

Systeme

Cours n°1

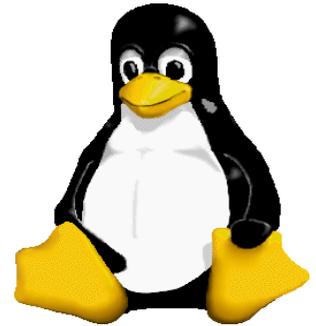
**Isis TRUCK,
Université Paris 8**

Remerciements

Ce cours est très fortement inspiré du cours de Marcel Bosc.

Les originaux sont disponibles aux formats OpenOffice et Powerpoint :

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



Avant-propos

- 12 (ou 15) séances = 11 (ou 14) CM/TP et 1 examen
- **présence obligatoire**
- Le cours est à lire **seul**, on se concentrera en séance surtout sur les exercices

Plan de la 1ère partie

- Qu'est-ce qu'un ordinateur?
- le matériel
- les systèmes d'exploitation
- fichiers et répertoires

Partie pratique

- environnement graphique
- logiciels importants
- la ligne de commande
- où trouver de l'aide ?

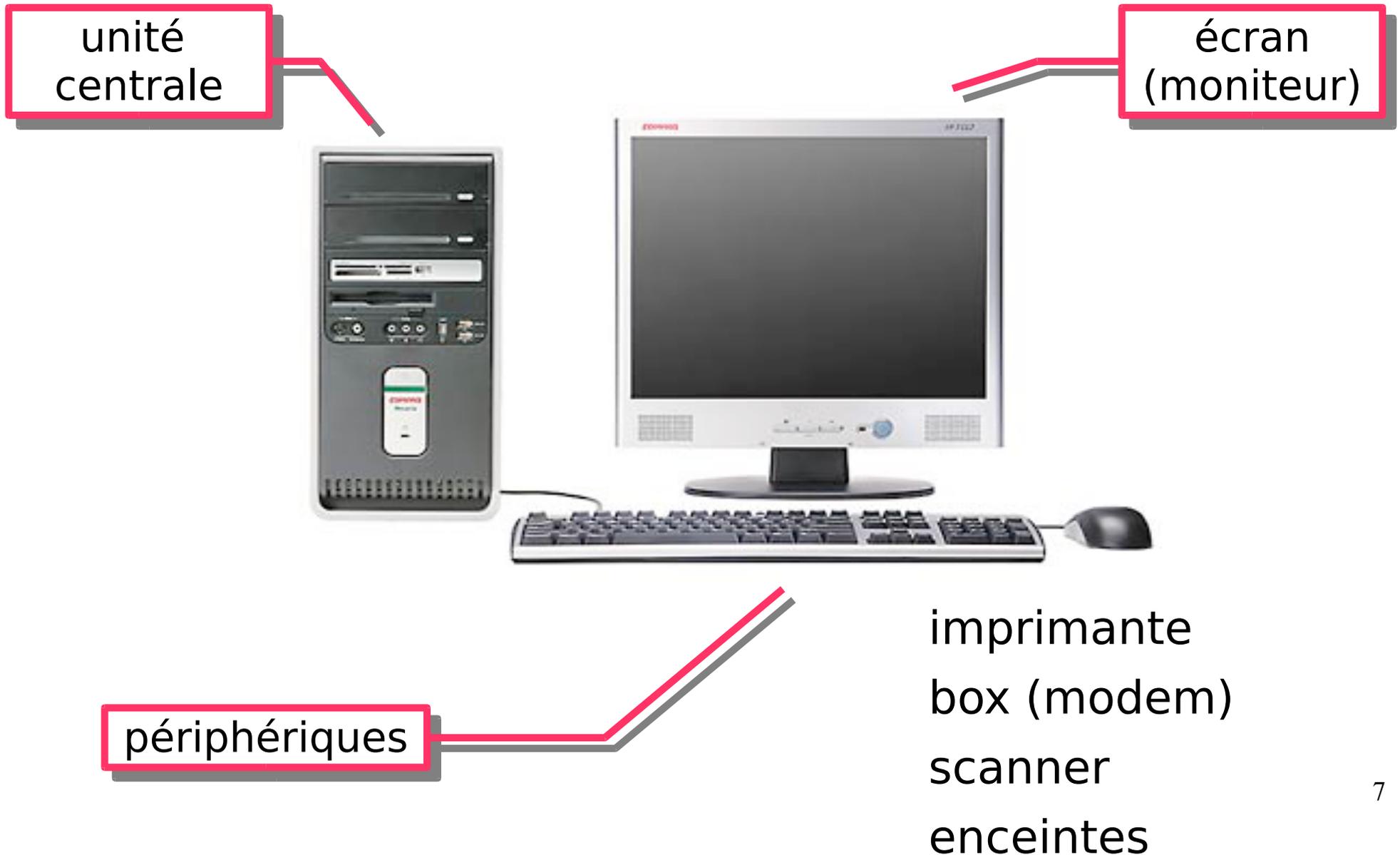
Qu'est-ce qu'un ordinateur?



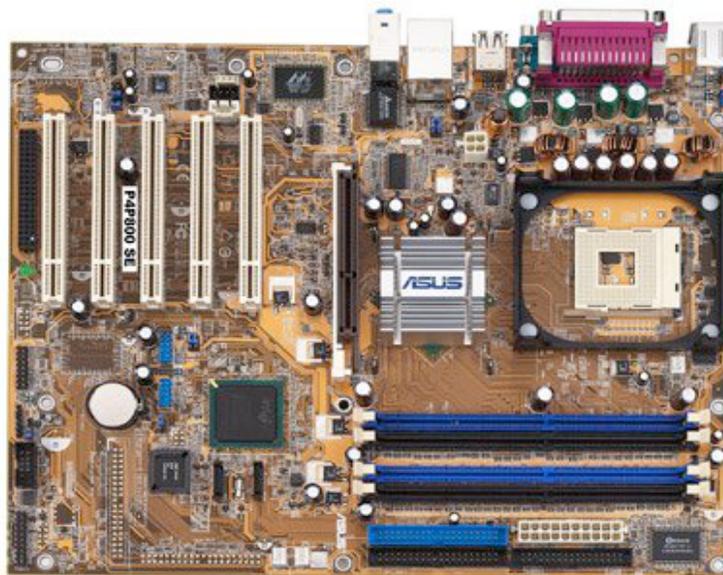
différents types d'ordinateurs



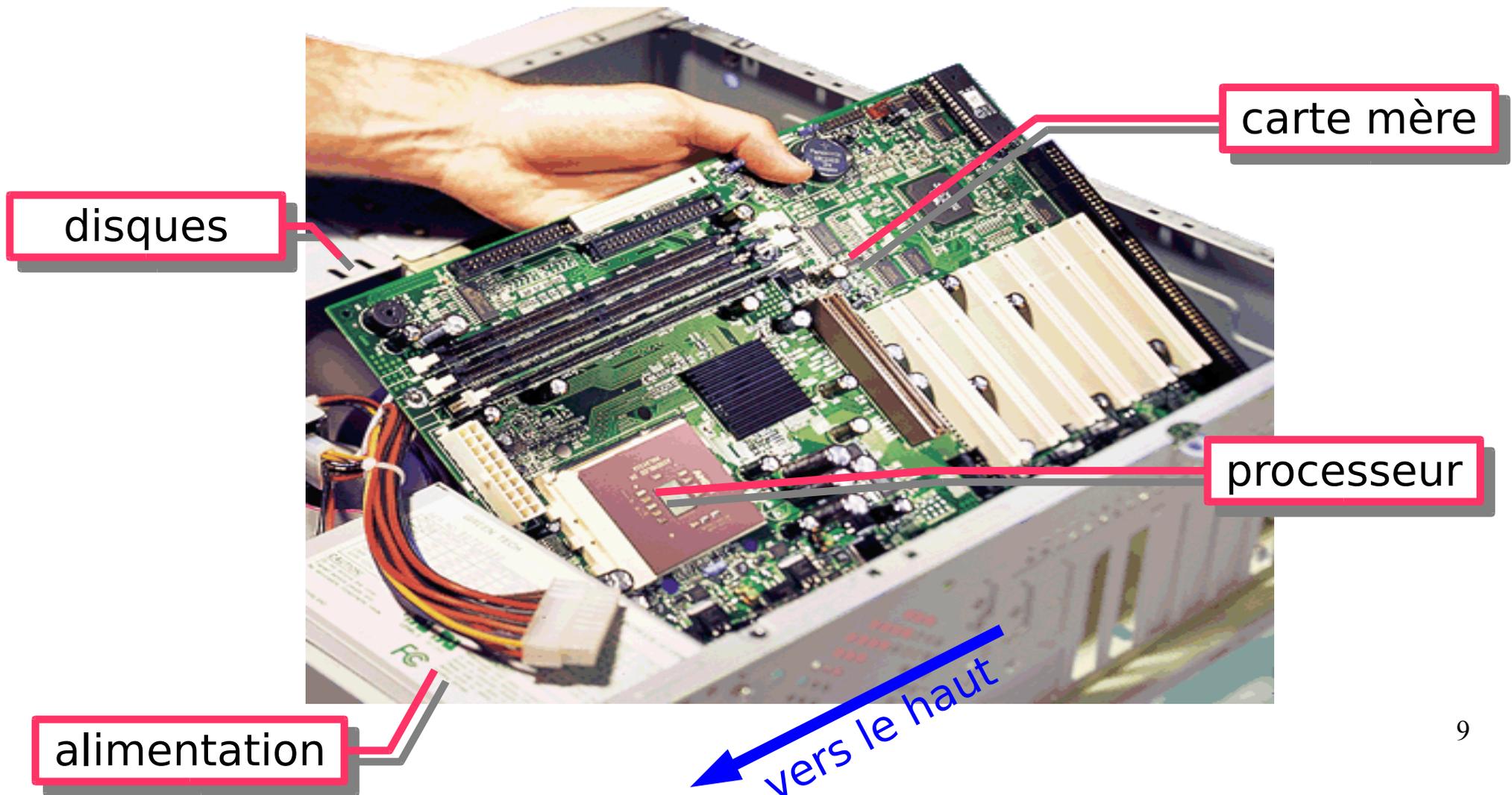
l'ordinateur de bureau



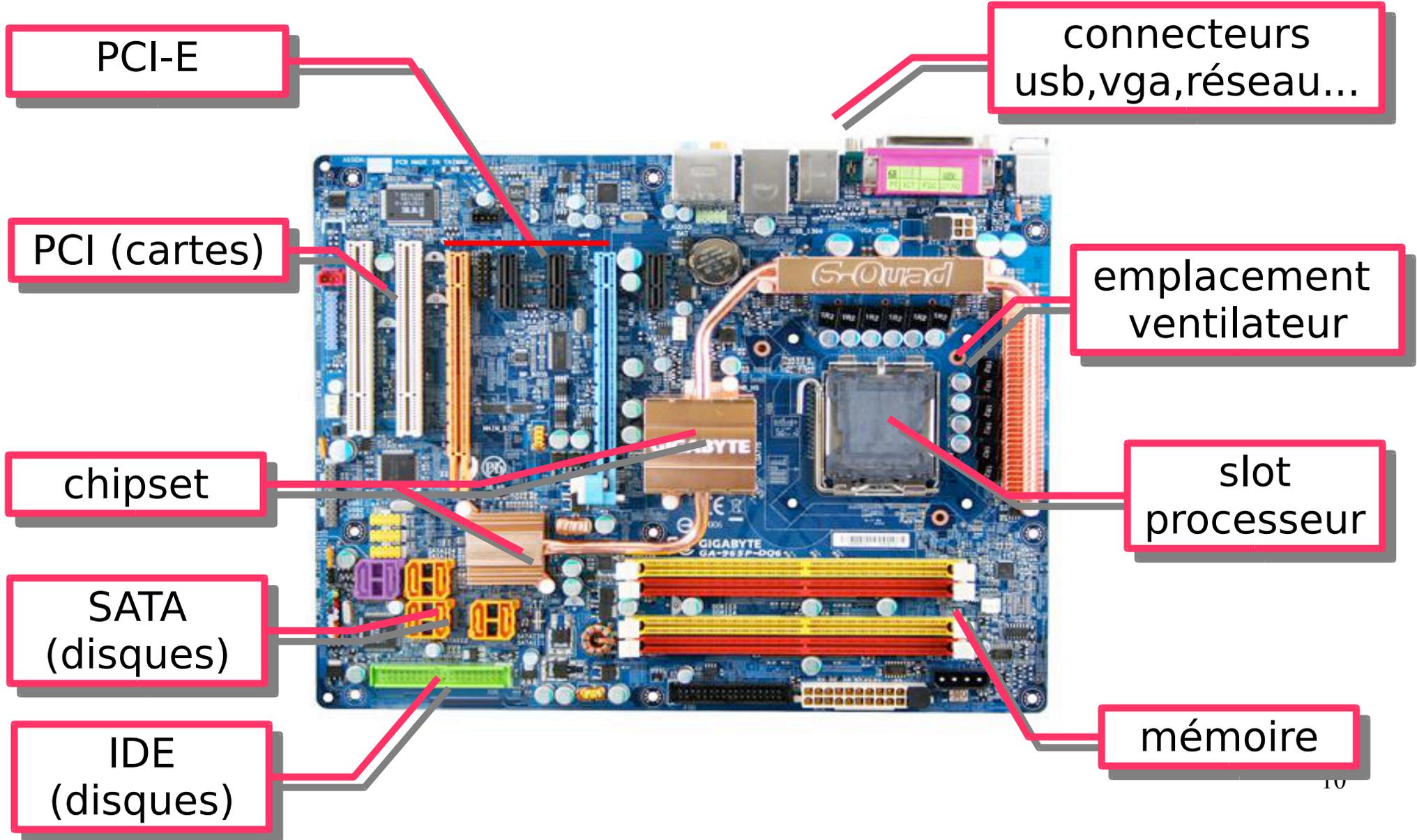
Le matériel



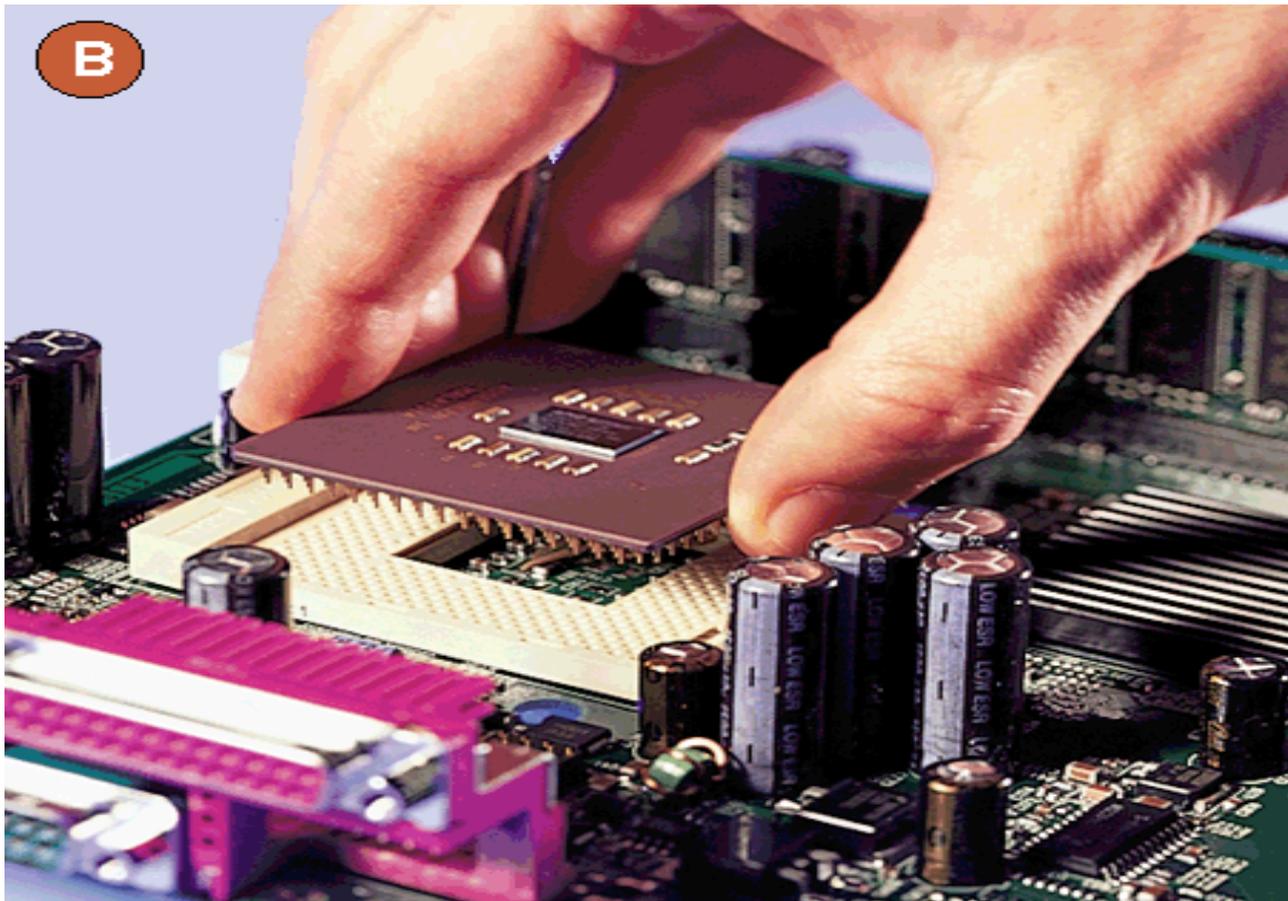
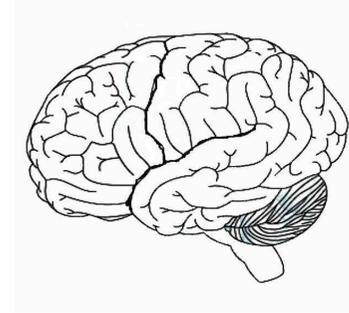
Dans le boîtier



