

Sommaire

Connexion à Internet multi-comptes Partie 3 : Automatisation partielle (ip-up et ip-down).....	1
Rappel des épisodes précédents.....	1
ip-up.....	1
ip-down.....	3
Script de connexion/déconnexion.....	3
Conclusion.....	4
Copyright.....	5
Autres ressources.....	6

Connexion à Internet multi-comptes

Partie 3 : Automatisation partielle (ip-up et ip-down)

Connexion à Internet multi-comptes

Partie 3 : Automatisation partielle (ip-up et ip-down)

par Fred

Jusqu'à maintenant nous avons vu comment nous connecter à Internet et comment recevoir nos mail, mais la connexion reste une procédure complexe pour l'utilisateur moyen.

Rappel des épisodes précédents

Nous savons nous connecter à Internet par la commande :

```
/usr/sbin/pppd call connexion0
```

Nous savons nous déconnecter d'Internet par la commande :

```
kill `cat /var/run/ppp*.pid`
```

qui doit être lancée par l'utilisateur qui a établi la connexion.

Nous savons récupérer nos mails par la commande :

```
fetchmail -v
```

qui, elle, doit être lancée depuis le compte de root.

Nous savons comment envoyer nos mails en attente :

```
/usr/sbin/sendmail -q
```

Comment faire pour automatiser 'tout' cela ?

ip-up

Lorsque la connexion est s'établit, pppd lance le script (s'il existe) : `/etc/ppp/ip-up`. Donc, pour que quelque chose se passe lorsque vous vous connectez, il suffit de créer un script ayant ce nom.

Souvent, votre distribution fournit déjà un tel script qui se charge d'exécuter `/etc/ppp/ip-up.local`. Si c'est le cas (lisez le contenu du script) vous pouvez pour faire ce qui va suivre :

- ou bien effacer le fichier `/etc/ppp/ip-up` préexistant ainsi que `/etc/ppp/ip-up.local` et créer celui que je vais vous indiquer (et que j'espère vous comprendrez) ; plutôt que d'effacer, renommez ces fichiers avec par exemple un suffixe `.bak`, cela vous permettra de revenir en arrière le cas échéant.

Reseau-cxion net-multi3 ipupdown

- ou bien effacer `/etc/ppp/ip-up.local` et faire ce que je vais décrire pour `/etc/ppp/ip-up` sur `/etc/ppp/ip-up.local` à la place (c'est plus sûr du point de vue de la distribution, mais il y a des chances pour que vous ne sachiez pas exactement ce que fait `/etc/ppp/ip-up` – moi je n'aime pas ça).

Ces considérations philosophiques étant tenues, passons aux choses sérieuses. Que souhaitons-nous qu'il se passe à chaque connexion ? Je vois plusieurs choses :

- que les mails soient automatiquement rapatriés sur tous les comptes,
- que les mails que nous recevons pendant que nous sommes connectés soient eux aussi rapatriés sur les comptes idoines,
- que le DNS de notre ISP soit activé,
- que les mails en attente soient envoyés,
- prévenir l'utilisateur que la liaison est effectivement active (il se passe toujours un certain temps avant que la liaison ne s'établisse...).

Comme nous utilisons X Window, nous allons faire en sorte d'utiliser des boîtes de dialogues en mode graphique, cela fera plus 'pro'. Comme nous ne sommes pas des programmeurs avertis, nous allons nous contenter du langage `/bin/sh`, c'est à dire le `[../admin/shell.php3 shell] bash !` Pour lui faire afficher des boîtes de dialogue, je vous propose d'utiliser l'excellent `Xdialog` : un utilitaire qui permet d'ouvrir une boîte de dialogue depuis un script shell (installation par : `./configure --prefix=/opt/apps ; make ; make install`).

Je n'expliquerai pas comment il fonctionne (essentiellement comme `dialog`, pour connaître les options tapez : `/opt/apps/bin/Xdialog`).

