

DAEMON D'IMPRESSION LPR

Félix Hauri

avril 2004

Programme

- Introduction
- Documentation
- Principe
- Connexions
- Deuxième étape : La file d'attente
- LPR-BSD (lpr-ppd) Configuration
- Filtres
- GNU Ghostscript
- Pratiquement
- Epilogue
- Questions

Introduction

Raccorder une imprimante à un système Un*x est l'une des tâches les plus complexes à assumer pour un administrateur.

1. Cheminement des données.
2. Format des données.
3. Convergence et files d'attente.

Introduction Raccorder une imprimante à un ordinateur peut sembler simple, pourtant cette tâche a une toute autre réputation sous Un*x. Pour que cela se passe correctement, pas moins de trois notions doivent être parfaitement maîtrisées.

En effet, une imprimante est un robot qui dessine à la demande, selon un langage particulier. Si le langage utilisé n'est pas compris par l'imprimante, alors deux cas peuvent se présenter : soit l'imprimante interprète les ordres de manière erronée et fait *n'importe quoi* (dessins bizarres, débiter des dizaines de pages, etc.) soit l'imprimante n'interprète rien et alors, ne fait rien du tout.

Documentation

En plus de la documentation habituelle : *manpages*, *infopages*, *howtos*, */usr/share/doc*, etc. Retenez l'URL suivante :

`http://www.linuxprinting.org/ voire`

`http://www.linuxprinting.org/printer_list.cgi`

linuxprinting.org Une base de donnée interactive vous permettra de trouver la meilleure configuration pour la plupart des imprimantes actuelles...

Cette base peut être enrichie de vos propres expériences.

Documentation complémentaire

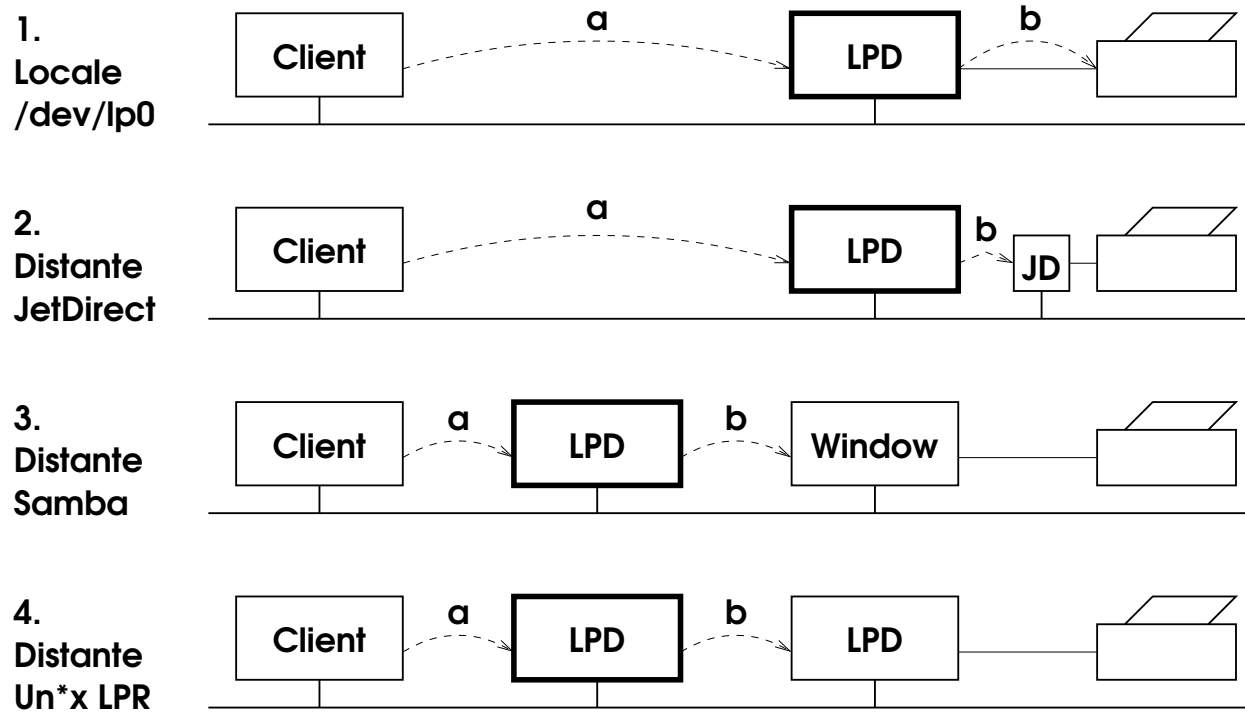
RTFM de l'imprimante !

Voire surtout les *caractéristique techniques*.

Les imprimantes (toutes à ma connaissance ;-)) sont fournies avec un minimum de documentation.

Il faut la lire. Tout particulièrement le chapitre *caractéristiques techniques* dans lequel on doit pouvoir trouver un voire plusieurs *modes de compatibilités* ou plus simplement *langage d'impression*.

Principe



Dans les faits, deux cas distincts : Imprimante locale ou distante

Le schéma de principe met en évidence trois étapes. à savoir :

1. *liaison a* L'acheminement des données depuis les applications clientes vers le serveur d'impression,
2. *liaison b* L'acheminement des données depuis le serveur d'impression vers les imprimantes.
3. *LPD* Pour Line Printer Daemon, le système de gestion des queues.

