

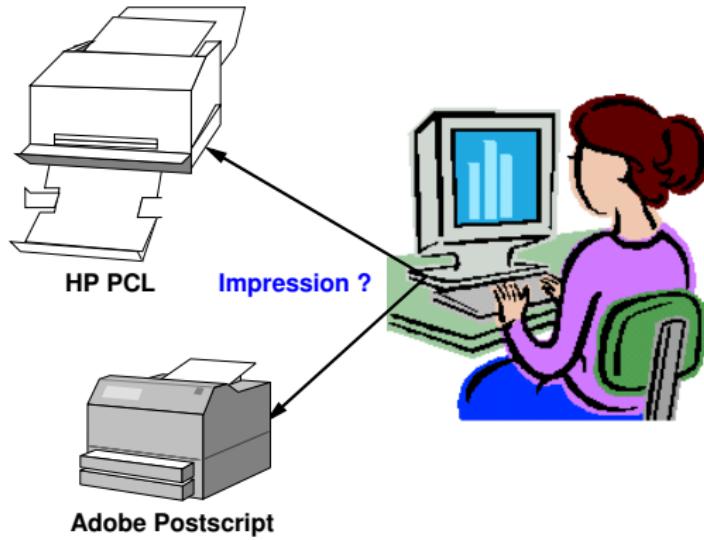
Introduction aux systèmes d'exploitation

Frédéric Goualard

Laboratoire d'Informatique de Nantes-Atlantique, UMR CNRS 6241



- ▶ Transparents, sujets de TDs et TPs sur *madoc*
- ▶ Les transparents ne présentent qu'une petite partie de ce qui sera abordé
- ▶ Tous les exercices effectués ne sont pas sur les sujets distribués
- ▶ Questions ?
 - ▶ Frédéric Goualard, bureau 208 (2^e étage), département Informatique
 - ▶ Frederic.Goualard@univ-nantes.fr
 - ▶ Forum du module sur *madoc*
- ▶ Toutes les informations seront mises sur *madoc* (déplacement de cours, ...)

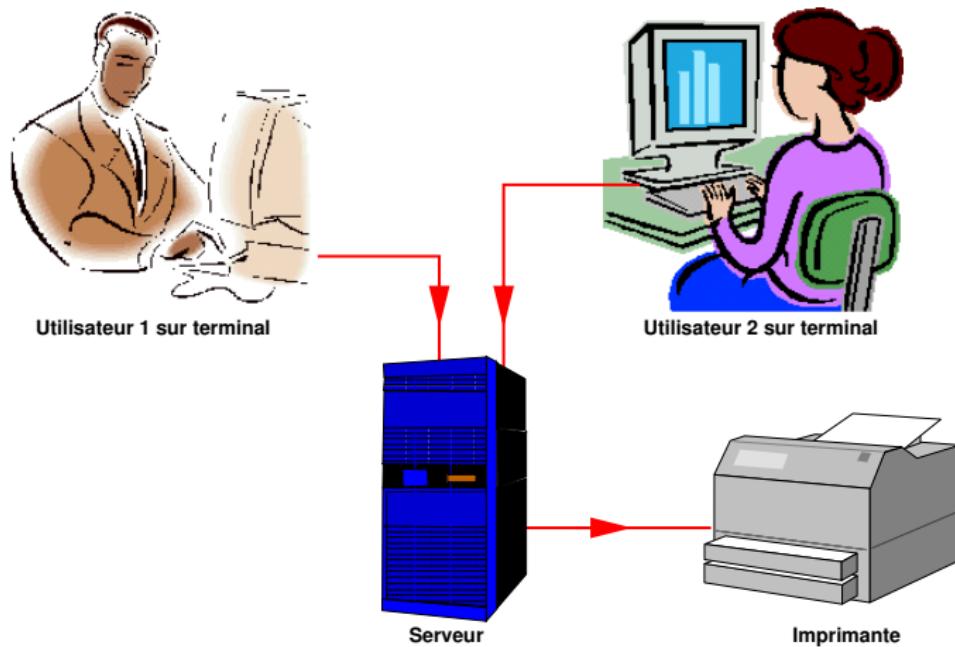


► Impression

⇒ connaître les détails de l'imprimante connectée ?



