

Ce projet a pour objectif de créer un programmeur de tâches appelée « pcron ».

## 1 FONCTIONNEMENT ET SYNTAXE

- La commande « pcron » est chargée de faire exécuter par le système toutes tâches (commandes et scripts) définies et planifiées à l'avance. Ces tâches allant de la simple commande aux scripts complexes peuvent ainsi être exécutées à heure fixe et même de façon périodique, et font l'objet de messages de compte-rendu.
- La commande lit régulièrement les fichiers présents dans le répertoire /etc/pcron/ et le fichier /etc/pcrontab pour voir si des tâches doivent être exécutées.
- Chaque action de « pcron » ajoute une ligne de message dans le fichier /var/log/pcron.
- Par défaut si une commande lancée par la commande « pcron » produit un affichage, il est dirigé vers la sortie standard.
- L'usage de la commande « pcron » est en principe réservé à l'administrateur. On peut toutefois autoriser certains utilisateurs. Pour cela, on en dresse la liste sur des lignes successives dans le fichier /etc/pcron.allow, et de façon symétrique, on peut mettre dans /etc/pcron.deny la liste des utilisateurs non autorisés. Exemple : seuls Jean et Charles ont la permission d'utiliser le service « pcron » :

```
$ cat pcron.allow
```

```
Jean  
Charles
```

- Une commande « pcrontab » permet la programmation du service « pcron ». La syntaxe de cette commande est la suivante :

```
pcrontab [-u user] {-l | -r | -e}
```

*pcrontab -u toto -l* affiche le fichier pcrontab de l'utilisateur toto situé dans le répertoire /etc/pcron/

*tcontab -u toto -r* efface le fichier pcrontab de l'utilisateur toto

*pcrontab -u toto -e* crée ou édite (pour modification) un fichier temporaire dans /tmp ouvert dans vi. Lors de la sauvegarde, le fichier est écrit dans /etc/pcron/pcrontabtoto.

- Chaque ligne du fichier pcrontab contient 7 champs. Les 6 premiers champs déterminent les moments d'exécution de la tâche décrite au 7ème champ.
- Les 6 premiers, séparés par des espaces, appelés champs temporels, décrivent la périodicité : 15 secondes (0-3), minutes (0-59), heures (0-23), jour du mois (1-31), mois de l'année (1-12), jour de la semaine (0-6, 0=dimanche)
- Le 7ème est la commande à exécuter, ce peut être naturellement un script.
- Un champ temporel peut contenir :
  - Une valeur précise et valide pour le champ (par exemple 15 dans le champ minutes)
  - Une liste de valeurs valides, séparées par deux points (1:3:5 dans le champ mois : janvier, mars, mai)
  - Un intervalle valide (1-5 dans le champ jour : du lundi au vendredi)
  - \* pour signifier toutes les valeurs possibles du champ (\* dans le champ minute : toutes les minutes)
  - \*/4 dans le champ heures : toutes les 4 heures.
  - Un ou plusieurs « ~nombre » peuvent être ajoutés afin de désactiver certaines valeurs dans l'intervalle. Par exemple, « 5-8~6~7 » est équivalent à « 5:8 ».

- Exemples

```
# Exécution toutes les quatre heures chaque  
1er et 15 de chaque mois  
0 0 */4 1,15 * * commande
```

```
# Provoquer un reboot la machine chaque 1er  
et 15 du mois à 2h 30 min 30 sec du matin  
2 30 2 1,15 * * /sbin/shutdown -r
```

```
# Appeler un script de sauvegarde tous les  
lundis à 3 h 15 du matin  
0 15 3 * * 1 /usr/bin/backup
```

```
# Exécution toutes les minutes passées de 15  
secondes  
1 * * * * * commande
```

```
# Exécution tous les matins du lundi au  
vendredi à 7 h 30  
0 30 7 * * 1-5 commande
```

```
# Exécution tous les quarts d'heure de 15h à  
19h du lundi au vendredi seulement en 1ère  
quinzaine de chaque mois de l'année sauf le  
mois d'avril et le mois de mai.  
0 0,15,30,45 15-19 1-15 1-12~4~5 1-5  
commande
```

```
# Trouver puis nettoyer le répertoire /tmp  
des vieux fichiers (non modifiés depuis 3  
jours) tous les 1ers jours de chaque mois à 2  
heures du matin  
0 0 2 1 * * find /tmp -atime 3 -exec rm -f {} \;
```

## 2 PRECISIONS

Toutes ressemblances avec des commandes existantes telles que les commandes crontab, anacron, fcron ou tous dérivés du service cron ne sont pas volontaires. En conséquence, il n'est pas conseillé d'utiliser ces commandes pour effectuer le projet.

## 3 EXTENSIONS

Toutes les extensions au sujet seront les bienvenues.  
L'interface utilisateur est laissée à votre discrétion.

## 4 RAPPORT ET PRÉSENTATION

Ce projet doit être effectué exclusivement en trinôme.

Un rapport final de quelques pages doit être rendu avant le lundi 24 février 2025. Une présentation et un test du Shell seront effectués pendant une quinzaine de minutes la dernière semaine du trimestre (du 24 au 27 février). Bon courage !!!