

BROUILLON

Résumé de la page de man de start-stop-daemon

Syntaxe

- `start-stop-daemon [<option> ...] <command>`

Commandes

Forme brève	Forme longue	
-S	<code>-start - <arguments> ...</code>	lance un programme et lui passe les arguments <arguments>
-K	<code>-stop</code>	arrête un programme
-T	<code>-status</code>	obtient l'état du programme
-H	<code>-help</code>	affiche de l'aide
-V	<code>-version</code>	affiche la version

Matching options (il en faut au moins une)

-p	<code>-pidfile <pid-file></code>	fichier pid à vérifier
-x	<code>-exec <executable></code>	programme à démarrer ou dont on veut vérifier s'il tourne
-n	<code>-name <process-name></code>	nom du processus à vérifier
-u	<code>--user <username uid></code>	propriétaire de processus à vérifier

Options

-g	<code>-group <group gid></code>	lancer ce processus sous ce groupe
-c	<code>--chuid <name uid[:group gid]></code>	se placer sous cet utilisateur ou ce groupe avant de lancer le processus
-s	<code>-signal <signal></code>	signal à envoyer (TERM par défaut)
-a	<code>-startas <pathname></code>	programme à lancer (par défaut : <executable>)
-r	<code>-chroot <directory></code>	s'enfermer (chroot) dans <répertoire> avant de démarrer
-d	<code>-chdir <directory></code>	aller à <directory> (/ par défaut)
-N	<code>-nicelevel <incr></code>	ajouter incr au niveau du processus
-P	<code>-procsched <policy[:prio]></code>	utiliser <policy> avec <prio> pour le gestionnaire de processus du noyau (prio = 0 par défaut)
-I	<code>-iosched <class[:prio]></code>	utiliser <class> avec <prio> pour pour l'ordonnanceur d'entrées/sorties (prio = 4 par défaut)
-k	<code>-umask <mask></code>	changer umask en <masque> avant de commencer
-b	<code>-background</code>	forcer le processus à se détacher

-C	-no-close	ne fermer aucun descripteur de fichier
-m	-make-pidfile	créer le fichier de pid avant de commencer
-R	-retry <schedule>	vérifier si le processus a été tué et et réessayer
-t	-test	mode de test, ne fait rien
-o	-oknodo	retourne 0 (au lieu de 1) si rien n'est fait fait
-q	-quiet	plus calme
-v	-verbose	plus bavard

Retry <schedule> is <item>|/<item>/... where <item> is one of

- `-<signal-num>|[-]<signal-name>`

: send that signal

- `<timeout>`

: wait that many seconds

- forever : repeat remainder forever

or <schedule> may be just <timeout>, meaning <signal>/<timeout>/KILL/<timeout>

The process scheduler <policy> can be one of:

other, fifo or rr

The IO scheduler <class> can be one of:

real-time, best-effort or idle

Exit status

0	= done
1	= nothing done (⇒ 0 if -oknodo)
2	= with -retry, processes would not die
3	= trouble

Exit status with --status

0	= program is running
1	= program is not running and the pid file exists
3	= program is not running
4	= unable to determine status

Nom

- **start-stop-daemon** - lance ou arrête des démons système

Synopsis

- `start-stop-daemon [option...] commande`

Description

- **start-stop-daemon** permet de gérer le lancement ou l'arrêt d'un processus système.
- Avec les bonnes options, **start-stop-daemon** peut trouver les instances d'un processus en fonctionnement.



- **start-stop-daemon** se comporte comme **killall** sauf utilisation de `-pidfile`.
- Il recherche dans le tableau des processus le nom ou le gid indiqué.
- Toute correspondance empêche `-start` de démarrer le démon.
- Les processus trouvés recevront le signal TERM (ou le signal indiqué par `-signal` or `-retry`).
- Si un démon doit survivre à un `-stop`, il faut indiquer un fichier pid (« pidfile »).

Commandes

courte	longue	explications
-S [--] paramètres	<code>--start</code> [--] paramètres	* S'il existe une instance du processus, start-stop-daemon se termine et retourne le code 1 * Sinon, une nouvelle instance est lancée. \\Tout argument donné après -- sur la ligne de commande est passé tel quel au programme lancé.
-K	<code>--stop</code>	* S'il existe une instance du processus, start-stop-daemon lui envoie le signal précisé par -signal et retourne le code 0. * Sinon, il se termine et retourne le code 1. * Si <code>-retry</code> est indiquée, start-stop-daemon recherche si le processus ou les processus se sont bien terminés.

courte	longue	explications
-T	<code>--status</code>	* S'il existe une instance du processus, start-stop-daemon retourne un code défini par les actions des scripts d'initialisation de la LSB (« LSB Init Script Actions »)
-H	<code>--help</code>	Affiche un message d'aide et quitte.
-V	<code>--version</code>	Affiche la version et le copyright du programme et quitte.

