

# Introduction système

Cours-5/5

Marcel.Bosc@iutv.univ-paris13.fr  
2005-2006

# Plan du cours

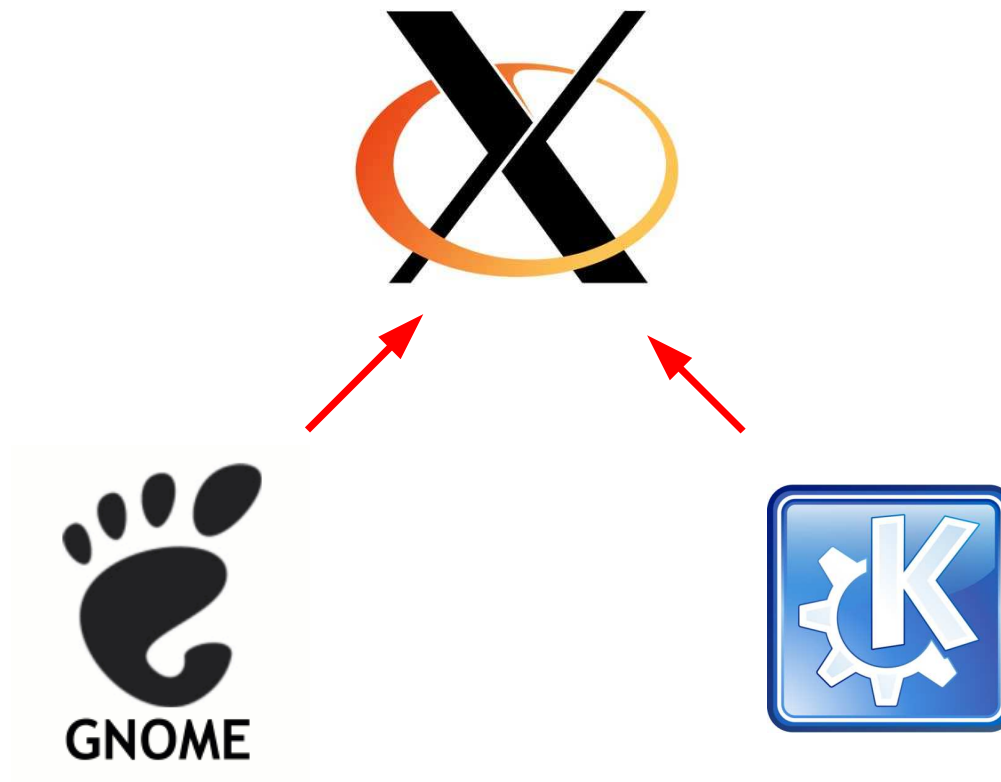
- environnements graphiques
- constructions bash
- librairies et paquets