D'autre part, on voit ressortir de ce schémas, deux principaux cas de figure,

- imprimante locale,
- impression via réseau.

En effet, si dans le cas d'une imprimante locale, on cherchera à faire réagir l'imprimante, dans le cas d'imprimante réseau, il faudra interroger les serveurs, dans la mesure du possible, c'ad : observer les diodes sur un boitier JetDirect, ou l'affichage de la queue de l'imprimante pour un serveur avec écran.

Problème récurrent

Si une imprimante ne réagit pas :

- Est-ce le langage utilisé qui ne correspond pas,
- est-ce le cheminement des données n'est pas correct ou
- est-ce le daemon de queue qui ne fonctionne pas ?

Il n'est pas très utile de se demander quel *driver* (langage d'impression) utiliser pour faire enfin réagir cette |censuré| imprimante si celle-ci ne reçoit aucune des données que l'on s'acharne à lui soumettre...

D'un autre point de vue, on pourra facilement faire mentir ma phrase d'introduction pour peu que l'on aborde le problème avec méthode, afin de ne travailler que sur des certitudes.

C'est pourquoi nous commencerons pas la fin, la liaison *b*.

Connexions

Premièrement : Assurer le contact avec l'imprimante.

- Port parallèle (standard),
- Port série (vieux),
- Port USB (série nouveau),
- Réseau Un*x (lpr),
- Réseau (HP JetDirect™),
- Réseau Window™ (samba),
- Réseau Macintosh™ (netatalk).
- Réseau Novell (NetWare™).

Communication avec l'imprimante Historiquement, les imprimantes recevaient leurs ordres depuis un port série RS-232 ou parallèle uni-directionnel (centronics).

Rapidement le besoin de partager ce type de ressources a fait apparaître des réponses chez un peu tous les *fabricants de réseaux*.

Actuellement, l'acheminement des données se fait le plus couramment par un port USB ou parallèle (bi-directionnel) pour les petites configurations et au moyen d'une interface réseau de type HP-JetDirect pour des configurations moyennes.

La première chose à faire sera de *connaître* l'imprimante à raccorder et au moins un moyen simple de la faire réagir.

La plupart des imprimantes étaient fabriquées avec un buzzer permettant d'avertir l'utilisateur en cas de problèmes...

La plupart des imprimantes fonctionnent aussi en mode texte. Dans ce cas et si buzzer il y a, alors le fait d'envoyer un *bell* (Ctrl-G = ASCII 007) fera émettre un bip par l'imprimante.

Si pas de beep, alors essayer le caractère *form feed* (éjecter page = Ctrl-L = ASCII 014 en octal)

Sinon, il faudra faire réagir l'imprimante en respectant son langage d'impression.

Page de test

1. Créer une page PostScript, avec mpage, a2ps, enscript ou \LaTeX :

- a) `latex testpage` , puis `a4paper` et `<return>`.
- b) `dvips -ta4 -f <testpage.dvi >tstpage.ps`

2. `gs -h`

pour une liste de *devices* (langages) supportés

3. `gs -q -sDEVICE=??? -sOutputFile=tstpage.raw \
-dNOPAUSE -dBATCH -dSAFER tstpage.ps -c quit`

Port parallèle Communiquer via le port parallèle nécessite la présence des modules “parport” et “lp”, dans le kernel.

En principe, l'action suivante :

```
echo -en "\\007 >/dev/lp0
```

... fait émettre un beep par l'imprimante,

Sinon utiliser le fichier *tstpage.raw* :

```
cat tstpage.raw >/dev/lp0
```

Les port parallèles actuels fonctionnent en mode “bi-directionnel”, ce qui permet de recevoir des informations de l'imprimante : Marque, type, capacités, ou défaut comme manque d'encre ou de papier, etc...

Pour profiter de ces possibilités, il faudra ajouter un kernel, le support *IEEE 1284*.

Port série Le port série est configuré par défaut dans le kernel Linux, ce port n'est pas très rapide, aujourd'hui, on lui préférera le port USB.

Il s'utilisera de la même manière :

```
echo -en "\\007 >/dev/ttyS0
```

ou

```
cat tstpage.raw >/dev/ttyS0
```


Port USB Le port série *nouveau*, avec des possibilités de transfert assez rapide et un moyen de raccorder plusieurs périphériques au même port (adressage).

Pour utiliser cette connexion, le support USB est nécessaire, avec USB-PRINTER.

```
echo -en "\\007" >/dev/usb/lp0
```

ou

```
cat tstpage.raw >/dev/usb/lp0
```

A noter qu'un port USB est plus rapide qu'un port parallèle.

Réseau lpr Le protocole LPD de Berkeley, bien que relativement simple, s'utilise au travers de clients comme *rlpr* ou une configuration LPD idoine.

Pour imprimer sur la queue d'impression *Queue* du serveur d'impression *Serveur* je peut utiliser :

```
rlpr -HServeur -PQueue tstpage.raw
```

Réseau JetDirect™ Très basique aussi, il s'agit tout simplement du port IP numéro 9100 qui est à l'écoute (*LISTENING*), en attente de données à imprimer.

