

---

# fcrontab

## Installer-gérer les tables fcron

fcrontab est le programme utilisé pour installer, éditer, lister et supprimer les tables utilisées par fcron. Comme fcron utilise en interne un format non lisible, l'utilisateur ne peut pas éditer directement son fcrontab. Un utilisateur peut installer un fcrontab s'il est listé dans **/etc/fcron.allow** et pas dans **/etc/fcron.deny**.

## OPTIONS

- u Spécifier l'utilisateur. ne devrait être utilisé que par root.
- l Liste le fcrontab de l'utilisateur courant
- e Éditer le fcrontab de l'utilisateur courant.
- r Supprimer le fcrontab de l'utilisateur courant.
- z Réinstaller le fcrontab de l'utilisateur courant.
- n Ignore la version précédente de fcrontab.
- c Spécifie le fichier de configuration
- d Mode debuggage

Chaque ligne dans un fichier crontab peut être :

- une configuration d'environnement
- une configuration d'options
- des entrées basées sur la durée totale de fonctionnement du système depuis la dernière exécution de la tâche
- des entrées basées sur un temps absolu (comme les entrées normales de crontab)
- des entrées lancées périodiquement

## Configuration de l'environnement

ils sont sous la forme : **name = value**. Quand fcron exécute une commande, il définit toujours **USER**, **HOME** et **SHELL** comme définit dans **/etc/passwd**. **HOME** et **SHELL** peuvent être remplacés par un paramètre dans fcrontab, mais pas **USER**. La variable **MAILTO** permet de spécifier à qui mailer les sorties.

## Entrées basées sur le temps écoulé depuis le démarrage du système

Les entrées des commandes qui ont été lancées une fois toutes les n minutes sont sous la forme :

**@options fréquence commande**

où la fréquence est une valeur de temps sous la forme :

**valeur+multiplieur[+valeur+multiplier[...]+valeur\_en\_minutes]**

exemple :

**"12h02", "3w2d5h1"**

il s'agit du temps d'exécution de fcron. les multiplieurs valides sont :

---

**mois(m), semaine(w), jours(d), heure(h) et seconde(s)**

**options** sert à placer une valeur de temps : il sera interprété comme **@first(<time>)**. Si **option** n'est pas spécifié, **frequency** est utilisé.

Ce type d'entrée ne garantit pas de date et heure d'exécution (vu que la tâche est retardé à chaque démarrage par le temps passé depuis l'arrêt), mais peut être pratique pour des tâches dépendant du nombre de choses faites par les utilisateurs.

Le temps restant avant la prochaine exécution est sauvegardé toutes les 1800 secondes (afin de limiter les dommages provoqués par un plantage) et lorsque fcron quitte après avoir reçu un signal SIGTERM, c.-à-d. lorsque l'on arrête le système.

## Exemples

recupérer vos mails toutes les 30 minutes :

**@ 30 getmails -all**

Effectuer quelques tests de sécurité toutes les 48 heures de fonctionnement du système, envoyer un mail au super-utilisateur

**@mailto(root),forcemail 2d /etc/security/msec/cron-sh/security.sh**

## Entrées basée sur une date et une heure

Le deuxième type d'entrée d'un fichier fcrontab commence par un caractère **&** optionnel qui peut être immédiatement suivi par un nombre optionnel définissant la fréquence d'exécution. il possède cinq champs de date et d'heure, ainsi qu'une commande shell :

**&options min hrs jour-du-mois mois jour-de-la-semaine commande**

Noter que la commande shell peut être précédée par un nom d'utilisateur. La fréquence est interprétée de la façon suivante : lancer cette commande après x correspondances des champs date et heure.

champ	Valeurs permises
minute	0-59
heure	0-23
jour du mois	1-31
mois	1-12 (ou leur noms)
jour de la semaine	0-7 (ou leur noms)

un \* signifie toutes les valeur. Les listes sont des valeur séparée par une virgule (1,4,8). Les intervalles sont séparés par un "-" (1-9), le pas est spécifié après un "/" (\* / 2), pour désactiver certaines valeur les précéder par " " (5-8 6 7)

Si un jour de mois et un jour de semaine sont donnés, la commande sera exécutée seulement si les deux correspondent au jour et à l'heure actuels à moins que l'option dayor n'ait été positionnée.