Rôle d'un système d'exploitation ? (2)



- ▶ Gestion des conflits ?



Système d'exploitation :

Premier programme chargé au démarrage (*boot*)

Tâches :

- ▶ Uniformisation des matériels utilisés :
 - ▶ *Disque dur* et non pas *disque Western Digital xxxx ou Quantum yyyy, ...*
 - ▶ Ensemble de blocs et non pas disque dur ou CD-ROM
- ▶ Protection des ressources



Système d'exploitation :

Premier programme chargé au démarrage (*boot*)

Tâches :

- ▶ Uniformisation des matériels utilisés :
 - ▶ Disque dur et non pas disque Western Digital xxxx ou Quantum yyyy, ...
 - ▶ Ensemble de blocs
- ▶ Protection des ressources

Ressource. Elément virtuel ou réel partageable (avec exclusion mutuelle ou non)

Exemple : RAM, disque dur, processeur, imprimante, ...



Système d'exploitation :

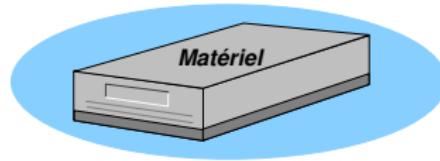
Premier programme chargé au démarrage (*boot*)

Tâches :

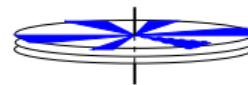
- ▶ Uniformisation des matériels utilisés :
 - ▶ Disque dur et non pas disque Western Digital xxxx ou Quantum yyyy, ...
 - ▶ Ensemble de blocs
 - ▶ Protection des ressources
- Moyen :
- ▶ Virtualisation des ressources

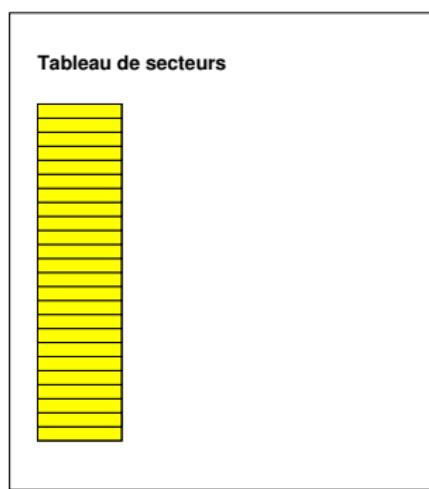
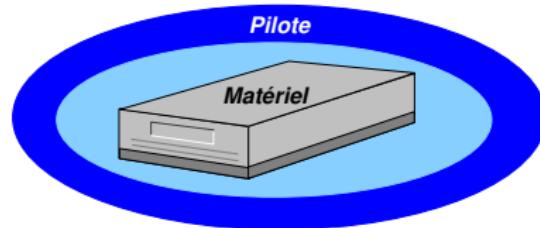
Ressource. Elément virtuel ou réel partageable (avec exclusion mutuelle ou non)

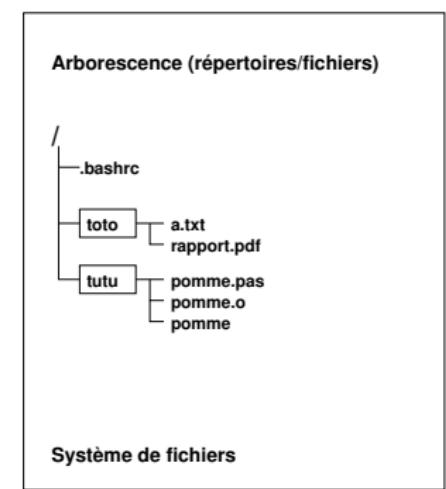
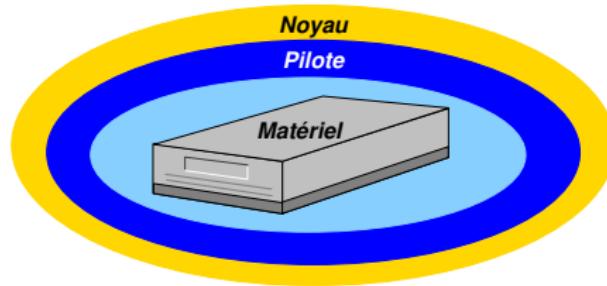
Exemple : RAM, disque dur, processeur, imprimante, ...

