

lkn Chapitre 11. Référence de ligne de commande de génération du noyau

Table of Contents

- [lkn Chapitre 11. Référence de ligne de commande de génération du noyau](#)
 - [Options informationnelles](#)
 - [Options de nettoyage](#)
 - [Options de configuration](#)
 - [Options de build](#)
 - [Options de package](#)
 - [Options de documentation](#)
 - [Options spécifiques à l'architecture](#)
 - [Options d'analyse](#)

Source: [Linux Kernel in a Nutshell: 11. Kernel build command line reference](#)

L'outil qui construit les générations de noyau est le programme **make**, auquel on passe une option qui spécifie ce que l'on veut construire. La Configuration de la construction passe en revue les Options de base nécessaires pour construire correctement le noyau, mais le système de construction du noyau a également un large éventail d'autres Options. Ce chapitre détaille ces Options et à quoi elles peuvent servir.

Toutes ces options sont transmises au programme **make** sur la ligne de commande, et un certain nombre d'entre elles peuvent être regroupées si on le souhaite. Par exemple:

```
make mrproper xconfig
```

Les options sont réparties en différents types dans les sections suivantes.

On peut obtenir un résumé de la plupart de ces options en exécutant, dans le répertoire de génération:

```
make help
```

Cela imprime une grande partie des options courantes décrites dans le reste de ce chapitre.

Options informationnelles

Ces options impriment la version du noyau, basée sur un certain nombre d'options différentes. Ils sont couramment utilisés par les scripts pour déterminer la version du noyau en cours de construction.

Option	Description
kernelrelease	Affiche la version actuelle du noyau, telle que déterminée par le système de génération.
kernelversion	Affiche la version actuelle du noyau, comme indiqué par le Makefile principal. Cela diffère de la Option kernelrelease en ce qu'il n'utilise aucune information de version supplémentaire basée sur les options de configuration ou les fichiers de version locale.

Options de nettoyage

Ces options suppriment simplement les fichiers des versions précédentes. Leur utilisation est fortement recommandée pour s'assurer de ne pas contaminer les nouvelles versions avec des fichiers restants qui peuvent avoir été créés avec différentes options. Ils diffèrent dans la quantité qu'ils enlèvent; parfois, on peut souhaiter conserver les fichiers que l'on a modifiés.

Option	Description
clean	Supprime la plupart des fichiers générés par le système de construction du noyau, mais conserve la configuration du noyau.
mrproper	Supprime tous les fichiers générés par le système de construction du noyau, y compris la configuration et certains fichiers de sauvegarde divers.
distclean	Fait tout ce que mrproper fait et supprime certains fichiers de sauvegarde de l'éditeur et les correctifs restants.

Options de configuration

Ces options permettent au noyau d'être configuré de différentes manières.

Option	Description
config	Met à jour la configuration actuelle du noyau en utilisant un programme orienté ligne.
menuconfig	Met à jour la configuration actuelle du noyau en utilisant un programme de menu basé sur du texte.
xconfig	Met à jour la configuration actuelle du noyau en utilisant un programme graphique basé sur QT.
gconfig	Met à jour la configuration actuelle du noyau en utilisant un programme graphique basé sur GTK+.
oldconfig	Met à jour la configuration actuelle du noyau en utilisant le fichier <code>.config</code> actuel et en demandant toutes les nouvelles options qui ont été ajoutées au noyau.
silentoldconfig	Tout comme oldconfig , mais n'imprime rien à l'écran sauf lorsqu'il faut répondre à une question.
randconfig	Génère une nouvelle configuration du noyau avec des réponses aléatoires à toutes les différentes options.
defconfig	Génère une nouvelle configuration du noyau avec la réponse par défaut utilisée pour toutes les options. Les valeurs par défaut sont extraites d'un fichier situé dans le fichier <code>arch/\$ARCH/defconfig¹⁾</code> , où \$ARCH fait référence à l'architecture spécifique pour laquelle le noyau est construit.
allmodconfig	Génère une nouvelle configuration du noyau dans laquelle les modules sont activés chaque fois que possible.
allyesconfig	Génère une nouvelle configuration du noyau avec toutes les options définies sur yes.

Option	Description
allnoconfig	Génère une nouvelle configuration du noyau avec toutes les options définies sur no.