## Exemples

Lancer ma\_commande tous les jours à 12 :05, 12 :35, 13 :05, 13 :35, 14 :05 et 14 :35

**& 05,35 12-14 \* \* \* macommande -u moi -o fichier**

Récupérer les mails toutes les heures aux minutes 20, 21, 22, et 24.

**20-24 23 \* \* \* \* getmail**

Sauvegarder mon travail de la journée toutes les nuits à 03 :45 avec une faible priorité, sauf le dimanche, envoyer la sortie par mail à jim et lancer cette tâche au démarrage si la machine était éteinte à 03 :45

**&nice(10),mailto(jim),bootrun 45 03 \* \* \* 0 "save —mon travail"**

---

# Entrées lancées périodiquement

Le troisième type d'entrée fcrontab commence par le caractère "%", suivi par un mot-clef parmi une des trois différentes listes, puis par des options.

**hourly , daily , monthly , weekly**

Indique à fcron de lancer la commande une fois entre le début et la fin de l'intervalle de temps correspondant.

**midhourly , middaily , nightly , midmonthly , midweekly**

identique aux précédant, excepté que les intervalles de temps sont définis au milieu de l'intervalle.

**exemple :**

```
%nightly,mail(no) * 21-23,3-5 echo "une entrée nightly"
```

```
mins , hours , days , mons , dow
```

**Lance cette commande une fois durant CHAQUE intervalle de temps spécifié, ignorant les champs suivants le mot-clef dans la définition de l'intervalle de temps**

**Un tel mot-clef est suivi par 5 champs de date et d'heure (les mêmes champs utilisés pour une ligne basée sur un temps absolu.**

**exemple :**

```
%hours * 0-22 * * * echo "Ok."
```

**note :**

```
%mins 15 2-4 * * * echo
```

**lancera echo tous les jours à 2 :15, 3 :15 ET 4 :15.**

```
%hours 15 2-4 * * * echo
```

**lancera echo seulement UNE FOIS soit à 2 :15, soit à 3 :15 OU BIEN à 4 :15.**

## Options

Les options peuvent être positionnées soit pour chaque ligne située après la déclaration, soit pour une ligne de manière individuelle. Dans le premier cas, la configuration est faite sur une ligne complète après un point d'exclamation "!", dans le second cas, elle est faite après l'un des symboles suivants "&", "%" ou "@", suivant le type de planification. Veuillez noter qu'une déclaration d'option dans une planification écrase la déclaration globale de la même option.

Les options sont séparées par des virgules ",", et leurs arguments, s'il y en a, sont placés entre parenthèses. Les espaces ne sont pas permises. Une déclaration d'option est de la forme :

**option[(arg1[,arg2][...])][,option[(arg1[...])]][...]**

**bootrun, b** Lancer une ligne "&" au démarrage defcron si celle-ci avait du être lancée pendant l'arrêt du système.

**dayand** Effectuer un ET logique entre le jour de la semaine et le jour du mois.

**dayor** Effectuer un OU logique entre le jour de la semaine et le jour du mois.

**exesev** Définir si une tâche peut être exécutée plusieurs fois simultanément

**first, f** Temps avant la première exécution d'une tâche basée sur le temps de fonctionnement du système. Utile dans les cas suivants : vous avez plusieurs tâches à exécuter, disons, toutes les heures. En positionnant différentes valeurs first pour chaque tâche, vous évitez que celles-ci soient lancées simultanément chaque fois. Vous pouvez également la mettre à 0, ce qui est pratique lorsqu'elle est utilisée avec l'option volatile.