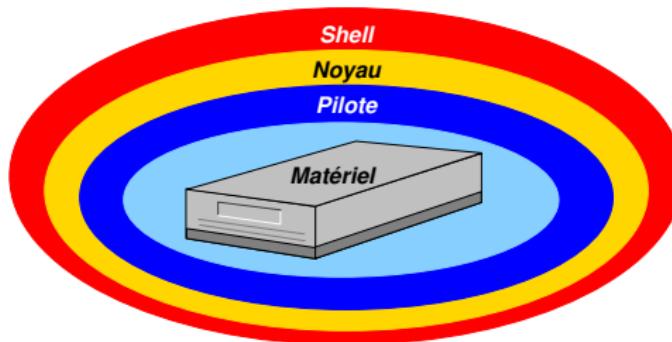


Western Digital WD2000BB
7200 tours/mn, 200 Gio
3 plateaux, 16383 cylindres



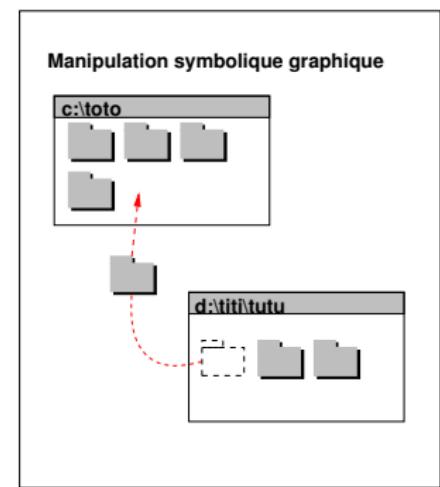
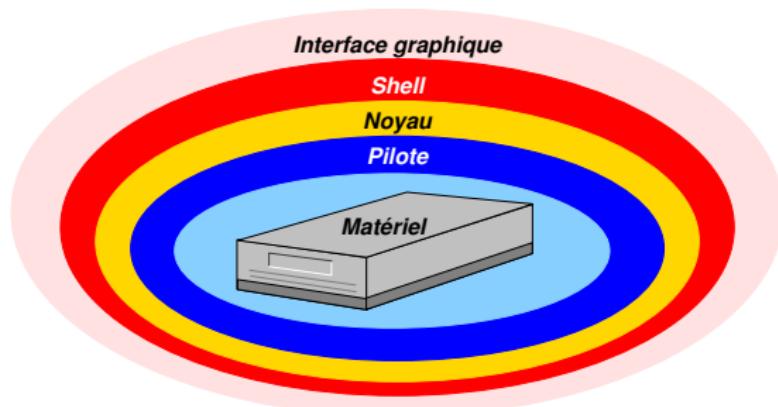