## partie pratique

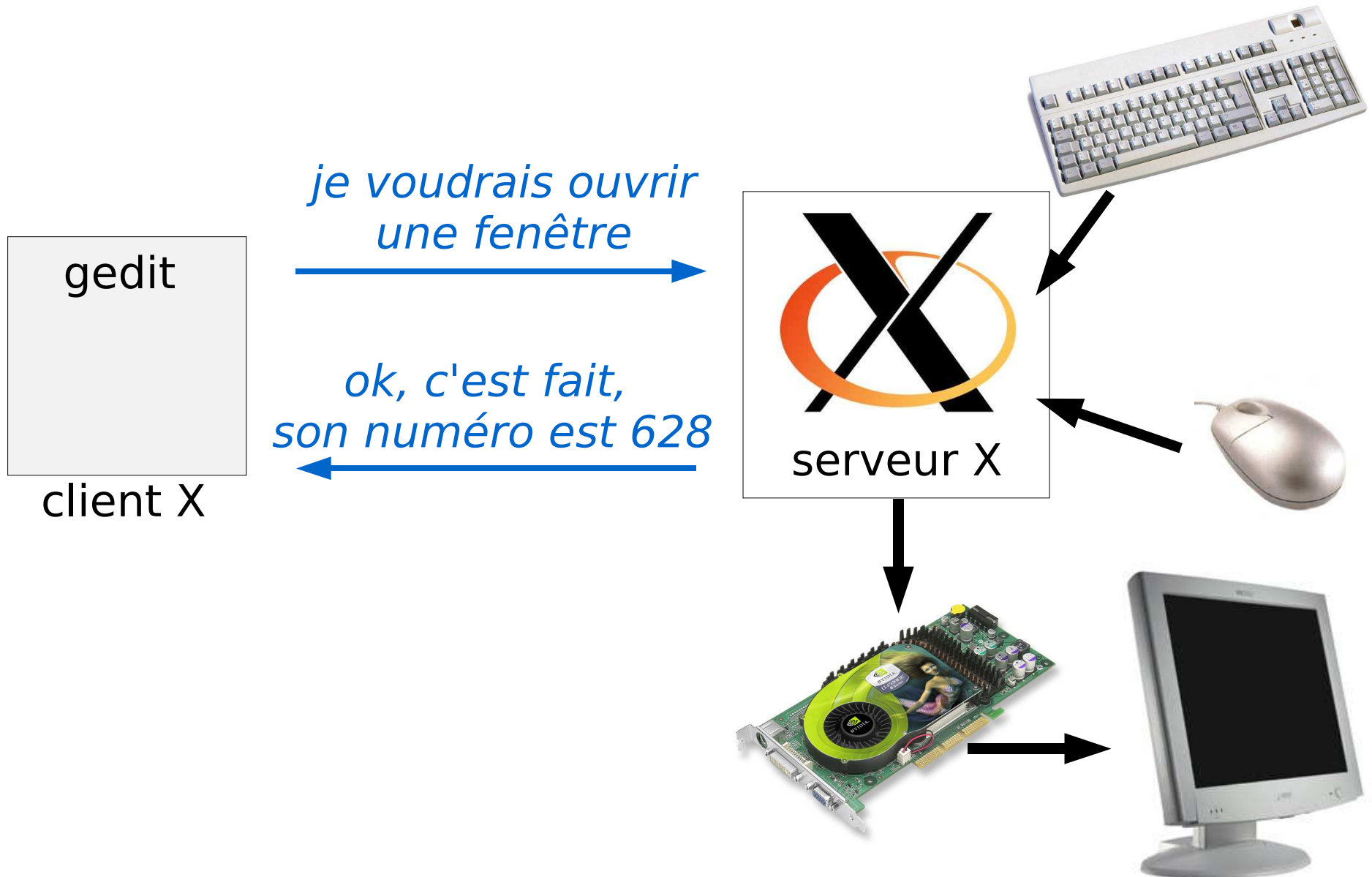
- quelques commandes

1ère partie

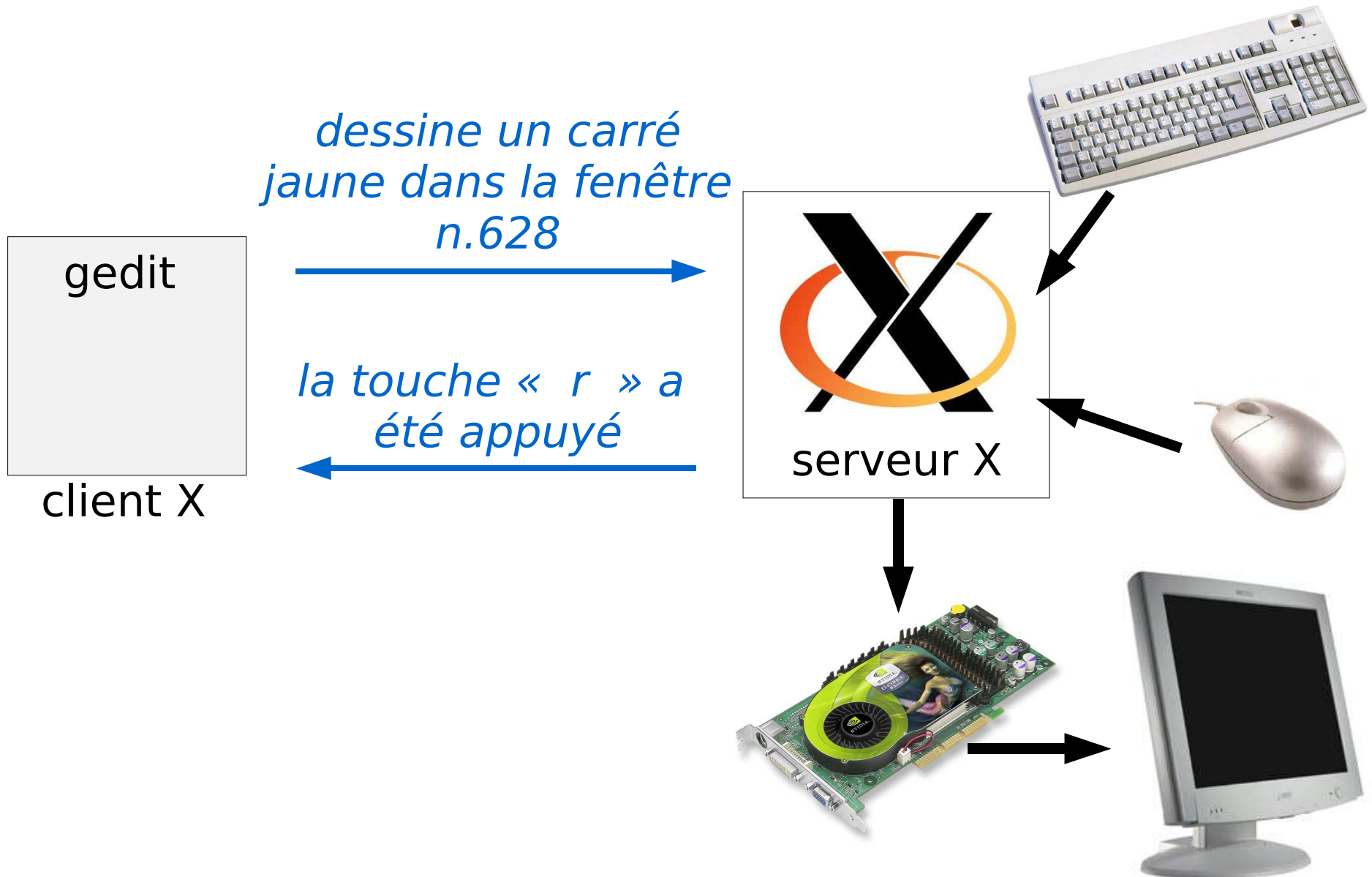
# environnements graphiques



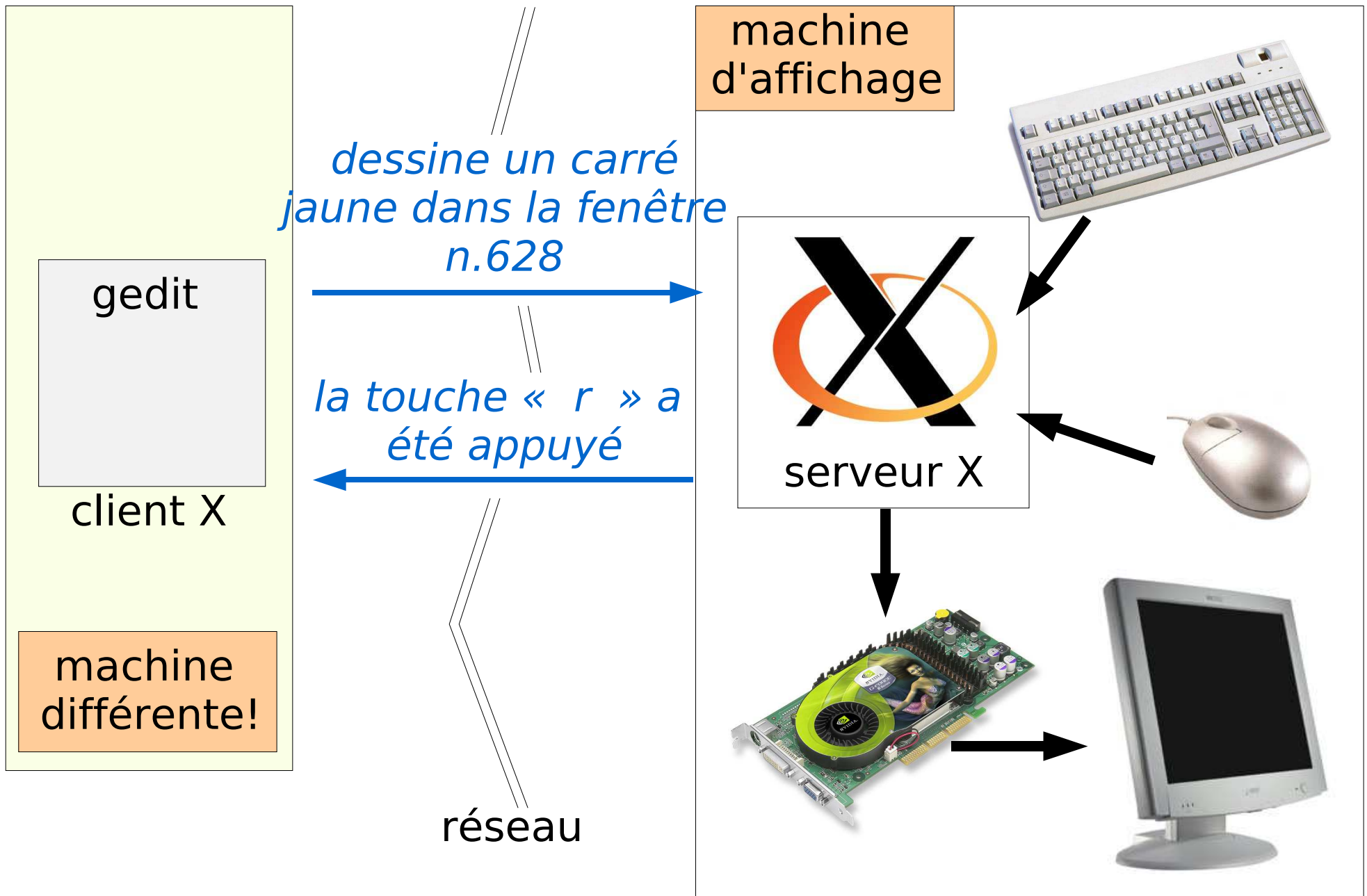
