

Sauvegarde/Restauration de Base de Données

ANNEXES

ANNEXE 1 : Redirection de messages

Redirections Enchaînement de commandes

Introduction

Lorsque l'on exécute un programme sous Linux/Unix, celui-ci va communiquer avec ce qui lui est extérieur de trois façons :

- Il attend des données sur son entrée standard (STDIN) qui portera le numéro 0 ;
- Il produit un affichage de données sur sa sortie standard (STDOUT) qui portera le numéro 1 ;
- Il produit un affichage sur sa sortie d'erreurs (STDERR) qui portera le numéro 2 ;

C'est ce que l'on appelle les flux standards.

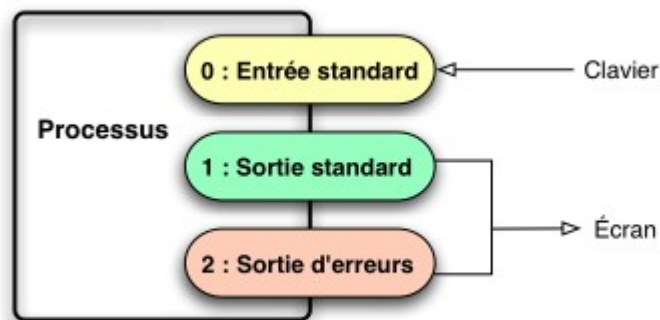


Illustration 1: Les flux standards avant redirection

Il est possible d'utiliser ou rediriger les flux d'un programme de différentes manières.

Redirection vers ou en provenance de fichiers

L'objectif est par exemple d'enregistrer le résultat d'une commande dans un fichier plutôt que de provoquer un affichage à l'écran.

Plusieurs types de redirection existent :

commande > fichier permet de rediriger le résultat d'une commande vers le fichier indiqué. Le fichier est écrasé s'il existait.

commande >> fichier fait la même opération mais sans écraser le contenu du fichier. Les nouvelles données sont ajoutées en fin de fichier.

commande n> fichier permet de rediriger le flux numéro n vers un fichier.

Remarque : **commande 1> fichier** est équivalent à **commande > fichier** car 1 est la sortie standard.

commande &> fichier permet de rediriger la sortie standard et l'erreur standard vers un seul et même fichier.

commande < fichier permet d'envoyer le contenu d'un fichier comme entrée d'une commande.

Redirection d'un flux vers un autre

Il peut être nécessaire de rediriger un flux dans un autre, de façon, par exemple, à obtenir tout l'affichage produit par une commande.

commande n>&m permet de dupliquer la sortie standard vers un des descripteurs de fichier.

Exemple : pour rediriger les erreurs vers le flux d'affichage normal, on fait

commande 2>&1

Enchaînement de commande : pipelines

il est possible d'envoyer la sortie standard d'une commande vers l'entrée standard d'une autre. On utilise pour cela un *tube* ou *pipeline*. Un pipeline est noté |

commande1 | commande2

Exemple : ls | grep mot

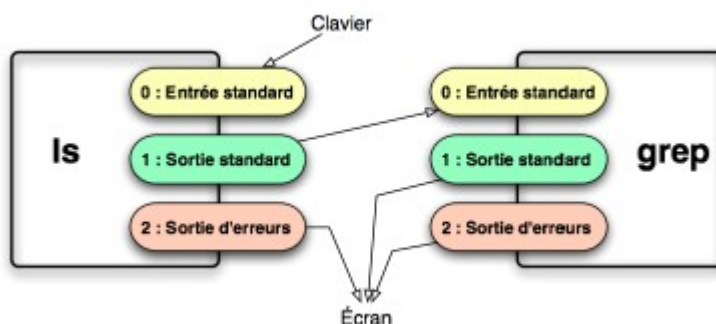


Illustration 2: Redirection de la sortie d'une commande vers l'entrée d'une autre

Enchaînement conditionnel de commandes

On peut vouloir exécuter une commande à la question que la précédente se soit déroulée normalement :

commande1 && commande2

On peut vouloir exécuter une commande seulement si la précédente ne s'est pas déroulée normalement :

commande1 || commande2

ANNEXE 2 : Utilisation du service cron

Exemple de fichier `/etc/crontab` :

