

[compléments_patch version1 ->il y a peut-être des erreurs. Merci de me les signaler. Eric Bachard]

1) Buts de ce programme

patch est un utilitaire GNU très utilisé par ceux qui codent. Ce programme permet d'appliquer des patchs. Le mot patch, à l'origine, désignait un morceau de programme binaire que l'on pouvait appliquer « juste à la place », c'est à dire faisant exactement la même longueur que a partie à changer afin de l'améliorer.

Un patch désigne aussi le fichier qui contient, de manière structurée (on parle alors de patch unifié) ou non, la différence entre une version et une autre d'un code source.

Lorsqu'un patch est appliqué, le programme original est remplacé par la nouvelle version et une archive de l'ancienne est faite (par défaut sous le nom du fichier + l'extension « .orig »)

2) Patch sur le web

Ces quelques liens intéressants relatifs à patch, résument bien l'utilisation de patch :

[1] : <<http://www.linux-france.org/article/lgazette/issue-32/lg-32-5.html>>

[2] : <<http://www.fr.linuxfromscratch.org/view/lfs-5.0-fr/appendixa/patch.html>>

3) Utilisation

Syntaxe : patch [options] [original [fichier_patch]]

Options importantes :

-p ou --strip[=*nombre_entier*] : Spécifie le nombre de répertoires à retrancher du nom du chemin qui précède. Si *nombre_entier* vaut 0, l'ensemble du chemin est retranché, et seul le nom du fichier est préservé ; une valeur 1 retranche le premier / ; chaque nombre supérieur retranche un répertoire supplémentaire à partir de la gauche.

-R ou --reverse : Crée un patch inversé, c'est à dire tente de restituer par étapes l'état antérieur en patchant avec des versions inversées de l'ancien et du nouveau fichier.

-r *fichier* ou --reject-file=*fichier* : Envoie les rejets (segments de code du patch que **patch** ne parvient pas à insérer dans le fichier original) vers *fichier*, nommé par défaut *original.rej*.

-u ou --unified : Interprète le fichier patch comme un fichier diff au contexte unifié (deux lignes de contexte par défaut).

-F *entier* ou --fuzz=*entier* : Spécifie le nombre de lignes qui peuvent être ignorées (*fuz zed over*) quand **patch** doit installer un bloc de code. La valeur par défaut est de deux lignes. Cette option n'a de sens que dans le cas de **diff** avec contexte (deux lignes de contexte avant et après par défaut).

-S ou --skip : Passe au prochain patch du fichier patch à appliquer : en général, un fichier patch contient toutes les modifications à appliquer en même temps, pour gagner du temps.

-s ou --silent ou encore --quiet : Désactive l'affichage des commentaires (quand on est sûr que le patch fonctionne, peut-être ?)

Voir la page de man de patch, pour les autres options, moins utilisées.

N.B. : En fait, on applique surtout un patch avec une redirection, ou un pipe quand le patch est (b) zippé (voir [1])

4) Exemples

4.1) patch -p0 < fichier_à_appliquer

Pour cela, il faut que le patch à appliquer soit dans le répertoire contenant le fichier à patcher

4.2) Application d'un patch du noyau Linux

a) récupérer les sources (un miroir de kernel.org est conseillé)

b) vérifier la signature des sources

c) installer les sources dans son répertoire /home

Exemple : ~/sources_linux

récupérer un patch permettant de passer d'une version à l'autre. Exemples Les sources du noyau 2.6 ont été chargées, et l'utilisateur souhaite utiliser le 2.7

--> il suffit de charger la patch permettant de passer du 2.6 au 2.7

Plus exactement ici (par exemple) :

<<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.6.tar.bz2>> pour les sources

<<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.6.tar.bz2.sign>> pour la signature

<<http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.7.bz2>> qui fait 2,7 Mo (au lieu de 33Mo pour les sources complètes du noyau 2.6.7 !!)

<<http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.7.bz2.sign>> pour la signature du patch

Ensuite, après avoir vérifié les signatures, il suffit d'appliquer le patch en plaçant celui-ci dans le répertoire ~/sources_linux/linux-2.6.6, qui a été créé après décompression de l'archive dans le répertoire ~/sources_linux (à l'aide de tar jvxf linux-2.6.6.tar.bz2)

Puis :

```
cd ~/sources_linux/linux-2.6.6
```

```
bunzip2 patch-2.6.7.bz2
```

```
patch -p1 < patch-2.6.7 # -p1 semble être nécessaire maintenant (ce qui évite d'avoir à donner un nom exact au répertoire qui contient les sources, ce qui est une bonne idée...
```

C'est tout ! Ensuite, make menuconfig...etc.

N.B. : les problèmes de liens symboliques pointant vers /usr/include/linux...etc ne sont pas abordés ici, mais sont à régler pour que tout fonctionne. Ce qui a été présenté n'est qu'un exemple pour illustrer.