Il existe des scripts perl du type *JetDirect.pl* ou autrement un outil à tout faire comme *netcat*

```
JetDirect.pl <tstpage.raw (Après configuration de l'outil)
```

ou

```
nc -q0 Serveur 9100 < tstpage.raw
```

Réseau samba Pour utiliser une imprimante *partagée* sur un réseau *samba*, il faut utiliser les outils *samba*...

```
smbclient //SERVER/PRINTER -A credential_file -c 'print -' <tstpage.raw
```

Réseau netatalk Pour imprimer au travers d'un réseau *netatalk*, on utilisera les outils *netatalk*

```
pap -p Server:LaserWriter@Zone < tstpage.raw
```

Réseau NetWare™ Enfin, imprimer via une queue NetWare sera réalisable avec l'aide du paquet *ncpfs*

```
nprint -S server -U user_name -P password -q queue_name -d <tstpage.raw
```

Deuxième étape : La file d'attente

Afin d'éviter les conflits en environnement multi-tâche et multi-utilisateur, un daemon spécialisé servira d'intermédiaire entre les applications et les imprimantes.

LPR BSD lpr/lpd line printer spooling system

GNUlpr GNU lpr printing system (basé sur BSD)

LPRng lpr/lpd printer spooling system

PDQ Print Don't Queue, Queueless printing system.

CUPS Common UNIX Printing System™

LPR Le plus *basique*, le daemon LPR de BSD (OpenBSD) permet essentiellement de gérer et comptabiliser des queues, appliquer des filtres pour des imprimantes locales (*/dev/lp0*, */dev/ttyS0*, */dev/usb/lp0*, etc) ou rediriger des impressions vers des serveurs d'impression BSD...

C'est tout... à priori...

LPR-BSD (lpr-ppd) Configuration

Trois axes de configuration : Entrées, queues, sorties.

- */etc/hosts.lpd* liste des clients autoriser à imprimer
- */etc/hosts.equiv* liste des clients autoriser à administrer
- */var/spool/lpd* répertoire principal des répertoires de queues
- */etc/printcap* fichier de configuration des queues et sorties
- */dev/???* Les sorties elles-même, qu'il faut s'assurer de
`chown root:lp /dev/lp0;chmod 660 /dev/lp0`

LPR-BSD (lpr-ppd) Le fichier *printcap*

Quelques paramètres du fichier */etc/printcap* :

sd Spool Directory, un sous-dir de */var/spool/lpd*.

lf Log File, journal des événements.

lp Line Printer, périphérique de sortie (*/dev/lp0*).

rm Remote Machine, nom du serveur distant.

rp Remote Printer, nom de la queue d'impression.

af Accounting Filter, Filtre de comptabilisation.

if Input Filtre, Filtre d'entrée.

sh Suppress Header, suppression des pages d'entêtes.

mx Maximum File Size. (0=non limité).

La syntaxe de */etc/printcap* est relativement simple.

Chaque queue est représentée sur une ligne, les champs sont séparés par des deux-points “:”, le premier champ est le nom de la queue. Plusieurs noms (alias) peuvent être séparés par une barre verticale.

Les fins de lignes peuvent être échappés afin de faire suivre une longue définition sur plusieurs lignes.

La sortie d'une queue se fait soit par *lp* pour une imprimante locale, soit par *rm*, ET *rp* pour une queue sur un serveur d'impression.

LPR-BSD (lpr-ppd) Outils

En exploitation, nous recourrons à

lpr Line Printer : Commande générale d'impression.

lpq Line Printer Query : Etat des files d'attente.

lprm Line Printer ReMove : Suppression d'un job.

lpc Line Printer Control :

start autoriser les impressions sur une queue et démarrer un démon d'impression.

stop arrêter un démon d'impression et désactiver une queue.

status afficher l'état d'une ou plusieurs queues.

help demander de l'aide concernant une commande.

Il est à noter que lorsque la sortie est un serveur d'impression, les filtres ne sont pas appliqués.

Pour *serveur d'impression*, il faut comprendre *serveur d'impression LPR*.

Pour tous les autres cas de figures, il faudra recourir à une *astuce* :

Astuce LPD

Afin de rediriger la sortie sur un script *exécutable*, en guise de *device* on utilise */dev/null* comme *destination finale*. Le champ *if* pour le nom du script et le champ *af* pour des arguments exceptionnels :

```
lp|Imprimante par défaut|printer0:\
    :sd=/var/spool/lpd/lp:\
    :sh:mx#0:\
    :lp=/dev/null:\
    :if=/var/spool/lpd/lp/filter.sh:\
    :af=options:
```

Le script contiendra simplement une commande permettant d'adresser l'imprimante via un réseau :

```
#!/bin/sh cat - |\  
  /usr/bin/netcat -q0 Ip_Ou_Nom_De_L_imprimante 9100
```