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow usercommand
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
00 01 * * * root /root/scripts/scan_virus
```

Chaque ligne du fichier correspond à une tâche à exécuter doit être décrite en indiquant tous les paramètres suivants :

Position du paramètre dans la ligne	Type	Valeur
1	minute de l'heure à laquelle doit s'exécuter la commande	entre 0 et 59, ou *
2	heure du jour à laquelle doit s'exécuter la commande	entre 0 et 23, ou *
3	jour du mois auquel doit s'exécuter la commande	entre 1 et 31, ou *
4	mois de l'année auquel doit s'exécuter la commande	entre 1 et 12, ou *
5	jour de la semaine auquel doit s'exécuter la commande	entre 0 et 7, ou * 0 et 7 représentent le dimanche
6	Utilisateur qui exécute la commande	Le plus souvent root
7	commande à exécuter	tout exécutable

Le caractère * représente chaque valeur possible pour le paramètre auquel il s'applique.

ANNEXE 3 : AIDE SUR LA COMMANDE DATE

date - Afficher ou configurer la date et l'heure du système.

SYNOPSIS

date [-d chaîne_date] [-f fichier] [-r fichier] [-R] [-s chaîne_date] [-u] [--date=chaîne_date] [--file=fichier] [--reference=fichier] [--rfc-822] [--set=chaîne_date] [--utc] [--universal] [--help] [--version] [+FORMAT] [MMJJhhmm[[SS]AA][.ss]]

DESCRIPTION

Cette page de manuel documente la version GNU de **date**.

date sans argument affiche la date et l'heure actuelles (avec le format indiqué par la directive '%c' décrite plus-bas).

Si un argument commençant par un '+' est indiqué, la date et l'heure sont affichées avec un format contrôlé par cet argument. La structure de la chaîne précisant le format est semblable à celle utilisée avec la fonction [strftime\(3\)](#).

Les caractères présents dans cette chaîne sont affichés tels quels, sauf les directives, commençant toutes par '%'.
Ces directives sont les suivantes :

% un caractère % littéral.
 n saut de ligne
 t tabulation horizontale

Champs horaires :

%H	heure (00..23)	%r	heure actuelle (sur 12 heures) (hh:mm:ss [AP]M)
%I	heure (01..12)	%s	secondes écoulées depuis le 01-01-1970 à 00:00:00 T.U (extension non standard)
%k	heure (0..23)	%S	secondes (00..61)
%l	heure (1..12)	%T	heure actuelle, (sur 24 heures) (hh:mm:ss)
%M	minute (00..59)	%X	représentation locale de l'heure (%H:%M:%S)
%p	notation locale pour AM ou PM.	%Z	fuseau horaire (par ex. MET), ou rien si le fuseau horaire n'est pas déterminé

Champs de date :

%a	abréviation locale du jour de la semaine (Dim .. Sam)	%h	comme %b
%A	nom local du jour de la semaine (Dimanche .. Samedi)	%j	jour de l'année (001..366)
%b	abréviation locale du nom du mois (Jan..Dec)	%m	mois (01..12)
%B	nom local du mois (Janvier .. Décembre)	%U	numéro de semaine dans l'année (00..53). La semaine commence le Dimanche.
%c	date et heure locales (Sat Nov 04 12:02:33 EST 1989)	%w	Jour de la semaine (0..6). Le 0 correspond au Dimanche.
%d	jour du mois (01..31)	%W	numéro de semaine dans l'année (00..53). La semaine commence le Lundi.
%D	date (mm/jj/aa)	%x	représentation locale de la date (mm/jj/aa)
		%y	deux derniers chiffres de l'année (00..99)
		%Y	année (1970...)

Par défaut, **date** complète les champs numériques avec des zéros. La version GNU de **date** dispose des options non standards suivantes :

- (tiret) ne pas compléter le champ
- _ (souligne) remplir le champ avec des espaces