La carte mère



Le processeur



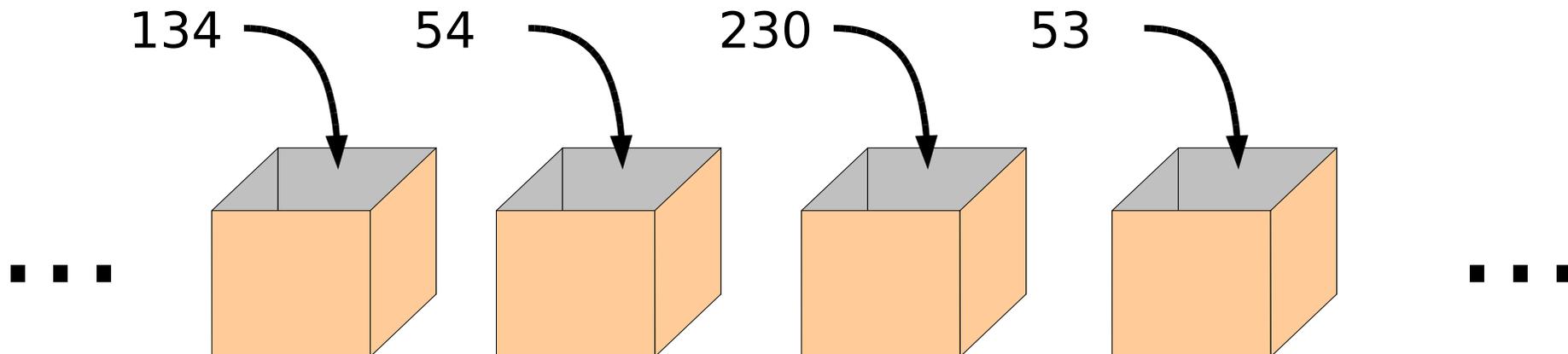
instructions
élémentaires

3,5 gigahertz (GHz)

1 gigahertz : 1
milliard de **cycles**
d'horloge par
seconde

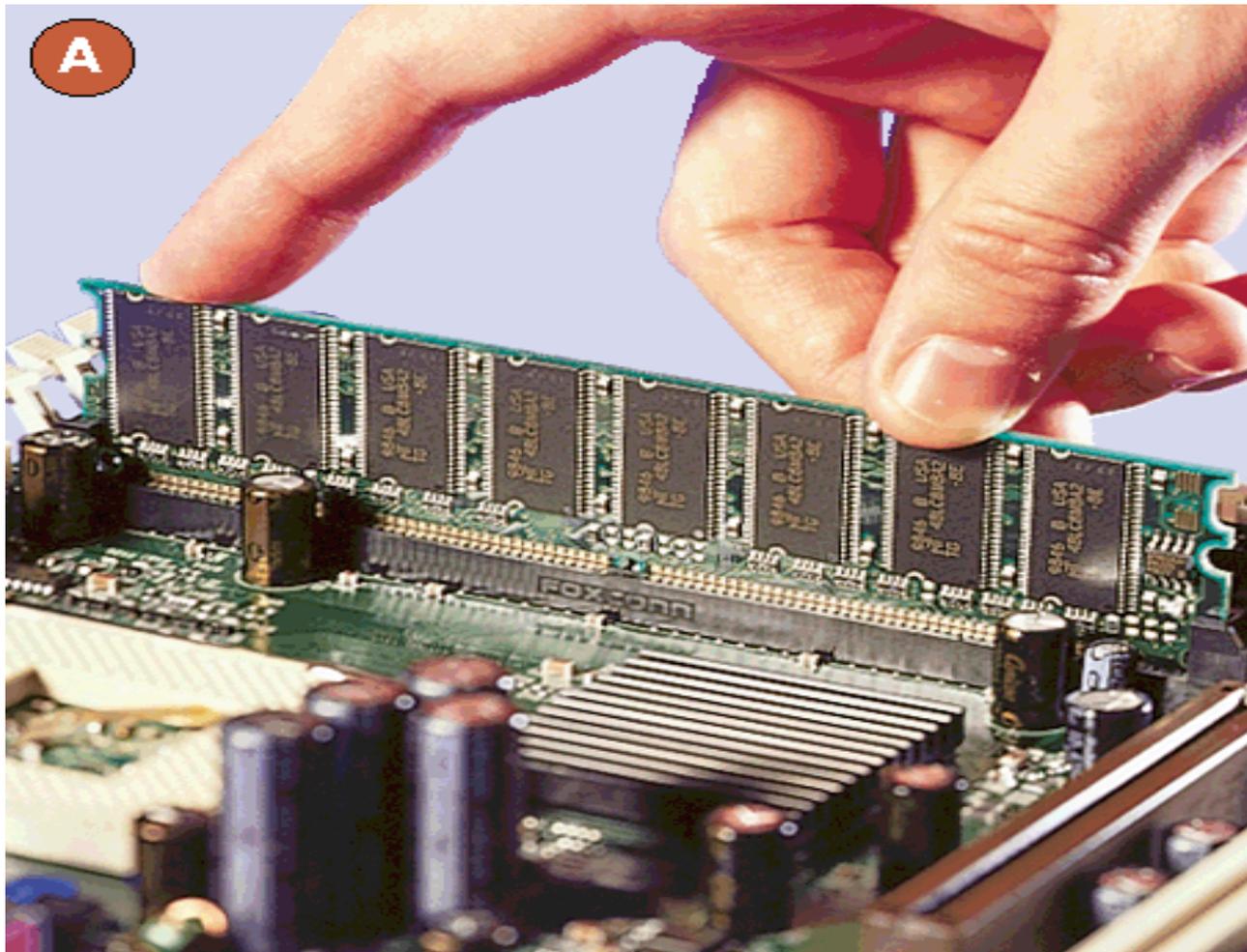
mémoire et octets

octet: un petit bout d'information pouvant représenter un nombre entre 0 et 255



mémoire

la mémoire vive (RAM)



- 8 gigaoctets

1 gigaoctet (Go):
1 milliard d'octets

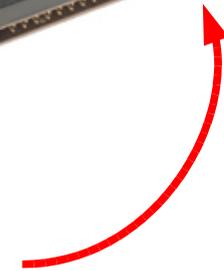
- volatile

Le disque dur

2 To (téraoctet - 1To = 10^{12} octets)

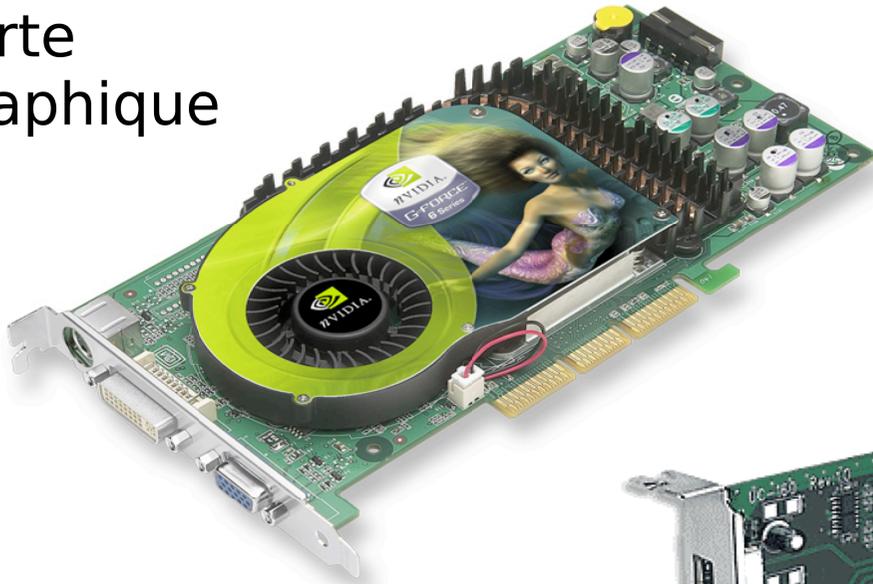


SATA ou IDE



Les cartes

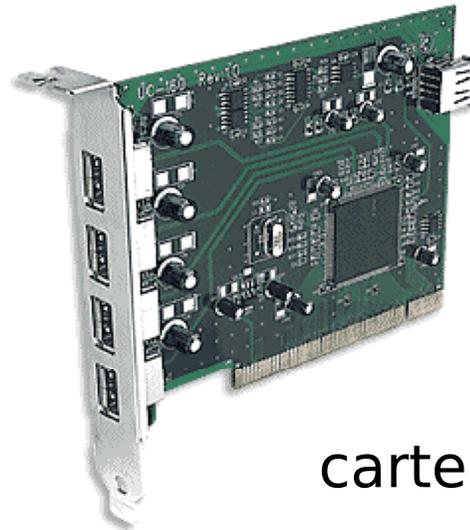
carte
graphique



carte réseau

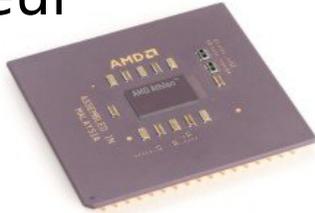


carte USB

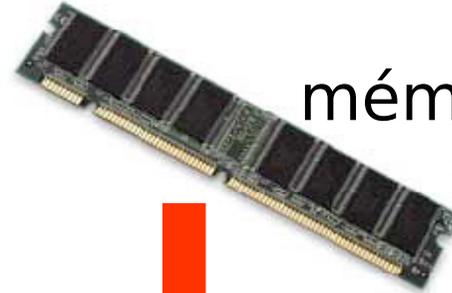


Fonctionnement

processeur



mémoire vive
(RAM)



bus

sur la carte
mère

carte graphique



afficher



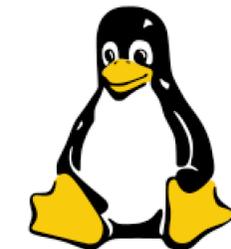
écran



disque-dur

enregistrer

Les systèmes d'exploitation



GNU/Linux

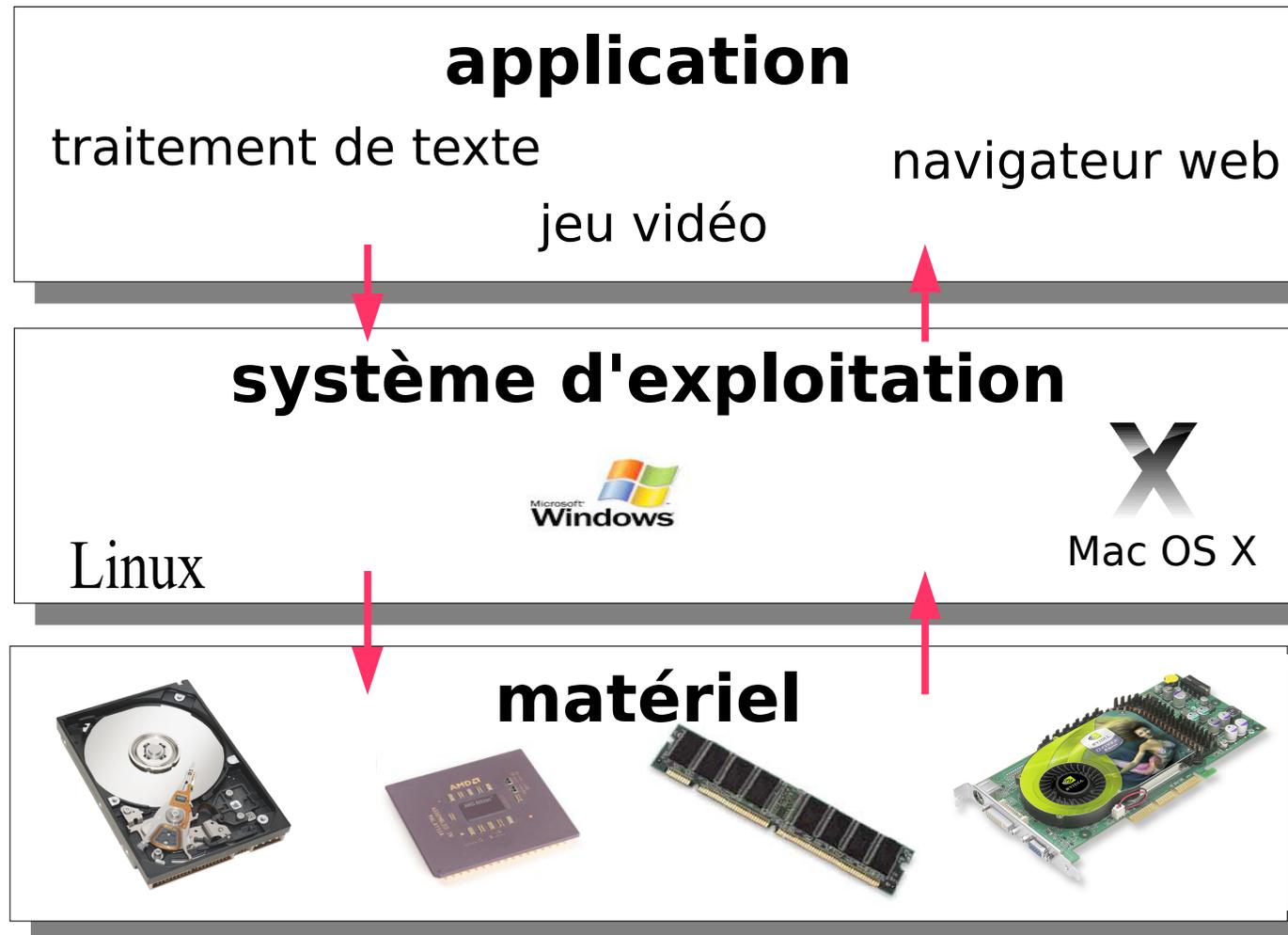


Mac OS X



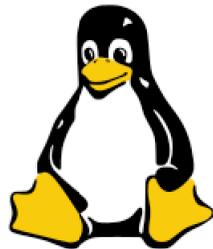
ANDROID

Qu'est-ce qu'un système d'exploitation?



systemes d'exploitation: exemples

logiciel libre



GNU/Linux

distributions:

ubuntu

debian

mandriva

fedora redhat

gentoo, ...



Mac OS X



XP, Vista, 7
Server...



ANDROID

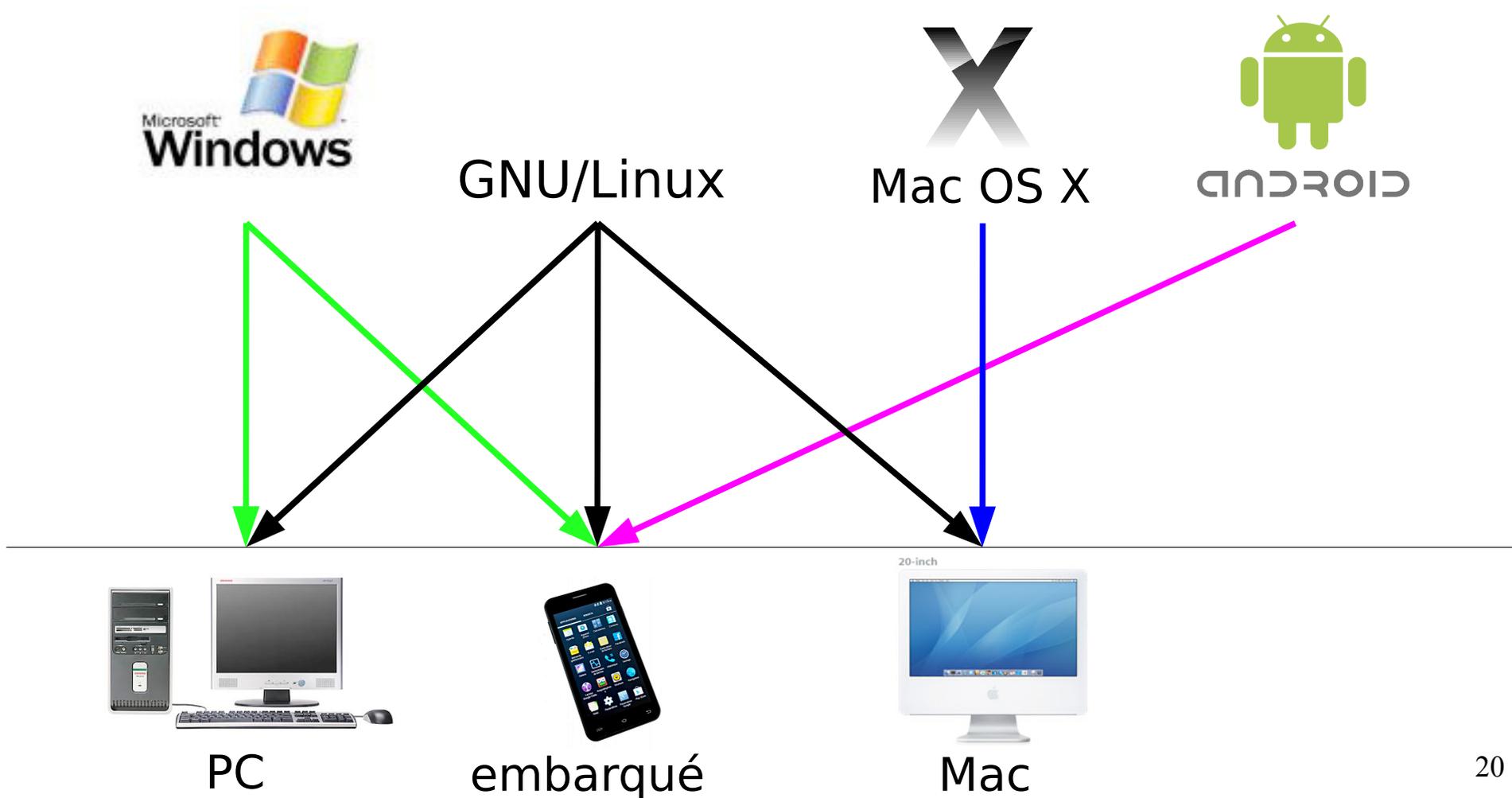
Linux *logiciel libre*



FreeBSD

logiciel libre

systemes d'exploitation ≠ plateforme matérielle



fichiers et répertoires

fichiers



suite d'octets
stockés sur un
support.