Options de correspondance

courte	longue	explications
-p	<code>--pidfile pid-file</code>	Contrôle si un processus a créé le fichier <i>fichier-pid</i> . Si l'ancien processus s'est terminé sans avoir effacé le fichier-pid, cette option peut donner des résultats imprévisibles
-x	<code>--exec exécutable</code>	Contrôle l'existence d'instances de cet exécutable. Le paramètre <i>exécutable</i> doit être un chemin absolu. Cela peut ne pas fonctionner avec des scripts interprétés (l'exécutable est alors l'interpréteur).
-n	<code>--name nomdeprocessus</code>	Contrôle l'existence de processus nommés <i>nom-de-processus</i> . Le nom-de-processus est en général le nom du fichier du processus. sur la plupart des systèmes, ce nom a une longueur limitée (15 caractères souvent, pour des raisons de portabilité).
-u	<code>--user nomutilisateur uid</code>	Contrôle l'existence de processus dont le propriétaire est <i>identifiant</i> ou <i>uid</i> . Si cette option est utilisée seule, tous les processus de cet utilisateur seront concernés par l'action.

Options

courte	longue	explications
-g	<code>--group group gid</code>	Modifie le group ou le gid au début du processus.
-s	<code>--signal signal</code>	L'action <code>--stop</code> définit le signal d'arrêt à envoyer au processus (par défaut : TERM).

courte	longue	explications	
-R	<code>--retry durée action-prévue</code>	<p>Avec l'action <code>-stop</code>, <code>start-stop-daemon</code> doit vérifier que les processus se sont terminés. Il le fait pour tous les processus correspondants qui tournent, jusqu'à ce qu'il n'y en ait plus.</p> <p>action-prévue Action à exécuter si le processus ne se termine pas.</p> <p>durée : l'action-prévue <code>signal/durée/KILL/durée</code> est utilisée, où <code>signal</code> est le signal indiqué par <code>-signal</code>.</p> <p>action-prévue est une liste d'au moins deux items séparés par des barres obliques (<code>/</code>) ; chaque item peut être de la forme <code>-signal-number</code> ou de la forme <code>[-]signal-name</code>, ce qui demande d'envoyer ce signal ; ou bien de la forme <code>durée</code>, ce qui demande d'attendre tant de secondes avant de terminer les processus, ou bien de la forme <code>forever</code>, ce qui demande de répéter constamment le reste de <code>action-prévue</code>, si nécessaire.</p> <p>Quand l'action-prévue se termine et sans le paramètre <code>forever</code>, <code>start-stop-daemon</code> se termine avec un code 2.</p> <p>Quand une action-prévue est indiquée, tout signal donné par <code>-signal</code> est ignoré.</p>	
-a	<code>--startas chemin/nom</code>	<p>Avec l'action <code>-start</code>, lance le processus spécifié par <code>chemin/nom</code>.</p> <p>Si rien n'est précisé, c'est par défaut l'argument donné à <code>-exec</code>.</p>	
-t	<code>--test</code>	Affiche les actions qui seraient entreprises et détermine la bonne valeur de retour, mais ne fait rien.	
-o	<code>--oknodo</code>	si rien n'est fait, retourne 0 au lieu de 1.	
-q	<code>--quiet</code>		

courte	longue	explications	
-c	<code>--chuid identifiant uid[:groupe gid]</code>	<p>Change pour cet utilisateur ou « uid » avant de lancer le processus.</p> <p>On peut aussi préciser un groupe en ajoutant un :, puis le groupe ou un « gid » de la même façon qu'avec la commande « chown » (utilisateur:groupe).</p> <p>Lorsqu'un utilisateur est indiqué mais pas de groupe alors le groupe primaire de celui-ci est utilisé.</p> <p>Quand on utilise cette option, on doit veiller à ce que les groupes primaires ainsi que les groupes secondaires existent bien, même si l'option -group n'est pas spécifiée.</p> <p>L'option -group sert seulement pour les groupes dont l'utilisateur n'est pas membre (c'est comme rendre membre d'un groupe-pour-processus des utilisateurs génériques comme nobody).</p>	
-r	<code>--chroot root</code>	<p>Change de répertoire racine pour root avant de lancer le processus. Remarquez que le « pidfile » est aussi écrit après le changement de racine.</p>	
-d	<code>--chdir path</code>	<p>Change de répertoire pour chemin avant de commencer le processus.</p> <p>Cela est fait après le changement de répertoire racine si l'option -r</p>	<p>-chroot est demandée. Si rien n'est demandé, start-stop-daemon changera de répertoire racine avant de commencer le processus.</p>

