

NOM

patch – appliquer un fichier diff à un original

SYNOPSIS

patch [*options*] [*fichier-original*] [*fichier-patch*]

mais habituellement simplement

patch **-p***nombre* <*fichier-patch*

DESCRIPTION

patch utilise un *fichier-patch* contenant un listing de différences produit par le programme **diff** et applique ces différences à un ou plusieurs fichiers originaux, en produisant des versions patchées. Normalement, les versions patchées remplacent les originaux. Des sauvegardes peuvent être créées ; voyez l'option **-b** ou **--backup**. Le nom des fichiers à patcher est habituellement composé à partir de celui du fichier patch, mais s'il n'y a qu'un seul fichier à patcher, il peut être spécifié sur la ligne de commandes comme *fichier-original*.

Au démarrage, **patch** essaie de déterminer le type du listing de diff, à moins qu'il ne soit contraint par une option **-c** (**--context**), **-e** (**--ed**), **-n** (**--normal**) ou **-u** (**--unified**). Les diffs contextuels (ancien style, nouveau style et unifié) et les diffs normaux sont appliqués par le programme **patch** lui-même, alors que les diffs **ed** sont simplement fournis à l'éditeur **ed**(1) via un tube.

patch essaie de sauter par dessus les déchets de tête, appliquer le diff, et ensuite passer les déchets de fin. Par conséquent, vous pourriez pouvoir alimenter facilement **patch** en lui injectant un article ou un message contenant un listing diff. Si le diff entier est fortement indenté, ou si un diff contextuel contient des lignes se terminant par CRLF ou est encapsulé une ou plusieurs fois en préfixant les lignes débutant par "-" par "-" comme spécifié par le RFC Internet 934, cela est pris en considération.

Avec les diffs contextuels, et dans une moindre mesure avec les diffs normaux, **patch** peut détecter si les numéros de lignes mentionnés dans le patch sont incorrects, et essayer de trouver l'endroit correct où appliquer chaque composant du patch (hunk). En première approximation, il prend le numéro de ligne mentionné par le composant, plus ou moins le décalage utilisé lors de l'application du composant précédent. Si ce n'est pas l'endroit correct, **patch** recherche en avant et en arrière un groupe de lignes correspondant au contexte donné par le composant. D'abord, **patch** recherche un endroit où toutes les lignes du contexte correspondent. S'il ne trouve pas de tel emplacement, et si c'est un diff contextuel, et que le facteur de bruit (fuzz factor) maximum vaut 1 ou plus, alors une autre analyse a lieu ; celle-ci ignore la première et la dernière ligne de contexte. Si cela échoue, et que le facteur de bruit maximum est supérieur ou égal à 2, les deux premières et les deux dernières lignes de contexte sont ignorées, et une autre analyse est faite (le facteur de bruit maximum par défaut est 2). Si **patch** ne peut trouver d'endroit où installer ce composant du patch, il place le composant dans un fichier de rejet, qui est normalement le nom du fichier de sortie plus un suffixe **.rej**, ou # si **.rej** générerait un nom de fichier trop long (si même la concaténation du seul caractère # rend le nom de fichier trop long, alors # remplace le dernier caractère du nom de fichier). (Le composant rejeté apparaît dans la forme diff contextuelle ordinaire quelle que soit la forme du patch d'entrée. Si l'entrée est un diff normal, beaucoup de contextes sont tout simplement vides.) Les numéros de ligne des composants dans le fichier de rejet peuvent être différents de ceux du fichier patch : ils reflètent l'endroit approximatif des composants ayant échoué dans le nouveau fichier plutôt que dans l'ancien.

Après chaque tentative d'application de composant, **patch** vous indique si le composant a échoué et, si c'est le cas, à quelle ligne (du nouveau fichier) le composant aurait dû être placé. Si le composant est installé à une ligne différente du numéro de ligne spécifié dans le diff, le décalage utilisé est indiqué. Un seul grand décalage *peut* indiquer qu'un composant a été installé au mauvais endroit. On vous signale également si un facteur de bruit a été utilisé pour trouver la correspondance, auquel cas vous devriez également être légèrement suspicieux. Si l'option **--verbose** est donnée, les composants correspondant exactement sont également mentionnés.