*Par exemple sur
un disque dur.*

tout est fichier!

- programmes
- documents
- texte
- fichiers de configuration
- ...

identifiés par un **nom**

noms de fichier

utilisez des noms explicites!

plus de 100 000 fichiers sur une machine

mauvais 😞

bon 😊

e1.c

fichier.odt

programme.java

toto.sh

exercice-1.c

notes-de-cours.odt

Exercice4.java

essai.sh

évités: espace, ponctuation, ^'!*%*#& ...

suffixes des noms de fichier

quel type de fichier?

mon-programme.c

page-web.html

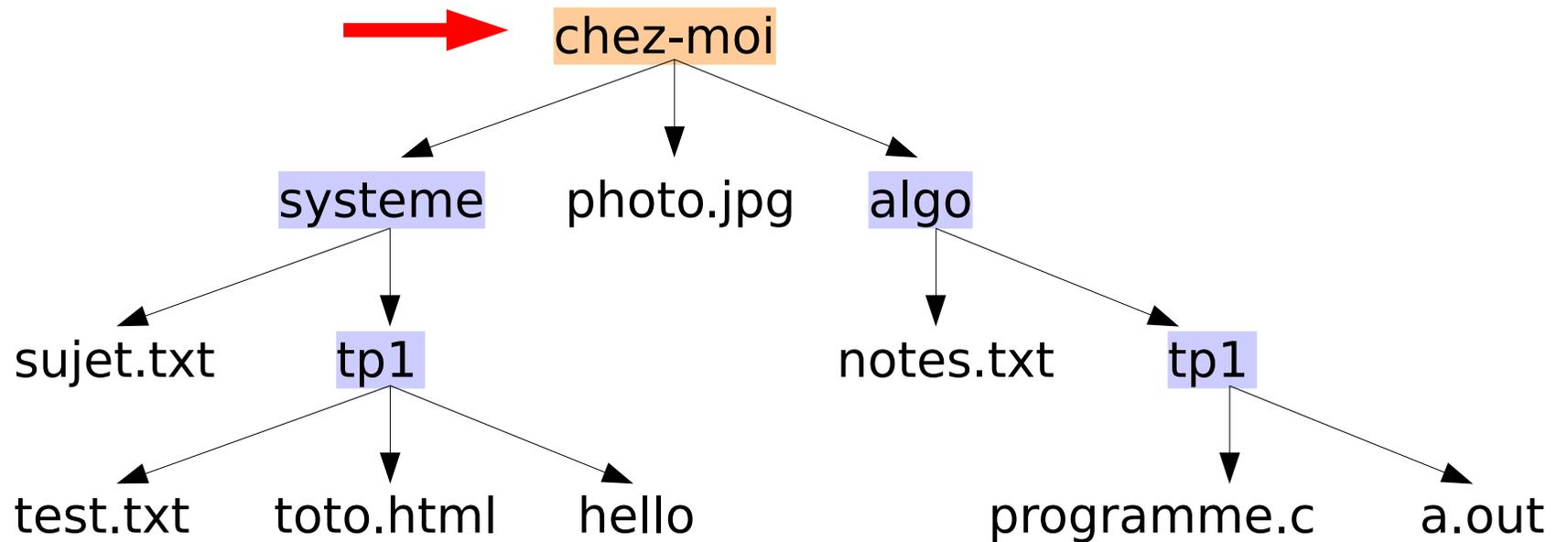
photo.jpg

musique.mp3

...

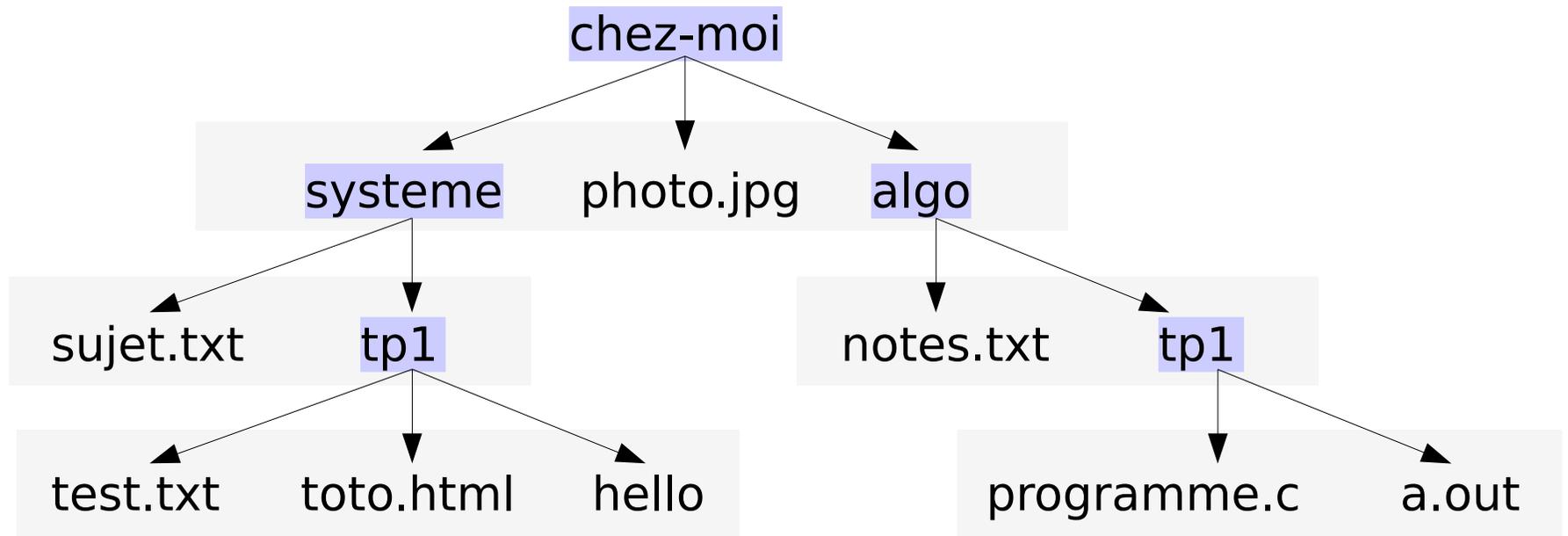
suffixe = extension

votre répertoire personnel



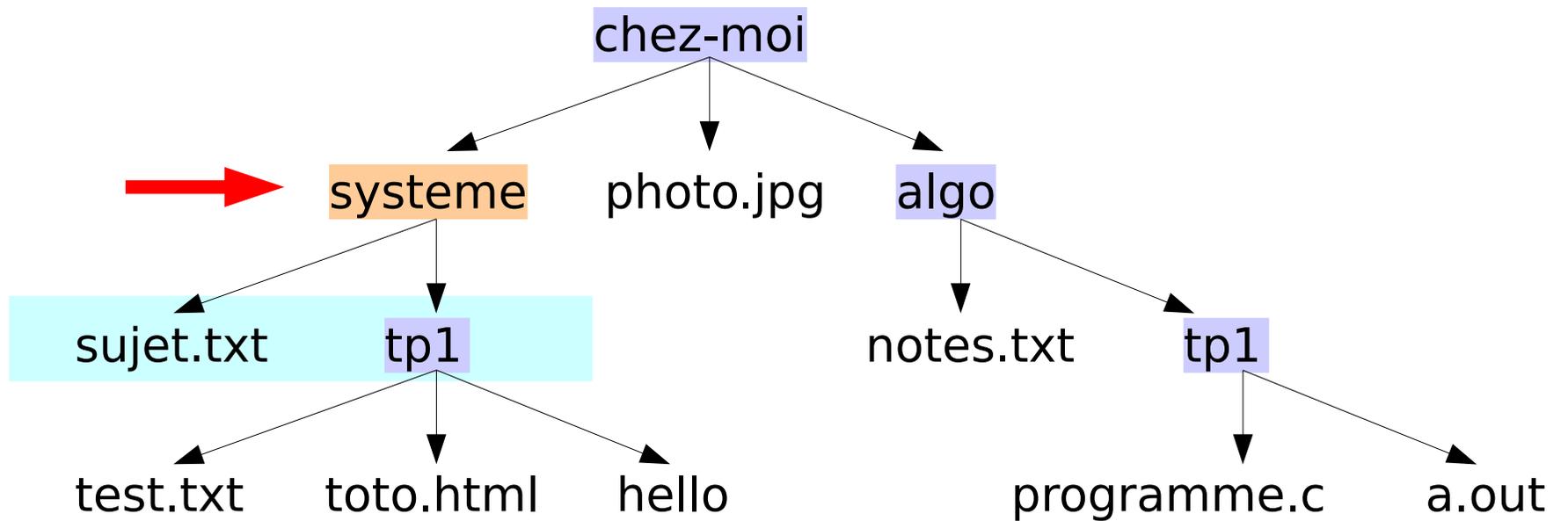
« répertoire personnel » = « home » = chez moi = ~

répertoires



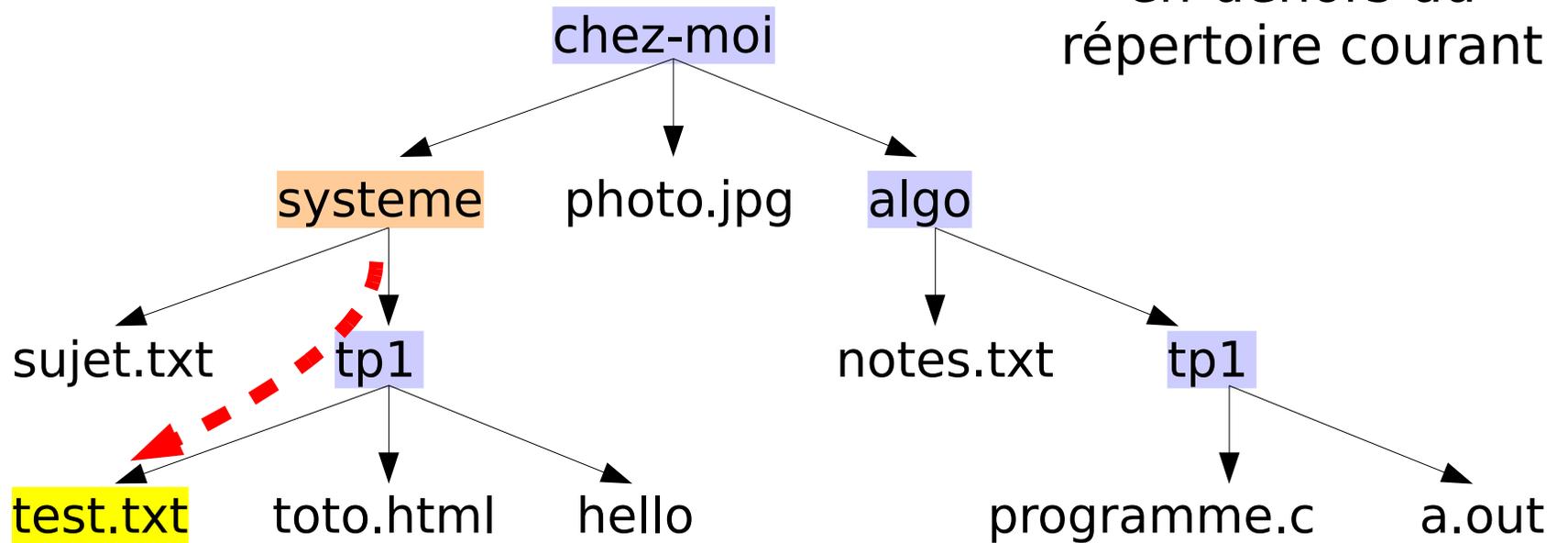
« dossier » = « répertoire »

le répertoire courant



chemins

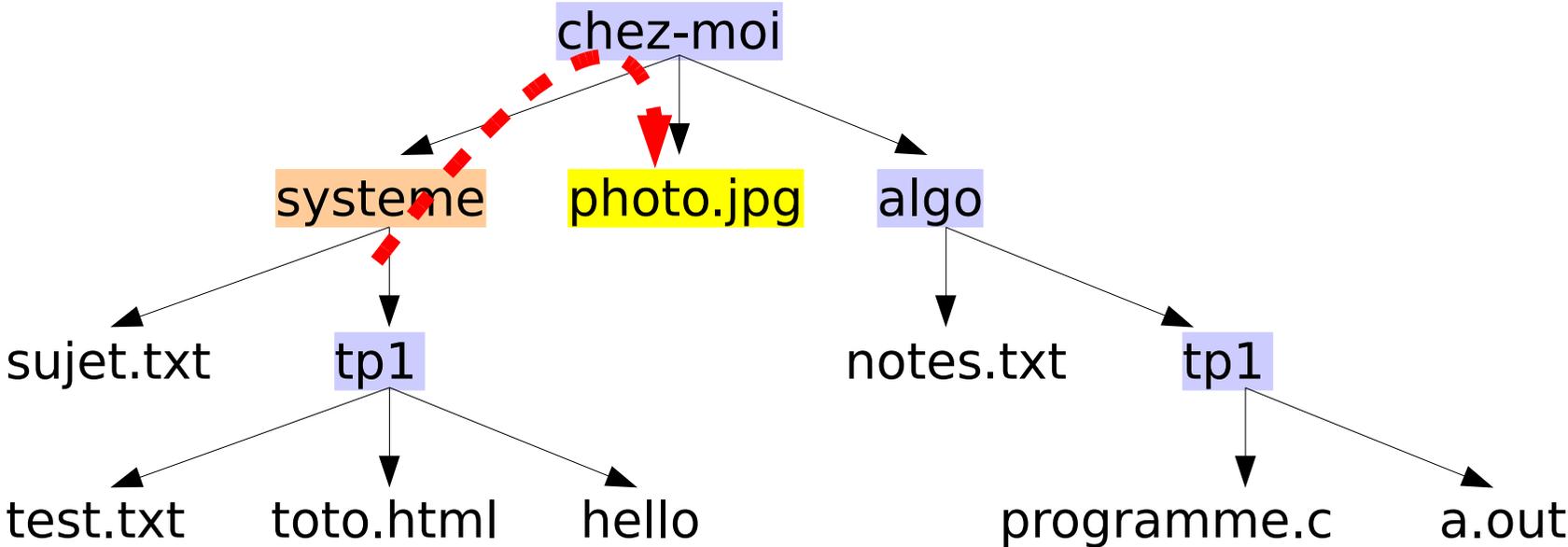
accéder à un fichier
en-dehors du
répertoire courant