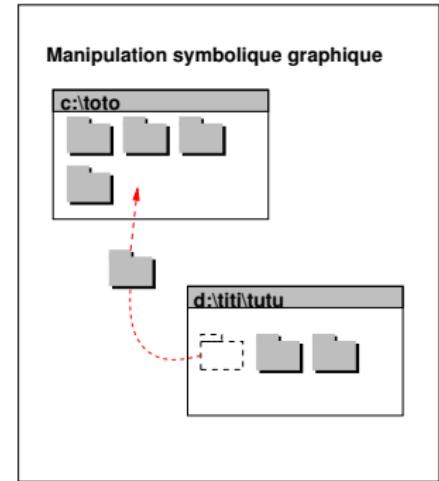
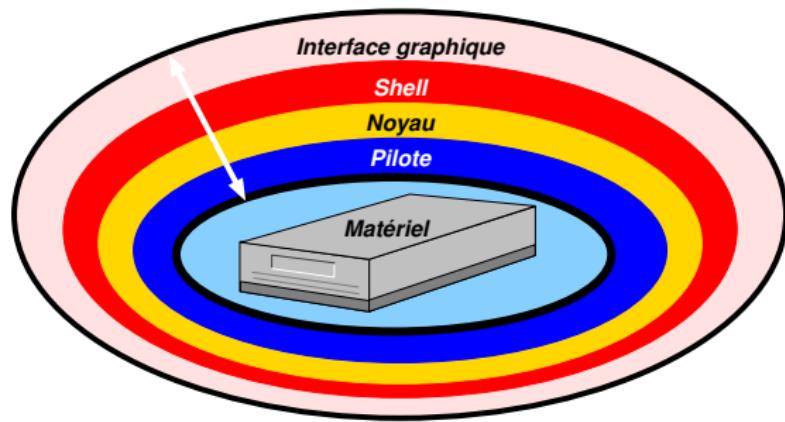




Manipulation symbolique textuelle

```
% ls -al  
% cp *.pas ~/tps/tp-4
```







- ▶ *Mono-utilisateur* (e.g. MS-DOS). Une seule personne peut travailler sur la machine à la fois
- ▶ *Multi-utilisateur* (e.g. Linux). Plusieurs personnes peuvent travailler sur une même machine via des terminaux
- ▶ *Mono-tâche* (e.g. MS-DOS). Un seul programme peut s'exécuter à la fois
- ▶ *Multi-tâche* (e.g. Windows XP). Plusieurs programmes s'exécutent *concurrentement*
- ▶ *Temps réel ou non* (e.g. RT-Mach). Un système temps-réel répond en un temps garanti à un stimulus. Utilisation dans des systèmes embarqués (e.g. gestion de centrale nucléaire)



- ▶ Premiers ordinateurs (ENIAC, EDSAC,...) : pas de système d'exploitation
 - Manipulation directe du matériel
 - ordinateurs mono-utilisateurs
- ▶ **1970.** UNIX (multi-tâche, multi-utilisateur)
- ▶ **1973.** Digital Research CP/M (Amstrad et premiers PCs)
- ▶ **1981.** MS-DOS 1.0 (\pm recopié sur CP/M)
- ▶ **1984.** MacOS 1.0. Système mono-utilisateur graphique
- ▶ **1985.** Ajout d'une interface graphique à MS-DOS : Windows 1.0 (mono-tâche, mono-utilisateur)
- ▶ **1991.** Linux, clône de UNIX



- ▶ Gestion des périphériques
- ▶ Gestion des processus (tâches)
- ▶ Gestion de la mémoire
- ▶ Gestion du système de fichiers
- ▶ Gestion des entrées/sorties

Couche interface avec l'utilisateur : le *shell*

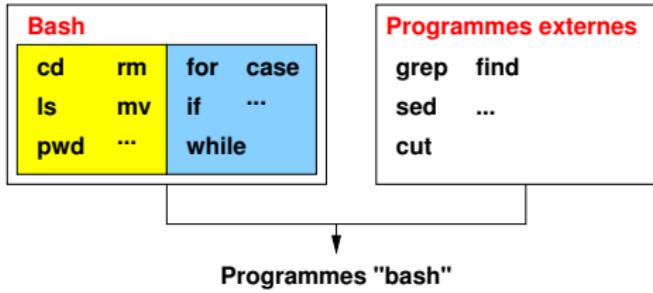
► Quoi ?

- Interpréteur de commandes
- Langage de programmation

► Pour quoi ?