Si aucun *fichier-original* n'est spécifié sur la ligne de commandes, **patch** essaie de déterminer à partir des déchets de tête le nom du fichier à éditer, en utilisant les règles suivantes :

D'abord, **patch** construit une liste ordonnée de noms de fichiers candidats comme suit :

- Si l'en-tête est celui d'un diff contextuel, **patch** extrait le nom des anciens et des nouveaux fichiers de l'en-tête. Un nom est ignoré s'il ne contient pas suffisamment de slashes pour satisfaire à l'option **-pnombre** ou **--strip=nombre**. Le nom **/dev/null** est également ignoré.
- S'il y a une ligne **Index:** dans les déchets de tête et que les anciens et nouveaux noms sont absents, ou que **patch** se conforme à POSIX, **patch** extraira le nom à partir de la ligne **Index:**.
- Dans les règles suivantes, les noms de fichiers candidats sont considérés être placés dans l'ordre (ancien, nouveau, index), quel que soit leur ordre d'apparition dans l'en-tête.

Ensuite, **patch** sélectionne un nom de fichier à partir de la liste de candidats comme suit :

- Si certains des fichiers nommés existent, **patch** sélectionne le premier nom s'il est conforme à POSIX, ou le meilleur nom sinon.
- Si **patch** n'ignore pas RCS, ClearCase et SCCS (voyez l'option **-g nombre** ou **--get=nombre**), et si aucun des fichiers nommés n'existe mais qu'un document maître RCS, ClearCase ou SCCS est trouvé, **patch** sélectionne le premier fichier nommé ayant un document maître RCS, ClearCase ou SCCS.
- Si aucun des fichiers nommés n'existe et qu'aucun document maître RCS, ClearCase ou SCCS n'a été trouvé, si des noms sont fournis, si **patch** ne se conforme pas à POSIX, et si le patch semble créer un fichier, **patch** sélectionne le meilleur nom requérant la création du plus petit nombre de répertoires.
- Si aucun nom de fichier ne résulte des heuristiques ci-dessus, on vous demandera le nom du fichier à patcher, et **patch** sélectionnera ce nom.

Pour déterminer le *meilleur* nom d'une liste non vide de noms de fichiers, **patch** considère d'abord tous les noms ayant le composant de nom de chemin le plus court ; parmi ceux-ci, il prend ensuite ceux de nom de base le plus court ; parmi les rescapés, il prend ensuite tous les noms les plus courts ; finalement, il utilise le premier nom restant.

En outre, si les déchets de tête contiennent une ligne **Prereq:**, **patch** extrait le premier mot de la ligne de prérequis (normalement un numéro de version) et examine le fichier original pour voir si ce mot peut être trouvé. Si ce n'est pas le cas, **patch** demande une confirmation avant de procéder.

Le résultat de tout ceci est que vous devriez pouvoir faire, alors que vous vous trouvez dans une interface de news (NdT : nouvelles Usenet), quelque chose du genre :

```
| patch -d /usr/src/local/blurfl
```

et patcher un fichier du répertoire **blurfl** directement à partir de l'article contenant le patch.

Si le fichier patch contient plus d'un patch, **patch** essaie d'appliquer chacun d'entre eux comme s'ils provenaient de fichiers patches différents. Cela signifie, entre autres choses, que l'on suppose que le nom du fichier à patcher doit être déterminé pour chaque listing de diff, et que les déchets précédant chaque listing de diff contiennent des choses intéressantes telles que les noms de fichiers et le niveau de révision, comme mentionné précédemment.

OPTIONS

-b ou **--backup**

Créer des fichiers de sauvegarde, c.-à-d. que lors du patchage d'un fichier, le fichier original est renommé ou copié au lieu d'être supprimé. Lors de la sauvegarde d'un fichier qui n'existe pas, un fichier de sauvegarde vide, non lisible, est créé comme substitut pour représenter le fichier non existant. Voyez l'option **-V** ou **--version-control** pour les détails sur la façon dont on détermine le nom des fichiers de sauvegarde.

--backup-if-mismatch

Sauvegarder un fichier si le patch ne correspond pas exactement au fichier et que les sauvegardes ne sont normalement pas requises. C'est le comportement par défaut à moins que **patch** ne se conforme à POSIX.

--no-backup-if-mismatch

Ne pas sauvegarder un fichier si le patch ne correspond pas exactement au fichier, et si les sauvegardes ne sont normalement pas requises. C'est le comportement par défaut si **patch** se conforme à POSIX.

-B *préf* or **--prefix=préf**

Le préfixe *préf* d'un nom de fichier lors de la génération de son nom de fichier de sauvegarde simple. Par exemple, avec **-B /junk/**, le nom de fichier de sauvegarde simple pour **src/patch/util.c** est **/junk/src/patch/util.c**.

