
dracut.cmdline

Options de ligne de commande kernel dracut

Le périphérique root utilisé par le kernel est spécifié dans le fichier de configuration de boot dans la ligne de commande kernel, comme toujours. La spécification traditionnelle `root=/dev/sda1` est toujours permise, mais non encouragée. Le périphérique root devrait être identifié par un label ou un uuid.

Les paramètres "rd.*" mentionnés sans '=' sont des paramètres booléens. Si l'assignement avec '=' est manquant, '=1' est assumé.

Standard

init=<path to real init> Spécifie le chemin du programme init à démarrer un fois l'initramfs terminé

root=<path to blockdevice> Spécifie le périphérique à utilisé pour '/'

rootfstype=<filesystem type> auto, si non spécifié

rootflags=<mount options> Spécifie des options de montage additionnels pour '/'. Non spécifié, lit /etc/fstab du vrai '/'.

ro force '/' et '/usr' en lecture seule

rw force '/' et '/usr' en lecture écriture

rootfallback=<path to blockdevice> Spécifie le périphérique block à utiliser comme périphérique root, si le root normal ne peut être trouvé

rd.auto rd.auto=1 Active l'auto-assemblage des périphérique spéciaux comme cryptoLUKS, dmraid, mdraid ou lvm.

rd.hostonly=0 Supprime tous ce qui est compilé dans la configuration de l'hôte. Cela permet de booter sur un changement de disque.

rd.cmdline=ask Demande à l'utilisateur des paramètres additionnels

rd.fstab=0 N'honore pas les options de montage spéciaux pour '/' dans /etc/fstab du vrai '/'

resume=<path to resume partition> Résume depuis une partition swap

rd.skipsck N'effectue pas de fsck pour '/' et '/usr'.

iso-scan/filename

Utiliser iso-scan/filename avec une iso Fedora/Red Hat/CentOS fonctionne en copiant les paramètres cmdline kernel

Divers

rd.driver.blacklist=<drivername> [,<drivername>,...] Ne pas charger le module kernel. Peut être spécifié plusieurs fois

rd.driver.pre=<drivername> [,<drivername>,...] Force le chargement du module spécifié. Peut être spécifié plusieurs fois

rd.driver.post=<drivername> [,<drivername>,...] Force le chargement du module une fois tous les modules automatique ont été chargé. Peut être spécifié plusieurs fois

rd.retry=<seconds> Délai pour retenter le initqueue pour configurer les périphériques. Défaut : 30s. après 2/3 du temps, les raids dégradés sont démarrés de force.

rd.timeout=<seconds> Délai d'attente pour que les périphériques apparaissent. Défaut : 0 - indéfinis. noter que ce délai ne devrait pas dépasser rd.retry

rd.noverifyssl Accèpte les certificats auto-signés pour les téléchargements ssl
rd.ctty=<terminal device> Spécifie le terminal de contrôle pour la console. Utile avec plusieurs console=*

debug

rd.info Affiche des informations supplémentaires même si 'quiet' est spécifié
rd.shell Permet d'être basculé dans un shell si le montage root échoue
ld.debug définis -x pour le shell dracut. Si systemd est actif dans l'initramfs, toute la sortie est loggée dans le journal système.
rd.memdebug= [0-3] Affiche l'utilisation mémoire, de 0 à 3
rd.break Bascule dans un shell à la fin
rd.break={cmdline|pre-udev|pre-trigger|initqueue|pre-mount|mount|pre-pivot|cleanup} Bascule dans un shell au breakpoint définis
rd.udev.info Définis le niveau de log udev à info
rd.udev.debug Définis le niveau de log udev à debug

I18N

rd.vconsole.keymap=<keymap base file name> Table de traduction clavier chargé par loadkeys ; pris depuis le répertoires de keymaps. Écris dans KEYMAP dans /etc/vconsole.conf dans l'initramfs
rd.vconsole.keymap.ext=<list of keymap base file names> Liste de keymaps supplémentaires, écris d-ans EXT_KEYMAP dans /etc/vconsole.conf
rd.vconsole.unicode (bool) Indique le mode utf-8. Sera écrit dans UNICODE dans /etc/vconsole.conf
rd.vconsole.font= Police de caractère de la console. Écris dans FONT dans /etc/vconsole.conf
rd.vconsole.font.map=<console map base file name> Voir setfont -m.
rd.vconsole.font.unimap=<unicode table base file name> Voir setfont -u
rd.locale.LANG=<locale> Pris depuis l'environnement.
rd.locale.LC_ALL=<locale> Pris depuis l'environnement

LVM

rd.lvm=0 Désactive la détection LVM
rd.lvm.vg=<volume group name> N'active que ce groupe de volume. Peut être spécifié plusieurs fois
rd.lvm.lv=<logical volume name> N'active que les volumes logique spécifiés. Peut être spécifié plusieurs fois
ld.lvm.conf=0 Supprime tout /etc/lvm/lvm.conf, qui pourrait exister dans l'initramfs

