

man grep (Commandes) - Afficher les lignes correspondant à un motif donné.



chrome

Le navigateur rapide
et gratuit de Google.

Oui, télécharger Chrome

NOM

grep, egrep, fgrep, rgrep - Afficher les lignes correspondant à un motif donné.

SYNOPSIS

grep [options] MOTIF [FICHIER...]

grep [options] [-e MOTIF | -f FICHIER] [FICHIER...]

DESCRIPTION

Grep recherche dans les *FICHIERS* indiqués (ou depuis l'entrée standard si aucun fichier n'est fourni, ou si le nom - est mentionné) les lignes correspondant à un certain *MOTIF*. Par défaut, **grep** affiche les lignes qui correspondent au motif.

De plus, il existe trois variantes du programme : **egrep**, **fgrep** et **rgrep**. **Egrep** est identique à **grep -E**. **Fgrep** est identique à **grep -F**. **Rgrep** est identique à **grep -r**.

OPTIONS

-A NOMBRE, --after-context=NOMBRE

Afficher *NOMBRE* lignes de contexte supplémentaires après les lignes qui concordent. Une ligne contenant -- est mise entre les groupes contigus de correspondances.

-a, --text

Traiter un fichier binaire comme s'il s'agissait de texte ; cette option est équivalente à l'option **--binary-files=text**.

-B NOMBRE, --before-context=NOMBRE

Afficher *NOMBRE* lignes de contexte supplémentaires avant les lignes qui concordent. Une ligne contenant -- est mise entre les groupes contigus de correspondances.

-C NOMBRE, --context=NOMBRE

Afficher *NOMBRE* lignes de contexte. Une ligne contenant -- est mise entre les groupes contigus de correspondances.

-b, --byte-offset

Avant chaque ligne, afficher sa position relative (en octets) au sein du fichier.

--binary-files=TYPE

Si les premiers octets d'un fichier indiquent qu'il contient des données binaires, supposer que le fichier est de type *TYPE*. Par défaut, *TYPE* est **binary** (binaire), et soit **grep** affiche un message d'une ligne disant que le fichier binaire concorde, soit il n'affiche rien s'il n'y a pas concordance. Si *TYPE* est **without-match** (sans concordance), **grep** suppose qu'un fichier binaire ne concorde jamais. C'est équivalent à l'option **-I**. Si *TYPE* est **text** (texte), **grep** traite un fichier binaire comme du texte. C'est équivalent à l'option **-a**. *Attention* : **grep --binary-files=text** peut afficher des données binaires indésirables, qui peuvent avoir des conséquences fâcheuses si la sortie est envoyée vers un terminal, et si le pilote de celui-ci interprète cet affichage comme étant des commandes.

--colour[=QUAND], --color[=QUAND]

Entourer la chaîne correspondante avec les marqueurs trouvés dans la variable d'environnement **GREP_COLOR**. Les valeurs possibles de *QUAND* sont **never** (jamais), **always** (toujours) ou **auto** (automatique).

-c, --count

Ne pas afficher les résultats normaux. À la place, afficher un décompte des lignes correspondantes pour chaque fichier. Avec l'option **-v**, **--invert-match** (voir plus bas), afficher le nombre de lignes ne correspondant pas au motif.

-d ACTION, --devices=ACTION

Si le fichier est un périphérique, une FIFO ou une socket, utiliser *ACTION* dessus. Par défaut, *ACTION* est **read** (lecture), ce qui signifie que les périphériques sont lus comme des fichiers normaux. Si *ACTION* est **skip** (escamotage), les périphériques ne sont pas pris en compte et aucun message n'est affiché.

-d ACTION, --directories=ACTION

Si le fichier est un répertoire, utiliser *ACTION*. Par défaut, *ACTION* est **read** (lecture), ce qui signifie que les répertoires sont lus comme des fichiers normaux. Si *ACTION* est **skip** (escamotage), les répertoires ne sont pas pris en compte et aucun message n'est affiché. Si *ACTION* est **recurse** (récursivité), **grep** lit tous les fichiers présents dans chaque répertoire, récursivement. C'est équivalent à l'option **-r**.