# serveur X



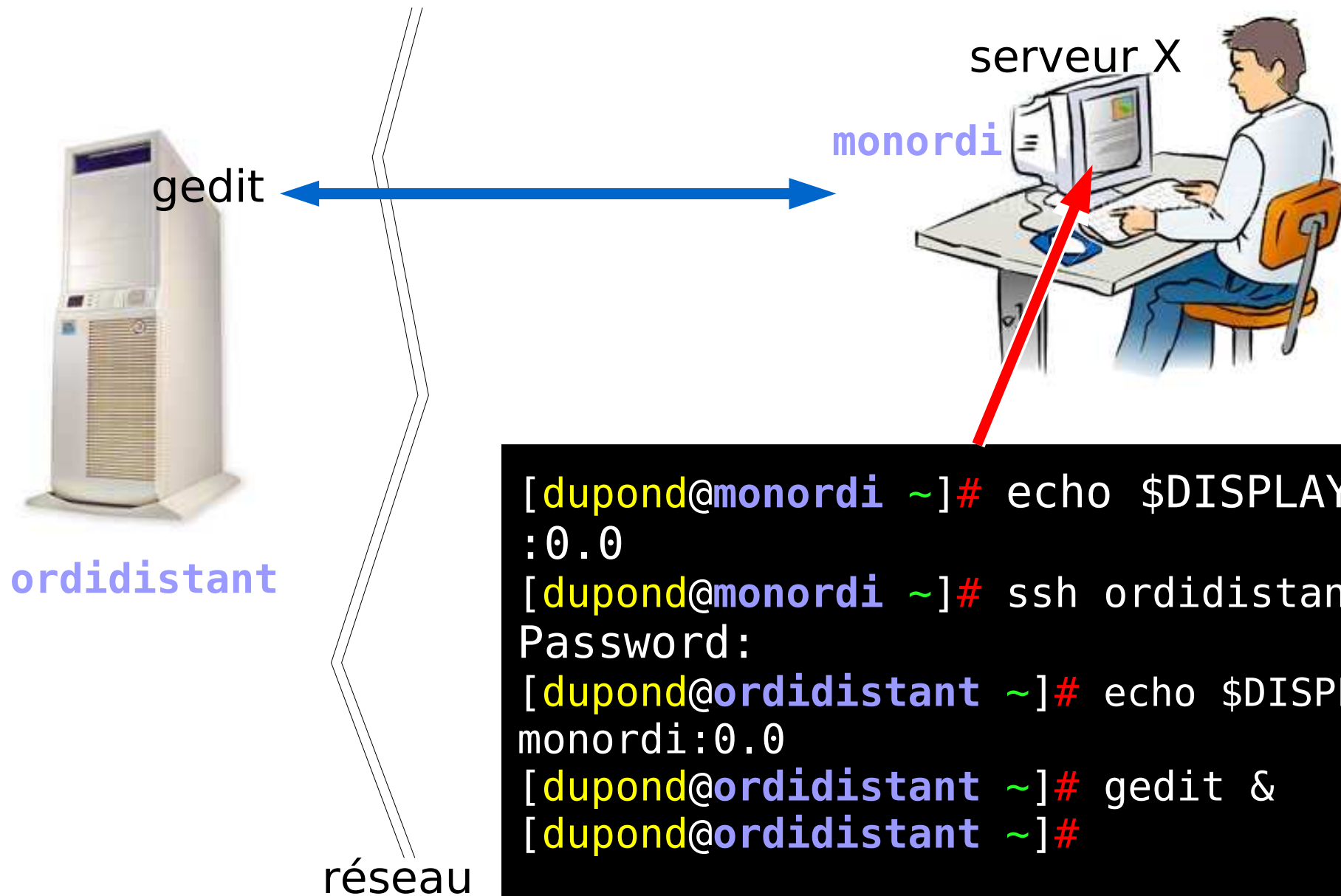
# serveur X



# transparence au réseau



# transparence au réseau



# environnements graphiques

environnements de bureau



GNOME

thèmes



kde

thèmes

WM:metacity

WM:Kwin

serveur X



X.org  
XFree86

applications

gnome

gedit,  
gnome-terminal,  
gpdf, ...

Firefox  
OpenOffice

kde

kate,  
konsole,  
kpdf, ...



