

Faire exécuter des scripts en fonction des événements fichiers avec Incron

Table of Contents

- [Présentation d'incron](#)
- [Installation d'incron](#)
- [Les événements auxquels incron réagit](#)
- [La table des règles : incrontab](#)
- [Exemple de génération d'un hash md5 de façon automatique](#)
- [Exemple de traitement automatique des téléchargements](#)

Afin de surveiller la modification, suppression... de fichiers ou répertoires, on utilise généralement des scripts lancés par cron à intervalles réguliers ou, moins fréquemment, on utilise inotify (intégré au noyau de puis le 2.6.13).

Ces solutions, si elles sont pertinentes, posent des problèmes soit de charge (exécution inutile de script avec cron) soit des difficultés d'extensibilité ou maintenance.

C'est là que Incron peut apporter une réponse simple et facile à mettre en œuvre.

Présentation d'incron

Alors que cron réagit en fonction des heures, minutes, mois. . . incron réagit en fonction des événements affectant les fichiers tels modification, accès, suppression, déplacement. . .

D'ailleurs incron n'est rien d'autre que l'acronyme de Inotify Cron.

Cette capacité à réagir aux événements fichiers va être mise en place au travers de incrontab tout comme cron au travers de crontab.

Installation d'incron

Sous Debian :

```
aptitude install incron inotify-tools
```

Pour les autres distributions la manipulation est similaire avec le gestionnaire de paquets de celles-ci.

Le fichier `/etc/incron.conf` contient les paramètres de fonctionnement d'incron. Ce fichier est très simple et facilement compréhensible.

Les événements auxquels incron réagit

Incron réagit à des événements dont **incron**, le démon d'incron, permet d'en avoir la liste très

facilement en saisissant la commande suivante :

```
incron -t
```

```
IN_ACCESS , IN_MODIFY , IN_ATTRIB , IN_CLOSE_WRITE , IN_CLOSE_NOWRITE ,
IN_OPEN ,
IN_MOVED_FROM , IN_MOVED_TO , IN_CREATE , IN_DELETE , IN_DELETE_SELF ,
IN_CLOSE ,
IN_MOVE , IN_ONESHOT , IN_ALL_EVENTS , IN_DONT_FOLLOW , IN_ONLYDIR ,
IN_MOVE_SELF
```

Ce résultat nécessite quelques explications. Dans les tableaux qui suivent apparaissent les différents événements acceptés par **Incron**. Comme on peut le voir ceux-ci ouvrent des perspectives intéressantes.

IN_ACCESS	Le fichier a été accédé en lecture (*).
IN_ATTRIB	Les métadonnées ont changé, par exemple, les permissions, l'horodatage, les attributs étendus, le nombre de liens (depuis Linux 2.6.25), uid, gid. . . (*)
IN_CLOSE_WRITE	Le fichier ouvert en écriture a été fermé (*).
IN_CLOSE_NOWRITE	Le fichier non ouvert en écriture a été fermé (*).
IN_CLOSE	équivalent à IN_CLOSE_WRITE IN_CLOSE_NOWRITE
IN_CREATE	Un fichier ou répertoire a été créé dans le répertoire surveillé (*).
IN_DELETE	Un fichier ou répertoire a été supprimé à partir du répertoire surveillé (*).
IN_DELETE_SELF	Le fichier ou répertoire surveillé a été lui-même supprimé.
IN_MODIFY	Le fichier a été modifié (*).
IN_MOVE_SELF	Le fichier ou répertoire surveillé a été lui-même déplacé.
IN_MOVED_FROM	Le fichier a été déplacé hors du répertoire surveillé (*).
IN_MOVED_TO	Un fichier a été déplacé dans le répertoire surveillé (*).
IN_MOVE	équivalent à IN_MOVED_FROM IN_MOVED_TO
IN_OPEN	Le fichier surveillé a été ouvert (*).
IN_ALL_EVENTS	est définie comme l'ensemble des événements ci-dessus.
IN_DONT_FOLLOW (depuis Linux 2.6.15)	Ne pas déréférencer le chemin s'il s'agit d'un lien symbolique.
IN_ONESHOT	Surveiller le chemin pour un événement, puis le retirer de la liste de surveillance.
IN_ONLYDIR (depuis Linux 2.6.15)	Surveiller le chemin si c'est un répertoire.
IN_ISDIR	La cible de cet événement est un répertoire.
IN_UNMOUNT	Le système de fichiers contenant l'objet surveillé a été démonté.

Lors de la surveillance d'un répertoire, les événements marqués d'un astérisque (*) ci-dessus peuvent se produire pour les fichiers dans le répertoire, auquel cas le nom retourné identifie le nom du fichier dans le répertoire.

La table des règles : incrontab

La table contenant les règles incron est invoquée par la commande `incrontab -e`.