--binary

Lire et écrire tous les fichiers dans le mode binaire, sauf pour la sortie standard et **/dev/tty**. Cette option n'a pas d'effet sur les systèmes conformes à POSIX. Sur les systèmes comme DOS où cette option a un effet, le patch devrait être généré par un **diff -a --binary**.

-c ou **--context**

Interpréter le fichier patch en tant que diff contextuel ordinaire.

-d rép ou **--directory=rép**

Se rendre immédiatement dans le répertoire *rép*, avant de faire quoi que ce soit d'autre.

-D define ou **--ifdef=define**

Utiliser la construction **#ifdef . . . #endif** pour marquer les changements, avec *define* comme symbole de différenciation.

--dry-run

Afficher les résultats de l'application des patchs sans réellement modifier le moindre fichier.

-e ou **--ed**

Interpréter le fichier patch comme un script **ed**.

-E ou **--remove-empty-files**

Supprimer les fichiers de sortie vides après que les patchs aient été appliqués. Cette option n'est normalement pas nécessaire, car **patch** peut examiner les horodates dans l'en-tête pour déterminer si un fichier devrait exister après le patchage. Néanmoins, si l'entrée n'est pas un diff contextuel ou si **patch** se conforme à POSIX, **patch** ne supprime pas les fichiers patchés vides à moins que cette option ne soit donnée. Quand **patch** supprime un fichier, il essaie également de supprimer tout répertoire ancêtre vide.

-f ou **--force**

Supposer que l'utilisateur sait exactement ce qu'il fait, et ne poser aucune question. Passer les patchs dont les en-têtes ne disent pas quel fichier il faut patcher, patcher les fichiers même s'ils ont la mauvaise version dans la ligne **Prereq:** du patch, et supposer que les patchs ne sont pas renversés même s'il semble que cela soit le cas. Cette option ne supprime pas les commentaires ; utilisez **-s** pour cela.

-F nombre ou **--fuzz=nombre**

Fixer le facteur de bruit maximum. Cette option ne s'applique qu'aux diffs qui ont du contexte, et indique à **patch** d'ignorer jusqu'à *nombre* lignes lorsqu'il recherche des endroits où installer un composant. Notez qu'un facteur de bruit plus grand augmente la probabilité d'un patch défectueux. Le facteur de bruit par défaut est 2, et ne peut être fixé à plus que le nombre de lignes de contexte du diff contextuel (généralement 3).

-g nombre ou **--get=nombre**

Cette option contrôle les actions de **patch** quand un fichier est contrôlé par RCS ou SCCS, et n'existe pas ou est accessible en lecture seule et correspond à la version par défaut, ou quand un fichier est sous le contrôle de ClearCase et n'existe pas. Si *nombre* est positif, **patch** obtient le fichier (ou l'extrait) à partir du système de contrôle de révisions ; s'il est nul, **patch** ignore RCS, ClearCase et SCCS et n'obtient pas le fichier ; s'il est négatif, **patch** demande à l'utilisateur s'il faut obtenir le fichier. La valeur par défaut de cette option est donnée par la valeur de la variable d'environnement **PATCH_GET** si elle est définie ; si ce n'est pas le cas, la valeur par défaut est zéro si **patch** se conforme à POSIX, ou négative sinon.

--help

Afficher un résumé des options et se terminer.

-i fichier-patch ou **--input=fichier-patch**

Lire le patch à partir de *fichier-patch*. Si le *fichier-patch* est **-**, lire depuis l'entrée standard (comportement par défaut).

-l ou **--ignore-whitespace**

Reconnaître lâchement les motifs, au cas où des tabulations ou des espaces aient été altérés dans vos fichiers. Une séquence d'un ou de plusieurs blancs dans le fichier patch correspond à n'importe quelle séquence du fichier original, et les séquences de blancs à la fin des lignes sont ignorées. Les caractères normaux doivent toujours correspondre exactement. Chaque ligne de contexte doit correspondre exactement à une ligne du fichier original.

-n ou **--normal**

Interpréter le fichier patch en tant que diff normal.

-N ou **--forward**

Ignorer les patches qui semblent avoir été renversés ou déjà appliqués. Voyez également **-R**.

-o *fichier-sortie* ou **--output=fichier-sortie**

Envoyer la sortie dans *fichier-sortie* au lieu de patcher les fichiers sur place.