2ème partie

# constructions bash

- substitution de commandes
- évaluation arithmétique
- boucles numériques

# substitution de commandes

fichier: *liste.txt*

```
doc.txt  
page.html  
essai.pdf
```

**`commande`** : exécute la commande et remplace les "`" par le résultat de la commande

```
echo `cat liste.txt`
```

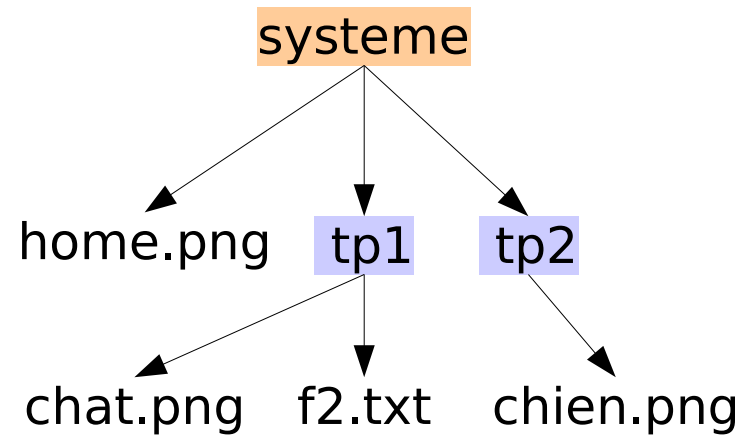
exécution de  
"cat liste.txt"

```
echo doc.txt page.html essai.pdf
```

**`commande`** ↔ **\$(commande)**

# substitution de commandes

exemple



```
rm `find . -name "*.png" `
```

exécution de  
la commande find

```
rm ./home.png ./tp1/chat.png ./tp2/chien.png
```

# substitution de commandes

exemple

fichier: *liste.txt*

```
home.png  
bird.jpg  
tux.png
```

```
echo `grep png liste.txt | sed 's/png/gif/g'`
```

exécution de  
la commande

```
echo home.gif tux.gif
```

# substitution de commandes

## exemples

```
# pour toutes les images png de l'arborescence
# créer une petite image et renommer l'original
for fichier in `find . -name "*.png" `; do
    rep=`dirname $fichier`
    nom=`basename $fichier`
    convert $fichier -geometry 100x100 $rep/petit-$nom
    mv $fichier $rep/original-$nom
done
```

# évaluation arithmétique

`$( (10+15) )` → 25

**`$(expression)`**:  
effectue les calculs  
arithmétiques spécifiés  
dans l'expression

```
n=5  
mv fichier.txt fichier$(($n+1)).txt
```



```
mv fichier.txt fichier6.txt
```

```
i=1;i=$(($i+1))
```

 → `i = 2`

# boucles numériques

```
for ((i=0;i<5;i++)); do  
    echo numéro: $i  
done
```



```
numéro: 0  
numéro: 1  
numéro: 2  
numéro: 3  
numéro: 4
```

3ème partie

# librairies et paquets

- librairies
- paquets
- gestion de paquets



# librairies : présentation

**librairie**: collection de fonctions compilées

rectangle.c

```
void dessiner_rectangle(...)
{
    ...
}
void dessiner_carre(...)
{
    ...
}
```

*compilation*

libdessin.a

cercle.c

```
void dessiner_cercle(...)
{
    ...
}
```

*compilation*

# librairies statiques

mon-programme.c

```
int main(...)  
{  
    dessiner_cercle(...)  
}
```

libdessin.a

*compilation*

*édition de liens*

contient  
"dessiner\_cercle"

mon-programme


exécution

# bibliothèques dynamiques

mon-programme.c

```
int main(...)  
{  
    dessiner_cercle(...)  
}
```

si 500 programmes  
utilisent "libdessin"

une seule copie de  
ses fonctions 

*compilation*

ne contient  
**pas** "dessiner\_cercle"

mon-programme

*édition  
de liens*

libdessin.so

exécution

# éléments d'un logiciel

- fichiers de données
- documentation
- librairies
- programmes exécutables
- ...

quelques fichiers de « gpdf »  
(sur 275)