La ligne décrivant les règles dans la table est séparée en quatre champs décrits ci-dessous.

- Le répertoire surveillé (ex. `:/home/toto/repertoire surveille`).
- Le ou les événements déclencheurs (ex. `:IN_CLOSE_WRITE`). Il est possible de préciser plusieurs événements. Dans ce cas, il suffit de les séparer par une virgule.
- La commande à exécuter (ex. `:/home/toto/scripts/mon_script.sh`).
- Les paramètres à passer à la commande (Ceux-ci sont facultatifs mais en pratique très utilisés, principalement les deux premiers):
 - `$`: chemin du répertoire surveillé.
 - `$#`: nom du fichier qui a déclenché l'événement.
 - `$$`: affiche le signe `$`.
 - `$%`: affiche le nom de l'événement.
 - `$`: affiche la valeur numérique de l'événement.



Il est impératif de saisir les chemins absolus.

Si l'on ne se souvient plus des règles en vigueur dans **incrontab**, il suffit de saisir la ligne suivante dans un terminal afin d'en avoir la liste.

```
incrontab -l
```

Exemple de génération d'un hash md5 de façon automatique

Cet exemple permet de générer la clef MD5 d'un fichier quelconque qui sera déplacé d'un répertoire source vers le répertoire cible qui sera surveillé par incron.

La clef sera écrite dans un fichier portant le même nom que celui dont on calcule la clef, mais se terminant par `".md5"`.

On suppose que le fichier source aura une extension comme `jpg`, `iso`, `mkv`. . .

Voici l'exemple de ligne saisie dans `incrontab`. Elle présuppose que le répertoire surveillé `md5` existe et que le script se situe dans le répertoire `/home/toto/scripts/`.

```
/home/toto/md5 IN_MOVED_TO /home/toto/scripts/incron_md5.sh @$ $#
```

Explication de la ligne:.

- **/home/toto/md5**: Il s'agit du répertoire que surveillé.
- **IN_MOVED_TO**: incron réagira à tout fichier qui sera déplacé d'un répertoire dans le répertoire surveillé.

- **/home/toto/scripts/incron_md5.sh**: Le script qui sera utilisé pour calculer la clef md5.
- **\$@ \$#**: On passe le chemin du répertoire surveillé et le nom du fichier qui a déclenché l'événement au script.

```
#!/bin/bash
# script générant les hash md5
# recoit en paramètre = $1 : le répertoire / $2 : le nom du fichier
# événement IN_MOVED_TO
#
# déplacement répertoire /home/toto/md5
cd "$1"
# création somme md5 seulement si c'est un fichier dont l'extension
n'est pas .md5
if [[ ${2##*.} != "md5" ]]
then
md5sum "$2" "${2%.*}.md5"
fi
```

Comme on peut le voir, le script n'est pas très compliqué en lui même. Et avec incron on peut automatiser nombre de choses, comme le calcul d'un hash md5 de l'iso de notre distribution préférée après téléchargement.

Exemple de traitement automatique des téléchargements

Cet exemple permet, lorsqu'on télécharge un iso, d'en calculer automatiquement le hash md5 et de placer l'iso et son hash md5 dans le répertoire d'archive.

Pour ce faire, on va surveiller le répertoire `/home/toto/download/` qui est le répertoire par défaut pour tous les téléchargements.

Une fois que le fichier sera entièrement téléchargé, on le bascule dans le répertoire `/home/toto/md5` afin de calculer son hash md5.

Ceci permettra de vérifier qu'il n'y a pas eu d'erreur de téléchargement.

On a déjà mis en place une partie de ce que l'on veut faire avec l'exemple précédent. Il ne reste qu'à mettre en place la surveillance du répertoire `/home/toto/download` et le déplacement du fichier une fois fini le téléchargement.

Pour cela ouvrir la table incron par la commande :

```
incrontab -e
```

Puis saisir la ligne suivante afin de surveiller le répertoire `/home/toto/download`.

```
/home/toto/download IN_CLOSE /home/toto/scripts/incron_download.sh $@ $#
```

Puis mettre en place le script `incron_download.sh`

```
#!/bin/bash
#####
# script générant les hash md5
# recoit en paramètre = $1 : le répertoire / $2 :le nom du fichier
# événement IN_CLOSE
#####
#
# déplacement répertoire /home/toto/download
cd "$1"
# Déplacement du fichier reçu en paramètre
# et du fichier de hash md5 vers le répertoire d'archive
#
if [[ ${2##*.} != "md5" ]]
then
md5sum "$2" "${2%.*}.md5"
fi
mv ${2%.*}.* /home/toto/Archive
```

From:

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

Permanent link:

<http://www.ouarte.garden/doku.php?id=notes:incron>

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