courte	longue	explications	
-b	<code>--background</code>	<p>Utilisé généralement pour les programmes qui ne « se détachent » pas d'eux-mêmes. Cette option oblige start-stop-daemon à se dupliquer (fork) avant de lancer le processus et l'oblige à passer en arrière-plan. AVERTISSEMENT : start-stop-daemon ne peut pas vérifier le code de sortie quand, pour une raison ou une autre, le processus échoue.</p> <p>C'est un expédient dont on se servira seulement pour des programmes dont la duplication n'a pas de sens ou bien des programmes dont le code n'est pas adaptable pour leur ajouter cette fonctionnalité.</p>	
-C	<code>--no-close</code>	<p>Ne fermer aucun descripteur de fichiers en forçant le démon à s'exécuter en arrière-plan. Utilisé à des fins de débogage afin de voir ce qu'affiche le processus ou pour rediriger les descripteurs de fichiers pour journaliser l'affichage du processus.</p> <p>N'est pertinent que lors de l'utilisation de <code>--background</code>.</p>	
-N	<code>--nicelevel int</code>	modifie la priorité du processus avant qu'il ne soit lancé.	
-P	<code>--procsched politique:priorité</code>	<p>Modifie la politique du programmeur de processus (« process scheduler policy ») et la priorité du processus avant de le démarrer.</p> <p>La priorité peut être indiquée de manière facultative avec : suivi de la valeur souhaitée.</p> <p>La priorité par défaut est 0.</p> <p>Les valeurs de politiques actuellement gérées sont other, fifo et rr.</p>	

courte	longue	explications	
-I	<code>--iosched classe:priorité</code>	<p>Modifie la classe du programmeur d'entrée/sortie (« IO scheduler ») et la priorité du processus avant de le démarrer. La priorité peut être indiquée de manière facultative avec : suivi de la valeur souhaitée.</p> <p>La priorité par défaut est 4, sauf si classe est idle, auquel cas priorité sera toujours égale à 7. Les valeurs de classe actuellement gérées sont idle, best-effort et real-time.</p>	
-k	<code>--umask mask</code>	modifie le masque utilisateur du processus avant qu'il ne soit lancé.	
-m	<code>--make-pidfile</code>	<p>Utilisé lors du lancement d'un programme qui ne crée pas son propre fichier identificateur « pid ».</p> <p>Cette option indique à start-stop-daemon de créer le fichier référencé par <code>-pidfile</code> et placer le « pid » dans ce fichier juste avant d'exécuter le processus.</p> <p>Il faut remarquer que ce fichier n'est pas supprimé quand le programme s'arrête.</p> <p>NOTE : il se peut que cette caractéristique ne marche pas dans tous les cas.</p> <p>Notamment quand le programme qui est exécuté se duplique. À cause de cela, cette option n'est habituellement utile que combinée avec l'option <code>-background</code>.</p>	
-v	<code>--verbose</code>	Affiche des messages nombreux.	

Codes de sortie

0	<p>L'action demandée a été effectuée.</p> <p>Si <code>-oknodo</code> était indiqué, il est également possible que rien ne se soit passé. Cela peut se produire si <code>-start</code> était indiqué et qu'un processus correspondant était déjà en train de s'exécuter ou <code>-stop</code> était indiqué et qu'aucun processus ne correspondait.</p>
1	Si <code>-oknodo</code> n'était pas indiqué et que rien ne s'est passé.
2	Si <code>-stop</code> et <code>-retry</code> étaient indiqués mais que la fin de la planification a été atteinte et que les processus étaient toujours en cours d'exécution.
3	Toute autre erreur.

Avec la commande `-status` :

0	Le programme est en cours d'exécution.
1	Le programme n'est pas en cours d'exécution et le fichier PID existe.
3	Le programme n'est pas en cours d'exécution.
4	Impossible de déterminer l'état du programme.

Exemple

Démarrer le démon **foo**, à moins qu'il ne soit déjà en cours d'exécution (un processus nommé **foo**, tournant sous le nom d'utilisateur **foo**, avec un pid dans **foo.pid**) :

```
start-stop-daemon --start --oknodo --user foo --name foo \  
  --pidfile /run/foo.pid --startas /usr/sbin/foo \  
  --chuid foo -- --daemon
```

Envoyer le signal SIGTERM à foo et attendre son arrêt durant 5 secondes :

```
start-stop-daemon --stop --oknodo --user foo --name foo \  
  --pidfile /run/foo.pid --retry 5
```

Démonstration d'un ordonnanceur personnalisé pour l'arrêt de foo :

```
start-stop-daemon --stop --oknodo --user food --name food \  
  --pidfile /run/food.pid --retry=TERM/30/KILL/5
```

From:

<https://nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

https://nfrappe.fr/doc-0/doku.php?id=logiciel:os:linux:commandes_linux:start-stop-daemon

Last update: **2022/08/13 22:14**

