

LKN Chapitre 7. Mettre à jour un noyau

Table of Contents

- [LKN Chapitre 7. Mettre à jour un noyau](#)
 - [Télécharger la nouvelle source](#)
 - [Téléchargement depuis kernel.org](#)
 - [Quel patch s'applique à quelle version?](#)
 - [Trouver le patch](#)
 - [Application du patch](#)
 - [Reconfigurer le noyau](#)

Source: [Linux Kernel in a Nutshell: 7 Upgrading a kernel](#)

Inévitablement, cela se produit, on a un noyau personnalisé, fonctionnant à merveille, sauf pour une petite chose qu'il faut corriger dans la dernière version des développeurs du noyau. Ou un problème de sécurité est détecté et une nouvelle version stable du noyau est rendue publique. Quoi qu'il en soit, on est confronté au problème de la mise à niveau du noyau et on ne veut pas perdre tout le temps et les efforts nécessaires pour créer cette configuration de noyau parfaite.

Ce chapitre va montrer à quel point il est facile de mettre à jour un noyau à partir d'une ancienne version, tout en conservant toutes les options de configuration de la précédente.

Tout d'abord, sauvegarder le fichier `.config` dans le répertoire source du noyau. On a consacré du temps et des efforts à le créer, et il doit être enregistré en cas de problème lors de la mise à niveau.

```
cd ~/linux/linux-2.6.17.11
cp .config ../good_config
```

Il n'y a que cinq étapes simples qui sont nécessaires pour mettre à niveau un noyau à partir d'un noyau précédemment construit:

- Télécharger le nouveau code source.
- Appliquer les modifications à l'ancienne arborescence source pour l'amener au niveau supérieur.
- Reconfigurez le noyau en fonction de la configuration précédente du noyau.
- Construire le nouveau noyau.
- Installer le nouveau noyau.

Les deux dernières étapes fonctionnent de la même manière que précédemment, on ne décrira donc que des trois premières étapes de ce chapitre.

Dans ce chapitre, on va supposer qu'on a construit une version de noyau 2.6.17.9 réussie et que vous souhaitez effectuer une mise à niveau vers la version 2.6.17.11.

Télécharger la nouvelle source

Les développeurs du noyau Linux se rendent compte que tous les utilisateurs ne souhaitent pas télécharger l'intégralité du code source dans le noyau pour chaque mise à jour. Ce serait une perte de bande passante et de temps. Pour cette raison, ils proposent un **patch**



Les fichiers de correctifs contiennent une liste des lignes à supprimer et des lignes à ajouter, avec un contexte dans le fichier indiquant où les modifications doivent être apportées. qui peut mettre à niveau une ancienne version du noyau vers une nouvelle. Ils contiennent une représentation des modifications nécessaires pour reconstruire les nouveaux fichiers, en fonction des anciens. Le programme **patch** prend le fichier et l'applique à l'arborescence d'origine, créant le nouvel arbre.

Téléchargement depuis kernel.org

Le site principal de kernel.org contient une liste des versions actuelles du noyau disponibles en téléchargement.

Pour télécharger l'intégralité du code source du noyau, on utilise le lien indiqué par **F**. Cependant, si on clique sur le nom de la version du noyau, on téléchargera un fichier correctif à la place:

C'est ce que l'on veut faire lors de la mise à niveau. Mais il faut déterminer le correctif à télécharger.

Quel patch s'applique à quelle version?

