

Administration système

Gestion diverses : terminaux, imprimantes

Rémy CARD & Tuyêt Trâm DANG NGOC

`<dntt@u-cergy.fr>`

Université de Cergy-Pontoise

2009–2010



1 Gestion des terminaux

2 X-Window

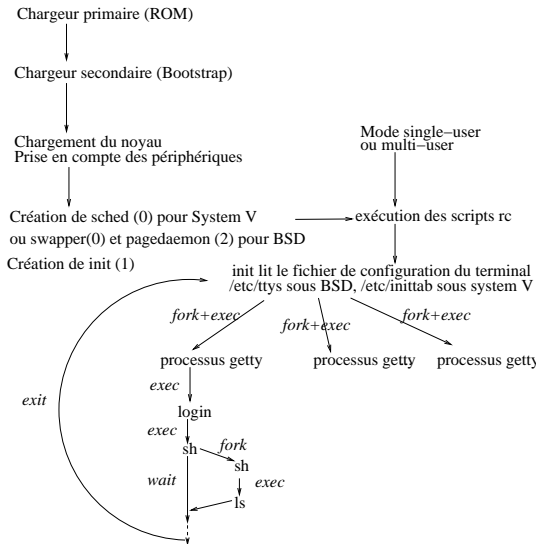
3 Imprimantes

4 Crédits

Gestion des terminaux

- Deux types de terminaux :
 - terminaux connectés directement au système
 - terminaux virtuels, utilisés par X-Window et les connexions par réseau
- Terminaux connectés :
 - activation d'un processus de connection
 - mise à disponibilité des informations de contrôle
- Terminaux virtuels :
 - mise à disponibilité des informations de contrôle

Démarrage des getty



Détail d'une connexion

- Lecture du nom d'utilisateur par `getty`
- Exécution du programme `login` par `getty`
- Lecture du mot de passe et validation par `login`
- Affichage de `/etc/motd` par `login`
- Positionnement de la variable d'environnement `TERM` et exécution du shell par `login`
- Exécution des fichiers d'initialisation par le shell

Fichiers de configuration

Système	Activation	Type de terminal	Paramètres	Gestionnaire
Solaris	_sactab	_sactab	_pmtab	ttymon
HP-UX	inittab	ttytype	gettydefs	getty
IRIX	inittab	ttytype	gettydefs	getty
SunOS	ttytab	ttytab	gettytab	getty
OSF/1	inittab	inittab	gettydefs	getty
*BSD	ttys	ttys	gettytab	getty

/etc/ttytab et /etc/ttys

- Systèmes basés sur 4.3BSD
- Description des terminaux directement connectés :
 - port
 - type du terminal
 - programme à exécuter

- Syntaxe de chaque ligne :

```
port    programme type_terminal on|off [secure]
```

- Exemple (`/etc/ttytab` sous SunOS) :

```
console '/usr/etc/getty std.9600'  sun      on secure
ttya    '/usr/etc/getty std.19200' vt100   on
ttyb    '/usr/etc/getty std.9600'  unknown off
```

- Après un changement de configuration : `kill -HUP 1`

/etc/inittab

- Systèmes basés sur System V
- Le programme `getty` peut être lancé par `init`
- Exemple :

```
co:234:respawn:/etc/getty console console  
t1:234:respawn:/etc/getty ttyS1 19200  
t2:234:off:/etc/getty ttyS2 9600
```


/etc/ttytype

- Fichier décrivant le type des terminaux connectés
- Syntaxe de chaque ligne :

type_terminalport

- Exemple :

```
wyse      console
dialup    ttyi1
dialup    ttyi2
vt320     ttyi2
```

/etc/gettytab

- Définition des informations de contrôle des ports
- Association de noms symboliques à la configuration des ports
- Syntaxe similaire à `printcap` et `termcap`
- Exemple :

```
default:\n    :lm=\r\n%h login\72 :sp#9600:\n2|std.9600|9600-baud:\n    :sp#9600:\nh|std.19200|19200-baud:\n    :sp#19200:
```

/etc/gettydefs

- Rôle indentique à `gettytab` ...