- Manipulation des fichiers
- Gestion des processus

NOMBREUSES VARIANTES : **sh**, **csh**, **tcsh**, **ksh**, **bash**





Exemple de motivation

```
1  #!/usr/bin/env bash
2
3  if [ $# -ne 1 ]; then
4      echo 'load-photos.sh <dir>' 1>&2
5      exit 1
6  fi
7  if [ ! -d "$1" ]; then
8      echo "Error: $1 not a directory" 1>&2
9      exit 1
10 fi
11
12 imgs=$(find "$1" -name "*.[jJ][pP][gG]")
13 if [ -z "$imgs" ]; then
14     echo "No photo found (*.jpg)"
15 else
16     cp $imgs .
17     for f in *.[jJ][pP][gG]; do
18         jhead -autorot "$f" > /dev/null
19         rep=$(jhead "$f" | grep "Date/Time" | sed 's/.*: \([0-9:]\+\)\.\*/\1/\;tr : _')
20         mkdir -p $rep
21         mv "$f" $rep
22     done
23 fi
```



- ▶ Variables :
 - ▶ Affectation : **a=5, toto="une jolie phrase"**
 - ▶ Utilisation : **echo \$a, b=\$toto**
- ▶ Commentaires : tout ce qui suit **#** jusqu'à la fin de la ligne
a=5 # ceci est un commentaire
- ▶ Pas d'évaluation explicite (tout est chaîne) :
a=5+6 # met "5+6" dans a
- ▶ Séquence de commandes : séparation par un retour-chariot ou un **";"** :
a=5; echo \$a



- ▶ Appel *synchrone* : rend la main à la fin de l'exécution
 > **mozilla**
 - ▶ Appel *asynchrone* : rend la main immédiatement
 (exécution en tâche de fond)
 > **mozilla &**
-
1. Lecture au clavier ou dans un fichier
 2. Expansions
 3. Exécution

```
> a = pomme  
> echo $a
```

affectation



Interprétation de commandes

- ▶ Appel *synchrone* : rend la main à la fin de l'exécution
 > **mozilla**
 - ▶ Appel *asynchrone* : rend la main immédiatement
 (exécution en tâche de fond)
 > **mozilla &**
-
1. Lecture au clavier ou dans un fichier
 2. Expansions
 3. Exécution

> **a = pomme**

> **echo \$a**

affectation

substitution de \$a par sa valeur

> **echo pomme**



- ▶ Appel *synchrone* : rend la main à la fin de l'exécution
 > **mozilla**
 - ▶ Appel *asynchrone* : rend la main immédiatement
 (exécution en tâche de fond)
 > **mozilla &**
-
1. Lecture au clavier ou dans un fichier
 2. Expansions
 3. Exécution

```
> a = pomme
> echo $a
pomme
```

```
> echo pomme
```

affectation
substitution de \$a par sa valeur
exécution de la commande 'echo'



Règles :

- ▶ Pas d'expansion à l'intérieur d'apostrophes :
`a=5; echo '$a'` affiche “\$a”
- ▶ Expansion à l'intérieur de guillemets :
`a=5; echo "$a"` affiche “5”

Types d'expansions :

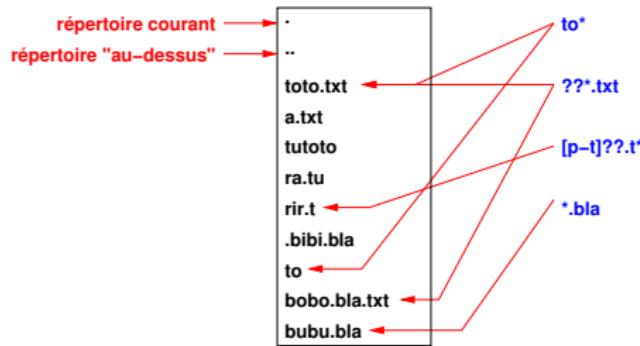
- ▶ Variables : `$x` expansé en le contenu de `x`
- ▶ Commandes : `$(cmd)` expansé en le résultat de l'exécution de `cmd` :
`echo "Chemin courant : $(pwd)"`
- ▶ Expressions arithmétiques : `$((expr))` :
`a=$((a+1))`
- ▶ Noms de fichiers : *globbing*