LUKS

ld.luks=0 Désactive la détection LUKS
rd.luks.uuid=<luks uuid> N'active que les partitions LUKS avec l'uuid donné.
rd.luks.allow-discards=<luks uuid> Permet l'utilisations des requêtes discards (TRIM) pour les partitions LUKS avec l'uuid donné.
rd.luks.allow-discards Permet l'utilisations des requêtes discards (TRIM) pour toutes les partitions LUKS
rd.luks.crypttab=0 Ne vérifie pas si la partition LUKS est dans /etc/crypttab

LUKS - key on removable device support

rd.luks.key=<keypath> :<keydev> :<luksdev> keypath est le chemin d'une clé à rechercher. si keypath fini par .gpg, c'est une clé symétrique GPG. keydev est un périphérique sur lequel la clé réside. Peut être un périphérique, UUID ou label. luksdev indique le périphérique luks auquel s'applique la clé.

md raid

rd.md=0 Désactive la détection raid

rd.md.ims=0 Désactive les raids imsm/ism

rd.md.ddf=0 Désactive les raids ddf

rd.md.conf=0 Ignore mdadm.conf inclus dans l'initramfs

rd.md.waitclean=1 Attends la fin des resynchro, récupération, ou reshape avant de continuer

rd.md.uuid=<md raid uuid> N'active que le raid avec l'uuid spécifié. peut être spécifié plusieurs fois

dm raid

rd.dm=0 Désactive la détection dm raid

rd.dm.uuid=<dm raid uuid> N'active que le raid avec l'uuid spécifié. peut être spécifié plusieurs fois

multipath

rd.multipath=0 Désactive la detection multipath

FIPS

rd.fips Active FIPS

rd.fips.skipkernel Ne vérifie pas le checksum de l'image kernel

Réseau

ip={dhcplonlany|dhcp6|auto6} spécifie la méthode d'obtention d'une adresse ip

ip=<interface> :{dhcplonlany|dhcp6|auto6} [:[<mtu>] [:<macaddr>]] idem, spécifique à une interface. peut être spécifié plusieurs fois

ip=<client-IP> : [<peer>] :<gateway-IP> :<netmask> :<client_hostname> :<interface> :{noneloff|dhcplonlany|dhcp6|auto6|libft} [: [<peer>]] Configuration réseau explicite. Peut être spécifié plusieurs fois

ip=<client-IP> : [<peer>] :<gateway-IP> :<netmask> :<client_hostname> :<interface> :{noneloff|dhcplonlany|dhcp6|auto6|libft} [: [<peer>]] configuration réseau explicite.

ifname=<interface> :<MAC> Assigne le nom spécifié au périphérique réseau.

rd.route=<net>/<netmask> :<gateway> [:<interface>] Ajouter une route statique.

bootdev=<interface> Spécifie l'interface réseau à utiliser pour les informations netroot et de routage. Requis si plusieurs lignes ip= sont utilisées

BOOTIF=<MAC> Spécifie l'interface réseau à utiliser pour les informations netroot et de routage.

rd.bootif=0 Désactive le parsing BOOTIF, qui est fournis par PXE

nameserver=<IP> [nameserver=<IP> ...] Spécifie les serveurs de noms à utiliser

rd.peerdns=0 Désactive le paramètre DNS des paramètres DHCP

biosdevname=0 désactive le renommage d'interface réseau biosdevname

rd.reednet=1 active l'événement réseau sans netroot définis

vlan=<vlanname> :<phydevice> Définis un périphérique vlan

bond=<bondname> [:<bondslaves> : [:<options>]] Définis une périphérique bond

team=<teammaster> :<teamslaves> Définis un périphérique team

bridge=<bridgename> :<ethnames> Définis un périphérique bridge

NFS

root= [<server-ip> :] <root-dir> [: <nfs-options>] Monte un partage NFS. si aucun serveur n'est donné, utilise dhcp

root={dhcp|dhcp6} permet de rechercher le chemin de root via dhcp

rd.nfs.domain=<NFSv4 domain name> Spécifie le domaine NFSv4. Écrase les paramètres dans /etc/idmap.conf

rd.net.dhcp.retry=<cnt> Si définis, dracut tente de se connecter via dhcp, cnt fois avant d'échouer. Défaut : 1

rd.net.timeout.dhcp=<arg> Si définis, dhclient est appelé sans "-timeout <arg>"

rd.net.timeout.iflink=<seconds> Délai d'attente d'activation des liens. Défaut : 60

rd.net.timeout.ifup=<seconds> Délai d'attente de l'état "up" des liens. Défaut : 20

rd.net.timeout.route=<seconds> Délai d'attente d'activation des routes. Défaut : 20

rd.net.timeout.ipv6dad=<seconds> Délai d'attente pour IPv6 DAD. Défaut : 50

rd.net.timeout.ipv6auto=<seconds> Délai d'attente pour l'adressage ipv6 automatique. Défaut : 40

rd.net.timeout.carrier=<seconds> Délai d'attente de reconnaissance du carrier. Défaut : 5