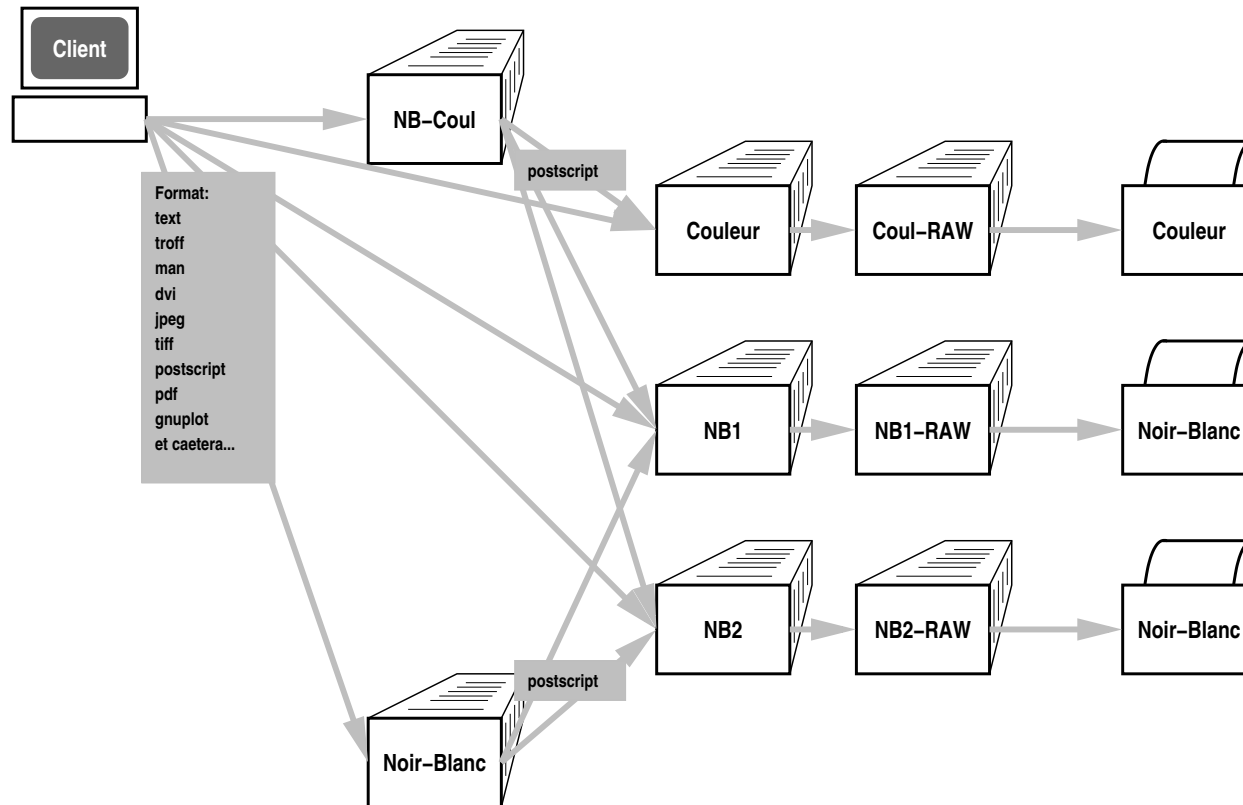
Il sera lancé de l'une des manière suivante :

```
/var/spool/lpd/lp/commande -x width -y length -n login -h host acct-file  
ou
```

```
.../commande [-c ] -w width -l length -i indent -n login -h host acct-file
```

Où les informations de largeur et de longueur sont à relativiser... on préférera utiliser des filtres ré-entrants puis compter et dimensionner les pages sortantes avec des outils PostScript avancés.

Exemple d'application



exemple Dans cet exemple, les clients accèdent aux imprimantes à choix :

NB-Coul Imprimante virtuelle économique qui choisira sa destination : première noir-blanc disponible pour les pages noir-blanc, et couleur pour les pages en couleur.

Couleur Imprimante couleur.

Noir-blanc Première imprimante noir-blanc disponible.

NB1 Imprimante noir-blanc numéro 1.

NB2 Imprimante noir-blanc numéro 2.

Le fichier */etc/printcap* pourrait ressembler à :

```
Coul-RAW:\
```

```
:sd=/var/spool/lpd/Coul-RAW:\
```

```
:sh:mx#0:\
```

```
:rm=ImprimanteCouleur:\ :rp=lp:
```

```
NB1-RAW:sd=/var/spool/lpd/NB1-RAW:sh:mx#0:rm=ImprimanteNoirBlanc:rp=lp:
```

```
NB2-RAW:sd=/var/spool/lpd/NB2-RAW:sh:mx#0:rm=ImprimanteNB2:rp=lp:
```

```
Couleur:\
```

```
:sd=/var/spool/lpd/Couleur:\
```

```
:sh:mx#0:\
```

```
:lp=/dev/null:\
```

```
:if=/usr/local/bin/postscript2Epson-Escp2:\
```

```
:af=Coul-RAW:
```

```
NB1:sd=/var/spool/lpd/NB1:sh:mx#0:lp=/dev/null:\
```

```
:if=/usr/local/bin/ps2pcl:af=NB1-RAW:
```

```
NB2:sd=/var/spool/lpd/NB2:sh:mx#0:lp=/dev/null:\n    :if=/usr/local/bin/ps2pcl:af=NB2-RAW:
```

```
Noir-Blanc:\n    :sd=/var/spool/lpd/Noir-Blanc:\n    :sh:mx#0:\n    :lp=/dev/null:\n    :if=/usr/local/bin/lpwrapper:\n    :af=NB1,NB2:
```

```
NB-Coul:\n    :sd=/var/spool/lpd/NB-Coul:\n    :sh:mx#0:\n    :lp=/dev/null:\n    :if=/usr/local/bin/colorsplitter:\n    :af=Noir-Blanc,Couleur:
```

LPRng

LPRng est une autre implémentation du même protocole, un peu plus avancée.

lpstat Interrogation de la queue.

lprngtool Outil graphique de configuration de printcap (X nécessaire).

checkpc Test du fichier */etc/printcap*

lp Envoi de plusieurs requêtes pour un *job*

cancel Envoi une requête d'abandon à une queue.