-E, --extended-regexp

Interpréter le *MOTIF* comme une expression rationnelle étendue (voir plus bas).

-e MOTIF, --regexp=MOTIF

Utiliser le *MOTIF* indiqué. Ceci permet de protéger les motifs commençant par -.

-F, --fixed-strings

Interpréter le *MOTIF* comme une liste de chaînes de caractères, séparées par des sauts de ligne. Chacune d'entre elles doit être recherchée.

-P, --perl-regexp

Interpréter le *MOTIF* comme étant une expression rationnelle de Perl.

-f FICHIER, --file=FICHIER[▶ Choisir sa pub](#)[Manuel en ligne](#)[GREP sort](#)[rechercher sur pwet.fr](#)[go](#)[sections du manuel](#)

■ Commandes

- [rtl](#)
- [sr](#)
- [p](#)
- [WN](#)
- [l](#)
- [L](#)
- [fun](#)
- [emacs21](#)
- [m](#)
- [cn](#)
- [emacs-snapshot](#)
- [X](#)
- [gmt](#)
- [grass](#)
- [ssl](#)
- [Z](#)
- [a](#)
- [x](#)
- [osd_clock](#)
- [posix](#)
- [nas](#)
- [PVM](#)
- [pccts](#)

- [Appels systèmes](#)
- [Fonctions bibliothèques](#)
- [Fichiers spéciaux](#)
- [Formats](#)
- [Jeux](#)
- [Conventions](#)
- [Administration système](#)
- [Fonctions noyau](#)
- [Nouveau](#)

[derniers billets](#)[Veille de la semaine #38 de 2018](#)[Veille de la semaine #37 de 2018](#)[Veille de la semaine #36 de 2018](#)[Veille de la semaine #32 de 2018](#)[Veille de la semaine #31 de 2018](#)[Veille de la semaine #30 de 2018](#)[Le code coverage, cet éternel incompris](#)[Veille de la semaine #29 de 2018](#)[Veille de la semaine #28 de 2018](#)[Veille de la semaine #27 de 2018](#)

Lire les motifs dans le *FICHIER* indiqué, un motif par ligne. Un fichier vide ne contient aucun motif, si bien qu'aucune concordance n'est trouvée.

- G, --basic-regexp
Interpréter le *MOTIF* comme une expression rationnelle simple (voir plus bas). C'est le comportement par défaut.
- H, --with-filename
Afficher le nom du fichier pour chaque concordance.
- h, --no-filename
Ne pas afficher le nom des fichiers dans les résultats lorsque plusieurs fichiers sont parcourus.
- help
Afficher un court message d'aide.
- l
Traiter un fichier binaire comme s'il ne contenait aucune concordance ; c'est équivalent à l'option --binary-files=without-match.
- i, --ignore-case
Ignorer les différences majuscules/minuscules aussi bien dans le *MOTIF* que dans les fichiers.
- L, --files-without-match
Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels aucun résultat n'aurait été affiché. La recherche à l'intérieur de chaque fichier s'arrête dès la première concordance.
- l, --files-with-matches
Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels des résultats auraient été affichés. La recherche à l'intérieur de chaque fichier s'arrête dès la première concordance.
- m *NOMBRE*, --max-count=*NOMBRE*
Arrêter de lire un fichier après avoir lu *NOMBRE* lignes qui correspondent. Si l'entrée est l'entrée standard prise depuis un fichier normal, et si *NOMBRE* lignes qui correspondent sont affichées, **grep** s'assure avant de s'arrêter que l'entrée standard est positionnée juste après la dernière ligne qui concorde, même s'il y a des lignes de contexte supplémentaires qui viennent ensuite. Ceci permet au processus d'appel de redémarrer une recherche. Quand **grep** s'arrête après *NOMBRE* lignes de concordance, il affiche éventuellement des lignes de contexte. Quand l'option -c ou --count est utilisée, **grep** n'affiche pas plus que *NOMBRE* lignes. Quand l'option -v ou --invert-match est aussi utilisée, **grep** s'arrête après avoir affiché *NOMBRE* lignes qui ne concordent pas.
- mmap
Si possible, utiliser la fonction système [mmap\(2\)](#) pour lire les données, à la place de la fonction système [read\(2\)](#) utilisée par défaut. Dans certaines circonstances, --mmap permet de meilleures performances. Cependant, --mmap peut avoir un comportement indéfini (y compris la génération d'un *core dump*) si un fichier voit sa taille réduite pendant que **grep** le lit, ou si une erreur d'entrée-sortie survient.
- n, --line-number
Ajouter à chaque ligne de sortie un préfixe contenant son numéro dans le fichier.
- o, --only-matching
N'afficher que la partie d'une ligne qui correspond au *MOTIF*.
- label=*ÉTIQUETTE*
Afficher les données provenant de l'entrée standard comme si elles provenaient du fichier *ÉTIQUETTE*. C'est particulièrement utile pour des outils comme **zgrep**, par exemple, qui font :

