UPMC
PARIS UNIVERSITAS

Préparation LPI

Exam 101

104.7. Rechercher les fichier
-Connaître l'arborescence Linux

104.7 – Domaines de connaissances

- Poids : 2
- Connaître la localisation des fichiers d'une arborescence Linux fondée sur le standard FHS (Filesystem Hierarchy Standard)
- Commandes pour rechercher fichiers et commandes

Sommaire

- File system Hierachy Standard - FHS
- find
- locate – updatedb
- whereis
- which
- type

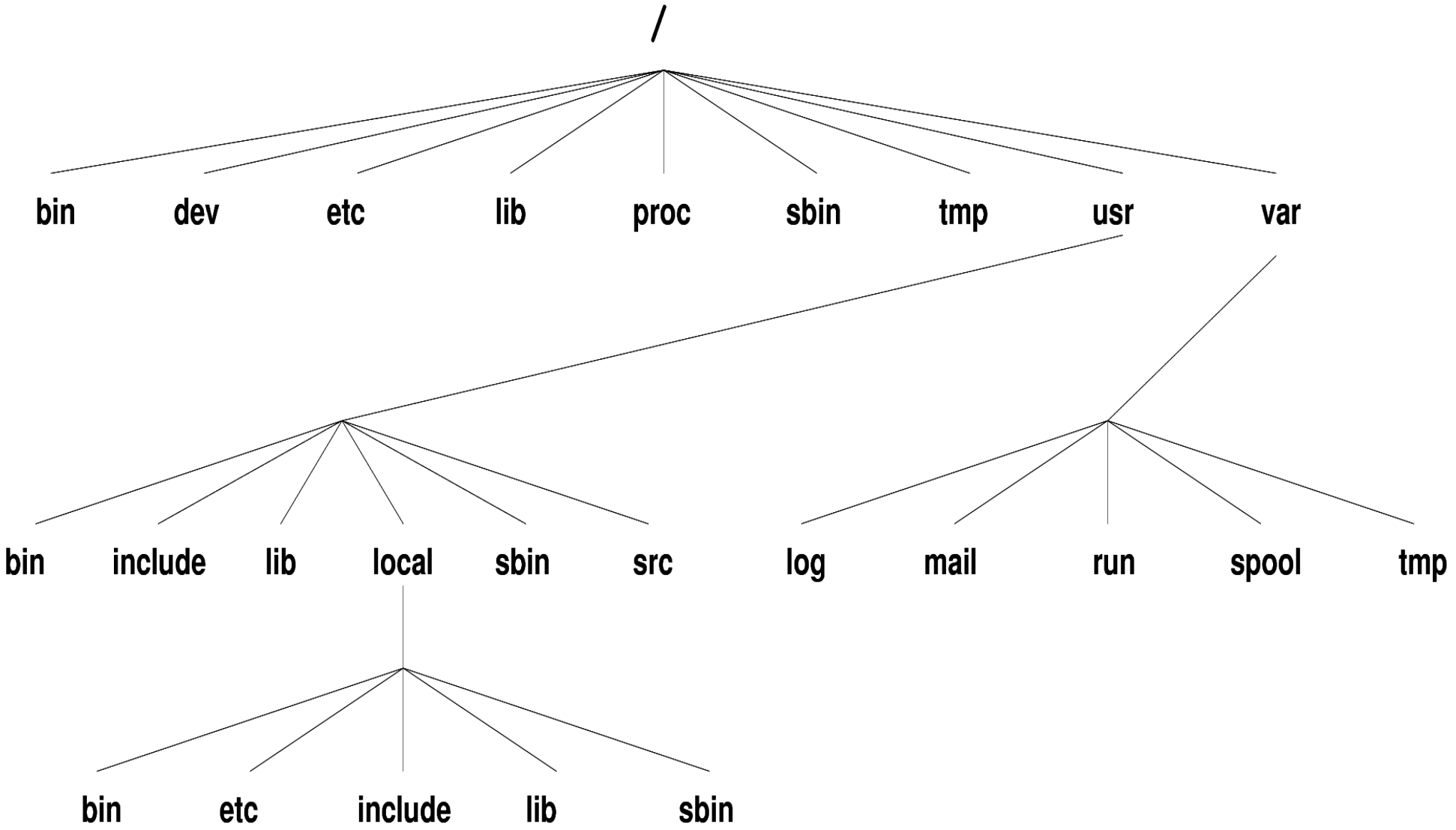
104.7 - FHS

- Intérêt de la connaissance de l'arborescence unix
  - connaitre le contenu et l'utilité des répertoires
  - savoir configurer les PATH (utilisateur + root)
  - surveiller les répertoires susceptibles de grossir rapidement
- Hiérarchie entre les différents unix (linux également) peu homogène
- Tentative de normalisation avec FHS (FileSystem Hierarchy Standard)
  - `http://www.pathname.com/fhs`
- La répartition des fichiers est basée sur 2 caractéristiques indépendantes mais mixables:
  - statique / variable
  - partageable / non partageable