- ... mais syntaxe différente

- Syntaxe :

label# initialisation# terminaison# message# suivant

- Exemple :

console# B9600 HUPCL # B9600 SANE #login: #console

19200# B19200 HUPCL # B19200 SANE #login: #9600

9600# B9600 HUPCL # B9600 SANE #login: #4800

9600# B9600 HUPCL # B9600 SANE #login: #4800

- Après modification :

`getty -c gettydefs`

Paramètres des terminaux

- Deux bases de données définissent les paramètres des terminaux :
 - `/etc/termcap` sous BSD
 - `/usr/lib/terminfo` (ou `/usr/share/terminfo` ou `/usr/share/lib/terminfo`) sous System V
- Ces bases définissent :
 - les paramètres des terminaux (nombre de lignes, de colonnes, ...)
 - les caractères de contrôle

/etc/termcap

- Suite d'entrées
- Syntaxe d'une entrée :
nom1|nom2...:paramètres
- Syntaxe similaire à `printcap`
- Exemple :

```
d0|vt100|dec vt100:\
    :co#80:li#24:ho=\E[H:\
    :ku=\EOA:kd=\EOB:
```

terminfo

- Ensemble de fichiers binaires décrivant des terminaux
- Chaque entrée est un fichier situé dans le répertoire de la base
- Compilation d'un fichier de description : `tic`
- Décompilation d'une entrée compilée : `infocmp`
- Exemple de description :

```
vt100|dec vt100,  
    cols#80, lines#24, home=\E[H,  
    kcuul=\EOA, kdcudl=\EOB
```

Caractéristiques du terminal

- Un utilisateur peut utiliser `stty` pour modifier les caractéristiques de son terminal
- Syntaxe : `stty option [valeur] ...`
- Principales options :
 - `N` : vitesse de la ligne
 - `rows N` : nombre de colonnes
 - `lines N` : nombre de lignes
 - `erase C` : caractère d'effacement de caractère
 - `intr C` : caractère d'interruption
 - `susp C` : caractère de suspension
 - `oddp`, `evenp` : parité
 - `sane` : réinitialisation des paramètres

1 Gestion des terminaux

2 **X-Window**

3 Imprimantes

4 Crédits

X-Window

Configuration X11 - Fichier /etc/X11/xorg.conf

```
# Repertoires comportant les polices
Section "Files"
    FontPath          "/usr/share/fonts/X11/misc"
    FontPath          "/usr/X11R6/lib/X11/fonts/misc"
EndSection

# Les modules à charger
Section "Module"
    Load             "i2c"
    Load             "bitmap"
EndSection
```

Configuration X11 - Fichier /etc/X11/xorg.conf

```
# Périphérique : carte video
Section "Device"
    Identifier      "Ma Carte vidéo"
    Driver          "vesa"
    BusID           "PCI:0:2:0"
EndSection

# L'ecran
Section "Monitor"
    Identifier      "Écran générique"
    Option          "DPMS"
EndSection

# L'affichage : une carte video + un ecran + un mode d'affichage
# (nb couleur en puissance de 2, resolution)
Section "Screen"
    Identifier      "Mon Affichage"
    Device          "Ma Carte vidéo"
    Monitor         "Écran générique"
    DefaultDepth    24
    SubSection "Display"
        Depth      16
        Modes       "1920x1440" "1600x1200" "1280x1024" "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth      24
        Modes       "1920x1440" "1600x1200" "1280x1024" "1024x768" "800x600" "640x480"
    EndSubSection
EndSection
```

Configuration X11 - Fichier /etc/X11/xorg.conf

```
# Configuration d'un périphérique d'entree (clavier, souris)
```

```
Section "InputDevice"
```

```
    Identifier "nom du peripherique"
```

```
    Driver      "nom_du_driver"
```

```
    Option      "option du driver pour le peripherique"
```

```
    Option      "option2 du driver pour le peripherique"
```

```
EndSection
```

```
# Une config serveur : un affichage, un clavier, une souris
```

```
Section "ServerLayout"
```

```
    Identifier      "Default Layout"
```

```
    Screen          "Mon Affichage"
```

```
    InputDevice     "Mon Clavier"
```

```
    InputDevice     "Ma Souris"
```

```
EndSection
```

1 Gestion des terminaux

2 X-Window

3 Imprimantes

4 Crédits

Gestion des imprimantes

- Deux modèles
- BSD :
 - `lpr`
 - `lpd`
 - `lpc`
- System V
 - `lp`
 - `lpsched`
 - `lpadmin`
 - `lpshut`
 - ...