Un fichier de correctif du noyau uniquement mettra à niveau le code source d'une version spécifique vers une autre version spécifique. Voici comment appliquer les différents fichiers de patch:

- Les correctifs stables du noyau s'appliquent à la version de base du noyau. Cela signifie que le correctif 2.6.17.**10** ne s'appliquera qu'à la version 2.6.17 du noyau. Le correctif du noyau 2.6.17.**10** ne s'appliquera pas au noyau 2.6.17.**9** ou à toute autre version.
- Les correctifs de base du noyau (Base kernel release) s'appliquent uniquement à la version précédente du noyau de base. Cela signifie que le correctif 2.6.**18** ne s'appliquera qu'à la version 2.6.**17** du noyau. Il ne s'appliquera pas à la dernière version du noyau 2.6.**17.y**, ni à aucune autre version.
- Mise à niveau des correctifs incrémentiels d'une version spécifique vers la prochaine version. Cela permet aux développeurs de ne pas avoir à rétrograder leur noyau puis à le mettre à niveau, juste pour passer de la dernière version stable à la prochaine version stable (les correctifs de version stable sont uniquement pour le noyau de base, pas la version stable précédente.) possible, il est recommandé d'utiliser les correctifs incrémentiels pour se faciliter la vie.

Trouver le patch

Comme on veut passer de la version du noyau 2.6.17.9 à la version 2.6.17.11, il faut télécharger deux correctifs différents. On a besoin d'un correctif de la version 2.6.17.9 à la version 2.6.17.10, puis de la version 2.6.17.10 à la version 2.6.17.11.



Lorsqu'on doit mettre à niveau plus de deux versions, il est recommandé de sauvegarder les étapes, de reculer, puis de mettre à niveau vers l'avant. Dans ce cas, on pourrait revenir en arrière de 2.6.17.9 à 2.6.17 puis avancer de 2.6.17 à 2.6.17.11.

Les correctifs du noyau stable et du noyau sont situés dans la même structure de répertoires que les arborescences sources principales. Tous les correctifs incrémentiels se trouvent un niveau plus bas, dans le sous-répertoire `incr`. Donc, pour trouver le patch qui va de 2.6.17.9 à 2.6.17.10, regarder dans le répertoire `/pub/linux/kernel/v2.6/incr` pour trouver les fichiers dont on a besoin

```
cd ~/linux

lftp ftp.kernel.org/pub/linux/kernel/v2.6/incr

cd ok, cwd=/pub/linux/kernel/v2.6/incr
lftp ftp.kernel.org:/pub/linux/kernel/v2.6/incr>
ls *2.6.17.9*.bz2

-rw-rw-r-- 1 536      536      2872 Aug 22 19:23
patch-2.6.17.9-10.bz2
lftp ftp.kernel.org:/pub/linux/kernel/v2.6/incr>
get patch-2.6.17.9-10.bz2

2872 bytes transferred
lftp ftp.kernel.org:/pub/linux/kernel/v2.6/incr>
get patch-2.6.17.10-11.bz2

7901 bytes transferred
lftp ftp.kernel.org:/pub/linux/kernel/v2.6/incr>
exit
ls -F

good_config linux-2.6.17.9/ patch-2.6.17.10-11.bz2  patch-2.6.17.9-10.bz2
```

Application du patch

Comme les patches que l'on a téléchargés sont compressés, la première chose à faire est de les décompresser avec la commande **bzip2**:

```
bzip2 -dv patch-2.6.17.9-10.bz2

patch-2.6.17.9-10.bz2: done

bzip2 -dv patch-2.6.17.10-11.bz2

patch-2.6.17.10-11.bz2: done

ls -F

good_config linux-2.6.17.9/  patch-2.6.17.10-11  patch-2.6.17.9-10
```

Maintenant, on doit appliquer les fichiers de correctifs au répertoire du noyau. Aller dans le répertoire:

```
cd linux-2.6.17.9
```

et exécuter le programme de correctif pour appliquer le premier correctif en déplaçant l'arborescence source de la version 2.6.17.9 vers la version 2.6.17.10:

```
patch -p1 < ../patch-2.6.17.9-10

patching file Makefile
patching file block/elevator.c
patching file fs/udf/super.c
patching file fs/udf/truncate.c
patching file include/net/sctp/sctp.h
patching file include/net/sctp/sm.h
patching file net/sctp/sm_make_chunk.c
patching file net/sctp/sm_statefuns.c
patching file net/sctp/socket.c
```