Le *globbing*

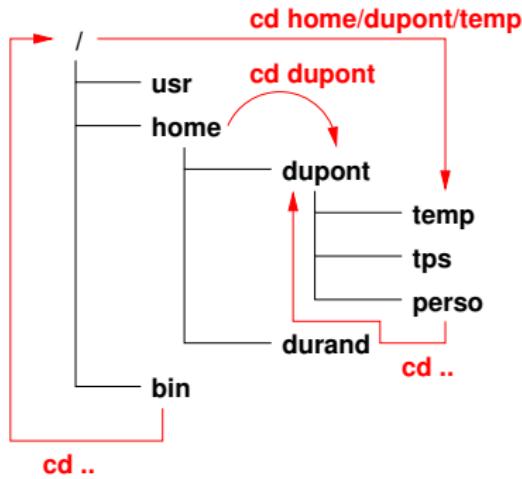
- ▶ Moyen de spécifier une liste de fichiers en intension avec un *patron*
- ▶ Trois jokers :
 - ▶ * : prend la place de 0 ou *n* caractères quelconques
 - ▶ ? : prend la place d'un caractère quelconque
 - ▶ [...] : prend la place de n'importe quel caractère dans l'ensemble “...”
 - ▶ {...} : expansion de liste

Exemples :





L'arborescence des répertoires



Commandes de base

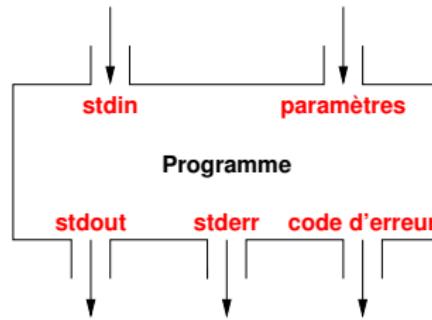
- déplacement : `mv /home/durand/*.txt .`
- copie : `cp ..//toto.txt ~dupont/temp`
- effacement : `rm *`
- listage : `ls /bin`
`ls`
`ls ../*.*`
- Position courante : `pwd`



Les commandes

- ▶ Commandes internes : **cd**, **mv**, ...
- ▶ Commandes externes : **find**, **grep**, ...
- ▶ Scripts **bash** de l'utilisateur

Entrées/sorties des commandes :



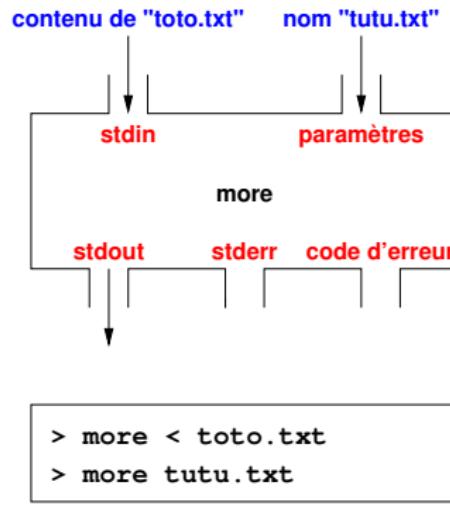
- ▶ Code d'erreur : dans \$?



Les commandes

- ▶ Commandes internes : **cd**, **mv**, ...
- ▶ Commandes externes : **find**, **grep**, ...
- ▶ Scripts **bash** de l'utilisateur

Entrées/sorties des commandes :