**forcemail** Envoie la sortie par mail même si celle-ci est vide.

**lavg** Définit les valeurs de la charge moyenne du système sur 1,5 et 15 minute (dans cet ordre) en dessous desquelles la tâche pourra être lancée. Ces valeurs ne peuvent avoir qu'une seule décimal. Mettre une valeur à 0 pour ignorer la charge moyenne correspondante.

**lavg1, lavg5, lavg15** Définir le seuil des valeurs de charge moyenne du système sur 1, 5 ou 15 minutes. Définir l'une d'elles à 0 pour ignorer la charge moyenne correspondante.

**lavgand** Effectuer un ET logique entre les valeurs de charge moyenne du système 1, 5 et 15 minutes.

**lavgonce** Une tâche peut-elle être mise plusieurs fois simultanément dans la file d'attente de charge moyenne du système.

**lavgor** Effectuer un OU logique entre les valeurs de charge moyenne du système à 1, 5 et 15 minutes.

**mail, m** Envoyer la sortie par mail

**mailto** Spécifier l'adresse mail.

**nice,n** Modifier la priorité de la tâche

**nolog** Journaliser seulement les erreurs des tâches correspondantes

**noticenotrun** fcron devrait-il prévenir l'utilisateur par mail de la non exécution d'une tâche % ou d'une tâche & ? (à cause de l'arrêt du système dans les deux cas ou d'une charge moyenne trop élevée dans le second cas).

**random** Dans une ligne lancée périodiquement, cette option répond à la question : Cette tâche devrait-elle être lancée aussitôt que possible dans son intervalle de temps d'exécution (plus sûr), ou bien fcron doit-il définir une date et heure aléatoire d'exécution dans ce même intervalle de temps ?. Veuillez noter que si cette option est positionnée, la tâche peut ne pas être lancée si fcron n'est pas en cours d'exécution durant tout l'intervalle d'exécution. En outre, sachez qu'il peut être relativement facile pour une personne compétente de deviner l'heure de la prochaine exécution d'une tâche utilisant l'option random : il est donc préférable de ne pas baser la sécurité de quelque chose de sensible sur l'aléa de cette option. Cependant, cela ne devrait pas poser de problème pour la plupart de ses utilisations.

**reset** Positionner toutes les options à leur valeurs par défaut

**runas** Lancer avec les permissions et l'environnement de l'utilisateur spécifié (seul root peut utiliser cette option)

**runfreq, r** Lancer toutes le "runfreq" correspondances de date et d'heure. (cette options est ignorée dans le cas d'entrées basées sur le temps écoulé depuis le démarrage du système).

**serial, s** fcron fait tourner au plus 1 tâche séquentielle en même temps (càd. pour lesquelles l'option serial est définie), et autant de tâches qui sont à la fois séquentielles et dont l'exécution dépend aussi de la charge système (autrement dit pour lesquelles les options serial et lavg sont définies). Cette valeur peut être modifiée grâce à l'option -m de fcron. Cette option trouve toute son utilité dans le cas de tâches lourdes afin d'éviter une surcharge du système.

**serialonce** Définit si un tâche peut être mise plusieurs fois dans la file d'attente séquentielle.

**stdout** Si fcron s'exécute en arrière plan, permet la sortie sur stderr/stdout plutôt que par mail.

**strict** Lorsqu'une tâche % de la file d'attente de charge moyenne du système est à la fin de l'intervalle de temps d'exécution, doit-elle être retirée de la file d'attente (strict(true), ainsi la tâche ne sera pas exécutée), ou bien doit-elle y rester jusqu'à ce que la charge moyenne du système permette son exécution (strict(false)) ?

**timezone** Lancer la tâche dans le fuseau horaire fourni. timezone-name est une chaîne qui doit être valide du point de vue de la variable d'environnement TZ

**until** Définir le délai d'attente maximal pour les valeurs de charge système. Si le délai est dépassé, la tâche sera lancée indépendamment de la charge système. Définir until à 0 pour désactiver le délai d'attente.