chemin: `tp1/test.txt`

séparateur: `"/"`

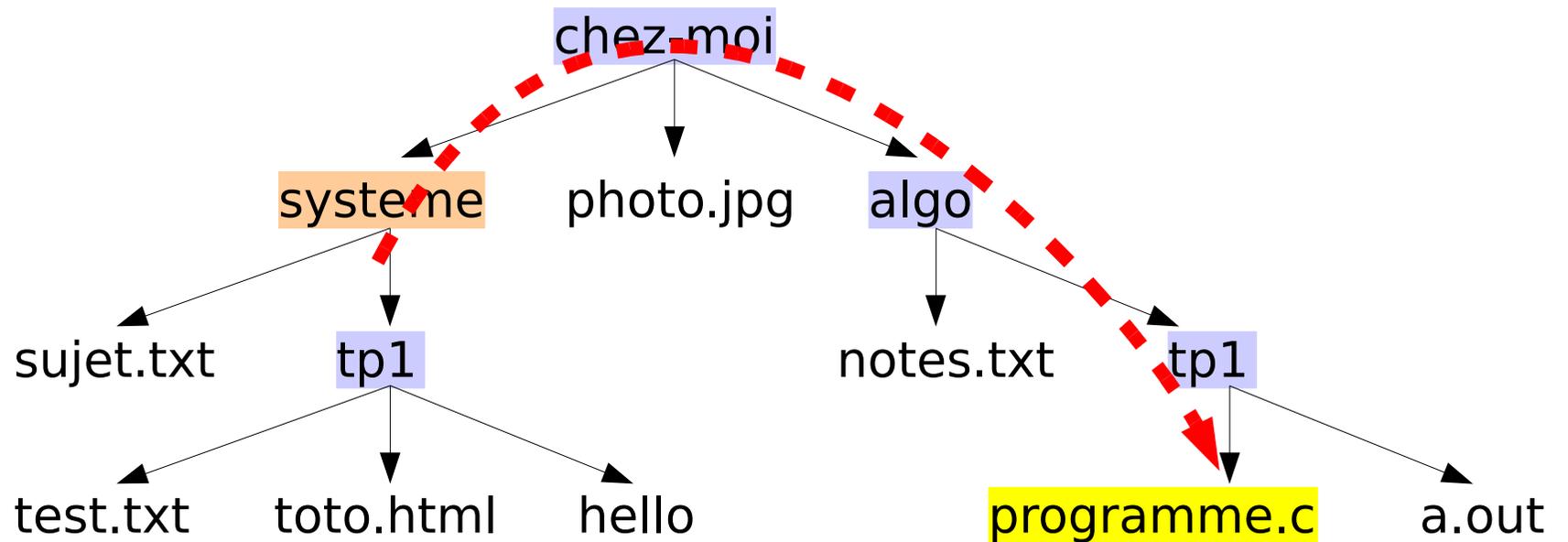
chemins



chemin: ../photo.jpg

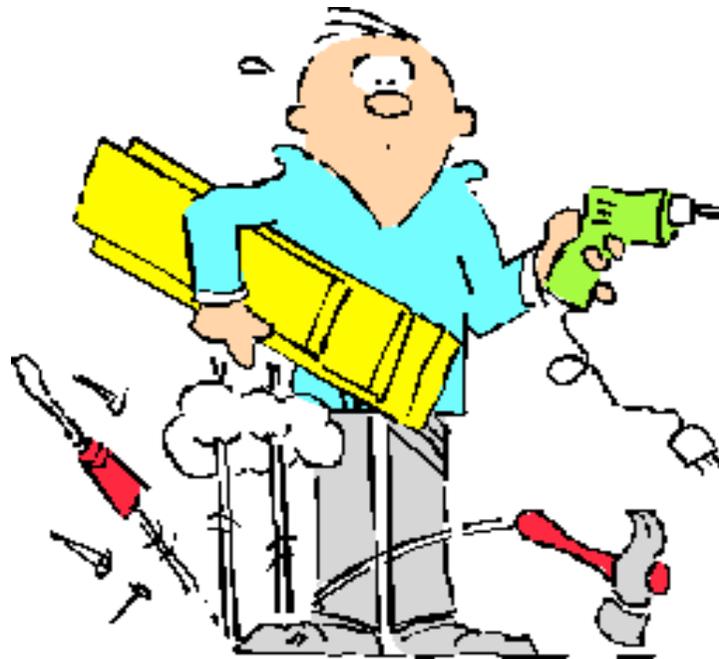
remonter: ".."

chemins



chemin: `../algo/tp1/programme.c`

partie pratique : environnement graphique

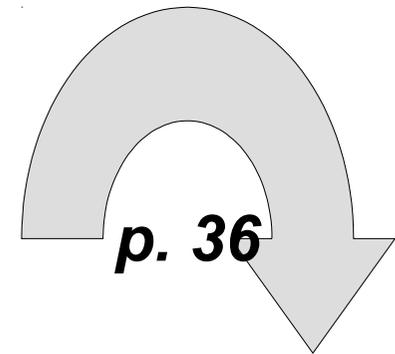


logiciels importants

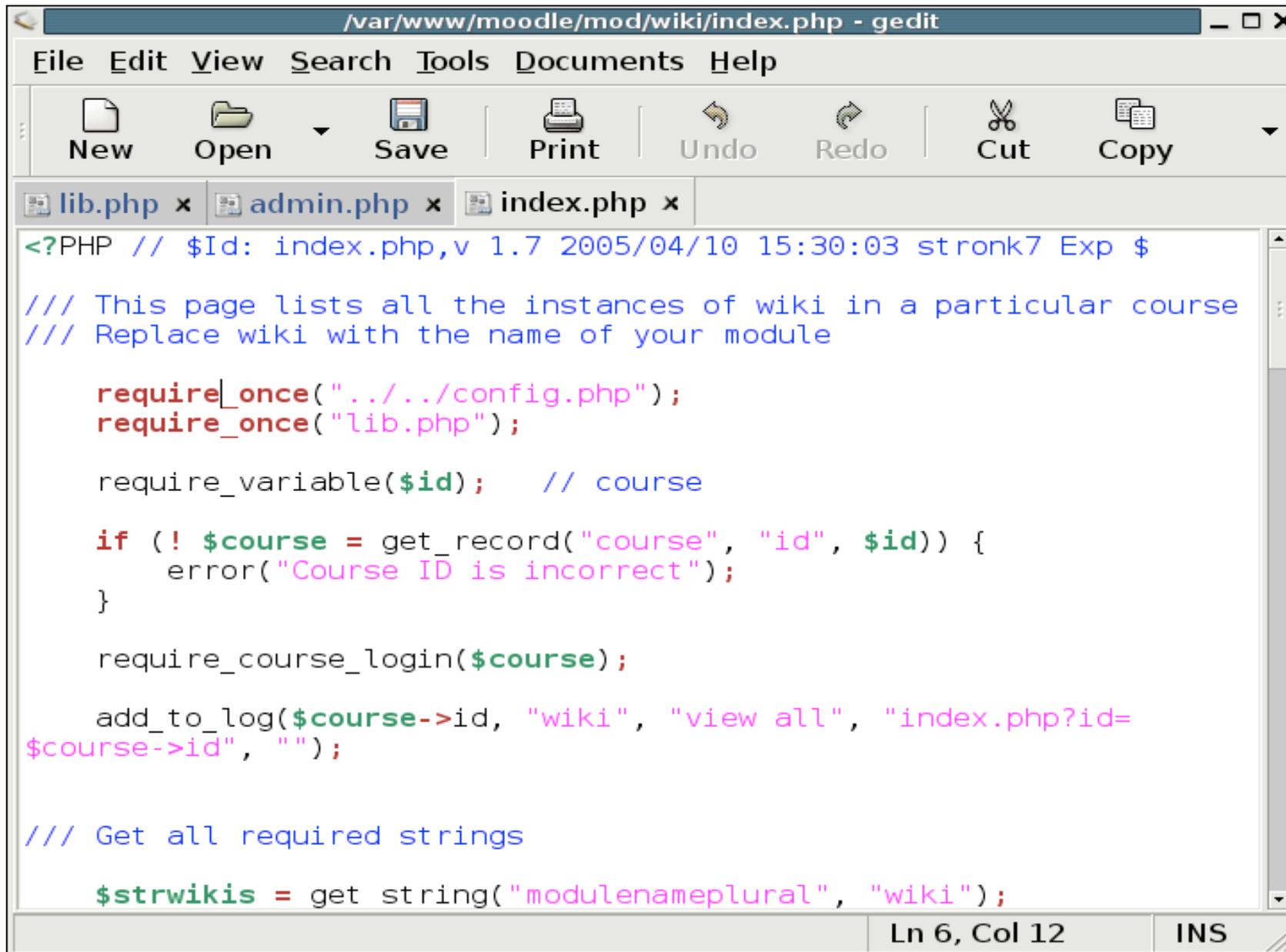
navigateur web

éditeur de texte

terminal



éditeur de texte



The image shows a screenshot of the gedit text editor. The window title is "/var/www/moodle/mod/wiki/index.php - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for New, Open, Save, Print, Undo, Redo, Cut, and Copy. The editor has three tabs: lib.php, admin.php, and index.php. The main text area contains PHP code for a Moodle wiki module. The code includes comments, require_once statements for config.php and lib.php, a require_variable call for \$id, an if statement to check if a course record exists, a require_course_login call, an add_to_log call, and a get_string call for the module name plural. The status bar at the bottom indicates "Ln 6, Col 12" and "INS" mode.

```
<?PHP // $Id: index.php,v 1.7 2005/04/10 15:30:03 stronk7 Exp $  
  
/// This page lists all the instances of wiki in a particular course  
/// Replace wiki with the name of your module  
  
require_once("../../config.php");  
require_once("lib.php");  
  
require_variable($id); // course  
  
if (! $course = get_record("course", "id", $id)) {  
    error("Course ID is incorrect");  
}  
  
require_course_login($course);  
  
add_to_log($course->id, "wiki", "view all", "index.php?id=  
$course->id", "");  
  
/// Get all required strings  
  
$strwikis = get_string("modulenameplural", "wiki");
```

outil de
base
très
utilisé!

texte
brut

traitement / éditeur de texte

traitement de texte:

documents avec mise en forme
lecture par un humain

- Open Office
- MS Word
- ...

éditeur de texte:

texte brut
lisible par un programme
utilisation:

- écrire des programmes
- configurer
- ...

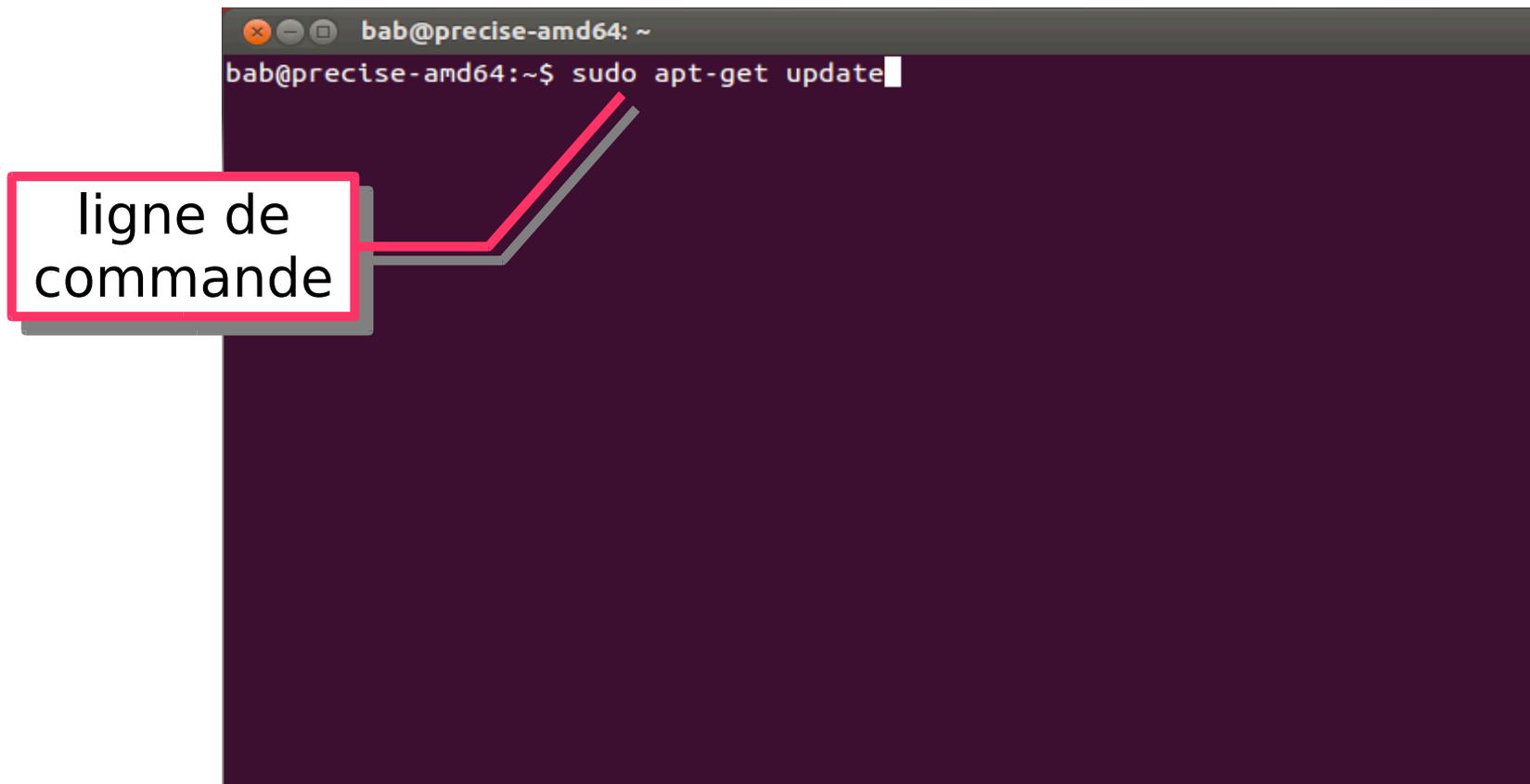
- gedit
- jedit
- vim (vi)
- xemacs
- nano



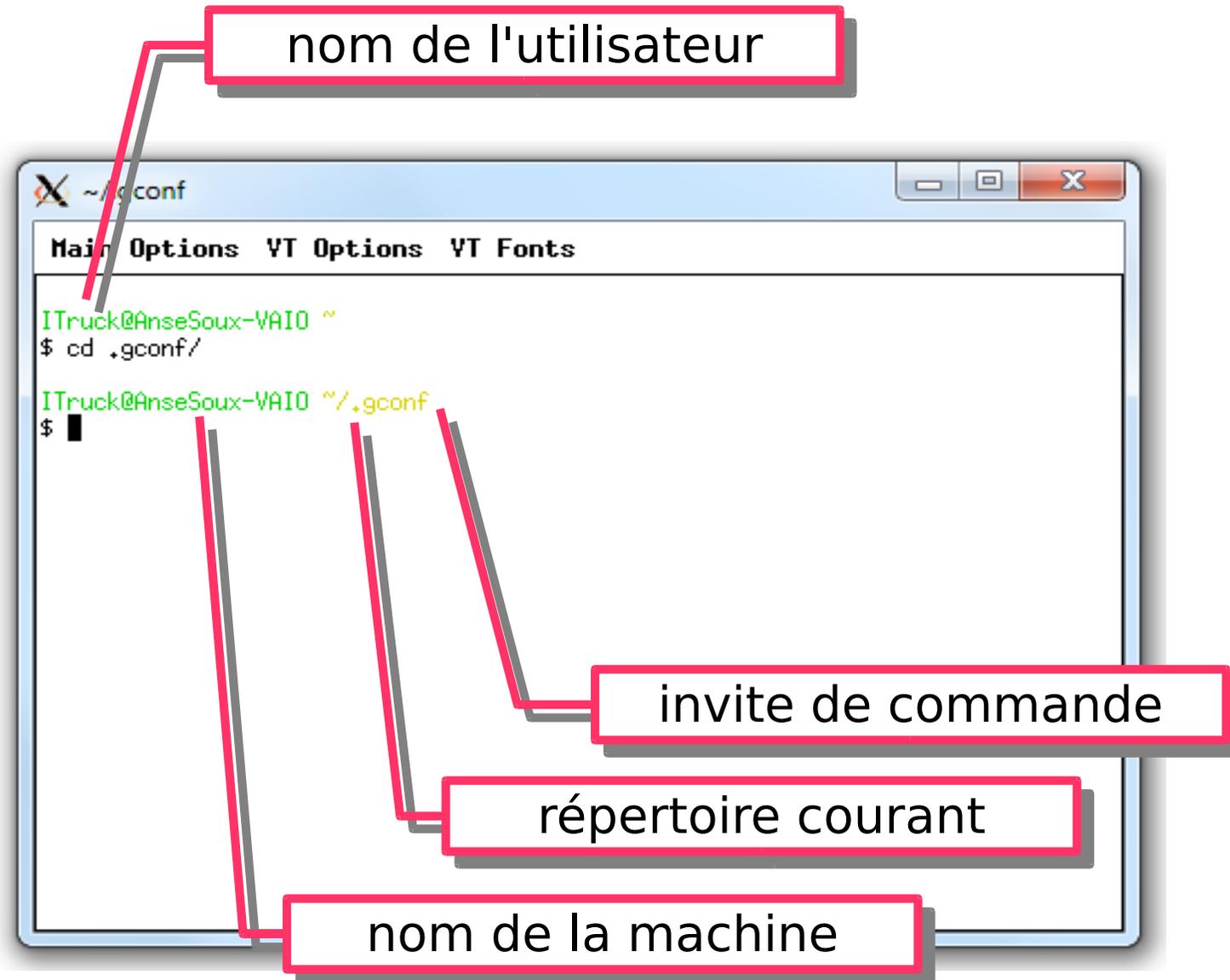
apprendre à bien
s'en servir ³⁵

le terminal

=> xterm, aterm, rxvt, konsole... selon l'environnement et la distribution de linux



le terminal: ligne de commande



la ligne de commande

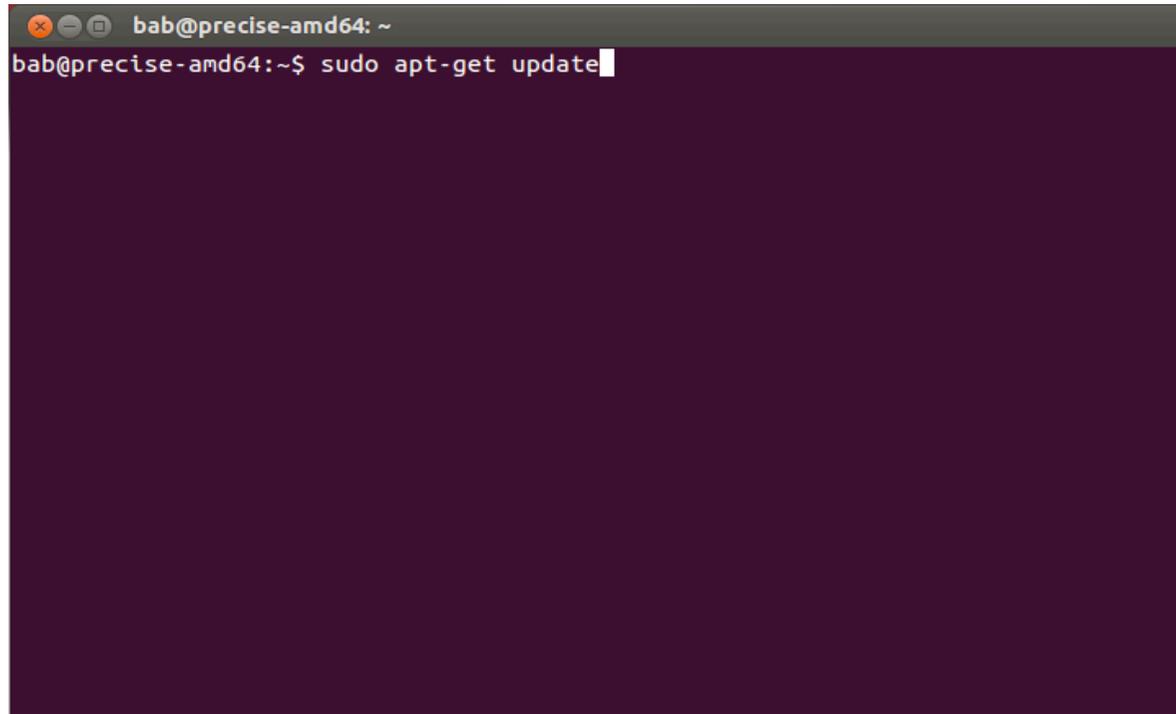
ligne de commande

principale interface au système
d'exploitation

- dialoguer avec le système d'exploitation
- manipuler des fichiers
- lancer des programmes
- administrer le système

très important 

ligne de commande : principe

A terminal window with a dark purple background. The title bar at the top shows a window icon, a close button, and the text 'bab@precise-amd64: ~'. The main area of the terminal shows the prompt 'bab@precise-amd64:~\$' followed by the command 'sudo apt-get update' with a white cursor at the end of the line.

```
bab@precise-amd64: ~  
bab@precise-amd64:~$ sudo apt-get update
```