 gzip -cd foo.gz | grep --label=truc quelquechose
- line-buffered
Vider la sortie après chaque ligne. Ceci peut détériorer les performances.
- q, --quiet, --silent
Ne pas afficher les résultats normaux. Sortir immédiatement avec un code de retour égal à zéro si aucune concordance n'est trouvée, et même si une erreur est détectée. Voyez aussi l'option -s ou --no-messages.
- R, -r, --recursive
Lire tous les fichiers à l'intérieur de chaque répertoire, récursivement. C'est équivalent à l'option -d recurse.
- include=*MOTIF*
Procéder récursivement dans les répertoires, pour les fichiers qui correspondent au *MOTIF*.
- exclude=*MOTIF*
Procéder récursivement dans les répertoires, pour les fichiers qui ne correspondent pas au *MOTIF*.
- s, --no-messages
Ne pas afficher les messages d'erreur concernant les fichiers inexistantes ou illisibles. Note de portabilité : à la différence de GNU **grep**, le **grep** traditionnel n'est pas conforme au standard POSIX.2, car le **grep** traditionnel n'a pas d'option -q et son option -s agit comme l'option -q du **grep** de GNU. Les scripts shell destinés à être utilisés avec ces différentes versions devraient éviter d'utiliser les options -q et -s, et devraient plutôt rediriger la sortie vers /dev/null.
- U, --binary
Traiter les fichiers comme s'ils étaient des fichiers binaires. Par défaut, sous MS-DOS et MS-Windows, **grep** détermine le type de fichier en regardant le contenu des 32 premiers kilo-octets. Si **grep** décide que le fichier est un fichier de texte, il enlève les retours chariot (CR) du contenu du fichier original (afin que les expressions avec ^ et \$ fonctionnent correctement). L'option -U modifie ce comportement, tous les fichiers sont alors lus et traités tels quels. Si le fichier est un fichier de texte avec des paires CR-LF en fin de ligne, certaines expressions rationnelles peuvent échouer. Cette option n'a aucun effet sur des plates-formes autres que MS-DOS et MS-Windows.
- u, --unix-byte-offsets
Afficher la position relative en octets dans le style Unix. Avec cette option, **grep** affiche la position relative en octets comme si le fichier était un fichier de texte de type Unix, c'est-à-dire avec les caractères CR supprimés. Ceci permet d'avoir un résultat identique à celui retourné par **grep** sur une machine Unix. Cette option n'a d'effet que si elle est utilisée conjointement avec l'option -b, et sur une machine MS-DOS ou MS-Windows.
- V, --version
Afficher le numéro de version de **grep** sur la sortie d'erreur standard. Ce numéro de version devra être inclus dans tous les rapports de bogues (voir plus bas).
- v, --invert-match
Inverser la mise en concordance, pour sélectionner les lignes ne correspondant pas au motif.
- w, --word-regexp

dernières galeries

[Le prieuré de Marcevol](#)
[eZ Conference and eZ Awards 2007](#)
[Le Château de Viviores](#)
[Le Lac de Porté-Puymorens et Andorre](#)
[Le Jardin des Plantes et la faculté de médecine de Montpellier](#)
[Nokia FISE 2006 : BMX Street](#)
[Nokia FISE 2006 : FMX](#)
[L'abbaye Saint-Félix-de-Montceau et la montagne de la Gardiole](#)
[Petit village catalan : Bouleternère](#)
[Mondial de l'automobile 2006](#)

Ne sélectionner que les lignes contenant une concordance formant un mot complet. La sous-chaîne correspondante doit donc être soit au début de la ligne, soit être précédée d'un caractère ne pouvant entrer dans la constitution d'un mot. De même elle doit se trouver soit à la fin de la ligne, soit être suivie par un caractère ne pouvant entrer dans la constitution d'un mot. Les caractères composant les mots sont les lettres, les chiffres et le souligné (« _ »).