104.7 - FHS

- fichiers statiques : fichiers qui ne changent pas sans l'intervention de l'administrateur (binaires, librairies, pages de manuel,...) -
- fichiers variables : fichiers qui sont modifiés de manière naturelle au cours de l'activité du système (messagerie, journaux, verrous,...)
- fichiers partageables : fichiers stockés sur une machine et utilisables depuis une autre machine (répertoire utilisateurs – home, messagerie, binaires,...)
- fichiers non partageable : fichiers stockés sur une machine et inutilisable depuis une autre (fichiers de configuration, noyau, verrous,...)
- différences entre répertoires système et répertoires utilisateurs

104.7 - FHS


/
bin
dev
etc
lib
proc
sbin
tmp
usr
var
bin
include
lib
local
sbin
src
log
mail
run
spool
tmp
bin
etc
include
lib
sbin

- Système de fichier root
  - doit contenir l'ensemble des fichiers permettant de démarrer, restaurer et réparer le système
bin	commandes essentielles au système
boot	fichiers nécessaires au démarrage (noyau)
dev	fichiers de périphérique
etc	fichiers de configuration
lib	librairies et modules essentiels au système
media	point de montage pour les périphériques amovibles
mnt	point de montage pour les systèmes de fichiers temporaires
opt	applications
sbin	binaires essentiels au système
srv	données pour services proposés par le système
tmp	fichiers temporaires
usr	Seconde hiérachie
var	Données variables

- /**bin** (binaries)
  - contient les commandes utilisables par l'administrateur ou l'utilisateur qui doivent être disponibles quand aucun autre système de fichier est monté (single mode)
  - exemples : chmod, chown, cp, login, mkdir, mknod, ps, ...
- /**sbin** (system binaries)
  - contient les binaires système essentiels utilisés lors du démarrage et utilisables par l'administrateur
  - les binaires systèmes qui ne nécessitent pas que / soit monté se retrouvent dans /usr/sbin ou /usr/local/sbin
  - exemples : shutdown, init, fsck, halt, mkfs,...

- **/boot**
  - contient les fichiers nécessaires au démarrage du système : les fichiers du noyau en particulier
  - on les retrouvent parfois dans /
- /**dev** (devices)
  - les fichiers spéciaux de périphérique permettant d'accéder aux ressources physiques de la machine
  - répertoire plus ou moins hiérarchisé en fonction des Unix
  - script de création des fichiers de périphériques : /dev/MAKEDEV
  - remplacé aujourd'hui par des solutions plus souples et plus proche de la dynamique du système (udev, devfs)

- pseudo filesystems (/proc /sysfs ...)
  - /**proc** : présente les informations variables sur l'état du système (CPU,...), des périphériques...
  - /**sysfs** : (ou /sys) filesystem virtuel chargé de présenter la configuration matérielle du système de manière plus structurée de /proc
    - possibilité d'accéder à tout ce qui se passe dans le noyau, périphériques, pilotes
    - plusieurs catégories : block, bus, class, devices, firmware, fs, kernel, module, power
- /**srv**
  - Contient des données mises à disposition par le système (ftp, rsync, www, cvs)

- **/etc**
  - contient les fichiers de configuration
  - de plus en plus, regroupement des fichiers dans des sous-répertoires propres à chaque application ou service
  - /etc/opt, /etc/X11, /etc/sgml, /etc/xml
- **/home**
  - contient habituellement les répertoires des utilisateurs
  - les fichiers de configuration liés à un utilisateur sont la plupart du temps des fichiers ou des répertoires cachés (commencent par un point « . »)
- **/lib**
  - contient les librairies nécessaires au démarrage et aux commandes localisées dans /bin et /sbin