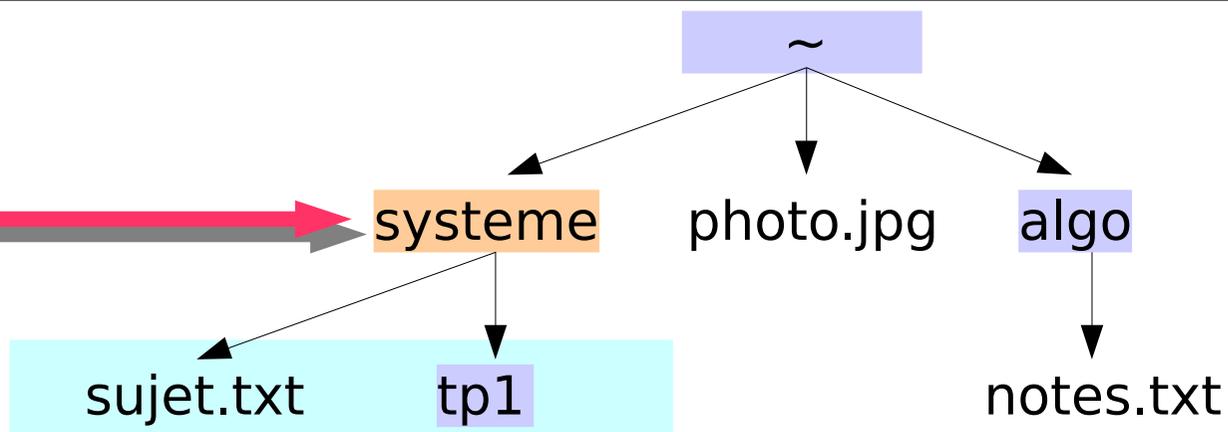
- 1) on tape une commande, suivie de la touche entrée
- 2) la commande s'exécute
- 3) parfois, la commande répond (texte)

première commande

la commande **ls**:
afficher le contenu du répertoire courant

répertoire courant

```
[dupond@r10102 ~/systeme]# ls
sujet.txt tp1
[dupond@r10102 ~/systeme]#
```



Testez-le !

la commande **cd**

cd: « change directory »
change le répertoire courant

```
[dupond@r10102 ~]# cd systeme
```

validez avec
« entrée »

répertoire
courant avant

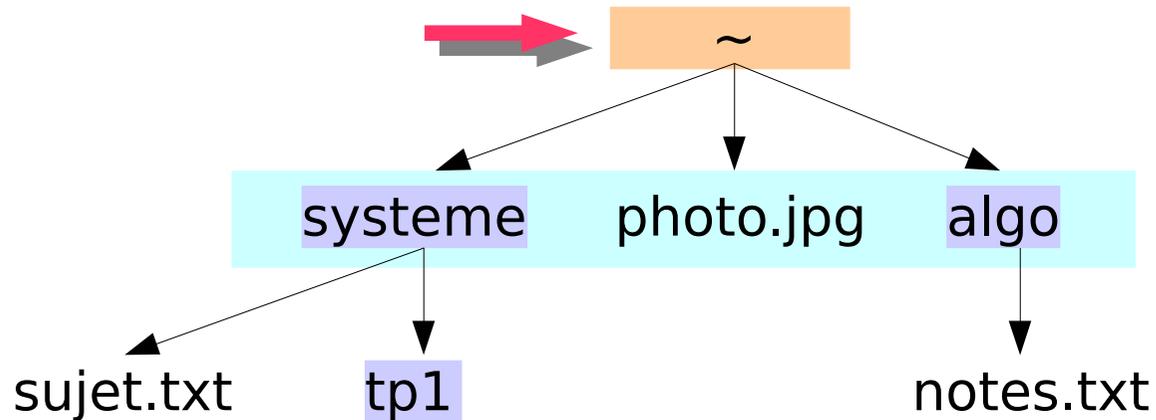


```
[dupond@r10102 ~/systeme]#
```

répertoire
courant après

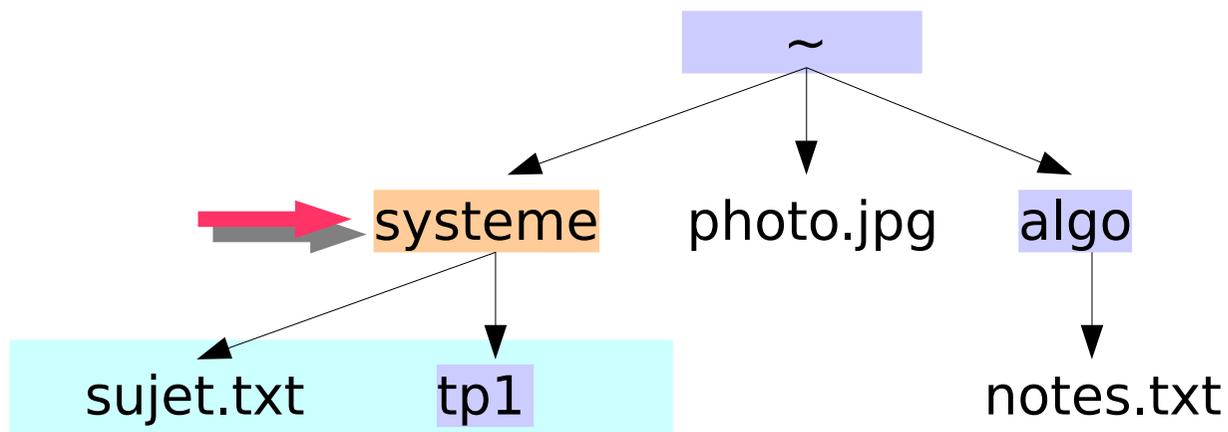
commande **cd** : explication

Avant



`cd systeme`

Après



la commande « **cd ..** »

.. = le répertoire
« au-dessus »

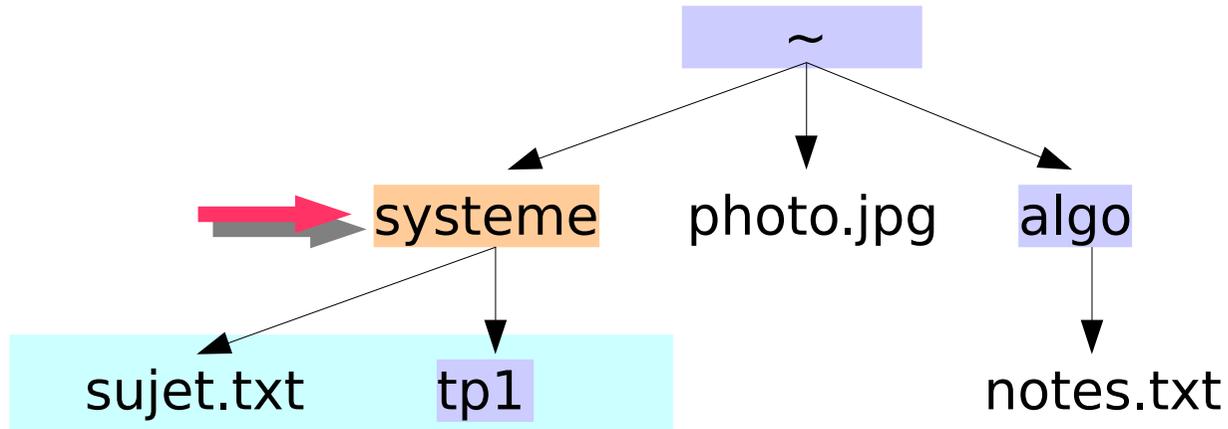
```
[dupond@r10102 ~/systeme]# cd ..  
[dupond@r10102 ~]#
```

« répertoire parent » = « au dessus » = ..

Testez-le !

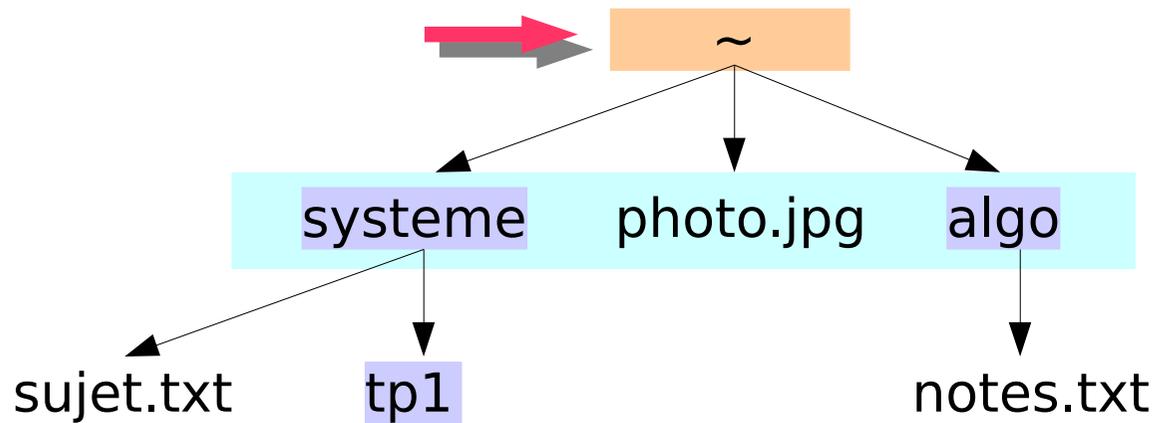
Commande `cd ..` : explication

Avant



`cd ..`

Après



messages d'erreur

```
[dupond@r10102 ~]# cd zozozo  
bash: cd: zozozo: Aucun fichier ou répertoire de ce  
type
```



message
d'erreur



lire attentivement les messages d'erreur

structure d'une commande

```
[dupond@r10102 ~]# cd systeme
```

nom de la commande

argument de
la commande

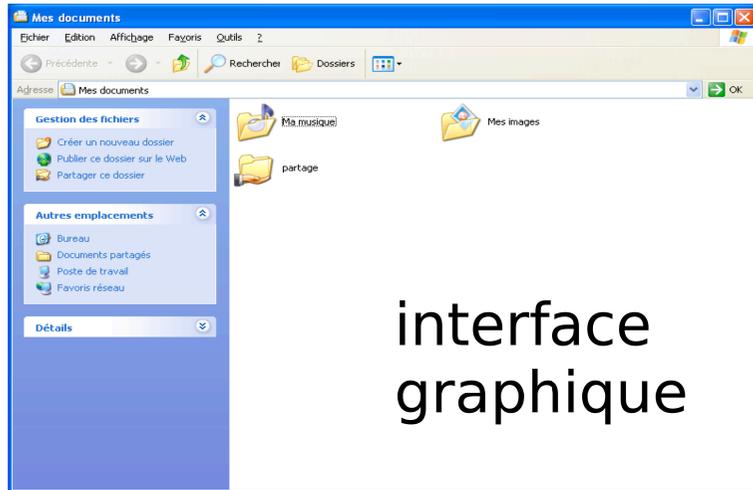
```
[dupond@r10102 ~]# rm -r toto titi zozo
```

nom de la commande
(rm = effacer)

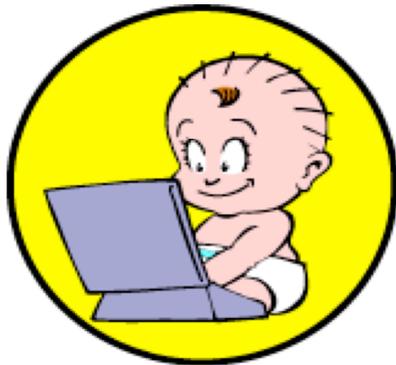
arguments de
la commande

option:
(-r = récursif)

interface graphique?



- plus simple
- non informaticiens
- rigide



```
[dupond@r10102 ~/public_html]# cd ..  
[dupond@r10102 ~]# cd zozozo  
bash: cd: zozozo: Aucun fichier ou répertoire de ce type  
[dupond@r10102 ~]# cd /etc  
[dupond@r10102 /etc]# cd  
[dupond@r10102 ~]# cd /etc  
[dupond@r10102 /etc]# cd ~  
[dupond@r10102 ~]# cd /etc  
[dupond@r10102 /etc]# cd ~/public_html
```

ligne de commande

- plus compliqué à apprendre
- informaticiens, professionnels
- *beaucoup* plus **puissant**
- programmation
- flexible



commandes de base

- cd** : (*change directory*) se positionne sur le répertoire désigné
- ls** : *liste* le contenu d'un répertoire
- man** : retourne le *mode d'emploi* (*manual*) de la commande s'il existe
- cp** : *copie* du fichier source vers la destination
- rm** : (*remove*) *supprime* la référence du fichier dans le répertoire
- rmdir** : *suppression* d'un répertoire vide
- pwd** : (*print working directory*) affiche le nom du répertoire *courant*
- mkdir** : (*make directory*) *crée* un répertoire
- less** : (*or more* – mais obsolète) affiche le contenu d'un fichier page par page

où trouver de l'aide



aide sur une commande: **man**

syntaxe: man nom-de-la-commande

```
[dupond@r10102 ~]# man ls
```



```
LS(1)          Manuel de l'utilisateur Linux          LS(1)

NOM
    ls, dir, vdir - Afficher le contenu d'un répertoire.

SYNOPSIS
    ls [options] [fichier...]

Options: [-labcdfgiklmnopqrstuvwxABCDEFGHNLNQRSUX]
        [--help] [--version] [--]

DESCRIPTION
    La commande ls affiche tout d'abord l'ensemble de ses arguments fichiers autres que des répertoires. Puis ls affiche l'ensemble des fichiers contenus dans chaque répertoire indiqué. Si aucun argument
    . . .
```

aide sur une commande: **man**

syntaxe: man nom-de-la-commande

```
[dupond@r10102 ~]# man ls
```



```
LS(1) Manuel de l'utilisateur Linux LS(1)
```

```
NOM
```

```
ls, dir,  
toire.
```

```
er-
```

**man est donc une commande
qui donne de l'aide sur une
commande, passée en
paramètre**

```
SYNOPSIS
```

```
ls [options]
```

```
Options: [-  
[--help] [--version] [--]
```

```
DESCRIPTION
```

```
La commande ls affiche tout d'abord l'ensemble de  
ses arguments fichiers autres que des répertoires.  
Puis ls affiche l'ensemble des fichiers contenus  
dans chaque répertoire indiqué. Si aucun argument
```

option aide: --help

syntaxe: nom-de-la-commande --help

```
[dupond@r10102 ~]# ls --help
```

```
Usage: ls [OPTION]... [FICHIER]...
```

```
Afficher les informations au sujet des FICHIERS (du répertoire courant par défaut). Trier les entrées alphabétiquement si aucune des options -cftuSUX ou --sort n'est utilisée.
```

```
Les arguments obligatoires pour les options de formes longues le sont aussi
```

```
pour les options de formes courtes.
```

-a, --all	ne pas cacher les entrées débutant par .
-A, --almost-all	ne pas inclure dans la liste . et ..
--author	afficher l'auteur de chaque fichier
-b, --escape	afficher en octal les caractères non-graphiques en utilisant des séquences d'échappement

obtenir de l'aide

moteurs de recherche

- Google
- Yahoo
- Ecosia



apprendre à chercher
l'information sur le web



*Il est souvent préférable de taper les
requêtes en anglais => plus de chances
d'avoir une réponse pertinente*

2e partie

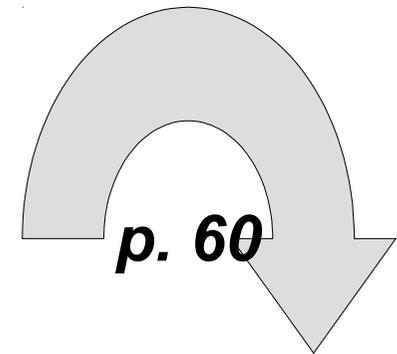
Plan de la 2e partie

- qu'est-ce qu'un programme?
- rôles du système d'exploitation
- les processus
- les types de fichiers
- systèmes multi-utilisateurs

partie pratique

- ligne de commande:
se simplifier la vie
- commandes de base

qu'est-ce qu'un programme?



les instructions

*Un programme contient des instructions
comme dans une recette de cuisine*



1. Détailler les carottes en petits tronçons.
2. Les faire revenir 10 minutes dans une casserole avec de l'huile d'olive, le temps qu'elles colorent un peu.
3. Ajouter le 1/2 litre de bouillon de volaille et laisser cuire 20 bonnes minutes, pour que les carottes deviennent fondantes.
4. Ajouter la crème et le lait et mixer le tout pour obtenir un mélange bien onctueux. Rectifier l'assaisonnement.