-pnombre ou **--strip=nombre**

Enlever le plus petit préfixe contenant *nombre* slashes de la tête de chaque nom de fichier trouvé dans le fichier patch. Une séquence d'un ou de plusieurs slashes adjacents compte pour un slash unique. Cela contrôle la façon dont les noms trouvés dans le fichier patch sont traités, au cas où vous conserveriez vos fichiers dans un répertoire différent de celui qui a envoyé le patch. Par exemple, en supposant que le nom du fichier dans le fichier patch était

/u/howard/src/blurfl/blurfl.c

spécifier **-p0** donne le nom de fichier entier non modifié, **-p1** donne

u/howard/src/blurfl/blurfl.c

sans le slash de tête, **-p4** donne

blurfl/blurfl.c

et ne pas spécifier de **-p** du tout vous donne **blurfl.c**. Ce que vous obtenez finalement est recherché soit dans le répertoire courant, soit dans le répertoire spécifié par l'option **-d**.

--posix

Se conformer plus strictement au standard POSIX, comme suit :

- Opter pour le premier fichier existant à partir de la liste (ancien, nouveau, index) quand on devine les noms de fichiers à partir des en-têtes diff.
- Ne pas supprimer les fichiers vides après le patchage.
- Ne pas demander s'il faut obtenir les fichiers depuis RCS, ClearCase ou SCCS.
- Requérir que toutes les options précèdent les fichiers sur la ligne de commandes.
- Ne pas sauvegarder les fichiers quand il y a une discordance.

--quoting-style=mot

Utiliser le style *mot* de protection des noms en sortie. Le *mot* devrait être un des suivants :

literal Sortir les noms tels quels.

shell Protéger les noms pour le shell s'ils contiennent des métacaractères du shell ou susciteraient une sortie ambiguë.

shell-always

Protéger les noms pour le shell, même s'ils ne devraient normalement pas requérir de protection.

c Protéger les noms comme pour une chaîne de caractères dans le langage C.

escape Protéger comme avec **c**, mais omettre les guillemets entourants.

Vous pouvez spécifier la valeur par défaut de l'option **--quoting-style** avec la variable d'environnement **QUOTING_STYLE**. Si cette variable d'environnement n'est pas définie, la valeur par défaut est **shell**.

-r *fichier-rejet* ou **--reject-file=fichier-rejet**

Placer les rejets dans *fichier-rejet* au lieu du fichier **.rej** par défaut.

-R ou **--reverse**

Supposer que ce patch a été créé alors que les anciens et nouveaux fichiers avaient été intervertis. (Oui, j'ai peur que cela se produise occasionnellement, la nature humaine étant ce qu'elle est.) **patch** essaie d'intervertir chaque composant avant de l'appliquer. Les rejets apparaissent dans le format « échangé ». L'option **-R** ne fonctionne pas avec les scripts **ed** car il y a trop peu d'informations pour reconstruire l'opération inverse.

Si le premier composant d'un patch échoue, **patch** le renverse pour voir s'il peut être appliqué de cette façon. Si c'est le cas, il vous demande si vous voulez que l'option **-R** soit utilisée. Si vous ne le souhaitez pas, le patch continue à être appliqué normalement. (Note : cette méthode ne peut détecter un patch reversé si c'est un diff normal et que la première commande est une concaténation (elle aurait dû être un effacement) puisque la concaténation réussit toujours, étant donné qu'un contexte nul convient partout. Heureusement, la plupart des patches ajoutent ou modifient des lignes plutôt que d'en supprimer, de sorte que la plupart des diffs normaux renversés commencent par un effacement, qui échoue, ce qui déclenche l'heuristique.)

-s ou **--silent** ou **--quiet**

Travailler silencieusement, à moins qu'une erreur ne se produise.

-t ou **--batch**

Supprimer les questions comme **-f**, mais employer des suppositions différentes : omettre les patches dont l'en-tête ne contient pas de noms de fichiers (comme pour **-f**) ; omettre les patches pour lesquels le fichier a la mauvaise version pour la ligne **Prereq** ; et supposer que les patches sont renversés s'il semble que cela soit le cas.

-T ou **--set-time**

Fixer les dates de dernière modification et de dernier accès des fichiers patchés à partir des horodates contenues dans les en-têtes diff contextuels, en supposant que les en-têtes diff contextuels utilisent l'heure locale. Cette option n'est pas recommandée, car les patches qui utilisent l'heure locale ne peuvent pas facilement être utilisés par des personnes situées dans d'autres fuseaux horaires, et parce que les horodates utilisant l'heure locale sont ambiguës quand les horloges locales sont retardées lors du passage de l'heure d'été à heure d'hiver. Au lieu d'utiliser cette option, générez des patches utilisant le temps UTC et utilisez l'option **-Z** ou **--set-utc** à la place.