/usr/share/pixmaps/gnome-pdf.png

/usr/share/man/man1/gpdf.1.gz

/usr/bin/gpdf

# paquets : présentation

formats



.deb

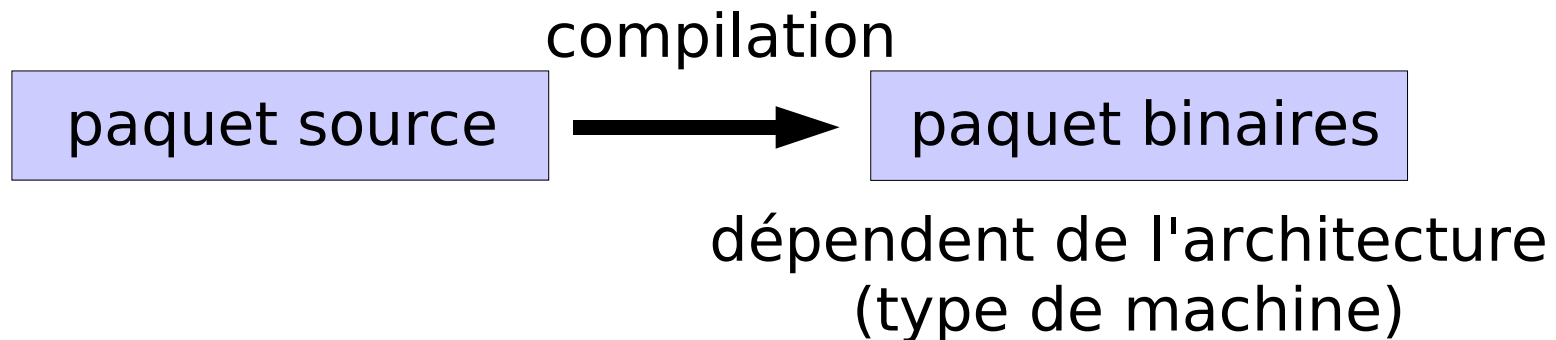
debian  
ubuntu



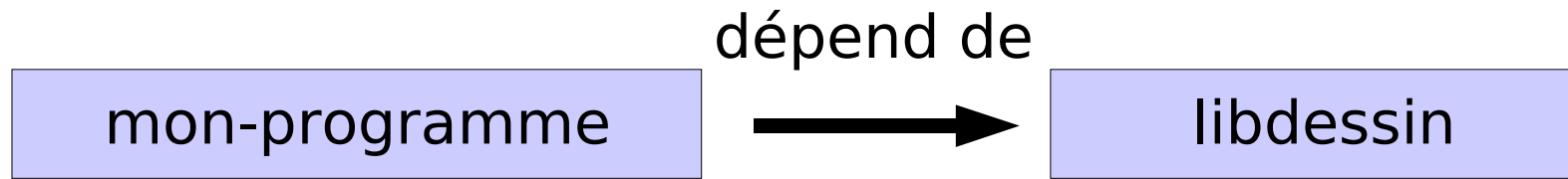
fedora  
mandriva

**paquet:** archive contenant des fichiers et des instructions pour les installer

comme une archive .tar  
+ quelques informations

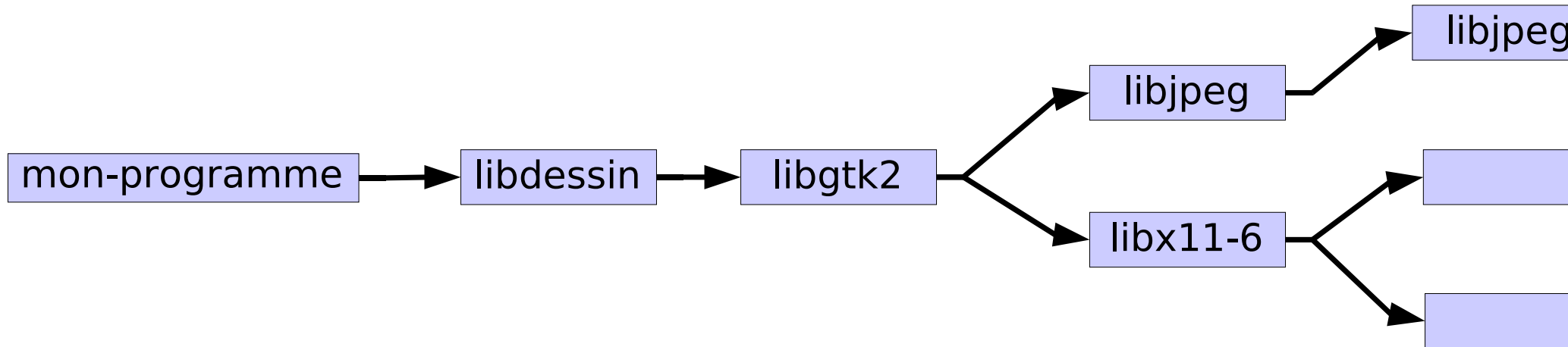