Si l'on fournit un argument ne commençant pas par '+', **date** le considère comme une heure à utiliser pour configurer l'horloge système. Cet argument doit être constitué uniquement de chiffres, ayant la signification suivante :

MM	mois	SS	Deux premiers chiffres de l'année (facultatif)
JJ	jour du mois	AA	Deux derniers chiffres de l'année (facultatif)
hh	heure	ss	secondes (facultatif)
mm	minute		

Seul le Super-Utilisateur peut configurer l'horloge du système.

OPTIONS

-d chaîne_date, --date=chaîne_date

Afficher la date et l'heure indiquées par la chaîne *chaîne_date*. L'affichage est fait avec le format par défaut, à moins qu'un argument commençant par un '+' ne fournisse un autre format

-f fichier, --file=fichier

Comme -d mais en utilisant successivement toutes les dates indiquées dans le fichier.

-r fichier, --reference=fichier

Afficher la date de dernière modification du fichier.

-R, --rfc-822

Afficher la date dans un format conforme à la RFC 822.

-s chaîne_date, --set=chaîne_date

Configurer l'horloge système avec la date et l'heure indiquées par *chaîne_date*, qui peut contenir éventuellement des noms littéraux de mois, de fuseau horaire, 'am' ou 'pm', etc...

-u, --utc, --universal

Afficher ou configurer l'heure en utilisant le Temps Universel (heure du méridien de Greenwich) plutôt que le temps local.

--help

Afficher un message d'aide sur la sortie standard, et se terminer normalement.

--version

Afficher un numéro de version sur la sortie standard, et se terminer normalement.

EXEMPLES

- Pour afficher la date de l'avant-veille : `date --date '2 days ago'`
- Pour afficher la date du jour à venir dans 3 mois plus un jour : `date --date '3 months 1 day'`
- Pour afficher le numéro du jour de Noël de l'année en cours : `date --date '25 Dec' +%j`
- Pour afficher la date actuelle dans un format incluant le jour du mois suivi du nom complet du mois : `date '+%d %B'`
- Il est parfois indésirable que les 9 premiers jours du mois soient complétés par un zéro. l'argument '%d' est responsable de ce complément sur deux chiffres. Par exemple ``date -d 1-may '+%d %B'`` donnera ``01 May``.
- Pour afficher la même chose sans le zéro de complément, on peut utiliser l'option non standard ``-``.
- `date -d 1-may '+%-d %B'`

ANNEXE 4 : Commande ftp

Lancer le programme ftp et se connecter à un serveur FTP

En ligne de commande, entrez au choix:

\$ ftp <nom_ou_ip_du_serveur_ftp>
\$ ftp ftp> open <nom_ou_ip_du_serveur_ftp>

Lorsque vous vous connectez à un site FTP, il vous est demandé un nom d'utilisateur (ou login) et mot de passe. Dans ftp vous pouvez vous identifier avec user <login> <mot_de_passe>

Explorer et voir la liste des fichiers

Pour afficher la liste des fichiers, utilisez la commande **ls**. Cette commande est exécutée sur le serveur distant, donc les options de la ligne de commande peuvent varier selon les serveurs. Les options les plus courantes de la commande **ls** sont en général utilisées, vérifiez la page de manuel de la commande **ls** pour plus de détails.

La commande **cd** permet de parcourir l'arborescence de votre racine sur le serveur ftp

Changer votre répertoire local : si vous voulez télécharger ou transférer des fichier qui ne sont pas dans votre répertoire local. Vous pouvez changer de répertoire local en utilisant la commande **lcd** :

exemple :