Vérifier que le correctif a vraiment fonctionné correctement et qu'il n'y a pas d'erreur ou d'avertissement dans la sortie du programme de correctif. C'est aussi une bonne idée de regarder le **Makefile** du noyau pour voir la version du noyau:

```
head -n 5 Makefile

VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 17
EXTRAVERSION = .10
NAME=Crazed Snow-Weasel
```

Maintenant que le noyau est au niveau de la version 2.6.17.**10**, faire la même chose qu'avant et appliquer le correctif pour l'amener au niveau 2.6.17.**11**:

```
patch -p1 < ../patch-2.6.17.10-11

patching file Makefile
patching file arch/ia64/kernel/sys_ia64.c
patching file arch/sparc/kernel/sys_sparc.c
patching file arch/sparc64/kernel/sys_sparc.c
patching file drivers/char/tpm/tpm_tis.c
patching file drivers/ieee1394/ohci1394.c
patching file drivers/md/dm-mpath.c
patching file drivers/md/raid1.c
patching file drivers/net/sky2.c
patching file drivers/pci/quirks.c
patching file drivers/serial/Kconfig
patching file fs/befs/linuxvfs.c
patching file fs/ext3/super.c
patching file include/asm-generic/mman.h
patching file include/asm-ia64/mman.h
patching file include/asm-sparc/mman.h
patching file include/asm-sparc64/mman.h
patching file kernel/timer.c
patching file lib/spinlock_debug.c
patching file mm/mmap.c
patching file mm/swapfile.c
patching file net/bridge/netfilter/ebt_ulog.c
patching file net/core/dst.c
patching file net/core/rtnetlink.c
patching file net/ipv4/fib_semantics.c
patching file net/ipv4/netfilter/arp_tables.c
patching file net/ipv4/netfilter/ip_tables.c
patching file net/ipv4/netfilter/ipt_ULOG.c
patching file net/ipv4/route.c
patching file net/ipv4/af_ipx.c
patching file net/netfilter/nfnetlink_log.c
```

Vérifier à nouveau que la sortie du programme de correctif n'a montré aucune erreur et regardez le Makefile:

```
head -n 5 Makefile

VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 17
EXTRAVERSION = .11
NAME=Crazed Snow-Weasel
```

Maintenant que le code source est correctement mis à jour vers la version que l'on souhaite utiliser, c'est une bonne idée de revenir en arrière et de changer le nom du répertoire pour faire référence au numéro de version du noyau afin qu'aucune confusion ne se produise ultérieurement:

```
cd ..
mv linux-2.6.17.9 linux-2.6.17.11
ls -F

good_config linux-2.6.17.11/ patch-2.6.17.10-11 patch-2.6.17.9-10
```

Reconfigurer le noyau

Auparavant, on a utilisé la méthode **make menuconfig** ou **gconfig** ou **xconfig** pour changer différentes options de configuration. Mais une fois qu'on a une configuration fonctionnelle, la seule chose qui est nécessaire est de la mettre à jour avec les nouvelles options qui ont été ajoutées au noyau depuis la dernière version. Pour ce faire, les options **make oldconfig** et **make silentoldconfig** doivent être utilisées.

- **make oldconfig** prend la configuration actuelle du noyau dans le fichier `.config` et la met à jour en fonction de la nouvelle version du noyau. Pour ce faire, il imprime toutes les questions de configuration et y répond si l'option est déjà gérée dans le fichier de configuration. S'il existe une nouvelle option, le programme s'arrête et demande à l'utilisateur à quoi la nouvelle valeur de configuration doit être définie. Après avoir répondu à l'invite, le programme continue jusqu'à ce que la configuration complète du noyau soit terminée.
- **make silentoldconfig** fonctionne exactement de la même manière que **oldconfig**, mais il n'imprime rien à l'écran, sauf s'il doit poser une question sur une nouvelle option de configuration.