# dépendances entre paquets



pour installer **mon-programme**, il faut  
d'abord installer **libdessin**

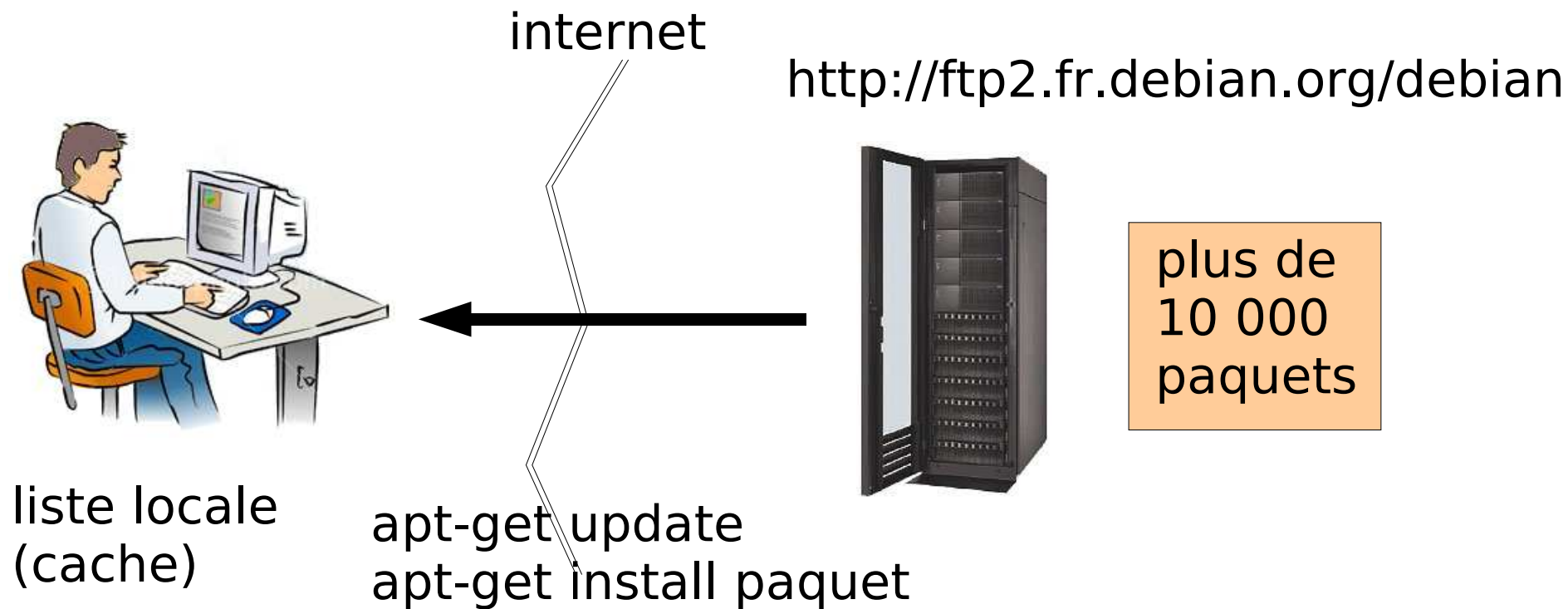
---



dépendances en chaîne!



# ou trouver des paquets?



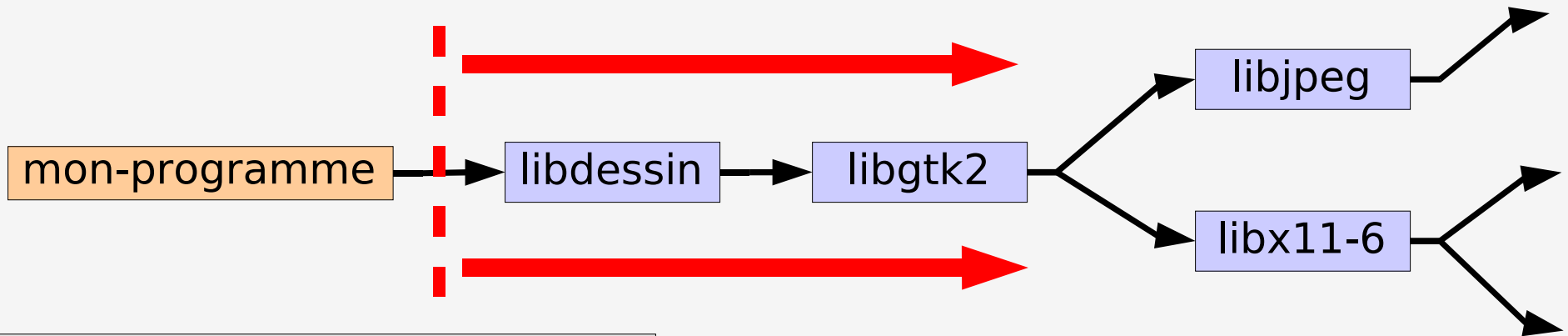
/etc/apt/sources.list

```
...  
deb http://ftp2.fr.debian.org/debian/ testing main  
...
```

aussi sur  
cd/dvd

# gestion de paquets

gestion des dépendances:



```
apt-get install mon-programme
```

mise à jour

versions:

mozilla-firefox-1.0.4-2

```
apt-get dist-upgrade
```

interfaces graphiques  
debian: synaptic

debian: apt-get  
mandriva: urpmi  
fedora: yum



# partie pratique



1ère partie

quelques commandes

# la commande: **top**

*afficher la liste de processus interactivement*

PID	USER	NI	RES	S	%CPU	%MEM	TIME+	COMMAND
3848	root	0	88m	S	44.5	4.4	9:29.61	XFree86
4401	bosc	0	13m	S	2.9	0.7	0:18.05	gnome-termi
4487	bosc	0	133m	S	0.6	6.6	12:33.99	soffice.bin
3372	root	0	2620	S	0.2	0.1	0:12.40	cupsd
4408	bosc	0	8280	S	0.2	0.4	0:07.54	clock-applet
20459	bosc	0	55m	S	0.2	2.8	0:06.43	acroread
20881	bosc	0	1092	R	0.2	0.1	0:00.45	top
1	root	0	512	S	0.0	0.0	0:00.60	init
2	root	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	19	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	-10	0	S	0.0	0.0	0:03.30	events/0
5	root	-10	0	S	0.0	0.0	0:00.00	khelper
6	root	-10	0	S	0.0	0.0	0:00.00	kacpid
60	root	-10	0	S	0.0	0.0	0:00.21	kblockd/0

# la commande: **cat**

*afficher un ou plusieurs fichier*

très simple!

essai.txt

```
bonjour toto  
bla bla bla
```

```
[dupond@r10102 ~/essai]# cat essai.txt  
bonjour toto  
bla bla bla  
[dupond@r10102 ~/essai]# cat essai.txt | grep bla  
bla bla bla  
[dupond@r10102 ~/essai]#
```

# suspension d'un processus

control-z : suspend  
la tâche courante

```
[dupond@r10102 ~/essai]# gedit
```

control-z

```
[2]+ Stopped gedit
```

suspension

```
[dupond@r10102 ~/essai]# bg
```

```
[2]+ gedit &
```

bg: background  
passer en tâche  
de fond

```
[dupond@r10102 ~/essai]# fg
```

fg: foreground  
retour au premier plan  
(on perd la main)

# la commande: **find**

rappel

*rechercher des fichiers dans une arborescence*

rechercher les fichiers terminant par .jpg

```
find . -iname "*.jpg"
```

rechercher les fichiers .h contenant "color" dans /usr/include

```
find /usr/include -iname "*color*.h"
```



commande très utile!

# la commande: **find**

**-exec** : exécuter une commande à chaque fichier trouvé

```
find . -iname "*.jpg" -exec cp {} ~/images \;
```

la commande à exécuter



{ } est remplacé par le nom  
du fichier trouvé



\; fin de la commande



exemple:

« find » trouve deux fichiers gnome.jpg et tux.jpg :

```
cp gnome.jpg ~/images  
cp tux.jpg   ~/images
```

# la commande: **tar**

*manipuler des archives*

désarchiver

le nom du fichier archive  
d'ou on veut extraire

```
[dupond@r10102 ~]# tar xzvf archive.tar.gz
```

**x**:extraire

**f**: fichier archive

**z**: compresser

**v**: verbose



# la commande: **tar**

*manipuler des archives*

créer une archive

le nom du fichier archive  
qu'on va créer

le répertoire qu'on  
veut archiver

```
[dupond@r10102 ~]# tar czvf archive.tar.gz repertoire
```

**c**: créer nouvelle  
archive

**z**: compresser

**f**: fichier archive

**v**: verbose

# la commande: **sort**

*trier des données*

texte.txt

```
bonjour  
au revoir  
courir
```

numeros.txt

```
100 /home  
30  /usr  
40  /var
```

```
[dupond@r10102 ~/essai]# sort texte.txt
```

```
au revoir
```

```
bonjour
```

```
courir
```

tri alphabétique

```
[dupond@r10102 ~/essai]# sort -n numeros.txt
```

```
30  /usr
```

```
40  /var
```

```
100 /home
```

n: tri numérique

```
[dupond@r10102 ~/essai]#
```

# la commande: cut

*afficher certains champs d'une ligne*

numeros.txt

```
100 /home,x  
30  /usr ,y  
40  /var ,z
```

**d**: délimiteur  
entre champs

**f**: numéro  
du champs

```
[dupond@r10102 ~/essai]# cut -d ' ' -f 1 numeros.txt  
100  
30  
40  
[dupond@r10102 ~/essai]# cut -d ',' -f 2 numeros.txt  
x  
y  
z  
[dupond@r10102 ~/essai]#
```

# la commande: **du**

*disk usage: place disque occupée*

récuratif

```
[root@r10102 ~/systeme]# du
24      ./tp1
272     ./tp2/fichiers-tp2
288     ./tp2
316     .
[root@r10102 ~]#
```

le répertoire courant  
et ses sous-répertoires  
occupent 316 KO

options:

- h: facile à lire
- s: juste le total

ce document est distribué librement :

- sous licence GNU FDL :

<http://www.gnu.org/copyleft/fdl.html>

- les originaux sont disponibles aux formats OpenOffice et powerpoint

<http://www-info.iutv.univ-paris13.fr/~bosc>