-x, --line-regexp

Ne sélectionner que les concordances qui occupent une ligne entière.

-y

Synonyme obsolète pour -i.

-Z, --null

Afficher un octet nul (le caractère ASCII NUL) à la place du caractère qui suit d'ordinaire le nom du fichier. Par exemple, **grep -IZ** affiche un octet nul après chaque nom de fichier, à la place du saut de ligne. Cette option permet de rendre la sortie non ambiguë, même quand les noms de fichiers contiennent des caractères inhabituels, comme des sauts de ligne. Cette option peut être utilisée avec des commandes telles que **find -print0**, **perl -0**, **sort -z**, et **xargs -0** pour traiter des fichiers avec des noms quelconques, même ceux contenant des sauts de ligne.

-z, --null-data

Considérer que l'entrée est un ensemble de lignes, chacune terminée par un octet nul (le caractère ASCII NUL) au lieu d'un saut de ligne. Comme l'option -Z ou --null, cette option peut être combinée avec des commandes comme **sort -z** pour traiter des fichiers ayant un nom quelconque.

EXPRESSIONS RATIONNELLES

Une expression rationnelle (*regular expression*) est un motif qui permet de décrire un ensemble de chaînes. Les expressions rationnelles sont construites comme des opérations arithmétiques ; elles utilisent différents opérateurs pour combiner des expressions plus petites.

Grep comprend deux versions différentes pour la syntaxe des expressions rationnelles : « simple » (*basic*) et « étendue » (*extended*). Dans la version GNU de **grep**, il n'y a pas de différence dans les fonctionnalités disponibles, quelle que soit la syntaxe utilisée. Dans d'autres implantations, les expressions rationnelles simples sont moins puissantes. La description ci-dessous correspond aux expressions étendues, les différences avec les expressions simples étant résumées ensuite.

Les briques élémentaires sont les expressions rationnelles correspondant à un seul caractère. La plupart des caractères, y compris les lettres et les chiffres, sont des expressions rationnelles qui concordent avec eux-mêmes. Tout méta-caractère ayant une signification spéciale doit être protégé en le faisant précéder d'une contre-oblique (*backslash*).

Une liste de caractères, encadrée par [et] peut être mise en correspondance avec n'importe quel caractère appartenant à la liste. Si le premier caractère de la liste est l'accent circonflexe ^ alors la mise en correspondance se fait avec n'importe quel caractère *absent* de la liste. Par exemple, l'expression rationnelle **[0123456789]** concorde avec n'importe quel chiffre.

Entre ces crochets, un intervalle de caractères peut être indiqué en donnant le premier et le dernier caractère, séparés par un tiret. Il correspond à n'importe quel caractère compris entre le premier et le dernier caractère (ceux-ci inclus), l'ordre des caractères dépendant des paramètres régionaux (*locale*, en anglais) en cours. Ainsi avec la valeur par défaut (appelée « C »), **[a-d]** est équivalent à **[abcd]**. Avec beaucoup de paramètres régionaux, les caractères sont triés en suivant l'ordre des dictionnaires, et **[a-d]** n'est alors pas équivalent à **[abcd]**, mais à **[aBbCcDd]**, par exemple. Pour que ces listes aient le comportement usuel de C, vous pouvez positionner la variable d'environnement **LC_ALL** à la valeur **C**.