Impression BSD

- Imprimantes gérées par le démon `lpd`
- Impression par la commande `lpr`
- Impression d'un fichier :
 - `lpr fichier`
 - transmission du fichier à `lpd` par `lpr`
 - Traitement par `lpd`
- Sélection de l'imprimante :
 - option de `lpr`
 - `-Pimprimante`

Fonctionnement de lpd

- Fichier de configuration : `/etc/printcap`
- Attente de requête d'impression
- Deux possibilités :
 - imprimante distante : transmission à une autre machine
 - imprimante locale : mise en file d'attente
- File d'attente locale :
 - fichier de contrôle (`cf*`)
 - fichier de données (`df*`)

Le fichier /etc/printcap

- Définition des imprimantes
- Chaque imprimante est définie par une suite de champs
- Principaux champs :
 - `sd=répertoire` : répertoire contenant la file d'attente
 - `lp=fichier_spécial` : périphérique de l'imprimante
 - `lf=fichier` : fichier contenant les traces des erreurs
 - `af=fichier` : fichier de comptabilité (pac)
 - `rm=machine`,
`rp=imprimante` : localisation de l'imprimante
 - `if=programme`,
`nf=programme`,
`of=programme` : filtres

Exemple

```
laser_pourrie:\n:lp=/dev/ttya:sd=/var/spool/laser_pourrie:sh:\n:lf=/var/log/laser_pourrie:\n:af=/var/adm/laser_pourrie:\n:br#9600:rw:fc#0000374:fs#0000003:xc#0:\n:xs#0040040:mx#0:sf:sb:\n:if=/usr/local/lib/filtre:\n:lf=/var/log/laser:af=/var/adm/laser:
```

Gestion de la file d'attente

- Affichage : `lpq`
- Suppression : `lprm`
- Gestion par l'administrateur : `lpc`
 - gestion interactive
 - plusieurs commandes

Impression System V

- Impression gérées par `lpsched`
- Impression par la commande `lp`
- Plusieurs commandes d'administration :
 - dans `/usr/lib`
 - dans `/usr/sbin` (Solaris 2)
- Deux types de destinations :
 - imprimantes
 - classes
 - paramètre de `lp` : `-d destination`

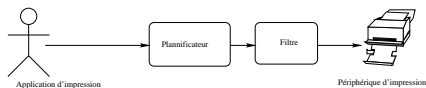
Types de systèmes d'impression

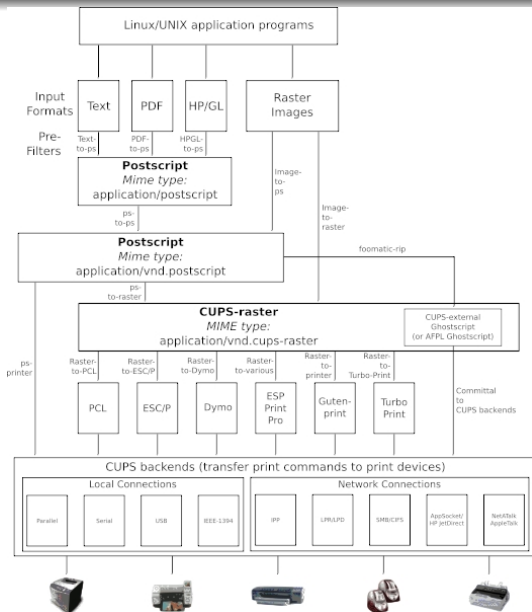
- Solaris 2 : System V
- HP-UX : System V
- IRIX : System V
- SunOS : BSD
- OSF/1 : BSD (+ commandes System V)
- *BSD : BSD
- Linux : BSD

Common Unix Printing System (CUPS)

Le Common Unix Printing System (CUPS) est un système modulaire d'impression informatique pour les systèmes d'exploitation Unix et assimilés qui permet à un ordinateur de se comporter comme un puissant serveur d'impression. Un ordinateur qui utilise CUPS est un serveur qui peut accepter des travaux d'impression envoyés par des ordinateurs clients, les traiter, et les envoyer à l'imprimante qui convient.

(source : Wikipédia)





- Support des systèmes d'impressions BSD

Exemple `/etc/printcap`

```
labo|HP LaserJet 4250:rm=192.68.42.254:rp=labo:  
couleur|HP Color LaserJet 2550:rm=192.68.42.254:rp=couleur:  
secretariat|secretariat:rm=192.68.42.254:rp=secretariat:
```


1 Gestion des terminaux

2 X-Window

3 Imprimantes

4 Crédits

Crédits

Ce cours a largement été inspiré des supports de cours de Rémy Card, Ingénieur de Recherche à l'université de Versailles-Saint-Quentin-en-Yvelines.

The CUPS diagram is based on GNU Free Documentation License. It was made by Glenn Davis (SVG), Ta bu shi da yu (PNG), and Kurt Pfeifle (ASCII)