**volatile** Activé, fcron ne retient pas le temps écoulé depuis la dernière exécution des tâches « volatiles » entre deux démarrages de fcron, et agit comme si les lignes avaient été fraîchement ajoutées à chaque démarrage de fcron. Utile lorsque fcron est lancé à partir d'un script qui tourne, par exemple, seulement pendant une connexion à Internet non permanente : l'exécution des tâches volatiles est alors basée sur le temps depuis le début de la connexion et le lancement de fcron plutôt que sur le temps absolu. Enfin, veuillez noter que cette option s'associe bien à l'option first.

## Exemple de fichier fcrontab

Utiliser /bin/bash

**SHELL=/bin/bash**

Envoyer la sortie par mail à l'utilisateur thib, quelque soit le propriétaire de ce fcrontab.

**!mailto(thib)**

Définir une variable équivalente à "Salut thib et paul !" les caractères newline sont échappés avec une barre oblique inverse "\n" et les guillemets servent à conserver les espaces de début et de fin de chaîne

**TEXT= " Salut\**

**thib et\**

**paul! "**

On veut utiliser la file d'attente séquentielle mais pas l'option bootrun

**!serial(true),b(0)**

Lancer la première fois après 5 minutes de fonctionnement du système, puis toutes les heures par la suite

**@first(5) 1h echo "Lancer toutes les heures"**

Lancer tous les jours

**@ 1d echo "fcron quotidien"**

Lancer une fois dans la matinée et une fois dans l'après-midi si le système est en fonctionnement à n'importe quel moment pendant ces intervalles de temps

**%hours \* 8-12,14-18 \* \* \* echo "Hé patron, je bosse aujourd'hui!"**

---

Lancer une fois par semaine à l'heure du déjeuner  
**%weekly \* 12-13 echo "J'ai laissé allumé mon système au moins une fois\  
à l'heure du déjeuner cette semaine."**

Lancer tous les samedi et dimanche à 9 :05  
**5 9 \* \* sat,sun echo "Bonjour Thibault!"**

Lancer tous les jours pairs du mois de mars à 18 :00, sauf le 16  
**0 18 2-30/2 16 Mar \* echo "Il est temps de se rentrer à la maison!"**

La ligne précédente est équivalente à  
**& 0 18 2-30/2 16 Mar \* echo "Il est temps de se rentrer à la maison!"**

Mettre toutes les options à leur valeur par défaut et définir runfreq pour les lignes qui vont suivre  
**!reset,runfreq(7)**

Lancer une fois toutes les 7 correspondances (grâce à la déclaration précédente), ainsi, si le système est en fonctionnement tous les jours à 10 heures,  
**La commande sera lancée une fois par semaine**

**& 0 10 \* \* \* echo "Si vous avez vu ce message pour la dernière fois il y a 7 jours,\  
c'est que cette machine a fonctionné tous les jours à 10 heures durant la dernière semaine.\  
Si vous avez eu ce message il y a 8 jours, c'est que cette machine n'a pas fonctionné\  
un jour à 10 :00 heures depuis le dernier message, etc."**

**Attendre toutes les heures que la charge système moyenne sur 5 minutes soit inférieure à 0.9**  
**@lavg5(0.9) 1h echo "La charge moyenne système est faible"**

**Attendre au maximum 5 heures chaque jour une baisse de la charge système**  
**@lavgand,lavg(1,2.0,3.0),until(5h) 1d echo "La charge système vient de diminuer"**

**Attendre le meilleur moment pour lancer une tâche lourde**  
**@lavgor,lavg(0.8,1.2,1.5),nice(10) 1w echo "C'est une très lourde tâche"**

**Lancer une fois toutes les nuits soit entre 21 :00 et 23 :00, soit entre 3 :00 et 6 :00**  
**%nightly,lavg(1.5,2,2) \* 21-23,3-6 echo "Il est temps de récupérer\  
la dernière version de Mozilla!"**