Cuisinier = processeur



processeur



les instructions

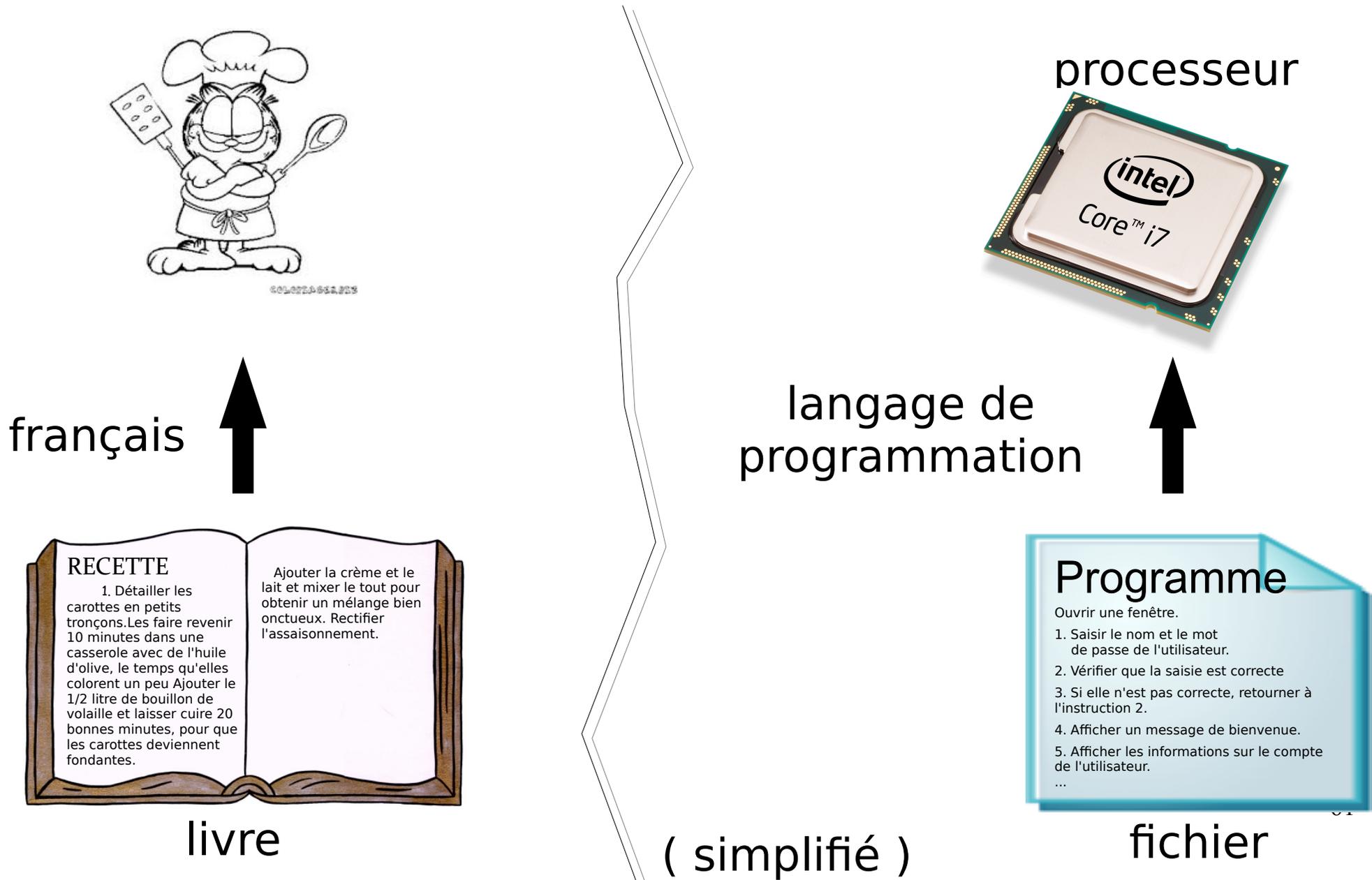
Instructions :

1. Afficher une fenêtre.
2. Demander à l'utilisateur de saisir un mot de passe
3. Vérifier que la saisie est correcte
4. Si elle n'est pas correcte, retourner à l'instruction 2.
5. Afficher un message de bienvenue.
6. Afficher les informations sur le compte de l'utilisateur.
7. Afficher un menu des opérations possibles



...

langage des instructions



rôles du système d'exploitation

- introduction
- simplifier l'accès au matériel
- partage des ressources

A quoi sert un système d'exploitation?

application

traitement de texte

navigateur web

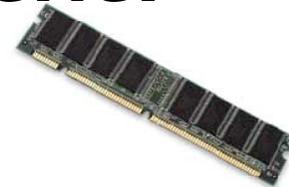
jeu vidéo

système d'exploitation

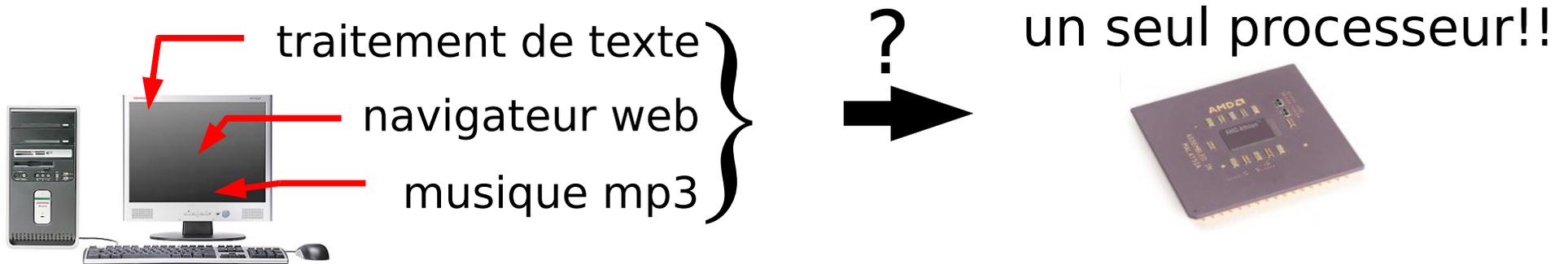
accès simplifié
au matériel

partage des
ressources

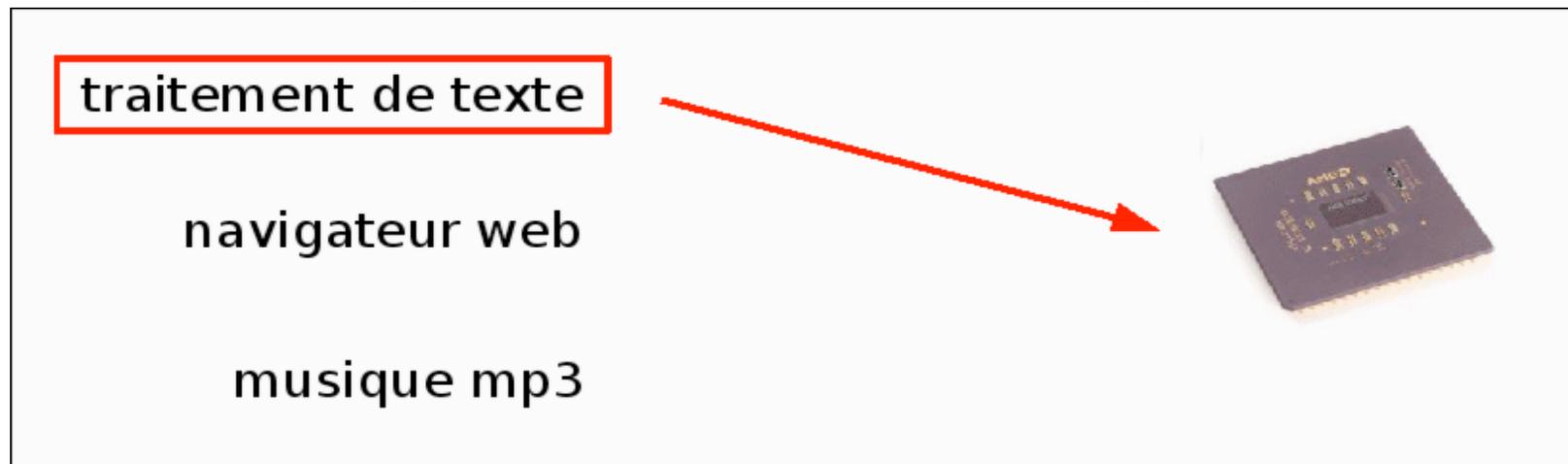
matériel



Partage du processeur



chacun son tour



très vite!



sentiment que c'est
« simultané »



multitâche

Simplifier l'accès au matériel



programmeur

« *dessine une ligne* »

API: interface de programmation d'une application

système d'exploitation

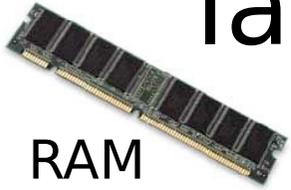
pilote

« *commande n°4A2 avec les registres 73 et 0C* »

carte graphique



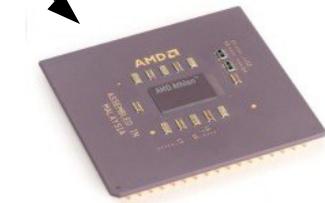
la mémoire, qu'est-ce que c'est ?



RAM



calcul:
 $11 + 8 \rightarrow 19$



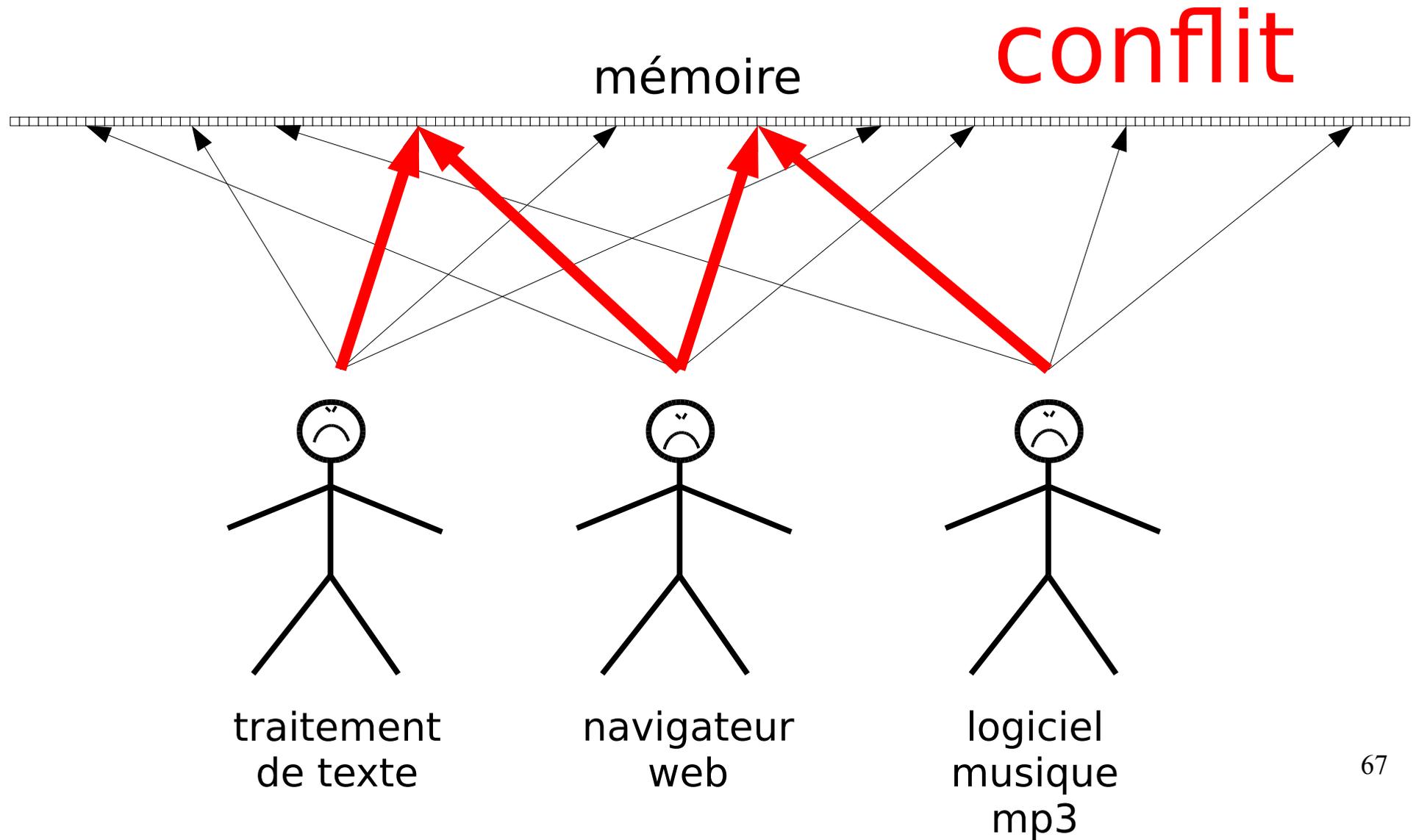
processeur

- programmes
- données

- rapide
- volatile
- taille limitée

(très simplifié!)

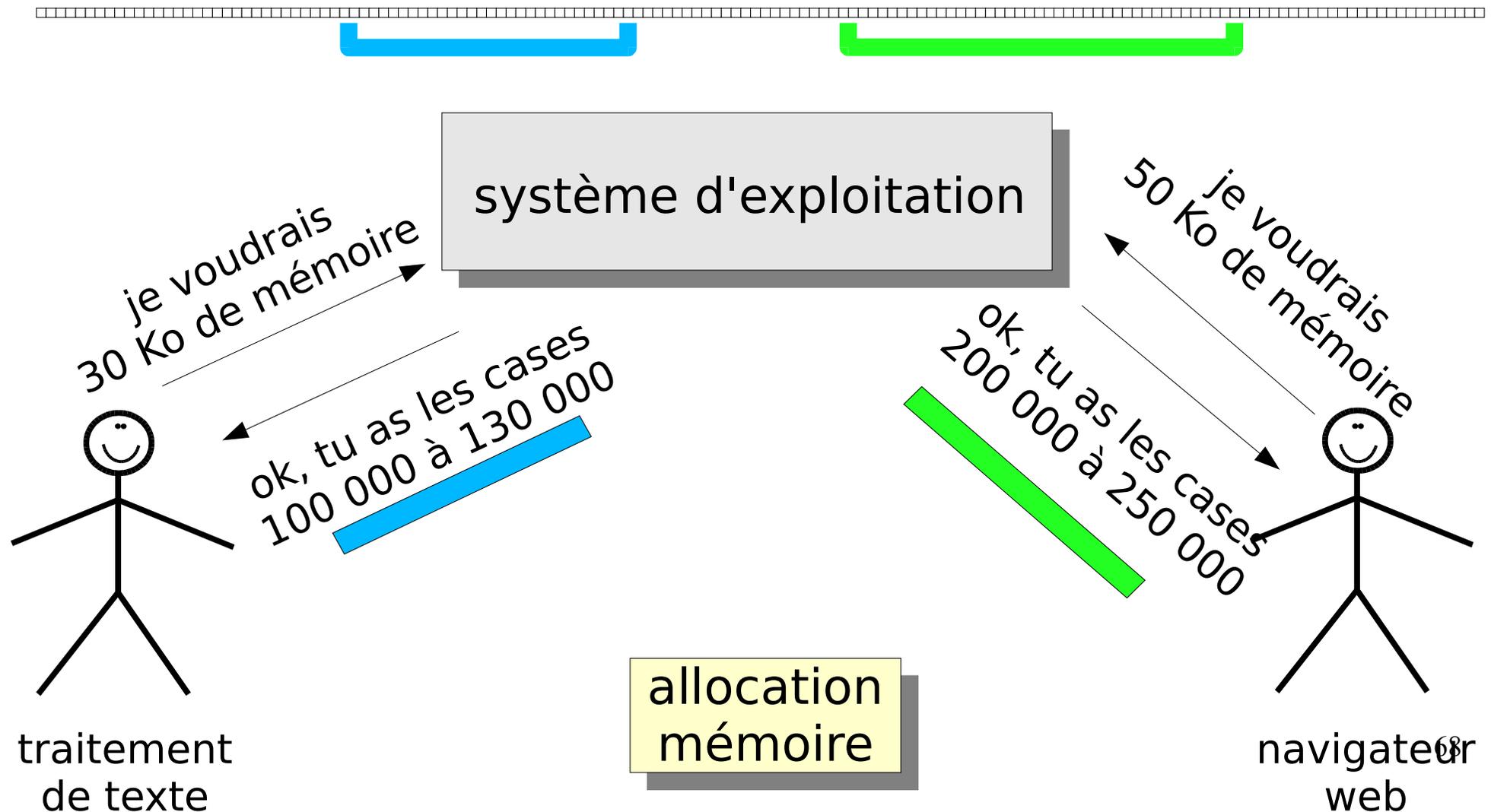
Partage de la mémoire



Partage de la mémoire

le rôle du SE (sys. expl.)

mémoire



les processus

rappel: structure d'une commande

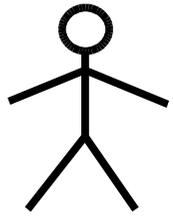
```
[dupond@r10102 ~]# rm -r toto titi zozo
```

nom de la commande
(rm = effacer)

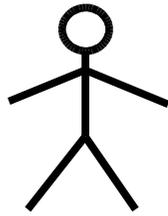
arguments de
la commande

option:
(-r = récursif)

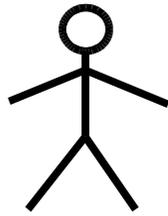
La notion de processus



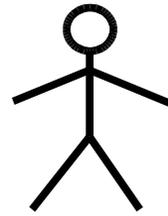
traitement
de texte



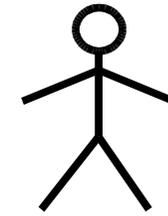
navigateur
web



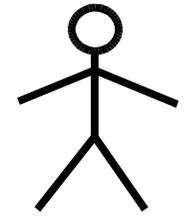
logiciel
musique
mp3



gestionnaire
de fenêtres



gestionnaire
de terminaux



serveur
web

... des dizaines de processus simultanés!