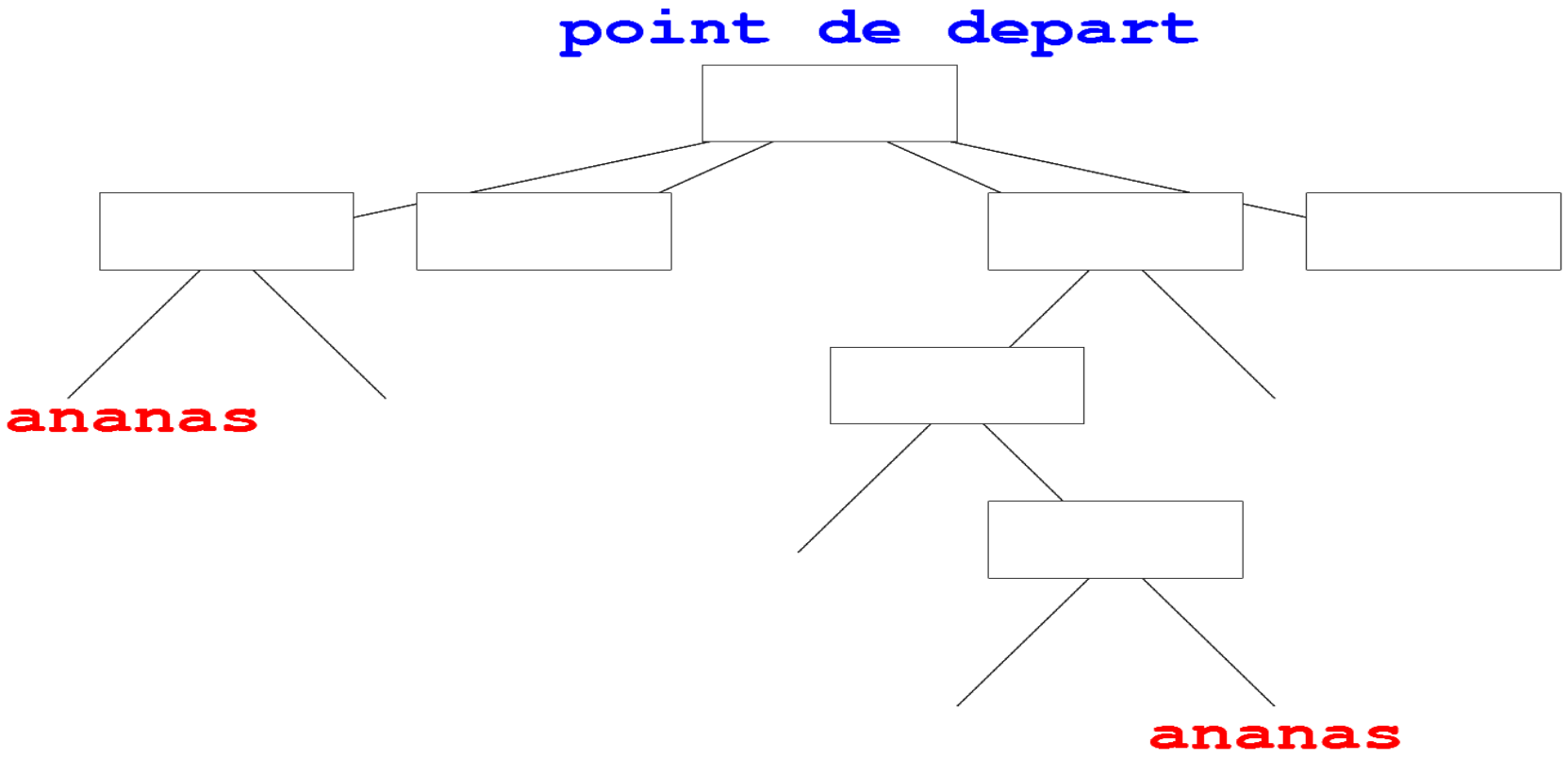
- **/media**
  - contient les répertoires accueillant les points de montage des périphériques amovibles (disques externes, clés USB, cdrom)
- **/mnt**
  - contient les répertoires accueillant les points de montage temporaires
- **/opt**
  - contient dans des répertoires spécifiques la (quasi)totalité des fichiers lié à une application
  - fichiers de configuration et données variables peuvent se trouver respectivement dans /etc/opt et /var/opt

- **/root**
  - le répertoire personnel de l'utilisateur root
- **/tmp**
  - contient les fichiers temporaires
  - ne pas stocker dans ce répertoire de données que l'on souhaite conserver

- hiérarchie /**usr**
  - doit contenir les fichiers qui sont partageables et en lecture seule
  - bin commandes utilisateur
    include fichiers d'en-têtes des programmes C
    lib librairies nécessaires aux commandes situées dans /usr/bin et /usr/sbin
    fichiers de configuration
    local hiérarchie destinée à accueillir les applications locale, spécifique à la machine et à priori non partagées avec d'autres machines
    sbin commandes utilisables par l'administrateur non essentielles au système
    share stocke habituellement les fichiers statiques des applications – pages de manuels
    src code source de certaines application - noyau

