

# Impression

Sous Linux, l'accès aux imprimantes ne se fait qu'au travers d'un **spooler d'imprimante**. Un spooler est un gestionnaire de file d'attente, ou « queue », dans laquelle sont placés les travaux d'impression. A chaque imprimante est associée une file d'attente. Linux envoie ces travaux à l'imprimante concernée dans l'ordre d'arrivée. Cette façon de faire permet de résoudre les conflits d'accès au cas où plusieurs utilisateurs tentent d'accéder à la même imprimante au même moment. Le processus d'impression proprement dit s'effectue en arrière-plan, ce qui permet à l'utilisateur de reprendre la main immédiatement, sans avoir à attendre la fin de son impression. Nous n'entrerons pas dans les détails de la création d'une file d'attente (ceci est du ressort de l'administrateur), mais nous présenterons les commandes permettant à l'utilisateur de manipuler une file d'attente d'impression : envoyer un ou plusieurs travaux (*lpr*), consulter l'état de la file d'attente (*lpq*), supprimer un ou plusieurs travaux de la file d'attente (*lprm*). Un travail d'impression peut être un fichier ou la sortie standard d'une commande.

## 9.1 IMPRESSION

### 9.1.1 Imprimante par défaut

La commande *lpr* permet d'envoyer un travail d'impression dans une file d'attente. Utilisée sans aucune option, *lpr* utilise l'imprimante par défaut :

```
xstra> lpr file1
xstra> ls -l /etc | lpr
xstra> lpr file1 file2 file3
```

### 9.1.2 Choix de l'imprimante

Il est possible que plusieurs imprimantes soient connectées au système. Dans ce cas, il est nécessaire de désigner l'imprimante de destination par l'option `PNom` où `Nom` spécifie le nom de l'imprimante.

```
| xstra> lpr Pbigbrother file1 file2
```

Cette commande envoie `file1` et `file2` à l'imprimante nommée `bigbrother`. Le nom des imprimantes est déterminé par l'administrateur du système. La variable d'environnement `PRINTER`, si elle existe, est utilisée par `lpr`, `lpq` et `lprm` pour désigner l'imprimante destination.

### 9.1.3 Options

La commande `lpr` accepte d'autres options dont les plus courantes sont :

- #num      imprimer en `num` exemplaires.
- m         Mail : le spooler notifie la fin du travail d'impression en envoyant un mail à l'utilisateur.
- p         Print : procède à une mise en forme identique à la commande `pr` (voir plus loin).
- r         Remove : supprime le fichier après l'avoir imprimé (pratique si le fichier à imprimer est temporaire).
- s         Symbolique : plutôt que de copier le fichier à imprimer dans le répertoire du spooler, `lpr` crée dans ce répertoire un lien symbolique sur le fichier à imprimer. Utile si le fichier à imprimer est très grand. Attention : ne pas modifier ou supprimer le fichier avant la fin de l'impression

## 9.2 ÉTAT DU SPOOLER D'IMPRIMANTE

La commande `lpq option [travaux] [utilisateur]` indique l'état du spooler d'imprimante. Sans option ni argument, indique l'état de tous les travaux soumis par l'utilisateur. Cette commande permet de retrouver l'identification de chaque travail, identificateur utilisable dans `lprm`. Les options suivantes sont acceptées par `lpq` :

- l         Long : donne plus d'information sur chaque travail.
- Pnom     Printer : spécifie une imprimante particulière (nom). `lpq` utilise aussi la variable d'environnement `PRINTER`.
- a         Liste l'ensemble des imprimantes disponibles et leur état.

### Exemple

```
| xstra> lpq -a | grep Printer
Printer: laser2@vivaldi (dest laser2@ljt.u strasbg.fr)
|xstra>
```

## 9.3 SUPPRESSION D'UN TRAVAIL

Il est parfois nécessaire de supprimer un travail déjà placé dans une file d'attente. Cette fonction est réalisée à l'aide de la commande `lprm job` où `job` est l'identification du travail d'impression dans le spooler fournie par `lpq`.

### Exemple

```
xstra> lpr f*
xstra> lpq
Rank Owner Job Files Total Size
1st colin 26 f1, f2, f3 5228 bytes
xstra> lprm 26
$ Suppression de l'impression de f1, f2, f3
```

Les options suivantes sont acceptées par `lprm` :

- `lprm` supprime tous les travaux d'impression appartenant à l'utilisateur. Souvent utilisé.
- `PNom` Printer : spécifie une imprimante particulière (Nom). `lprm` utilise aussi la variable d'environnement `PRINTER`.