- s'exécutent « en même temps »
- partagent les ressources
(mémoire, processeur, périphériques ...)
- chacun sa propre région mémoire

processus: exemples

la commande ps

même programme
lancé deux
fois : 2 processus

```
[dupond@r10102 ~]# ps -a
  PID TTY          TIME CMD
 24866 pts/3        00:00:03 gedit
 26631 pts/8        00:12:42 soffice.bin
 10987 pts/0        00:00:02 gpdf
 14908 pts/6        00:00:00 gedit
 14904 pts/10       00:00:00 man
 14910 pts/10       00:00:00 bash
 15102 pts/8        00:00:00 ps
```

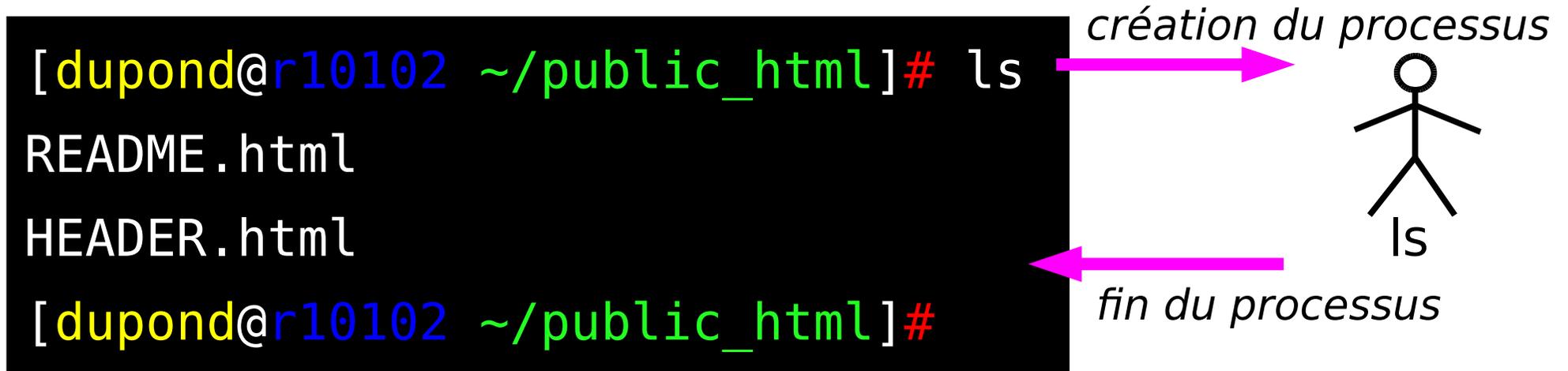
les numéros
des processus

les commandes
correspondant
à chaque processus

créer un processus

Vous créez systématiquement un processus lorsque, par ex. :

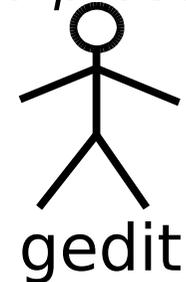
- vous lancez un programme
- vous tapez une commande
- ...



créer un processus en tâche de fond

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

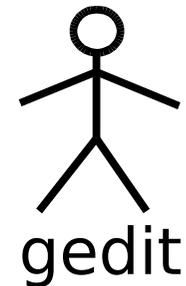
création du processus



on a perdu la main! 😞

```
[dupond@r10102 ~]# gedit &
```

création du processus
en tâche de fond



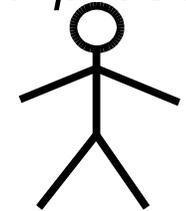
on récupère la main



créer un processus en tâche de fond

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

création du processus

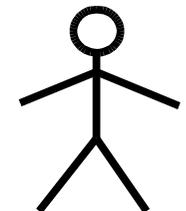


gedit

on a perdu la main! 😞

```
[dupond@r10102 ~]# gedit &
```

création du processus
en tâche de fond



gedit⁷⁵

on récupère la main



75

tuer un processus

(à la main)



la commande: **kill**

éditeur
de texte

```
[dupond@r10102 ~]# ps -a
  PID TTY          TIME CMD
24866 pts/3        00:00:03 man
26631 pts/8        00:12:42 gedit
[dupond@r10102 ~]# kill 26631
```

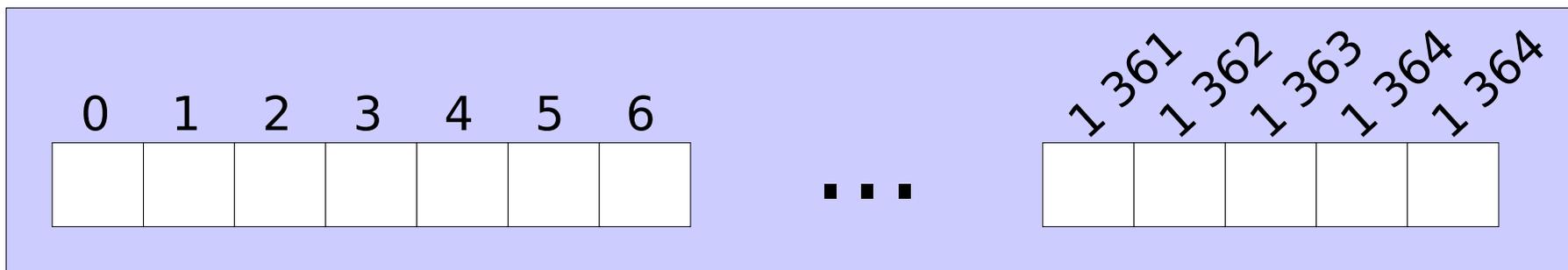
numéro de processus de
l'éditeur de texte

les types de fichiers

- fichiers texte
- fichiers binaires
- fichiers compressés
- archives

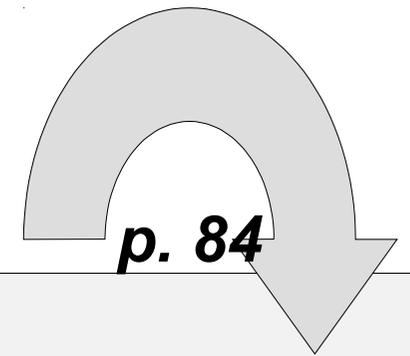
fichiers

fichier: suite
d'octets stockés
sur un support.
(rappel)



taille du fichier : 1 365 octets

fichiers texte



fichier: `bonjour, ca va?`



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
b	o	n	j	o	u	r	,		c	a		v	a	?

utiliser un éditeur de texte



exemples:

sources

```
programme.java  
programme.c  
programme.php  
script-shell
```

divers

```
page.html  
style.css  
donnees.xml
```

configuration

```
.bashrc  
/etc/passwd  
httpd.conf
```

formats de fichiers binaires

pas d'éditeur de texte ☹️

créés et gérés par des programmes spécifiques

exemple simple: image gif ↔ logiciels: gimp, photoshop...

taille de l'image
et autres infos

données binaires de l'image



autres exemples:

multimedia

photo.jpg
musique.mp3
video.avi
dessin.png

bureautique

document.doc
document.pdf
document.odp
calcul.xls

divers

executable
libc.so
libz.dll
compress.gz



fichiers compressés

réduire la taille d'un fichier

Comment c'est possible?

réduire les informations redondantes

exemple très simple:

0	1	2	3	4	5	6	7	8	9	10	11	12
5	5	5	5	1	1	1	1	2	2	2	2	2

→ 13 octets 😞



4 fois 5

4 fois 1

5 fois 2

0	1	2	3	4	5
4	5	4	1	5	2

→ 6 octets⁸¹ 😊

fichiers compressés

le fichier d'origine
fait 24 Ko

```
[dupond@r10102 ~]# ls -l page.html
-rw-r--r--  1 dupond dupond 23957 14:01 page.html
[dupond@r10102 ~]# gzip page.html
[dupond@r10102 ~]# ls -l page.html.gz
-rw-r--r--  1 dupond dupond 3038 14:01 page.html.gz
```

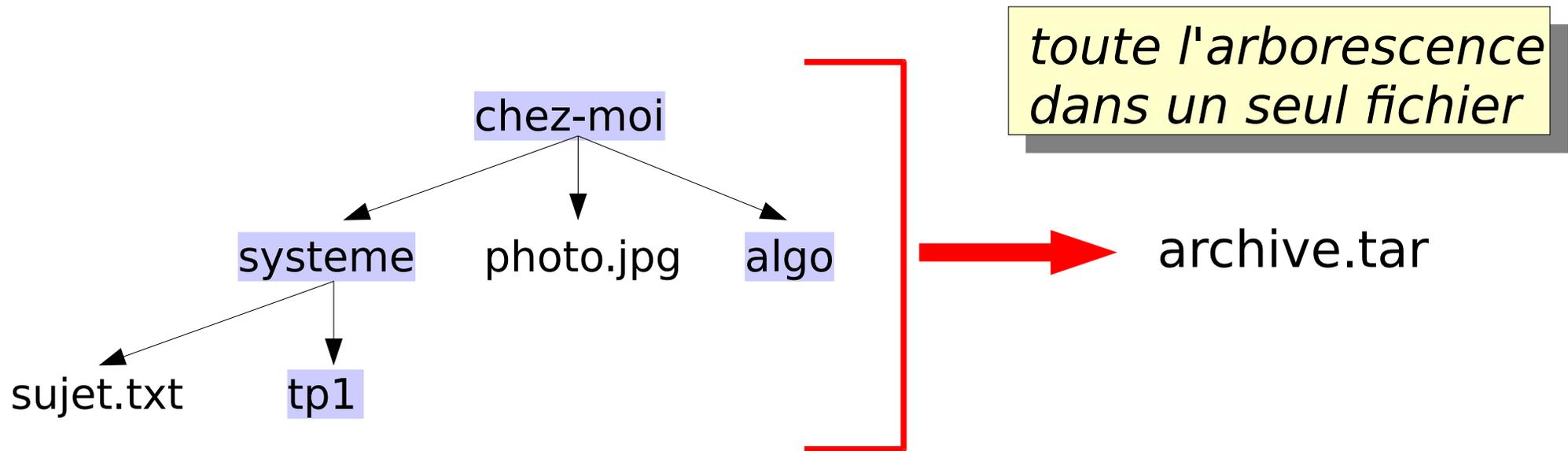
le fichier compressé
ne fait que 3 Ko !

exemples :

compress.gz
compress.bz2
compress.Z
compress.z

compress.rar
compress.zip

les fichiers archives



exemples:

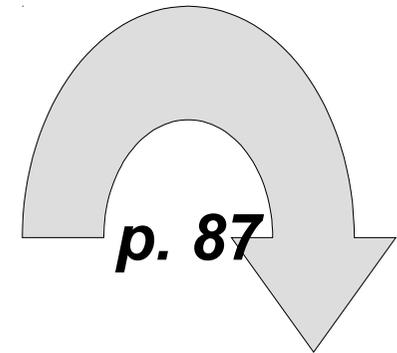
archive.tar
→ archive.tar.gz
⇐ archive.tgz

archive.zip
archive.rar
archive.jar

souvent:
archive+compression

systemes multi-utilisateurs

- présentation
- droits d'accès des fichiers

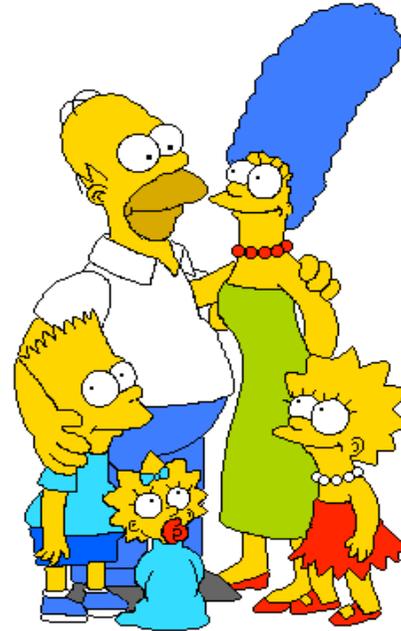


systemes multi-utilisateurs

pourquoi?



ordinateur familial

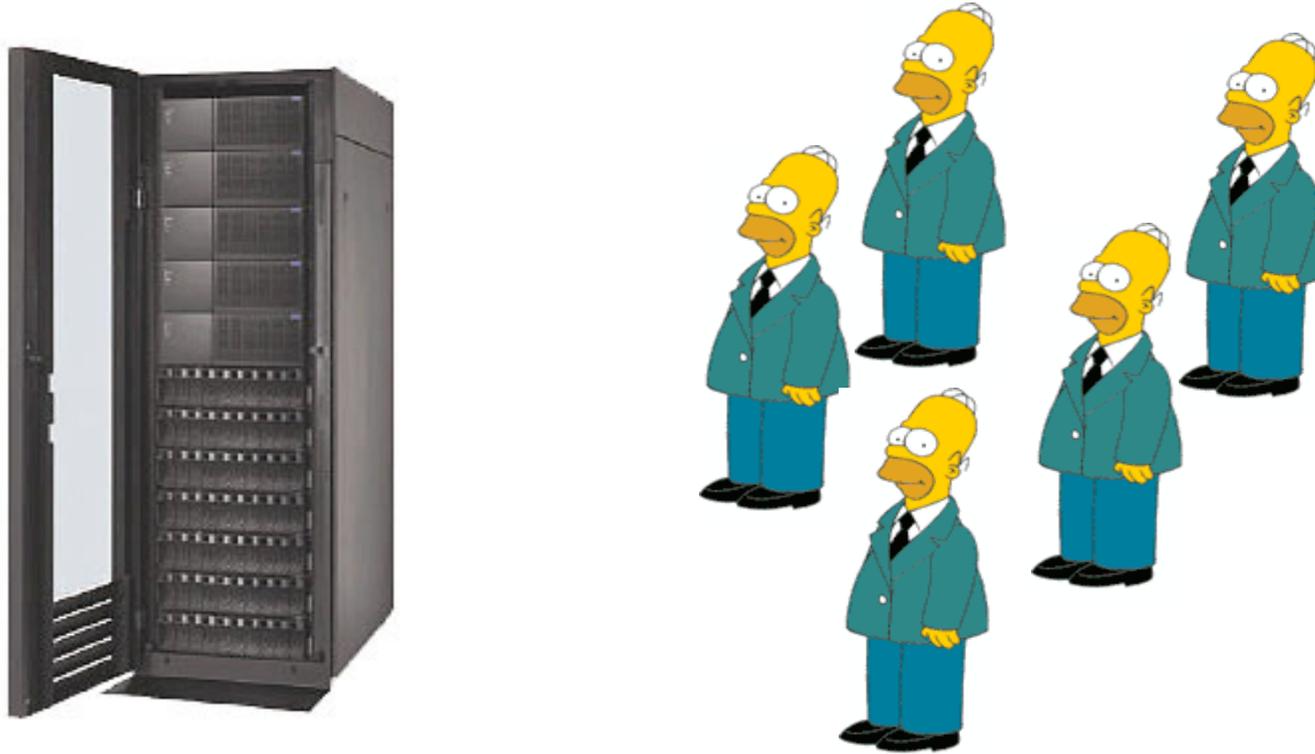


éviter que:

- votre petit frère détruit vos fichiers
- vos parents lisent vos courriers
- votre petite soeur administre la machine

systemes multi-utilisateurs

serveur



- gérer de nombreux utilisateurs
- groupes d'utilisateurs

exemple:
comptes prof

systemes multi-utilisateurs

comment?