LPRng présente des méthodes complexes pour gérer les comptes et facturations des jobs d'impression.

Les concepteurs ont cherché à réaliser un outil souple et sécurisé.

Développé conjointement au filtre *IFHP* il se repose sur ce dernier pour les options de lancement de Ghostscript.

Globalement il fonctionne de la même manière que LPR-BSD, avec une quantité d'options plus avancées essentiellement concernant la comptabilisation des jobs.

PDQ Print, Don't Queue !

- Pas de queues, juste des process relancés pendant une durée déterminée.
- Configuration via */etc/pdq/printrc* et/ou */.printrc*
- Les outils :
 - *pdq* - Imprime un ou plusieurs fichiers
 - *xpdq* - Utilitaire graphique pour impression, gestion des jobs et des configurations d'imprimantes
 - *lpd_print* - Envoie un fichier sur une queue lpd distante
 - *lpd_status* - Affiche l'état d'une queue lpd distante
 - *lpd_cancel* - Supprime une impressions d'une queue lpd distante

Autre approche, issue d'une lassitude à voir imprimer le lundi matin des jobs lancés le vendredi...

PDQ Présente l'avantage de permettre l'ajout d'imprimante sans nécessiter les droit *root*. De plus, son outil graphique *xpdq* permet de guider l'utilisateur pour l'ajout d'imprimantes.

xpdq utilise la librairie graphique *GTK* (pour *Gimp ToolKit*) ce qui est intéressant puisque Ghostscript utilise *Gimp* pour post-traiter les impressions.

Globalement, PDQ est une solution intéressante pour des configurations *desktop*.

A noter que pour juste éviter l'installation d'un serveur LPD, *rlpr* est un outil qui permet d'accéder à des queues LPD distantes.

CUPS - Common UNIX Printing System™

- IPP - Support de *Internet Printing Protocol*
- LPR - Support de BSD-LPR via *compatibility server*
- config. via */etc/cups/cupsd.conf* (syntaxe inspirée de Apache)
- Configuration *graphique* via navigateur web
- Configuration également en ligne de commande.
- Divers outils avancés comme :
 - cups-calibrate - calibration des couleurs
 - lpoptions - options passées à *lpr* ou *lp* par défaut
 - lppasswd - dérivé de *htpasswd*
 - lpadmin - inspiré de *System V Print Subsystem*

CUPS™ Nouveau *monstre sacré* du monde *Linux*, CUPS est un serveur *complet*, qui permet de gérer intégralement les configuration, manipulation, associations de queues d'impression.

Il s'agit d'un moteur très complet qui se configure et s'utilise via une interface web. Par défaut, après l'installation, on accède à son environnement avec l'aide d'un navigateur standard, en tapant l'url : `http://localhost:631/`

La configuration web est facile et conviviale.

La configuration du moteur web est confinés dans le fichier `/etc/cups/cupsd.conf`, dont la syntaxe ressemble à celle du fichier `/etc/apache/httpd.conf`.

Pour imprimer sur un serveur cups, le client doit être autorisé via une directive *Allow from adrs.ip.du.client*.

Il intègre un mécanisme de *classe* d'imprimante qui permet de distribuer les jobs sur un groupe d'imprimantes, en fonction de leur disponibilité.

Afin de l'utiliser au mieux, une version dérivée de Ghostscript permet de créer un format bitmap intermédiaire que les filtres *ESP* pourront retraiter.

Afin d'accéder à certaine imprimantes particulières, cups utilise des filtres propriétaires, qui sont à acquérir auprès de *Easy Software Products* <http://www.easysw.com>

Pour indiquer une destination, CUPS utilise des *Uniform Resource Identifiers* :

CUPS - Syntaxe URI

- parallel:/dev/lp0
- serial:/dev/ttyS1?baud=38400+size=8+parity=none+flow=soft
- usb:/dev/usb/lp0
- scsi:/dev/scsi/sc1d6l0
- socket://hostname :port
- tftp://hostname/path
- ftp://hostname/path
- http://hostname[:port]/path
- ipp://hostname/path
- smb://hostname/printer

Les outils :

cupsd (8) - common unix printing system daemon
lpinfo (8) - show available devices or drivers
lpadmin (8) - configure cups printers and classes
lp (1) - print files
lpstat (1) - print cups status information
lpmove (8) - move a job to a new destination
cups-calibrate (8) - ESP Printer Calibration Tool
lppoptions (1) - display or set printer options and defaults
lppasswd (1) - add, change, or delete digest passwords.
cupsaddsmb (8) - export printers to samba for windows clients
accept, cupsaccept, cupsreject,
reject (8) - accept/reject jobs sent to a destination
cupsdisable, cupsenable, disable,
enable (8) - stop/start printers and classes
cancel (1) - send cancel requests to an LPRng print service
lpc (8), lpr, lpr,
lprm (1) - Compatible LPR-BSD commands

Filtres

```
+-----+      +-----+
Tout > filtre > ps > ghostscript > Imprimante
+-----+      +-----+
```

- Vers PostScript : a2ps, mpage, enscript, Automagic, magicfilter, etc.
- PostScript vers imprimante simple : Ghostscript, et uni-print
- PostScript vers imprimante avancé : Ghostscript + gimp, stp, ijs, hpijs.