-u ou **--unified**

Considérer que le fichier patch est un diff contextuel unifié.

-v ou **--version**

Afficher l'en-tête de révision et le niveau de patches de **patch**, et se terminer.

-V *méthode* ou **--version-control=méthode**

Utiliser la *méthode* pour déterminer le nom des fichiers de sauvegarde. La méthode peut également être fournie via la variable d'environnement **PATCH_VERSION_CONTROL** (ou, si elle n'est pas définie, via **VERSION_CONTROL**), qui est surchargée par cette option. Cette méthode n'indique pas si des fichiers de sauvegarde doivent être générés ; elle n'affecte que les noms des fichiers de sauvegarde qui sont créés.

La valeur de *méthode* est identique à celle de la variable « version-control » de GNU Emacs : **patch** reconnaît également des synonymes plus descriptifs. Les valeurs valides pour *méthode* sont (les abréviations uniques sont acceptées) :

existing ou **nil**

Créer des sauvegardes numérotées des fichiers qui en ont déjà, sinon de simples sauvegardes. C'est le comportement par défaut.

numbered ou **t**

Créer des sauvegardes numérotées. Le nom du fichier de sauvegarde numéroté pour *F* est *F.N* où *N* est le numéro de version.

simple ou **never**

Créer des sauvegardes simples. Les options **-B** ou **--prefix**, **-Y** ou **--basename-prefix**, et **-z** ou **--suffix** spécifient le nom du fichier de sauvegarde simple. Si aucune de ces options n'est spécifiée, alors un suffixe de sauvegarde simple est utilisé ; il représente la valeur de la variable d'environnement **SIMPLE_BACKUP_SUFFIX** si elle est définie, ou est **.orig** sinon.

Avec les sauvegardes numérotées ou simples, si le nom du fichier de sauvegarde est trop long, le suffixe de sauvegarde **~** est utilisé à la place ; si même la concaténation d'un **~** rendrait le nom trop long, alors **~** remplace le dernier caractère du nom de fichier.

--verbose

Produire des informations supplémentaires sur le travail en cours.

-x nombre ou **--debug=nombre**

Spécifier des drapeaux de débogage qui intéressent uniquement les patcheurs de **patch**.

-Y préf ou **--basename-prefix=préf**

Préfixer *préf* au nom de base d'un nom de fichier lors de la génération de son nom de fichier de sauvegarde simple. Par exemple, avec **-Y .del/**, le nom du fichier de sauvegarde simple pour **src/patch/util.c** est **src/patch/.del/util.c**.

-z suffixe ou **--suffix=suffixe**

Utiliser le *suffixe* comme suffixe de sauvegarde simple. Par exemple, avec **-z -**, le nom du fichier de sauvegarde simple pour **src/patch/util.c** est **src/patch/util.c-**. Le suffixe de sauvegarde simple peut également être spécifié via la variable d'environnement **SIMPLE_BACKUP_SUFFIX**, qui est surchargée par cette option.

-Z ou **--set-utc**

Fixer les dates de dernier accès et de dernière modification des fichiers patchés à partir des horodates données dans les en-têtes diff contextuels, en supposant que les en-têtes diff contextuels utilisent le Coordinated Universal Time (temps UTC, mieux connu sous le nom de GMT). Voyez également l'option **-T** ou **--set-time**.

Les options **-Z** ou **--set-utc**, et **-T** ou **--set-time** empêchent normalement de fixer une date d'un fichier si la date originale du fichier ne correspond pas à celle donnée dans l'en-tête du patch, ou si son contenu ne correspond pas exactement au patch. Néanmoins, si l'option **-f** ou **--force** est spécifiée, la date du fichier est tout de même fixée.

À cause des limitations du format de sortie de **diff**, ces options ne peuvent mettre à jour les dates des fichiers dont le contenu n'a pas changé. De plus, si vous utilisez ces options, vous devriez supprimer (p.ex. avec **make clean**) tous les fichiers qui dépendent des fichiers patchés, afin que des invocations ultérieures de **make** ne soient pas dupées par les dates des fichiers patchés.

ENVIRONNEMENT**PATCH_GET**

Spécifie si **patch** doit obtenir les fichiers manquants ou en lecture seule depuis RCS, ClearCase ou SCCS par défaut ; voyez l'option **-g** ou **--get**.

POSIXLY_CORRECT

Si elle est définie, **patch** se conforme plus strictement au standard POSIX par défaut : voyez l'option **--posix**.