- hiérarchie /**var**
  - doit contenir l'ensemble des fichiers variables
  - cache données en cache
    lib informations d'état des applications
    lock verrous sur les ressources partagées
    local données variables pour la hiérarchie /usr/local/
    log journaux du système
    run données du système depuis son démarrage . Ex : fichiers PID
    log code source de certaines application – noyau
    spool données en attente de traitement ultérieur
    ex : mail, impressions
    tmp fichiers temporaires qui doivent subsister à un redémarrage du système
    mail fichiers de boîtes à lettres des utilisateurs
    se trouvent dans /var/spool/mail sur certains systèmes

- La recherche démarre à partir d'un répertoire de départ et sur la base de critères définis par des expressions



104.7 – Commande find

- Commande « `find` »
  - syntaxe : `find répertoires expressions`
  - `répertoires` indique le ou les répertoires à partir desquels `find` démarre la recherche
  - Les `expressions` indiquent :
    - des conditions
    - des actions à réaliser sur les fichiers trouvés
- Recherche d'un nom
  - « `-name nom` » : nom spécifié avec les méta-caractères du shell
  - exemples

```
$ find . -name '*.txt' -print
$ find . -name 'httpd.conf' -print
```

104.7 – Commande find

- Recherche sur les droits d'accès
  - « `-perm permissions` » : les permissions (en octal ou symbolique) doivent être strictement celles indiquées
  - « `-perm -permissions` » : tous les bits indiqués doivent être positionnés
  - « `-perm /permissions` » : au moins l'un des bits indiqués doit être positionné
  - exemples

    `$ find . perm 664 -print` (fichiers avec rw-rw-r-- exactement)

    `$ find / -perm 664 -print` (fichiers rwxrwxrwx correspondent également)

104.7 – Commande find

- Recherche sur les propriétaires des objets
  - « `-user nomuser` » : le propriétaire de l'objet doit être « nomuser »
  - exemple
    ```
    $ find /home -user corsini -print
    ```
- Recherche sur les tailles d'objets
  - « `-size [+|-]tailleunité` » : les fichiers doivent avoir pour taille exactement tailleunité (+ : plus de tailleunité; – : moins de tailleunité)
  - où `taille` est un nombre et `unité` = « c » pour octet, « k » pour kilooctet, « M » pour megaoctet et « G » pour Gigaoctet
  - exemple (les fichiers de plus de 30 Mo)
    ```
    $ find /home/toto -size 30M -print
    ```

104.7 – Commande find

- Recherche sur les types d'objets
  - « `-type code` » :
  - ou `code` =
    - « `f` » : fichier régulier
    - « `d` » : répertoire
    - « `l` » : lien symbolique
    - « `c` » : fichier mode caractère
    - « `b` » : fichier mode bloc
  - exemples

```
$ find . -type f -exec file '{}' \;
```

104.7 – Commande find

- Recherche sur les dates d'objets
  - « `-atime [+|-]nombre` » : fichiers accédés il y a « nombre » jours (+ : plus de « nombre » jours, - moins de « nombre » jours)
  - « `-mtime [+|-]nombre` » : fichiers modifiés il y a « nombre » jours
  - « `-ctime [+|-]nombre` » : fichiers crées il y a « nombre » jours
  - ou `n` = 24 heures
  - exemples

    `$ find . -mtime -3 -print` (fichiers modifiés il y a moins de 3 jours)

    `$ find . -mtime +3 -print` (fichiers modifiés il y a plus de 3 jours)

104.7 – Commande find

- Affichage des objets trouvés
  - « -print »
  - « -ls »

```
[aoi@test]$ find . -mtime -2 -type f -print
./exemple2.txt
./exemple.txt
./exemple2.txt~

[aoi@test]$ find . -mtime -2 -type f -ls
1666079    1 -rw-r--r--   1 franck   franck        142 mar 26
15:21 ./exemple2.txt
1666077    1 -rw-r--r--   1 franck   franck         40 mar 26
15:12 ./exemple.txt
1666078    1 -rw-r--r--   1 franck   franck         77 mar 26
15:19 ./exemple2.txt~
```

- Composition de critères de recherche
  - ou logique
    « `-cond1 -o cond2` »
  - et logique
    « `-cond1 -a cond2` » (**-a** facultatif)
  - groupement d'expressions
    « `\( -cond1 -[ao] cond2 \)` » (attention à la protection des paranthèses avec « \ »)
    Attention : il faut laisser un espace avant et après les paranthèses
  - exemple

```
$ find . \( -name " *.jpg "  -o -name
" *.jpeg " \) -a -size +100M -print
```