CIFS

root=cifs :// [<username> [: <password>] @] <server-ip> : <root-dir> Définis un partage cifs

cifsuser=<username> username cifs

cifspass=<password> password cifs

iSCSI

root=iscsi : [<username> : <password> [: <reverse> : <password>] @] [<servername>] : [<protocol>] : [<port>] [: [<iscsi_iface_name>]]
Définis un accès iSCSI pour root

root= ???

netroot=iscsi : [<username> : <password> [: <reverse> : <password>] @] [<servername>] : [<protocol>] : [<port>] [: [<iscsi_iface_name>]]
Plusieurs options netroot sont permis pour définir plusieurs disques iscsi

root= ??? rd.iscsi.initiator=<initiator> rd.iscsi.target.name=<target name> rd.iscsi.target.ip=<target ip>

rd.iscsi.target.port=<target port> rd.iscsi.target.group=<target group> rd.iscsi.username=<username>

rd.iscsi.password=<password> rd.iscsi.in.username=<in username> rd.iscsi.in.password=<in password> Spécifier manuellement tous les paramètres iscsistart.

root= ??? netroot=iscsi rd.iscsi.firmware=1 Li le paramètre iscsi depuis le firmware BIOS

rd.iscsi.param=<param> Le paramètre est passé à iscsistart. peut être spécifié plusieurs fois
rd.iscsi.ibft rd.iscsi.ibft=1 Active l'autoconfiguration iBFT pour les interfaces
rd.iscsi.waitnet=0 Désactive l'attente que toutes les interfaces soient actives avant de tenter une connexion aux targets iSCSI
rd.iscsi.testroute=0 Désactive la vérification, si la route vers de target iSCSI est possible avant de tenter une connexion

FCoE

fcoe=<eddlinterface|MAC> :{dcb|nodcb} Définis une connexion FCoE via l'interface spécifiée.

NBD

root=??? netroot=nbd <server> :<port/exportname> [:<fstype> [:<mountopts> [:<nbdopts>]]] Mounte un partage nbd.
root=dhcp dhcp root-path=nbd <server> :<port/exportname> [:<fstype> [:<mountopts> [:<nbdopts>]]] récupère les informations nbd via dhcp.

DASD

rd.dasd=... Même syntaxe que le paramètre du module kernel.

ZFCP

rd.zfcp=<zfcpc adaptor device bus ID>,<WWPN>,<FCPLUN> Peut être spécifié plusieurs fois
rd.zfcp.conf=0 ignore zfcp.conf inclus dans l'initramfs

ZNET

rd.znet=<nettype>,<subchannels>,<options> Peut être spécifié plusieurs fois

Booter des images live

dracut offre plusieurs options pour les images live.

SquashFS Le système boot avec un fs readonly depuis le squashfs et applique un dm rw par dessus le fs readonly.

non compressé Quand le système live est installé avec `–skipcompress`, le système root `ext3fs.img`, est étendu à l'installation et aucun squashfs n'est utilisé durant le boot.

image writable Le système récupère une image compressée, l'extrait dans `/run/initramfs/fsimg/rootfs.img`, la connecte dans un périphérique loop, créé un dm writable à `/dev/mapper/live-rw` et le monte comme volume rw à `/`.

rd.writable.fsimg=1 Active le support de système de fichier rw.

root=live :<url> Boot une image live récupérée depuis l'url spécifié. (http, https, ftp, torrent, tftp)

rd.live.debug=1 Active la sortie debug du processus de boot

rd.live.dir=<path> Spécifie le répertoire dans le squashfs où ext3fs.img ou rootfs.img peuvent être trouvés. Défaut : LiveOS

rd.live.squashimg=<filename of SquashFS image> Spécifie le fichier d'une image squash du système de fichier root. Défaut : squashfs.img

rd.live.ram=1 Copie l'image complète en RAM et l'utilise pour booter. Utile quand l'image réside sur un DVD par ex. qui doit être éjecté ensuite.

rd.live.overlay=<devspec> :(<pathspec>|auto)|none Permet l'utilisation d'overlay permanent.

rd.live.overlay.size=<size MiB> Spécifie une taille d'overlay persistant. Défaut : 512

rd.live.overlay.readonly=1 Spécifie un overlay rw non persistant stacké sur un snapshot ro.

rd.live.overlay.reset=1 Spécifie qu'un overlay persistant devrait être réinitialisés au boot.

rd.live.overlay.thin=1 Active l'utilisation de snapshots thin.

Plymouth

plymouth.enable=0 Désactive plymouth complètement

rd.plymouth=0 Désactive plymouth seulement dans l'initramfs

Clés kernel

masterkey=<kernel master key path name> Définis la clé maître

masterkeytype=<kernel master key type> Définis le type de clé maître

evmkey=<EVM key path name> Définis le chemin de la clé EVM

ecryptfskey=<eCryptfs key path name> Définis le chemin de la clé eCryptfs