Habituellement, quand on upgrade entre différentes versions des versions stables, aucune nouvelle option de configuration n'est ajoutée, car il s'agit d'une série de noyaux stables. Si cela se produit, il n'y a pas de nouvelles questions auxquelles il faut répondre pour la configuration du noyau, donc le programme continue avec succès sans aucune intervention de l'utilisateur. Un exemple de cela passe de la version 2.6.17.9 à la version 2.6.17.11:

```
cd linux-2.6.17.11
make silentoldconfig

scripts/kconfig/conf -s arch/i386/Kconfig
#
# using defaults found in .config
#
```

Voici un exemple où une nouvelle option de noyau apparaît dans une nouvelle version. L'option du noyau pour activer le débogage **Mutex** est nouvelle lors du basculement entre certaines versions du noyau. Voici la sortie lorsque cela s'est produit:

```
make silentoldconfig
scripts/kconfig/conf -s arch/i386/Kconfig
#
# using defaults found in .config
#
*
```

```
* Restart config...
*
*
* Kernel hacking
*
Show timing information on printks (PRINTK_TIME) [Y/n/?] y
Magic SysRq key (MAGIC_SYSRQ) [Y/n/?] y
Kernel debugging (DEBUG_KERNEL) [Y/n/?] y
  Kernel log buffer size (16 => 64KB, 17 => 128KB) (LOG_BUF_SHIFT) [16] 16
  Detect Soft Lockups (DETECT_SOFTLOCKUP) [Y/n/?] y
  Collect scheduler statistics (SCHEDSTATS) [N/y/?] n
  Debug slab memory allocations (DEBUG_SLAB) [Y/n/?] y
    Memory leak debugging (DEBUG_SLAB_LEAK) [Y/n] y
  Mutex debugging, deadlock detection (DEBUG_MUTEXES) [N/y/?] (NEW)
y
```

Le programme de configuration s'est arrêté à cette option et a demandé à l'utilisateur de choisir une option. Appuyer ensuite sur **y** et le programme continue:

```
Spinlock debugging (DEBUG_SPINLOCK) [Y/n/?] y
Sleep-inside-spinlock checking (DEBUG_SPINLOCK_SLEEP) [Y/n/?] y
kobject debugging (DEBUG_KOBJECT) [N/y/?] n
Highmem debugging (DEBUG_HIGHMEM) [N/y/?] n
Compile the kernel with debug info (DEBUG_INFO) [N/y/?] n
Debug Filesystem (DEBUG_FS) [Y/?] y
Debug VM (DEBUG_VM) [N/y/?] n
Compile the kernel with frame pointers (FRAME_POINTER) [N/y/?] n
Compile the kernel with frame unwind information (UNWIND_INFO) [N/y/?] n
Force gcc to inline functions marked 'inline' (FORCED_INLINING) [N/y/?] n
torture tests for RCU (RCU_TORTURE_TEST) [N/m/y/?] n
Check for stack overflows (DEBUG_STACKOVERFLOW) [N/y/?] n
Stack utilization instrumentation (DEBUG_STACK_USAGE) [N/y/?] n
Stack backtraces per line (STACK_BACKTRACE_COLS) [2] 2
*
* Page alloc debug is incompatible with Software Suspend on i386
*
Write protect kernel read-only data structures (DEBUG_RODATA) [N/y/?] n
Use 4Kb for kernel stacks instead of 8Kb (4KSTACKS) [N/y/?] n
```

La mise à niveau de la configuration du noyau pour une nouvelle version est donc aussi simple que d'utiliser une autre option de configuration. Avec cette méthode, vous n'avez pas besoin d'utiliser les programmes de configuration graphiques ou textuels pour toute nouvelle mise à jour du noyau.



le programme **ketchup** a été créé pour gérer tout cela automatiquement. Voir la section intitulée «ketchup» pour plus de détails sur le fonctionnement de ce programme et son utilisation.

From:

<http://vps101364.serveur-vps.net/> - **dwndoc**

Permanent link:

<http://vps101364.serveur-vps.net/doku.php?id=lfs:lkn-c07-upgrading-a-kernel>

Last update: **2025/02/19 10:59**