- définir des comptes utilisateurs
- authentification
- fichiers
 - propriétaire, groupe
- ressources du système (administration)

root: le compte de l'administrateur système sous UNIX

droits d'accès aux fichiers

catégories

utilisateur : **U**ser
groupe : **G**roup
les autres : **O**thers

droits

lecture : **R**ead
écriture : **W**rite
exécution : **eX**ecute

G:groupe **O**:les
U:utilis. autres

-rw-r-----

```
[truck@r10102 ~/public_html]# ls -l README.html  
-rw-r----- 1 truck prof 17:44 README.html
```

droits d'accès

propriétaire

groupe du propriétaire

changer les droits : chmod

catégories

utilisateur
groupe
les autres

: **U**ser
: **G**roup
: **O**thers

droits

lecture : **R**ead
écriture : **W**rite
exécution : **eX**ecute

G:groupe **O**:les
U:utilis. autres
- rw - r - - **r** - -

+r

```
chmod o+r unfichier.txt
```

Attention !

*Seul le propriétaire
d'un fichier/rép peut
changer les droits de
ce fichier/rép.*

Cas particulier :

*l'utilisateur root peut,
quant à lui, changer les
droits de TOUS les
fichiers/rép.*

changer les droits : chmod

catégories

utilisateur	:	U ser
groupe	:	G roup
les autres	:	O thers

droits

lecture	:	R ead
écriture	:	W rite
execution	:	eX ecute

on peut combiner :

```
chmod o+rw unfichier.txt  
chmod og+r unfichier.txt
```

on peut enlever des droits:

```
chmod o-rw unfichier.txt
```



changer les droits : chmod

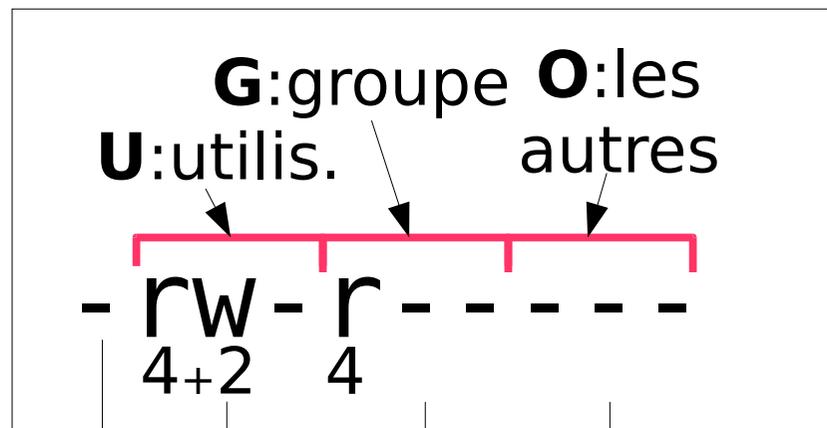
catégories

utilisateur : **U**ser
groupe : **G**roup
les autres : **O**thers

droits

lecture :4: **R**ead
écriture :2: **W**rite
exécution :1: e**X**ecute

droits
« numériques »



2^n

0 6 4 0

0640

chmod 0640 unfichier.txt

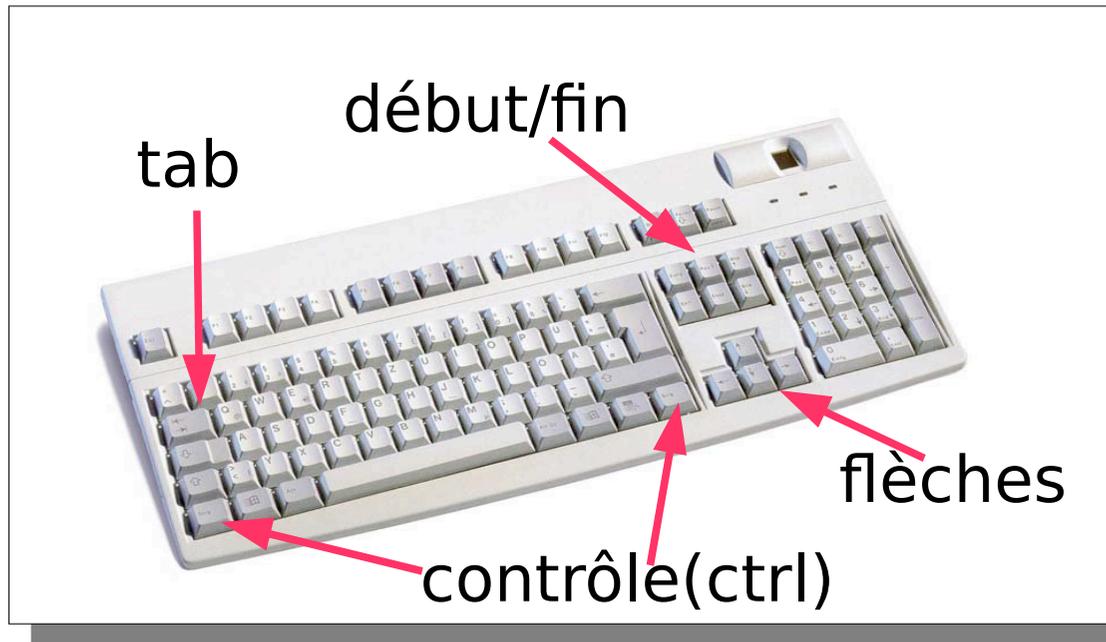
partie pratique

ligne de commande: se simplifier la vie

- raccourcis clavier
- Méta-caractères (qu'est-ce que c'est?)

raccourcis clavier

de la ligne de commande



Testez-les !

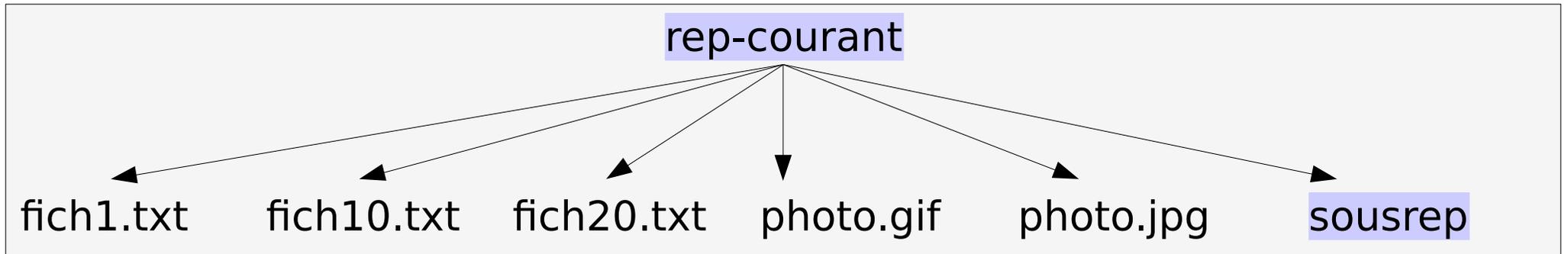
ctrl-c : tuer le processus courant

tab : compléter 😊

flèches : (haut/bas) dernières commandes

début/fin: début/fin de ligne

méta-caractères : pourquoi?



on veut copier tous les fichiers dans « **sousrep** » :

répertoire destination

```
[dupond@r10102 ~/rep-courant]# cp fich1.txt  
fich10.txt fich20.txt photo.gif photo.jpg sousrep ↵
```

fichiers source



trop long à écrire!



méta-caractère: *

répertoire destination

```
[dupond@r10102 ~/rep-courant]# cp *.txt photo.* sousrep
```

tous les fichiers
finissant par « .txt »

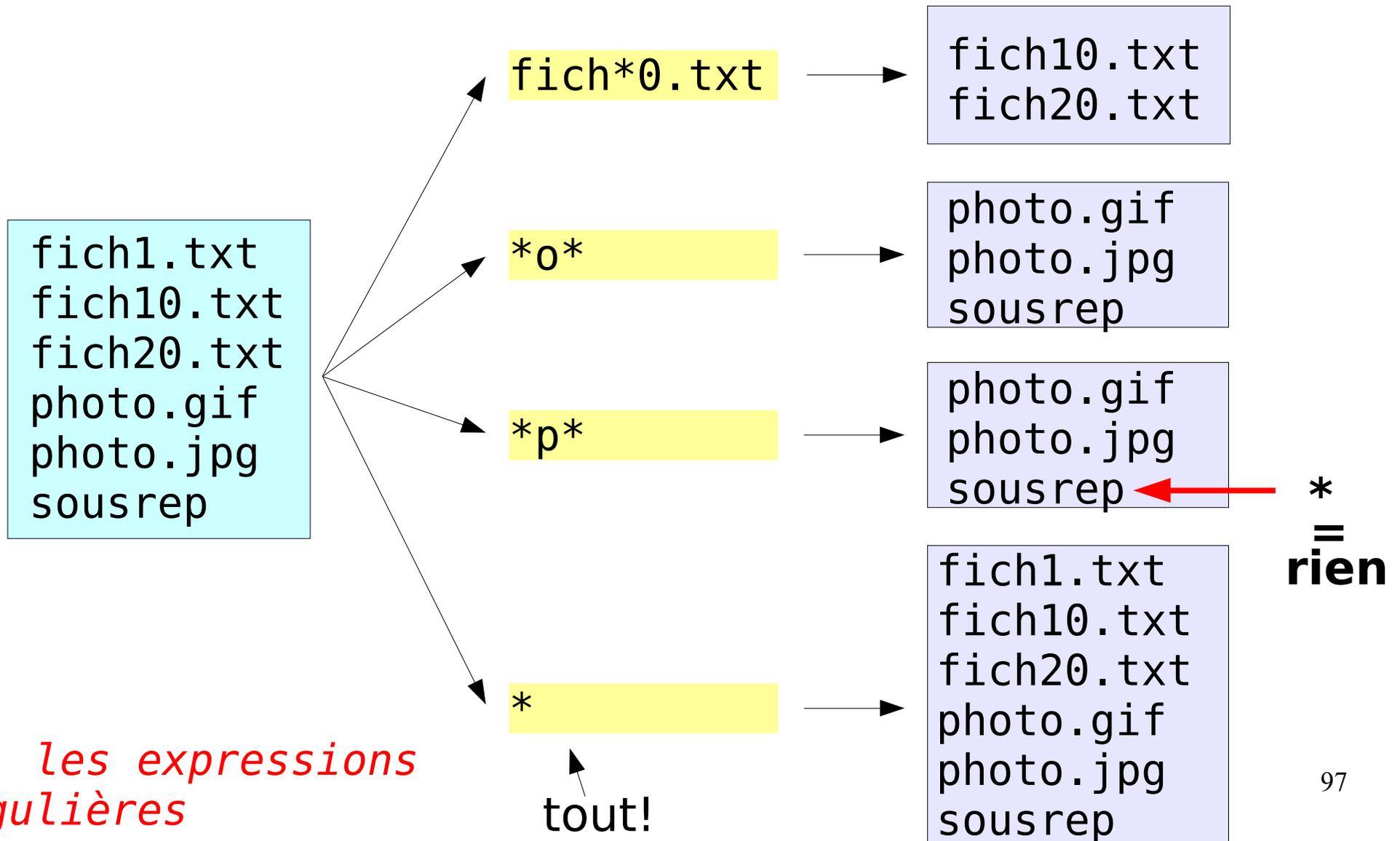
tous les fichiers
commençant
par « photo. »

* = n'importe
quelle suite
de caractères

remplacement par le shell:

*.txt	→	fich1.txt	fich10.txt	fich20.txt
photo.*	→	photo.gif	photo.jpg	

méta-caractère * : exemples



commandes de base

- ls
- rm
- cp

fichiers cachés

fichiers ou
répertoires dont le
nom commence par
un point

fichiers de configuration
dans votre répertoire personnel

- `.bash_profile`
- `.bash_history`
- `.bashrc`
- `.emacs`
- `.gnome`
- `.profile`
- etc.**

la commande `ls`

plus en détail

options:

- l : toute l'information
- a : tous les fichiers

```
[truck@r10102 ~/public_html]# ls -la
total 16
drwxr-xr-x  2 truck prof 4096 2015-07-12 17:45 .
drwxr-xr-x  8 truck prof 4096 2015-07-12 13:22 ..
-rw-r--r--  1 truck prof  161 2015-07-12 17:45 .hidden
-rw-r--r--  1 truck prof  161 2015-07-12 17:45 HEADER.html
-rw-r--r--  1 truck prof 1155 2015-07-12 17:44 README.html
```

droits d'accès

propriétaire

taille du
fichier en
octets

date de
modification

nom de
fichier

la commande `ls`

affiche toute l'information
des fichiers finissant par « .html »

```
[truck@r10102 ~/public_html]# ls -l *.html
-rw-r--r--  1 truck prof  161 2005-07-12 17:45 HEADER.html
-rw-r--r--  1 truck prof 1155 2005-07-12 17:44 README.html
```

public_html est un
répertoire:
ls affiche son **contenu**

```
[truck@r10102 ~]# ls public_html
HEADER.html README.html
```

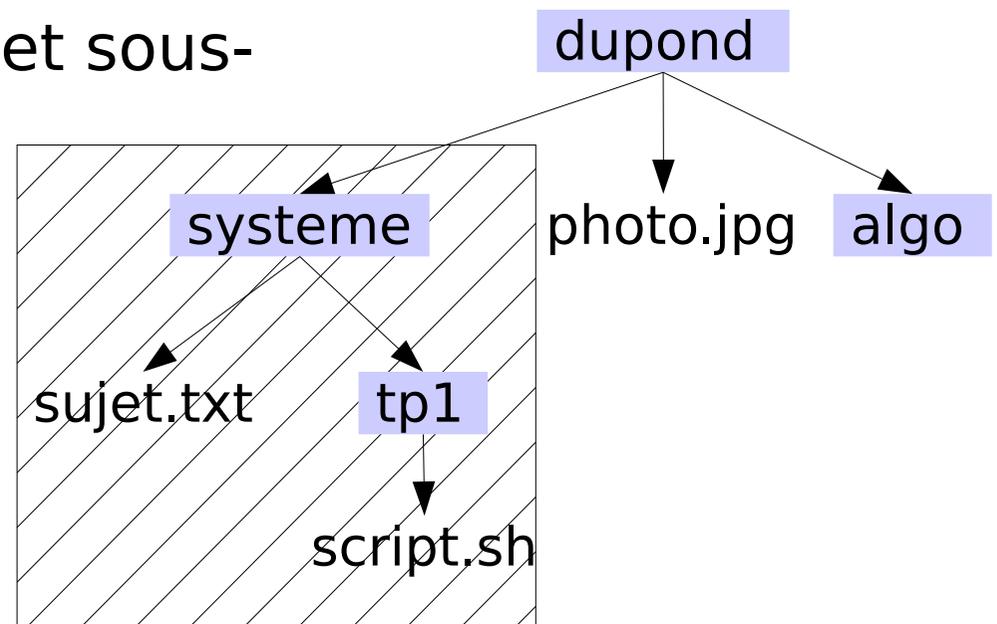
la commande **rm**

détruit un ou plusieurs fichier ou répertoires

options:

- r : récursif :
le répertoire, ses fichiers, et sous-répertoires, etc.
- f : forcer:
ne pose pas de questions

 **Dangereux!**



```
[truck@r10102 ~]# rm -rf systeme
```

la commande **rm**

effacer deux fichiers

```
[truck@r10102 ~]# rm public_html/README.html test/toto
```

effacer un répertoire

```
[truck@r10102 ~]# rm public_html  
rm: ne peut enlever `public_html': est un répertoire  
[truck@r10102 ~]# rm -r public_html  
[truck@r10102 ~]#
```

message d'erreur

ok

la commande **cp**

syntaxe:

```
cp source1 [ source2 source3 ... ] destination
```

options:

- r : récursif :
le répertoire, ses fichiers, et sous-répertoires, etc.

la commande **cp**

duplication simple: source -> destination

```
[truck@r10102 ~]# cp a b
```

crée un fichier « a » dans le répertoire public_html

```
[truck@r10102 ~]# cp a public_html
```

copie de plusieurs fichiers vers un répertoire

```
[truck@r10102 ~]# cp x y z public_html
```

la commande **cp**

duplication d'un répertoire

```
[truck@r10102 ~]# cp public_html toto  
cp: omission du répertoire `local' message d'erreur
```

```
[truck@r10102 ~]# cp -r public_html toto  
[truck@r10102 ~]# ok
```

la commande **echo**

Afficher sur la sortie standard (l'écran) les arguments:

```
[truck@r10102 ~]# echo bonjour comment ca va
bonjour comment ca va
[truck@r10102 ~]# echo "bonjour"
bonjour
[truck@r10102 ~]# cd public_html
[truck@r10102 ~/public_html]# echo "*"
*
[truck@r10102 ~/public_html]# echo *
HEADER.html README.html
[truck@r10102 ~/public_html]#
```

Donc les guillemets empêchent toute substitution