Enfin, il existe certaines classes de caractères prédéfinies. Leurs noms sont assez explicites : **[:alnum:]**, **[:alpha:]**, **[:cntrl:]**, **[:digit:]** (chiffres), **[:graph:]**, **[:lower:]** (minuscules), **[:print:]** (affichables), **[:punct:]** (ponctuation), **[:space:]** (espace), **[:upper:]** (majuscules), et **[:xdigit:]** (chiffres hexadécimaux). Par exemple, **[:alnum:]** correspond à **[0-9A-Za-z]**, à la différence près que le dernier dépend des paramètres régionaux C et du codage de caractères ASCII, alors que le premier est plus portable. Remarquez que les crochets dans les noms de classes font partie intégrante du nom symbolique, et qu'ils doivent donc être inclus en plus des crochets encadrant la liste. La plupart des méta-caractères perdent leur signification spéciale au sein des listes. Pour inclure un caractère], mettez-le en premier dans la liste. De même, pour inclure un caractère ^, placez-le n'importe où sauf au début de la liste. Enfin, pour inclure un -, placez-le en dernier.

Le point . correspond à n'importe quel caractère. Le symbole \w est un synonyme de **[:alnum:]** et \W un synonyme de **[^[:alnum:]]**.

L'accent circonflexe ^ et le symbole dollar \$ sont des méta-caractères correspondant respectivement à une chaîne vide au début et en fin de ligne. Les symboles \< et \> correspondent respectivement à une chaîne vide en début et en fin de mot. Le symbole \b correspond à une chaîne vide à l'extrémité d'un mot, et \B correspond à une chaîne vide ne se trouvant *pas* à une extrémité de mot.

Une expression rationnelle correspondant à un caractère unique peut être suivie par l'un des opérateurs de répétition suivants :

?

L'élément précédent est facultatif et peut être rencontré au plus une fois.

L'élément précédent peut être rencontré zéro ou plusieurs fois.

+

L'élément précédent peut être rencontré une ou plusieurs fois.

{n}

L'élément précédent doit être cherché exactement *n* fois.

{n,}

L'élément précédent doit être cherché *n* fois ou plus.

{n,m}

L'élément précédent doit être cherché au moins *n* fois, mais au plus *m* fois.

Deux expressions rationnelles peuvent être juxtaposées ; l'expression résultante correspondra à toute chaîne formée par la juxtaposition de deux sous-chaînes correspondant respectivement aux deux expressions.

Deux expressions rationnelles peuvent être reliées par l'opérateur infixe | ; l'expression résultante correspondra à toute chaîne concordant avec l'une ou l'autre des deux expressions.

Les répétitions ont priorité sur les juxtapositions, qui à leur tour ont priorité sur les alternatives. Une sous-expression peut être entourée par des parenthèses pour modifier ces règles de priorité.

La référence inverse `\n`, où *n* est un chiffre unique, correspond à la sous-chaîne déjà mise en correspondance avec la *n*-ième sous-expression rationnelle entre parenthèses.

Dans les expressions rationnelles simples, les méta-caractères `?`, `+`, `{`, `|`, `(`, et `)` perdent leur signification spéciale, il faut utiliser à la place leurs versions avec la contre-oblique `\?`, `\+`, `\{`, `\|`, `\(`, et `\)`.

La version traditionnelle d'**egrep** ne connaît pas le méta-caractère `{`, et certaines implantations d'**egrep** utilisent `\{` à la place, si bien que des scripts shell portables devraient éviter `{` dans les motifs d'**egrep** et utiliser `[{]` pour désigner un caractère `{`.

GNU **egrep** essaie d'émuler l'usage traditionnel en supposant que `{` n'est pas spécial au cas où il rendrait invalide l'expression qu'il commence. Par exemple, la commande shell **egrep** `'{1'` recherche la chaîne composée des deux caractères `{1` au lieu de signaler une erreur de syntaxe dans l'expression rationnelle. POSIX.2 permet ce comportement comme une extension à la norme, mais les scripts portables devraient l'éviter.

VARIABLES D'ENVIRONNEMENT

Le comportement de **grep** est modifié par les variables d'environnement suivantes : **LC_ALL**, **LC_truc** et **LANG**, dans cet ordre. La variable positionnée en premier détermine le choix des paramètres régionaux. Par exemple, si **LC_ALL** n'est pas positionnée, mais **LC_MESSAGES** vaut **fr_FR**, alors le français est utilisé pour l'affichage des messages. Le défaut C est utilisé si aucune variable d'environnement n'est trouvée, ou si le catalogue des paramètres régionaux n'est pas installé, ou bien si **grep** a été compilé sans le support pour les langues nationales (NLS).