Les Filtres Actuellement les possibilités graphiques et la rapidité des imprimantes font que nous renoncerons à les utiliser en *mode texte*, bien que théoriquement nettement plus rapide, cet avantage tend à devenir négligeable en regard de son manque de souplesse.

Nous chercherons donc à transformer nos données dans le langage graphique de notre imprimante (ce travail est à faire pour chaque différent type de matériel).

Pour ce faire, nous utiliserons GNU-Ghostscript qui se chargera de la dernière (ou avant dernière dans certains cas) conversion.

Ghostscript est un interpréteur PostScript auquel nous devons donc fournir du format PostScript.

La plupart des applications *finies* proposent d'imprimer les données dans ce format. Elles fonctionneront donc naturellement dans notre environnement.

Pour le reste, une quantité impressionnante de filtres spécialisés, de *troff* pour les manpages, à *html2ps* pour imprimer des URLs en passant par *jpeg2ps* pour des posters ou des images imbriquées ou encore *dvips* pour imprimer le présent cours...

Une commande GNU *file* permet de déterminer la nature d'un flux de données.

Il existe d'ailleurs plusieurs *intégrés* qui permettent d'automatiquement appliquer les filtres en fonction de la nature des données.

Quelque filtres et outils :

- `apsfilter` - Filtre *Magic* avec reconnaissance automatique.
- `magicfilter` - Filtre *Magic* avec reconnaissance automatique.
- `printfilters-ppd` - Filtre *Magic* avec reconnaissance automatique.
- `psutils` - Collection d'outils permettant de manipuler des fichiers `postscript`.
- `foomatic` - Base de données d'imprimante + outils de configuration.

La commande :

```
foomatic-configure -n BJC8200 -c smb://user:passwd@host/lpt1 \  
-p 123776 -d gimp-print  
créera dans /etc/printcap
```

```
BJC8200 | | Canon BJC-8200 | : \  
:sd=/var/spool/lpd/new: \  
:ppdfile=/etc/foomatic/new.ppd: \  
:lf=/var/log/lp-errs: \  
:lp=/dev/null: \  
:af=/etc/foomatic/lpd/new.lom: \  
:if=/usr/sbin/lpdomatic: \  
:sh: \  
:mx#0:
```

Avec quelque variante pour le cas où le daemon serait LPRng au lieu de LPR.

GNU Ghostscript

```
# gs -sDEVICE=bjc600 -dSAFER -dNOPAUSE -dBATCH \  
-q -sOutputFile=- -sPAPERSIZE=a4 - -c quit
```

- -c.. -d.. -s.. L'indice du paramètre : -c commande, -d drapeau, -s setting=val.
- -sOutputFile= Permet de préciser une destination.
- -dNOPAUSE Evite d'attendre que l'utilisateur appuie sur la touche "Return" entre chaque page.
- -dSAFER Evite l'accès en écriture sur les fichiers soumis.
- -c quit Quitte l'interpreteur.
- -dBATCH Lancé par un script, gs quittera après exécution.

- `-sDEVICE=` Sélection du mode de sortie.
- `-sOutputFile=` Nom du fichier de sortie plutôt que la sortie standard
- `@driver.upp` Raccourcis vers le driver “uniprint”.

Ghostscript est un monument. Sa documentation est conséquente aussi je ne m'attarderais pas. Attention cependant :

-sOUTPUTFILE NE MARCHE PAS ! La casse des paramètres et options DOIT être respectée.

La ligne de commande est lue séquentiellement par ghostscript.

`gs truc.ps -c quit` affichera `truc.ps` puis quittera l'interpréteur, tandis que `gs -c quit truc.ps` ne fera rien du tout.

uniprint ou @ Mode de sortie unifiée :

```
gs -sDEVICE=uniprint @bjc610a1.upp truc.ps
```

aura le même effet que :

```
gs @bjc610a1.upp truc.ps
```

La présence du "@" implique "uniprint".

Ghostscript quelque options

- *-rXX* résolution par (défaut 72 dpi)
- *-gXXxYY* taille de l'image en sortie.
- *-q* Be quiet. Silencieux conjugué à :
- *-sOutputFile=-* Utilise la sortie standard.
- Quelques DEVICES :
 - *X11* est le DEVICE par défaut.
 - *pnm* pour des images à manipuler avec *NetPBM*
 - *pdfwrite* converti en PDF (Portable Document File).

Pratiquement

Les imprimantes disponibles en salle de cours sont relativement représentatives : L'une comprend directement le postscript, l'autre ne fonctionne qu'en mode *PCL3*.

HP 4000 N	laser	Postscript, HP-PCL
HP 895Cxi	jet d'encre couleur	HP-PCL

concretement Nous allons nous servir de ces imprimantes pour comprendre les manipulations de queues et de filtres.