QUOTING_STYLE

Valeur par défaut pour l'option **--quoting-style**.

SIMPLE_BACKUP_SUFFIX

Extension à utiliser pour le nom des fichiers de sauvegarde simple au lieu de **.orig**.

TMPDIR, TMP, TEMP

Répertoire ou placer les fichiers temporaires ; **patch** utilise la première variable d'environnement de cette liste qui est définie. Si aucune n'est définie, le répertoire par défaut dépend du système ; il est normalement **/tmp** pour les hôtes Unix.

VERSION_CONTROL ou **PATCH_VERSION_CONTROL**

Sélectionne le style de contrôle de version ; voyez l'option **-v** ou **--version-control**.

FICHIERS

\$TMPDIR/p*

fichiers temporaires

/dev/tty

terminal de contrôle ; utilisé pour obtenir des réponses aux questions posées à l'utilisateur

VOIR AUSSI

diff(1), **ed(1)**

Marshall T. Rose and Einar A. Stefferud, Proposed Standard for Message Encapsulation, RFC Internet 934 <URL:ftp://ftp.isi.edu/in-notes/rfc934.txt> (1985-01).

NOTES DESTINÉES AUX ÉMETTEURS DE PATCHS

Il y a plusieurs choses que vous devriez garder à l'esprit si vous avez l'intention d'émettre des patches.

Créez votre patch de façon systématique. Une bonne méthode est la commande **diff -Naur ancien/nouveau** où *ancien* et *nouveau* identifient l'ancien et le nouveau répertoire respectivement. Les noms *ancien* et *nouveau* ne devraient pas contenir de slashes. Les en-têtes de la commande **diff** devraient posséder des dates et des heures dans le Temps Universel en utilisant le format Unix traditionnel, afin que les destinataires du patch puissent utiliser l'option **-Z** ou **--set-utc**. Voici une commande d'exemple, qui utilise la syntaxe du shell Bourne :

```
LC_ALL=C TZ=UTC0 diff -Naur gcc-2.7 gcc-2.8
```

Indiquez à vos destinataires la façon d'appliquer le patch en leur disant dans quel répertoire ils doivent se rendre, et quelles options de **patch** utiliser. La chaîne d'options **-Np1** est recommandée. Testez votre procédure en faisant semblant d'être un destinataire et en appliquant votre patch sur une copie des fichiers originaux.

Vous pouvez épargner un tas de cheveux blancs aux utilisateurs de votre patch en conservant un fichier **patchlevel.h** qui est patché pour incrémenter le niveau de patches en tant que premier diff du fichier patch que vous envoyez. Si vous placez une ligne **Prereq:** dans le patch, il ne vous laissera pas les appliquer dans le désordre sans avertissement.

Vous pouvez créer un fichier en envoyant un diff qui compare **/dev/null** ou un fichier vide daté de l'Epoch (01/01/1970 00:00:00 UTC) avec le fichier que vous voulez créer. Cela ne fonctionne que si le fichier que vous voulez créer n'existe pas déjà dans le répertoire cible. Inversement, vous pouvez supprimer un fichier en envoyant un diff contextuel qui compare le fichier à effacer avec un fichier vide daté de l'Epoch. Le fichier sera supprimé à moins que **patch** ne se conforme à POSIX et que l'option **-E** ou **--remove-empty-files** n'est pas donnée. Une manière commode de générer des patches qui créent ou suppriment des fichiers est d'utiliser l'option **-N** ou **--new-file** du **diff** GNU.

Si le destinataire est censé utiliser l'option **-p N**, n'envoyez pas de sortie ressemblant à ceci :

```
diff -Naur v2.0.29/prog/README prog/README
--- v2.0.29/prog/README  Mon Mar 10 15:13:12 1997
+++ prog/README  Mon Mar 17 14:58:22 1997
```

car les deux noms de fichiers ont un nombre différent de slashes, et différentes versions de **patch** interprètent les noms de fichiers différemment. Pour éviter une confusion, envoyez plutôt une sortie qui ressemble à ceci :

```
diff -Naur v2.0.29/prog/README v2.0.30/prog/README
--- v2.0.29/prog/README  Mon Mar 10 15:13:12 1997
+++ v2.0.30/prog/README  Mon Mar 17 14:58:22 1997
```

Évitez d'envoyer des patches qui comparent des noms de fichiers de sauvegarde comme **README.orig**, car cela pourrait embrouiller **patch** qui patcherait alors un fichier de sauvegarde au lieu du fichier réel. Au lieu de cela, envoyez des patches qui comparent les mêmes noms de base de fichiers dans des répertoires différents, comme p.ex. **old/README** et **new/README**.