```
ftp> lcd /home/toto/
```

Local directory now /home/toto

Télécharger et uploader des fichiers

Téléchargez avec la commande **get** <nom_du_fichier>

Si vous souhaitez télécharger plusieurs fichiers à la fois, utiliser la commande **mget**. Par exemple : ftp> mget lu*.tar

Uploader avec la commande **put** <nom_de_fichier>

exemple : ftp> put toto.tar.gz

local: toto.tar.gz remote: toto.tar.gz

200 PORT command successful.

150 Opening BINARY mode data connection for toto.tar.gz.

226 Transfer complete.

10257 bytes sent in 0.00316 secs (3.2e+03 Kbytes/sec)

Si vous souhaitez uploader plus d'un fichier à la fois, vous pouvez utiliser la commande **mput**.

Options de FTP :

- i Désactiver l'interrogation interactive lors de transferts de fichiers multiples.
- n Empêcher **ftp** de tenter la « connexion automatique » lors de la connexion initiale. Si l'auto-connexion est activée, **ftp** recherchera dans le fichier **.netrc** (voyez [netrc\(5\)](#)) du répertoire personnel de l'utilisateur une entrée décrivant un compte sur la machine distante. S'il n'en trouve pas, **ftp** demandera le nom de connexion (login) sur la machine distante (par défaut l'identité de l'utilisateur sur la machine locale) et, si nécessaire, un mot de passe et un compte avec lesquels se connecter.
- g Désactiver le développement des métacaractères dans les noms de fichiers.
- v option bavarde (verbeuse).
- d Activer le mode de débogage.

Les commandes FTP usuelles sont les suivantes :

Commande	Description
help	Affiche l'ensemble des commandes supportées par le serveur FTP
status	Permet de connaître certains paramètres de la machine cliente
binary	Cette commande vous fait basculer du mode ASCII (envoi de documents textes) au mode binary (envoi de fichiers en mode binaire, c'est-à-dire pour les fichiers non texte, comme des images ou des programmes)
ascii	Bascule du mode binary au mode ascii. Ce mode est le mode par défaut
type	Permet d'afficher le mode courant de transfert (binary ou ascii)
user	Vous permet de réouvrir une session sur le site FTP en cours avec un nom d'utilisateur différent. Un nouveau mot de passe vous sera alors demandé
ls	Permet de lister les fichiers présents dans le répertoire courant. La commande "ls -l" donne des informations supplémentaires sur les fichiers
pwd	Affiche le nom complet du répertoire courant
cd	Cette commande signifie <i>change directory</i> , elle permet de changer le répertoire courant. La commande "cd .." permet d'accéder au répertoire de niveau supérieur
mkdir	Le commande <i>mkdir</i> permet de créer un répertoire dans le répertoire courant. L'utilisation de cette commande est réservée aux utilisateurs ayant un accès le permettant
rmdir	Le commande <i>rmdir</i> permet de supprimer un répertoire dans le répertoire courant. L'utilisation de cette commande est réservée aux utilisateurs ayant un accès le permettant
get	Cette commande permet de récupérer un fichier présent sur le serveur <ul style="list-style-type: none"> • Si la commande est suivie d'un nom de fichier, le fichier distant est transféré sur la machine locale dans le répertoire local en cours • Si la commande est suivie de deux noms de fichiers, le fichier distant (le premier nom) est transféré sur la machine locale dans le répertoire local en cours, avec le nom de fichier précisé (le deuxième nom) Si jamais le nom de fichier contient des espaces il faut veiller à le saisir entre guillemets
put	Cette commande permet d'envoyer un fichier local sur le serveur <ul style="list-style-type: none"> • Si la commande est suivie d'un nom de fichier, le fichier local est transféré sur le serveur dans le répertoire distant en cours • Si la commande est suivie de deux noms de fichiers, le fichier local (le premier nom) est transféré sur le serveur dans le répertoire distant en cours, avec le nom de fichier précisé (le deuxième nom) Si jamais le nom de fichier contient des espaces il faut veiller à le saisir entre guillemet
open	Ferme la session en cours et ouvre une nouvelle session sur un autre serveur FTP
close	Ferme la session en cours, en laissant le logiciel FTP client actif
bye	Déconnecte le logiciel client du serveur FTP et le met en état inactif
quit	Déconnecte le logiciel client du serveur FTP et le met en état inactif

Pour aller plus loin : Le fichier .netrc

Le fichier .netrc doit être créé dans votre \$HOME il s'agit d'un fichier de configuration de FTP qui évite d'entrer systématiquement son LOGIN et PASSWORD. Les clients FTP sous Unix supporte généralement le fichier ".netrc". Le client `ftp` de MS-Windows généralement pas. Votre client FTP doit gérer l'utilisation de ".netrc" de lui-même, mais si le fichier ".netrc" existe, un transfert FTP ne demandera pas de nom d'utilisateur ni de mot de passe.

La syntaxe typique des lignes d'un fichier ".netrc" est donnée ci-dessous.

```
machine {NomCompletdelHôte} login {NomUtilisateur} password {MotDePasse}
default login {NomUtilisateur} password {MotDePasse}
```

NOTE : Comme ce fichier contient des mots de passe, assurez-vous que personne d'autre ne peut le lire ! La plupart des programmes refuseront d'utiliser un fichier ".netrc" que les autres peuvent lire. N'oubliez pas que l'administrateur du système peut tout de même lire votre fichier !