GREP_OPTIONS

Cette variable définit des options qui seront ajoutées avant les options de la ligne de commande. Par exemple, si **GREP_OPTIONS** vaut « `--binary-files=without-match --directories=skip` », **grep** se comporte comme si les deux options `--binary-files=without-match` et `--directories=skip` avaient été spécifiées avant les options explicites. Différentes options peuvent être séparées par des espaces, et une contre-oblique supprime la signification spéciale du caractère suivant, ce qui permet de spécifier une option contenant un espace ou une contre-oblique.

GREP_COLOR

Spécifie les marqueurs pour la mise en relief.

LC_ALL, LC_COLLATE, LANG

Ces variables spécifient le choix des paramètres régionaux pour **LC_COLLATE**, qui détermine l'ordre des caractères utilisé dans des intervalles tels que `[a-z]`.

LC_ALL, LC_CTYPE, LANG

Ces variables spécifient le choix des paramètres régionaux pour **LC_CTYPE**, qui détermine le codage de caractères utilisé, par exemple pour indiquer quels caractères sont considérés comme étant des espaces.

LC_ALL, LC_MESSAGES, LANG

Ces variables spécifient le choix des paramètres régionaux pour **LC_MESSAGES**, qui détermine la langue utilisée par **grep** pour ses messages. Avec le défaut C, les messages sont en américain.

POSIXLY_CORRECT

Si cette variable est positionnée, **grep** se comporte comme indiqué dans la norme POSIX.2. Sinon, **grep** se comporte plus comme les autres programmes GNU. POSIX.2 requiert que les options qui suivent des noms de fichiers soient considérées aussi comme des noms de fichiers. Par défaut, ces options sont déplacées avant la liste des opérandes et sont traitées comme des options. POSIX.2 requiert aussi que les options non reconnues soient considérées comme « illégales » ; mais comme elles n'enfreignent pas vraiment la loi, elles sont rapportées comme étant « invalides » par défaut. **POSIXLY_CORRECT** désactive aussi l'option `_N_GNU_nonoption_argv_flags_`, qui est décrite plus bas.

_N_GNU_nonoption_argv_flags_

(ici, *N* est l'identifiant numérique du processus de **grep**). Si le *i*-ième caractère de la valeur de cette variable d'environnement vaut 1, le *i*-ième opérande de **grep** n'est pas considéré comme étant une option, même s'il semble l'être. Un shell peut placer cette variable dans l'environnement de chaque commande lancée, pour spécifier quels opérandes sont le résultat du remplacement de méta-caractères et ne doivent donc pas être considérés comme des options. Ce comportement n'est présent qu'avec la bibliothèque C de GNU, et seulement si **POSIXLY_CORRECT** n'est pas positionnée.

DIAGNOSTICS

Normalement, le code de retour est 0 si des concordances ont été trouvées, et 1 si aucune concordance n'a été faite. Mais le code est 2 si une erreur est survenue, à moins que les options `-q`, `--quiet` ou `--silent` ne soient utilisées et qu'une ligne ne soit trouvée.

BOGUES

Envoyez les rapports de bogue ([ndt] en anglais !) à bug-gnu-utils@gnu.org. Assurez-vous d'inclure le mot « **grep** » quelque part dans le sujet du message.

Dans les constructions `{m,n}` de grandes valeurs de répétition peuvent pousser **grep** à utiliser beaucoup de mémoire. De plus, certaines autres expressions rationnelles tordues peuvent prendre un temps très long, et mener à un manque de mémoire.

Les références inverses sont très lentes et peuvent demander un temps très long.

TRADUCTION

Christophe Blaess, 1997.

Denis Barbier, 2003.

CETTE PAGE DOCUMENTE AUSSI :

- [egrep](#)
- [fgrep](#)
- [rgrep](#)

À PROPOS DE CETTE PAGE SUR LE BLOG :

- [Chercher remplacer dans tous les buffers de vim](#)
- [Un \(vieux\) script shell pour écouter ses messages sur le répondeur de sa Freebox](#)
- [Être à l'aise avec bash #5](#)

- [Être à l'aise avec bash #4](#)