104.7 – Commande find

- Execution d'une commande
  - « `-exec commande {} \;` » (attention à la protection du « ; » avec « \ »)
  - « `{}` » symbolise dans la commande le fichier trouvé
  - exemple : rechercher tous les fichiers de type régulier et les effacer

```
[aoi@test]$ ls -l
total 2
-rw-r--r-- 1 franck franck    0 mar 27 15:32 a
-rw-r--r-- 1 franck franck    0 mar 27 15:32 b
-rw-r--r-- 1 franck franck    0 mar 27 15:32 c
drwxr-xr-x 2 franck franck 1024 mar 27 15:33 d/
drwxr-xr-x 2 franck franck 1024 mar 27 15:33 e/
[aoi@test]$ find . -type f -exec rm {} \;
[aoi@test]$ ls -l
total 2
drwxr-xr-x 2 franck franck 1024 mar 27 15:33 d/
drwxr-xr-x 2 franck franck 1024 mar 27 15:33 e/
```

104.7 – Commande locate

- Commande `locate`
  `locate [OPTION]... PATTERN...`
  - Affiche la liste des fichiers qui satisfont un motif
  - Utilise soit le file globbing soit des regexp si l'option `–regexp` est spécifiée
  - Récupère ses informations depuis une ou plusieurs bases de données générée par la commande updatedb
  - Plus rapide que find mais pas forcément exhaustif
- Options
  - -i : ignore la casse
  - -L : suit les liens symboliques
  - -b : restreint la recherche au basename

104.7 – Commande locate

```
$ locate *conv
/usr/bin/iconv
/usr/bin/piconv
/usr/bin/preconv
/usr/lib/gconv
/usr/lib/man-db/manconv
/usr/lib/perl5/auto/Text/Iconv
/usr/sbin/grpconv
/usr/sbin/grpunconv
/usr/sbin/pwconv
/usr/sbin/pwunconv
```

104.7 – Commande locate

- Implicitement, locate transforme le motif recherché
  `MOTIF` est équivalent à `*MOTIF*`
- Pour restreindre la recherche uniquement au nom recherché, utilisation de \ devant le motif
  `locate -b '\MOTIF'`

```
$ locate a2ps | wc -l
246

$ locate -b a2ps | wc -l
45

$ locate -b '\a2ps'
/home/franck/cours/Archives/aoi/2006/unix/cours/AOI-1/1050/a2ps
/usr/bin/a2ps
/usr/lib/emacsen-common/packages/install/a2ps
/usr/lib/emacsen-common/packages/remove/a2ps
/usr/share/a2ps
/usr/share/doc/a2ps
/usr/share/emacs/site-lisp/a2ps
```

104.7 – Commande updatedb

- Commande `updatedb`
  - Commande qui crée la base de données utilisée par la commande locate
  - Généralement, lancée par un cron prériodiquement

    Conséquence : entre 2 exécution de updatedb, les nouveaux fichiers n'existent pas dans la base !
  - S'appuie sur le fichier de configuration `/etc/updadatedb.conf`
  - La base de données se trouve par défaut dans `/var/lib/mlocate/mlocate.db`

104.7 – Commande whereis

- Commande `whereis`
  `whereis [OPTION] NOM-COMMANDE`
  - Recherche les binaires, les fichiers sources et les pages de manuels
  - `whereis` sélectionne les répertoires habituellement utilisés pour accueillir des binaires

/{bin,sbin,etc}

/usr/{lib,bin,old,new,local,games,include,etc,src,man,sbin,
X386,TeX,g++-include}

/usr/local/{X386,TeX,X11,include,lib,man,etc,bin,games,emacs}

104.7 – Commande which

- Commande `which`
  `which  NOM DE FICHIER`
  - Recherche un binaire dont le nom correspond exactement au nom passé en argument et qui doit se trouver exclusivement dans les chemins définis dans la varibles PATH
  - Affiche le chemin absolu lorsque le binaire est trouvé

```
# which a2ps
/usr/bin/a2ps
```

104.7 – Commande which

- Commande `type`

  `type NOM-COMMANDE`

  - Commande interne au shell
  - Indique comment le shell interprétera la commande lorsqu'elle sera évaluée
  - Recherche dans les chemins de PATH

```
# type echo
echo is a shell builtin

$ type ls
ls est un alias vers « ls --color=auto »

$ type a2ps
a2ps est /usr/bin/a2ps
```

Quelques conseils LPI...

-
-