Pour imprimer une page sur l'imprimante HP 4000, nous n'aurons pas besoin de recourir à Ghostscript du fait que cette imprimante comprend le langage Postscript. Nous pourrions donc nous contenter d'envoyer notre page de test postscript directement sur l'imprimante :

```
nc -q0 adresse_ip_4000N 9100 <testpage.ps
```

Si tout se passe bien, nous pourrions créer notre queue.

A moins que la documentation du *serveur d'impression HP JetDirect* ne puisse nous indiquer un nom de queue LPR, nous utiliserons le port 9100.

A noter que plutôt que le script *script.sh* et *lp=/dev/null* nous devrions pouvoir utiliser *:rm=adresse_ip_4000N :rp=raw*.

Les noms de queue LPR sont souvent *raw*, *lp*, *lpt1* ou encore *printer...* La documentation est parfois utile.

```
root@debian # mkdir /var/spool/lpd/HP4000
```

```
root@debian # chown lp: $_
```

```
root@debian # chmod 2775 $_
```

```
root@debian # cd $_
```

```
root@debian # cat <<eof >script.sh
#!/bin/sh
nc -q0 adresse_ip_4000N 9100
eof
root@debian # chown lp: script.sh
root@debian # chmod 754 $_
root@debian # cat <<eof >>/etc/printcap
HP4000|HP 4000N|laser|lp:
    :sd=/var/spool/lpd/HP4000:\
    :sh:mx#0:\
    :lp=/dev/null:\
    :if=/var/spool/lpd/HP4000/script.sh:
eof
root@debian # lpr /tmp/testpage.ps      # pour finir.
```

Notre imprimante couleur n'est pas postscript !

stp, ijs, hpijs Là, ça se complique encore un peu : Le problème tiend dans la nature des encres censées être des primitives capable de composer toutes les couleurs...

Dans les fait, savant mélange chimique hautement sophistiqué, ces encres ne sont malheureusement pas exactes. De plus. les gouttelettes d'encres ne sont pas carrées et donc ne peuvent se conjuguer afin de créer un obscurcissement homogène.

Bref, traduire une couleur 24 bits en dispositions de gouttelettes d'encre en 4 couleurs est un calcul terriblement complexe, variant d'une qualité d'encre à l'autre et même d'un type de papier à l'autre.

C'est pourquoi plutôt que d'ajouter un moteur graphique à Ghostscript (qui est déjà conséquent), on a préféré le conjuguer à un moteur graphique existant, libre et très puissant.

Le bien nommé *GIMP* pour *GNU Image Manipulation Program* est effectivement très performant et tout à fait compétent pour *mapper* une image de $X \times Y \times 24bits$ en p. ex : $(X \times 3) \times (Y \times 4) \times 4couleurs$

Malheureusement en cours de développement, il passe par des stades difficilement compatible. Le point de départ étant ghostscript, pour ce *post-traitement* les développeurs ont eu recours à un format particulier càd à un DRIVER spécifique pour ghostscript.

D'une version à l'autre de *gimp-print*, le format à évolué, mais le marché des imprimantes aussi. Si bien qu'actuellement certaines imprimantes sont utilisable via la première version, tandis que d'autre nécessiterons la dernière...

Bref entre *stp*, *gimp-print*, *ijs* et les adaptations propriétaires comme *hpijs*, le choix est vaste et tortueux...

foomatic

LinuxPrinting.org nous propose d'installer *foomatic* et de télécharger le fichier *HP-DeskJet_895C-hpijs.ppd*.

Foomatic est un ensemble de filtres du type *magicfilter* couplé à une base de donnée qui recense toutes les imprimantes fonctionnant sous Linux.

foomatic ou manuel *foomatic* repose sur l'utilisation des *fichiers de description d'imprimante*, les fichiers **.ppd*.

En configurant les applications pour qu'elle utilisent le fichier *ppd* correspondant à l'imprimante, l'application ajoute des commandes et options destinées à l'interpréteur postscript (Ghostscript) au moment de l'impression.

Le filtre *foomatic* interprète le fichier d'impression postscript pour extraire la commande *gs* et les options à lancer pour traduire correctement le postscript.

La configuration de *printcap* diffère en cela que l'on rajoute un champ *:ty= :* qui n'est pas reconnu par le daemon LPD mais qui sera interprété par *foomatic* pour une config par défaut.

Pour s'affranchir de *foomatic* et créer des queues capables de traiter du *postscript générique*, le plus simple sera de rechercher les commandes et options à l'intérieur du fichier *ppd* proposé par :

```
http://www.linuxprinting.org/ppd-o-matic.cgi?driver=hpijs&  
printer=HP-DeskJet_895C&show=0
```

au bout de quelque *grep*, on en ressort quelque chose comme :

```
gs -q -dBATCH -dPARANOIDSAFER -dQUIET -dNOPAUSE -sDEVICE=ijs \  
-sIjsServer=hpijs%A%B%C -dIjsUseOutputFD%Z -sOutputFile=- -
```

reste à remplacer les *variables* %A, %B, %C et %Z.

```
A: -sDeviceManufacturer="HEWLETT-PACKARD" -sDeviceModel="DESKJET 895"
```

```
A: -dDEVICEWIDTHPOINTS=595 -dDEVICEHEIGHTPOINTS=842
```

```
B: -r600 -sIjsParams=Quality:Quality=2,Quality:ColorMode=2,Quality:MediaType=2,Quality:PenSet=2
```