Les options **allyesconfig**, **allmodconfig**, **allnoconfig** et **randconfig** tirent également parti de la variable d'environnement **KCONFIG_ALLCONFIG**. Si cette variable pointe vers un fichier, ce fichier sera utilisé comme une liste de valeurs de configuration que l'on veut définir sur une valeur spécifique. En d'autres termes, le fichier remplace le comportement normal des Options de création.

Par exemple, si le fichier `~/linux/must_be_set` contient les variables suivantes:

```
cat ~/linux/must_be_set
```

```
CONFIG_SWAP=y  
CONFIG_DEBUG_FS=y
```

et qu'on fait `make allnoconfig` avec la variable d'environnement **KCONFIG_ALLCONFIG** appropriée:

```
KCONFIG_ALLCONFIG=../must_be_set make allnoconfig  
grep CONFIG_SWAP .config
```

```
CONFIG_SWAP=y
```

alors les résultats incluent:

```
grep CONFIG_DEBUG_FS .config
```

```
CONFIG_DEBUG_FS=y
```

Sinon, cette variable n'aurait normalement pas été définie sur **y**.

Si la variable **KCONFIG_ALLCONFIG** n'est pas définie, le système de génération vérifie les fichiers dans le répertoire de génération de niveau supérieur nommé:

- `allmod.config`
- `allno.config`
- `allrandom.config`
- `allyes.config`

Si l'un de ces fichiers est présent, la génération les utilise comme des listes de valeurs de configuration qui doivent être forcées aux valeurs spécifiées. Si aucun de ces fichiers n'est trouvé, le système de génération recherche finalement un fichier appelé `all.config` pour une liste de valeurs de configuration forcée.

On peut utiliser ces différents fichiers pour mettre en place une bonne configuration de base connue qui fonctionnera toujours. Ensuite, les autres options de configuration peuvent être utilisées pour

générer différentes configurations de test pour la situation requise.

Options de build

Ces options construisent le noyau lui-même de différentes manières.

Option	Description
all	Construit toutes les différentes cibles nécessaires pour que ce noyau puisse être utilisé. Cela inclut à la fois les modules et la partie statique du noyau.
vmlinux	Construit juste la partie statique du noyau, pas tous les modules chargeables.
modules	Construit tous les modules du noyau chargeables pour cette configuration.
modules_install	Installe tous les modules à l'emplacement spécifié. Si aucun emplacement n'est spécifié avec la variable d'environnement INSTALL_MODULE_PATH , ils sont installés dans le répertoire racine par défaut de la machine.
dir/	Génère tous les fichiers dans le répertoire spécifié et dans tous les sous-répertoires en dessous.
dir/file.[o/i/s]	Génère uniquement le fichier spécifié.
dir/file.ko	Génère tous les fichiers nécessaires et les relie ensemble pour former le module spécifié.
tags	Construit toutes les balises nécessaires que les éditeurs de texte les plus courants peuvent utiliser lors de la modification du code source.
TAGS	Construit toutes les balises nécessaires que les éditeurs de texte les plus courants peuvent utiliser lors de la modification du code source.
cscope	Construit une image cscope , utile dans les recherches d'arborescence source, de l'arborescence source pour l'architecture spécifiée par le fichier de configuration (pas tous les fichiers source du noyau).

On peut également transmettre un certain nombre de variables d'environnement pour que cela change la construction. Ceux-ci peuvent être spécifiés pour presque toutes les cibles.

Variables d'environnement	Valeur	Description
V	0	Cela indique au système de génération de s'exécuter de manière silencieuse, en affichant uniquement le fichier en cours de génération, et non la commande entière en cours d'exécution afin de générer ce fichier. Il s'agit de l'option par défaut pour le système de génération.
V	1	Cela indique au système de génération de fonctionner de manière détaillée, en affichant la commande complète utilisée pour générer chacun des fichiers spécifiques.
O	dir	Cela indique au système de génération de localiser tous les fichiers de sortie dans le répertoire <code>dir</code> , y compris les fichiers de configuration du noyau. Cela permet au noyau d'être construit à partir d'un système de fichiers en lecture seule et de placer la sortie dans un autre emplacement.