Voilà ce que l'on peut mettre dans `/etc/ppp/ip-up` :

```
#!/bin/sh# le paramètre $6 correspond au paramètre
# ipparam du fichier /etc/ppp/peers/script# ---Configuration
du DNS---Début---
# ceci nécessite que l'option usepeerdns
# soit activée dans /etc/ppp/peers/script
# dans ce cas DNS1 et DNS2 reçoivent les
# adresse des deux dns du provider.
# sinon mettez ici vos deux dns :
#DNS1=123.123.123.123
#DNS2=213.213.213.213
echo "# fichier généré par ip-up." > /etc/resolv.conf
echo search org >> /etc/resolv.conf
echo nameserver $DNS1 >> /etc/resolv.conf
echo nameserver $DNS2 >> /etc/resolv.conf
# ---Configuration du DNS---Fin--- # récupération des mails :
/usr/bin/fetchmail
# et envoie des mails en attente :
/usr/sbin/sendmail -q# on récupère automatiquement les mails
toutes les 11 minutes
# (11 minutes car netcourrier n'aime pas qu'on consulte
# les comptes pop trop souvent !)
/usr/bin/fetchmail --daemon 660 # le plus de la semaine :
# mettre son pc à l'heure :
ntpdate ntp-sop.inria.fr # on est connecté, on le dit :
```

Reseau-cxion net-multi3 ipupdown

```
export DISPLAY=:0
/opt/apps/bin/Xdialog --title "Information" --center\
--infobox "La connexion est maintenant active." 0 0 3000
```

Ce script n'appelle pas de commentaire particulier, sinon que pour que le PC puisse être mis à l'heure il faut que le paquetage xntp3 soit installé.

Il faut savoir, avant de mettre n'importe quoi dans ce fichier, que celui-ci est toujours exécuté avec les droits de **root**, donc prudence. De plus ce script est exécuté avec un PATH minimal (PATH= " "), donc vous devez préciser le chemin complet des exécutable.

ip-down

De même que pppd lance ip-up à la connexion, il lance /etc/ppp/ip-down lorsque la connexion vient d'être interrompue. Lorsque la connexion est rompue, nous pouvons stopper le daemon fetchmail (en effet fetchmail ne peut plus que retourner des messages d'erreur), de plus il faut remettre le fichier /etc/resolv.conf à sa valeur par défaut. Voici le script /etc/ppp/ip-down que j'utilisais :

```
#!/bin/sh # comme on n'est plus connecté au net, on quitte
fetchmail
/usr/bin/fetchmail --quit # ---Configuration du DNS---Début---
echo "# fichier généré par ip-down." > /etc/resolv.conf
# mon domaine local est tux !
echo search tux >> /etc/resolv.conf
# ---Configuration du DNS---Fin--- # on est déconnecté, on le
dit :
export DISPLAY=:0
/opt/apps/bin/Xdialog --title "Information" --center\
--infobox "La connexion est maintenant inactive." 0 0 3000
```

Les mêmes commentaires de prudence que pour ip-up s'appliquent à ip-down.

Script de connexion/déconnexion

Pour se connecter, il n'est pas nécessaire de créer un script, mais nous allons en créer un, pour que la procédure de connexion soit la même que la procédure de déconnexion. Voilà les deux scripts :

- /etc/ppp/scripts/pppconnect
-

```
#!/bin/sh
/usr/sbin/pppd call connexion0
```

- /etc/ppp/scripts/pppdisconnect
-

```
#!/bin/sh  
kill `cat /var/run/ppp*.pid`
```

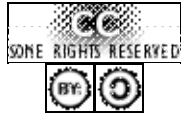
Conclusion

Bon, jusqu'à maintenant nous n'avons pas fait grand-chose pour la connexion multicomptes. C'est l'objet de la [multi4_compte.php3 prochaine partie].

Cette page est issue de la documentation 'pré-wiki' de Léa a été convertie avec HTML::WikiConverter. Elle fut créée par Frédéric Bonnaud le 30/05/2000.

Copyright

Copyright © 30/05/2000, Fr?ric Bonnaud



*Ce document est publié sous licence Creative Commons
Attribution, Partage à l'identique 2.0 :*
<http://creativecommons.org/licenses/by-sa/2.0/>

Autres ressources