Prenez bien soin de ne pas envoyer de patches renversés, car les gens pourraient se demander s'ils n'ont pas déjà appliqué le patch.

Évitez si possible que votre patch modifie des fichiers dérivés (p.ex. le fichier **configure** quand il y a une ligne **configure: configure.in** dans votre makefile), car le destinataire devrait de toute façon avoir la possibilité de régénérer les fichiers dérivés. Si vous devez envoyer des diffs de fichiers dérivés, générez-les en utilisant le temps UTC, faites en sorte que les destinataires appliquent le patch avec l'option **-Z** ou **--set-utc**, et faites-leur supprimer les fichiers non patchés qui dépendent des fichiers patchés (p.ex. avec **make clean**).

Bien que vous puissiez vous en sortir en plaçant 582 listings de diff dans un fichier, il peut être plus judicieux de grouper les patches apparentés dans des fichiers séparés au cas où quelque chose tourne mal.

DIAGNOSTICS

Les diagnostics indiquent généralement que **patch** n'a pas pu analyser votre fichier patch.

Si l'option **--verbose** est donnée, le message **Hmm...** indique qu'il reste du texte non traité dans le fichier patch et que **patch** essaie de deviner s'il contient un patch et, le cas échéant, de quel type il s'agit.

Le code de retour de **patch** est 0 si tous les composants ont été appliqués avec succès, 1 si certains d'entre eux n'ont pas pu être appliqués, et 2 en cas de problème plus sérieux. Lors de l'application d'un groupe de patches dans une boucle, il vous incombe de vérifier ce code de retour afin de ne pas appliquer ultérieurement un patch sur un fichier partiellement patché.

AVERTISSEMENTS

Les diffs contextuels ne peuvent pas représenter fiablement la création ou l'effacement de fichiers vides, de répertoires vides, ou de fichiers spéciaux comme les liens symboliques. Ils ne peuvent pas non plus représenter les métadonnées de fichiers comme la propriété, les permissions, ou si un fichier est un lien matériel vers un autre. Si des changements tels que ceux-ci sont également requis, des instructions séparées (p.ex. dans un script shell) permettant de les mettre en oeuvre devraient accompagner le patch.

patch ne peut dire si les numéros de ligne sont désactivés dans un script **ed**, et ne peut détecter les mauvais numéros de lignes dans un diff normal que lorsqu'il détecte un changement ou un effacement. Un diff contextuel utilisant un facteur de bruit de 3 peut avoir le même problème. En attendant qu'une interface interactive appropriée soit ajoutée, vous devriez probablement utiliser un diff contextuel dans ces cas pour voir si les changements ont du sens. Bien sûr, une compilation sans erreur est un bon indicateur que le patch a fonctionné, mais pas toujours.

patch produit habituellement les résultats escomptés, même quand il doit deviner beaucoup de choses. Néanmoins, des résultats corrects ne sont garantis que lorsque le patch est appliqué sur exactement la même version du fichier que celle à partir de laquelle le patch a été généré.

COMPATIBILITÉ

Le standard POSIX spécifie un comportement qui diffère du comportement traditionnel de **patch**. Vous devriez être conscient de ces différences si vous devez interopérer avec **patch** version 2.1 et précédentes, qui ne se conforment pas à POSIX.

- Dans le **patch** traditionnel, l'opérande de l'option **-p** était optionnel, et un **-p** nu était équivalent à **-p0**. L'option **-p** requiert maintenant un opérande, et **-p 0** est actuellement équivalent à **-p0**. Pour une compatibilité maximale, utilisez des options comme **-p0** et **-p1**.

De plus, le **patch** traditionnel comptait simplement les slashes lors de l'enlèvement des préfixes de chemin ; **patch** compte maintenant les composants du nom de chemin, c.-à-d. qu'une séquence d'un ou de plusieurs slashes adjacents compte désormais pour un seul slash. Pour une portabilité maximale, évitez d'envoyer des patches contenant // dans les noms de fichiers.

- Dans le **patch** traditionnel, les sauvegardes étaient activées par défaut. Ce comportement est maintenant activé par l'option **-b** ou **--backup**.

Inversement, dans le **patch** POSIX, les sauvegardes n'étaient jamais créées, même en cas de discorde. Dans le **patch** GNU, ce comportement est activé par l'option **--no-backup-if-mismatch**, ou en se conformant à POSIX avec l'option **--posix**, ou encore en définissant la variable d'environnement

POSIXLY_CORRECT.

