

LC_COLLATE Spécifie la locale pour la catégorie LC_COLLATE

LANG Spécifie la locale

LC_TYPE Spécifie la locale pour la catégorie LC_CTYPE

LC_MESSAGES Spécifie la locale pour la catégorie LC_MESSAGES

LANGUAGE Contient une liste séparée par ',' de langages qui remplacent les variables LC_ALL, LC_XXX et LANG

GREP_COLORS Spécifie les couleurs et autres attributs utilisés pour colorer la sortie :

sl= Lignes sélectionnées

cx= Lignes de contexte

rv booléen inversant la signification de sl et cx

mt=01;31 Texte non-vide dans un ligne matchant.

ms=01;31 Texte non vide matchant dans une ligne matchant

mc=01;31 texte non-vide matchant dans une ligne de contexte

fn=35 Sous-chaîne pour les noms de fichier préfixant une ligne

ln=32 Sous-chaîne pour les numéros de ligne préfixant une ligne

bn=32 Offset préfixant une ligne

se=36 Séparateur inséré entre les champs, entre les lignes de contexte et entre les groupes de lignes adjacentes

ne Booléen qui empêche d'effacer la fin d'une ligne avec Erase in Line (EL) à droite chaque fois qu'un élément colorisé prend fin.

POSIXLY_CORRECT grep se conforme à POSIX

_N_GNU_nonoption_argv_flags_ Où N est l'ID de processus grep. Si le I-ième caractère de cette variable est 1, ne pas considérer la I-ème opérande de grep comme option. Un shell peut définir cette variable pour chaque commande qu'il lance, en spécifiant quels opérandes sont le résultat d'une expansion de nom de fichier. Uniquement avec POSIXLY_CORRECT

Codes de sortie

Normalement, le code de sortie est 0 si une ligne est sélectionnée, 1 si aucune ligne n'est sélectionnée, et 2 si une erreur se produit.

Programmes grep

grep recherche dans les fichiers nommés en entrée à la recherche de lignes contenant un match. Par défaut, grep affiche les lignes matchants. Un fichier nommé '-' signifie l'entrée standard. Si aucune entrée n'est spécifiée, grep recherche le répertoire courant. Si une options spécifie la récursion, cependant, grep recherche l'entrée standard. Il y a 4 variantes majeurs de grep, contrôlés par les options suivantes :

-G, -basic-regexp Interprète le pattern comme expression régulière basique. c'est le mode pas défaut

- E, **-extended-regexp** Interprète le pattern comme expression régulière étendue.
- F, **-fixed-string** Interprète le pattern comme une liste de chaîne fixes au lieu d'expressions régulières, séparé par des newline, un d'entre eux doit matcher.
- P, **-perl-regexp** Interprète le pattern comme expression régulière compatible perl.

Expressions régulières

Une expression régulière est un pattern qui décrit un jeu de chaînes. Les expressions régulières sont construites analogiquement aux expressions arithmétiques, en utilisant divers opérateurs pour combiner des expressions plus petites. Grep comprend 3 types d'expressions régulières : basique, étendue, et perl. La suite décrit les expressions régulières étendue.

Structure fondamentale

- .
- ? L'élément précédent est optionnel et est matché une fois au plus
- * L'élément précédent est matché 0 ou plusieurs fois
- + L'élément précédent est matché 1 ou plusieurs fois
- {N} L'élément précédent est matché N fois
- {N,} L'élément précédent est matché au moins N fois
- {,M} L'élément précédent est matché au plus N fois
- {N,M} L'élément précédent est matché au moins N fois et au plus M fois.

Classes de caractère et expressions entre crochet

- [**:alnum :**] Caractères alphanumériques [=0-9A-Za-z]
- [**:alpha :**] Caractères alphabétiques [A-Za-z]
- [**:blank :**] Caractères blancs : espace et tabulation
- [**:cntrl :**] Caractères de contrôle
- [**:digit :**] Chiffres [0-9]
- [**:graph :**] Caractères graphiques : [:alnum :] et [:punct :]
- [**:print :**] Caractères imprimables. [:alnum :], [:punct :], et espace
- [**:punct :**] caractères de ponctuation ' ! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ' { | } ~ '
- [**:space :**] Caractères espace
- [**:lower :**] caractères minuscules [a-z]
- [**:upper :**] Caractères majuscules [A-Z]
- [**:xdigit :**] Chiffres hexadécimaux [0-9A-Fa-f]

La plupart des méta-caractères perdent leur signification spéciale dans les expressions entre crochet.

-] Termine une expression crochet si ce n'est pas le premier élément de la liste
- [. Représente le symbole d'ouverture de regroupement.
- .] Représente le symbole de fermeture de regroupement
- [= Représente l'ouverture de classe équivalente
- =] Représente la fermeture de classe équivalente

-
- [: Représente le symbole d'ouverture de classe de caractère
 -] Représente le symbole de fermeture de classe de caractère
 - Représente la plage si ce n'est pas le premier ou le dernier caractère
 - ^ Représente les caractères qui ne sont pas dans la liste.

Le caractère \ et des expressions spéciales

- \b Matche la chaîne vide au bord d'un mot
- \B Matche la chaîne vide qui n'est pas au bord d'un mot
- \< Matche la chaîne vide au début d'un mot
- \> Matche la chaîne vide à la fin d'un mot
- \w Matche un constituant d'un mot, synonyme de [_[:alnum:]]
- \W Match un non-constituant de mot, synonyme de [^[:alnum:]]
- \s Matche un espace blanc, synonyme de [[:space :]]
- \S Matche un non-espace, synonyme de [^[:space:]]

ancres

^ et \$ sont des méta-caractères qui matchent respectivement la chaîne vide au début de ligne et la fin de ligne.

Références et sous-expressions

Le back-référence, \N, où N est un simple chiffre, matche la sous-chaîne précédemment matchée par la N-ième sous-expression entre parenthèses de l'expression régulière. Par exemple, (a)\1 matche 'aa'.

Expressions régulières basiques et étendues

Dans les expressions régulières basiques, les méta-caractères '?', '+', '{', '|', '(', et ')' perdent leur signification spéciales, utiliser leur versions échappées \.

egrep ne supporte pas le méta-caractère { et certaines implémentations supportent \{, les scripts portables doivent éviter { dans les pattern grep -E, et utiliser [{}].

Exemples

Lister simplement les noms des fichiers qui matchent

```
grep -l 'main' *.c
```

Rechercher récursivement

```
grep -r 'hello' /home/gigi
```

Rechercher seulement dans les fichiers C

```
find /home/gigi -name '*.c' -print0 | xargs -0r grep -H 'hello'
```

Similairement

```
grep -rH --include='*.c' 'hello' /home/gigi
```

Recherche toutes les lignes matchant –cut here–

grep -e '-cut here-' *

Recherche seulement les instances de hello qui sont des mots entiers :

grep -w 'hello' *

Et ne matche pas Othello. Pour plus de contrôle, utilise \< et \>

grep 'hello\>' *

Afficher le contexte autour des lignes qui matchent

grep -C 2 'hello' *

Forcer grep à afficher le nom d'un fichier

grep 'eli' /etc/passwd /dev/null

ou encore

grep -H 'eli' /etc/passwd

Rechercher sur l'entrée standard et dans les fichiers

cat /etc/passwd | grep 'alain' - /etc/motd

Comment exprimer des palindromes en expressions régulières

grep -w -e '\(.\)\(.\)\2\1' file