## 9.4 MISE EN FORME

La commande `pr` peut être associée à `lpr` pour une mise en page du texte. Par défaut, `pr` produit des pages de 66 lignes précédées d'une en-tête de 5 lignes comportant la date, le nom du fichier imprimé et le numéro de page. `pr` peut aussi paginer le texte en plusieurs colonnes. La commande usuelle est :

```
xstra> pr file1 | lpr $ première solution
xstra> lpr -p file1 $ solution équivalente
```

## 9.5 IMPRESSION POSTSCRIPT

Pour imprimer sur une imprimante PostScript récente, la commande `lpr` convient dans tous les cas : l'imprimante détecte elle-même si le document est PostScript ou ASCII et l'imprime correctement. Si l'imprimante n'accepte que le Postscript, la commande `lpr` ne peut être utilisée que si le document à imprimer est déjà au format PostScript. Pour imprimer de l'ASCII, la commande `a2ps` remplacera `lpr`.

```
xstra> ls -l /etc | a2ps
[ stdin (plain): 3 pages on 2 sheets]
[ Total: 3 pages on 2 sheets] sent to the default printer
[ 1 line wrapped]
```

La commande `a2ps` (Ascii to PostScript) de GNU fait beaucoup plus que ne l'indique son nom. C'est un filtre très général de production de document PostScript

à partir de nombreux formats d'entrée, et qui comporte de nombreuses possibilités de mise en page. En pratique, avec une imprimante PostScript, il est préférable de n'utiliser que *a2ps*.

Les options les plus courantes de *a2ps* sont :

2	2 pages par feuille en mode paysage (défaut)
r	1 page par feuille en mode paysage
1	1 page par feuille en mode portrait
3	3 pages par feuille en mode portrait
P Nom	Printer : spécifie une imprimante particulière (Nom).
L'option P	reconnaît trois noms d'imprimantes spéciales :
void	désigne <i>/dev/null</i> et permet de savoir combien de pages seront imprimées (ou de tester la commande).
display	désigne Ghostview, le visualiseur PostScript. Si Ghostview n'est pas installé, <i>a2ps</i> produira une erreur.
file	désigne un fichier régulier ( <i> fich.c</i> sera imprimé dans le fichier <i> fich.ps</i> ).

## 9.6 EXERCICES

### Exercice 9.5.1

Le répertoire courant contient, entre autres, les fichiers : *bibal.h* *bibal.c* et *README*.

Pour imprimer ces trois fichiers en deux exemplaires, deux possibilités existent : imprimer chaque fichier en deux exemplaires ou imprimer en deux exemplaires l'ensemble des trois fichiers.

Quelles sont les commandes correspondantes ?

### Exercice 9.5.2

Comment savoir combien de feuilles serait nécessaire pour imprimer, à raison de deux ou trois pages par feuille, le résultat de la commande « `ls -l /usr/bin` » sur une imprimante PostScript ?

## Chapitre 10

# Gestion de l'espace disque

### 10.1 « FILE SYSTEM »

#### 10.1.1 Organisation des « file systems »

L'**organisation du système de fichiers** est une organisation logique basée sur une organisation physique constituée d'un ou plusieurs disques. Chacun est divisé en une ou plusieurs partitions physiques (P1, P2, etc.). L'espace disque physique disponible est ainsi découpé en disques logiques. Initialiser un disque logique consiste à créer sur ce disque un système de fichiers appelé *file system*.

Sous Unix/Linux, l'utilisateur ne voit pas les disques physiques, pas plus que les partitions. Il n'y a pas de lettres d'unités. L'arborescence de fichiers est construite au démarrage du système à partir de plusieurs *file systems*. Un *file system* particulier, le **root file system** contient la racine de l'arborescence, le noyau, les fichiers systèmes, et des répertoires vides (*/tmp*, */usr*,...) sur lesquels seront greffées des arborescences de fichiers se trouvant dans d'autres *file systems*. (Fig. 10.1).

Lors du démarrage d'une machine, après recherche du **disque système**, chaque *file system* est rattaché, à l'aide de la commande *mount*, à un répertoire existant appelé **point de montage** (*mount point*). La racine de ce *file system* prend alors pour nom celui du point de montage, et l'arborescence de fichier qu'il contient est alors totalement intégrée à l'arborescence de fichiers du système.

**La structure d'un file system** est globalement la même dans toutes les versions d'Unix/Linux : chaque *file system* est composé d'un grand nombre de blocs contigus. La structure de base d'un *file system* est détaillée à la figure 10.2.

- Le bloc 0 contient le boot et l'identification du disque.
- Le bloc 1, appelé aussi superbloc, contient des informations sur le *file system* :