L'option **-b** *suffixe* du **patch** traditionnel est équivalente aux options **-b -z** *suffixe* du **patch** GNU.

- Le **patch** traditionnel utilisait une méthode compliquée (et documentée de façon incomplète) pour deviner le nom du fichier à patcher à partir de l'en-tête du patch. Cette méthode ne se conformait pas à POSIX, et comportait quelques pièges. À l'heure actuelle, **patch** utilise une méthode différente, aussi compliquée (mais mieux documentée) qui se conforme optionnellement à POSIX ; nous espérons qu'elle comporte moins de pièges. Les deux méthodes sont compatibles si les noms de fichiers de l'en-tête du diff contextuel et la ligne **Index:** sont totalement identiques après la phase de suppression de préfixe. Votre patch est normalement compatible si les noms de fichiers de tous les en-têtes contiennent tous le même nombre de slashes.
- Quand le **patch** traditionnel posait une question à l'utilisateur, il envoyait la question sur la sortie d'erreur standard, et recherchait une réponse dans le premier fichier de la liste suivante qui était un terminal : l'erreur standard, la sortie standard, **/dev/tty**, et l'entrée standard. À l'heure actuelle, **patch** envoie ses questions sur la sortie standard et obtient les réponses à partir de **/dev/tty**. Certaines réponses par défaut ont été modifiées afin que **patch** n'entre pas dans une boucle infinie quand il utilise les réponses par défaut.
- Le **patch** traditionnel se terminait avec un code de retour qui comptabilisait le nombre de mauvais composants, ou avec le code de retour 1 en cas de problème sérieux. À l'heure actuelle, **patch** se termine avec un code de retour de 1 si certains composants ont échoué, ou 2 en cas de problème sérieux.
- Limitez-vous aux options suivantes quand vous envoyez des instructions destinées à être exécutées par quelqu'un exécutant le **patch** GNU, le **patch** traditionnel, ou un **patch** qui se conforme à POSIX. Les espaces sont significatives dans la liste suivante, et les opérandes sont requis.

```
-c
-d rép
-D define
-e
-l
-n
-N
-o fichier-sortie
-pnombre
-R
-r fichier-rejet
```

BOGUES

Rapportez s.v.p. les bogues via courrier électronique à <bug-gnu-utils@gnu.org>.

patch pourrait être plus intelligent en ce qui concerne les correspondances partielles, les décalages excessifs et l'inversion de code, mais cela requerrait une passe supplémentaire.

Si le code a été dupliqué (par exemple avec **#ifdef ANCIENCODE ... #else ... #endif**), **patch** est incapable de patcher les deux versions et, si jamais il fonctionne, patchera probablement le mauvais, et vous dira qu'il a réussi à démarrer.

Si vous appliquez un patch que vous avez déjà appliqué, **patch** pense que c'est un patch renversé et offre la possibilité de dés-appliquer le patch. Cela pourrait être interprété comme une fonctionnalité.

COPYRIGHT

Copyright © 1984, 1985, 1986, 1988 Larry Wall.

Copyright © 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998 Free Software Foundation, Inc.

L'autorisation est donnée de créer et de distribuer des copies textuelles de ce manuel, à condition que la notice de copyright et la notice de permission soient préservées dans toutes les copies.

L'autorisation est donnée de copier et distribuer des versions modifiées de ce manuel sous les conditions de copie textuelle, à condition que l'entièreté du travail dérivé résultant soit distribuée sous les termes d'une

autorisation identique à celle-ci.

L'autorisation est donnée de copier et distribuer des traductions de ce manuel dans n'importe quelle autre langue, sous les conditions ci-dessus pour les versions modifiées, sauf que cette notice de permission peut être incluse dans des traductions approuvées par les détenteurs du copyright au lieu de l'anglais originel.

AUTEURS

Larry Wall a écrit la version originale de **patch**. Paul Eggert a supprimé les limites arbitraires de **patch**, a ajouté le support des fichiers binaires, la spécification des dates du fichier, et l'effacement de fichiers, et l'a fait mieux se conformer à POSIX. Les autres contributeurs incluent Wayne Davison, qui a ajouté le support du diff unifié, et David MacKenzie, qui a ajouté la prise en charge de la configuration et de la sauvegarde.

TRADUCTION

Frédéric Delanoy <*delanoy_f at yahoo.com*>, 2002.