Variables d'environnement	Valeur	Description
C	1	Cela vérifie tous les fichiers C qui sont sur le point d'être créés avec l'outil sparse , qui détecte les erreurs de programmation courantes dans les fichiers source du noyau. sparse peut être téléchargé à l'aide de git depuis git://git.kernel.org/pub/scm/devel/sparse/sparse.git . Des instantanés peuvent être trouvés sur https://www.codemonkey.org.uk/projects/git-snapshots/sparse/ . On peut trouver plus d'informations sur l'utilisation de sparse dans le fichier <code>Documentation/sparse.txt</code> de l'arborescence des sources du noyau.
C	2	Cela force tous les fichiers C à être vérifiés avec l'outil sparse , même s'ils n'ont pas besoin d'être créés.

Options de packaging

Ces options regroupent un noyau intégré dans un package autonome qui peut être installé sur une large gamme de machines différentes.

Option	Description
rpm	Construit d'abord le noyau, puis le conditionne en tant que package RPM qui peut être installé.
rpm-pkg	Construit un package RPM source contenant le noyau de base.
binrpm-pkg	Construit un package RPM qui contient un noyau et des modules compilés.
deb-pkg	Construit un paquet Debian qui contient le noyau et les modules compilés.
tar-pkg	Construit une archive tar (tarball) qui contient le noyau et les modules compilés.
targz-pkg	Construit une archive tar compressée par gzip qui contient le noyau et les modules compilés.
tarbz2-pkg	Construit une archive tar compressée bzip2 qui contient le noyau et les modules compilés.

Options de documentation

Ces options construisent la documentation interne du noyau dans une variété de formats différents.

Option	Description
xmldocs	Génère la documentation du noyau sous forme de fichiers XML DocBook .
psdocs	Génère la documentation du noyau sous forme de fichiers PostScript .
pdfdocs	Construit la documentation du noyau sous forme de fichiers PDF .
htmldocs	Génère la documentation du noyau sous forme de fichiers HTML .
mandocs	Construit la documentation du noyau comme un ensemble de pages de manuel, qui peuvent ensuite être installées avec l'option installmandocs .

Options spécifiques à l'architecture

Chaque architecture de noyau a un ensemble de cibles spécifiques qui lui sont propres. L'architecture Intel 32 bits a les cibles suivantes disponibles.

Option	Description
bzImage	Crée une image de noyau compressée et la place dans le fichier <code>arch/i386/boot/bzImage</code> . Il s'agit de la cible par défaut pour la construction du noyau i386.
install	Installe l'image du noyau à l'aide du programme <code>/sbin/installkernel</code> spécifique à la distribution. Cela n'installe pas les modules du noyau; cela doit être fait avec l'option modules_install .
bzdisk	Crée une image de disquette de démarrage et l'écrit sur le périphérique <code>/dev/fd0</code> .
fdimage	Crée une image de disquette de démarrage et la place dans le fichier <code>arch/i386/boot/fdimage</code> . Le package mttools doit être présent sur le système pour que cela fonctionne correctement.
isoimage	Crée une image de démarrage du CD-ROM et la place dans le fichier <code>arch/i386/boot/image.iso</code> . Le paquet syslinux doit être présent sur le système pour que cela fonctionne correctement.

Options d'analyse

Ces options sont utiles pour essayer de trouver du code problématique dans le noyau. C'est une bonne idée de créer une liste d'espace de pile lors de la création de nouveau code pour déterminer que les modifications n'occupent pas trop d'espace de pile du noyau. L'option **namespacecheck** est utile pour déterminer si les modifications peuvent ajouter en toute sécurité ses symboles à l'espace de noms global du noyau.

Option	Description
checkstack	Génère une liste des fonctions qui utilisent le plus d'espace de la pile du noyau.
namespacecheck	Génère une liste de tous les symboles du noyau et de leurs espaces de noms. Ce sera une grande liste.

1)

Le système de construction du noyau parcourt tous les fichiers Kconfig (de tous les sous-répertoires), vérifiant toutes les options dans les fichiers Kconfig, si l'option est mentionnée dans le fichier `defconfig`, le système de construction place cette option dans `.config` avec la valeur choisie dans `defconfig` et donc écrase la valeur choisie

From:
<http://www.ouarte.garden/> - **dwndoc**

Permanent link:
<http://www.ouarte.garden/doku.php?id=lfs:lkn-c11-kernel-build-command-line-reference>

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