au final, on doit avoir quelque chose comme :

```
gs -q -d BATCH -d PARANOIDSAFER -d QUIET -d NOPAUSE -s DEVICE=ijs \
-s IjsServer=hpijs -s DeviceManufacturer="HEWLETT-PACKARD" \
-s DeviceModel="DESKJET 895" -d DEVICEWIDTHPOINTS=595 \
-d DEVICEHEIGHTPOINTS=842 -r600 \
-s IjsParams=Quality:Quality=2,Quality:ColorMode=2,Quality:Media\
aType=2,Quality:PenSet=2 \
-d IjsUseOutputFD -s OutputFile=- - <testpage.ps >testpage.pcl
```

Et donc les queues :

```
root@debian # mkdir /var/spool/lpd/HP895C-RAW
```

```
root@debian # chown lp: $_
```

```
root@debian # chmod 2775 $_
root@debian # cd $_
root@debian # cat <<eof >script.sh
#!/bin/sh
nc -q0 adresse_ip_HP895Cxi 9100
eof
root@debian # chown lp: script.sh
root@debian # chmod 754 $_
root@debian # cat <<eof >>/etc/printcap
HP895C-RAW:
    :sd=/var/spool/lpd/HP895C-RAW:\
    :sh:mx#0:\
    :lp=/dev/null:\
    :if=/var/spool/lpd/HP895C-RAW/script.sh:
eof
root@debian # mkdir /var/spool/lpd/HP895C-PS
root@debian # chown lp: $_
root@debian # chmod 2775 $_
```



```
root@debian # cd $_
root@debian # cat <<eof >script.sh
#!/bin/sh
gs -q -dBATCHE -dPARANOIDSAFER -dQUIET -dNOPAUSE -sDEVICE=ijs \
  -sIjsServer=hpijs -sDeviceManufacturer="HEWLETT-PACKARD" \
  -sDeviceModel="DESKJET 895" -dDEVICEWIDTHPOINTS=595 \
  -dDEVICEHEIGHTPOINTS=842 -r600 \
  -sIjsParams=Quality:Quality=2,Quality:ColorMode=2,Quality:Media\
  aType=2,Quality:PenSet=2 \
  -dIjsUseOutputFD -sOutputFile=- - | \
  lpr -P HP895C-RAW
eof
root@debian # chown lp: script.sh
root@debian # chmod 754 $_
root@debian # cat <<eof >>/etc/printcap
HP895C-PS-HQ|HP895C-PS:
  :sd=/var/spool/lpd/HP895C-PS:\
  :sh:mx#0:\
```

```
:lp=/dev/null:\
```

```
:if=/var/spool/lpd/HP895C-PS/script.sh:
```

```
eof
```

```
root@debian # lpr -P HP895C-PS /tmp/testpage.ps
```

```
root@debian #
```

```
root@debian # # Allez 'core une 'tite dernière:
```

```
root@debian #
```

```
root@debian # mkdir /var/spool/lpd/HP895C-Fast
```

```
root@debian # chown lp: $_
```

```
root@debian # chmod 2775 $_
```

```
root@debian # cd $_
```

```
root@debian # cat <<eof >script.sh
```

```
#!/bin/sh
```

```
gs -q -dBATCHe -dPARANOIDSAFER -dQUIET -dNOPAUSE -sDEVICE=ijs \
```

```
    -sIjsServer=hpijs -sDeviceManufacturer="HEWLETT-PACKARD" \
```

```
    -sDeviceModel="DESKJET 895" -dDEVICEWIDTHPOINTS=595 \
```

```
    -dDEVICEHEIGHTPOINTS=842 -r300 \
```

```
    -sIjsParams=Quality:Quality=0,Quality:ColorMode=2,Quality:Medi\
```

```
                                aType=0,Quality:PenSet=2 \  
-dIjsUseOutputFD -sOutputFile=- - |\  
eof  
root@debian # chown lp: script.sh  
root@debian # chmod 754 $_  
root@debian # cat <<eof >>/etc/printcap  
HP895C-Fast|HP895C-Draft:  
    :sd=/var/spool/lpd/HP895C-Fast:\br/>    :sh:mx#0:\br/>    :lp=/dev/null:\br/>    :if=/var/spool/lpd/HP895C-Fast/script.sh:  
eof  
root@debian # lpr -P HP895C-Fast /tmp/testpage.ps
```

Epilogue

Pour terminer liaison “a” sur le slide “Principe” représente le partage des imprimantes par d’autres hôtes clients.

Dans le cas de clients Un*x, la configuration d’un client LPD minimum suffira.

Pour les autres services, il faudra utiliser les serveurs respectifs. cf. samba, netatalk, etc.

Pour exemple, le fichier */etc/samba/smb.conf* présente entre autres :

```
printing = bsd  
print command = lpr -r -P'%p' %s  
lpq command = lpq -P'%p'  
lprm command = lprm -P'%p' %j
```

qui permettent d'indiquer à samba quels commandes utiliser et de quelle manière pour gérer les jobs d'impression.

\$Id: lpr.tex,v 1.2 2004/05/05 00:22:14 fh Exp \$