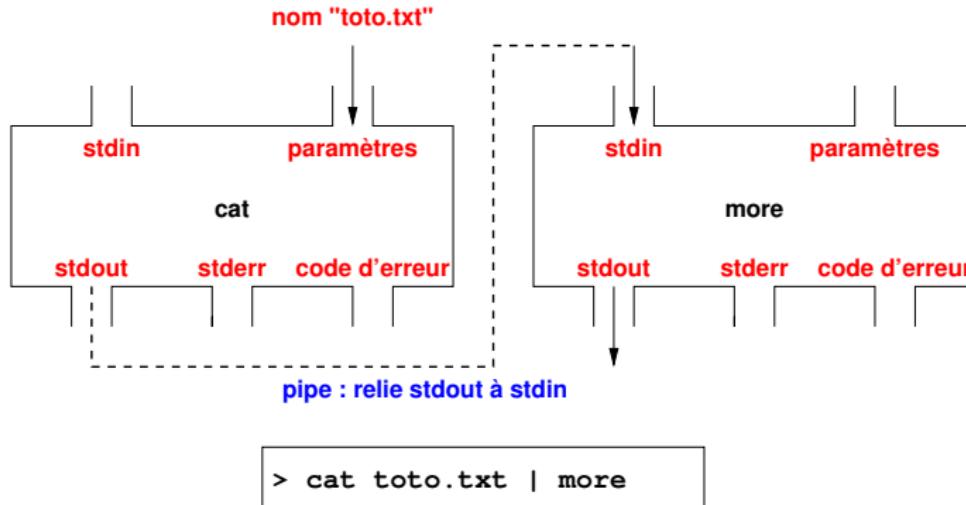
- ▶ Code d'erreur : dans \$?



Les commandes

- ▶ Commandes internes : **cd**, **mv**, ...
- ▶ Commandes externes : **find**, **grep**, ...
- ▶ Scripts **bash** de l'utilisateur

Entrées/sorties des commandes :



- ▶ Code d'erreur : dans \$?



- ▶ Redirection de l'entrée
(contenu d'un fichier envoyé sur **stdin**) :
less < toto.txt
- ▶ Redirection de la sortie
(envoi de ce qui sort par **stdout** dans un fichier) :
ls > liste.txt
- ▶ Redirection de la sortie d'erreur
(envoi de ce qui sort par **stderr** dans un fichier) :
gcc toto.c 2> erreurs.log
- ▶ Connection par un *pipe*
(envoi de ce qui sort de **stdout** de cmd1 vers **stdin** de cmd2) :
cat toto.txt | more



Contrôle de flot

<pre>for var in liste; do cmd₁ cmd₂ ... cmd_n done</pre>	<pre>while [test]; do cmd₁ cmd₂ ... cmd_n done</pre>	<pre>if [test]; then cmd₁ else cmd₂ fi</pre>
--	--	--

<pre>for x in *.txt; do echo \$x done</pre>	<pre>x=0 while [\$x -ne 4]; do x=\$((x+1)) done</pre>	<pre>x=banane if ["\$x" == "banane"]; then echo "banane trouvée" else echo "rien trouvé" fi</pre>
---	---	---

Tests

Nombres	(-eq =), (-ne ≠), (-le ≤), (-lt <), (-ge ≥), (-gt >)
Chaînes	(== =), (!= ≠)



Les scripts

- ▶ Fichier texte contenant une séquence de commandes
- ▶ Un script s'utilise comme une nouvelle commande
- ▶ Paramètres du script : accès par **\$1, \$2, ...**
- ▶ nombre de paramètres passés : dans **\$#**

Exemple :

monrm.sh

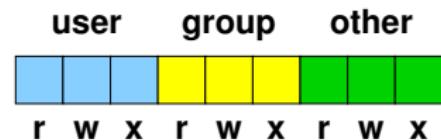
```
#!/usr/bin/env bash
if [ $# -ne 1 ]; then
    echo "Utilisation: monrm.sh fichier"
    exit 1 # code d'erreur 1
fi
echo "Effacer $1 (o/n) ?"
read choix
while [ "$choix" != "o" -a "$choix" != "n" ]; do
    read choix
done
if [ "$choix" == "o" ]; then
    rm -f $1
    echo "$1 effacé"
fi
# exit 0 implicite
```

```
> monrm.sh tutu.txt
Effacer tutu.txt (o/n) ? o
tutu.txt effacé
>
```



La gestion des droits

- ▶ Droits associés à chaque fichier/répertoire
 - ▶ Lecture (**r**)
 - ▶ Écriture (**w**)
 - ▶ Exécution (**x**)
- ▶ Trois types de personnes
 - ▶ Propriétaire (**u**)
 - ▶ Groupe (**g**)
 - ▶ Autres (**o**)
- ▶ Modification des droits : **chmod**



Exemples :

```
# ajout du droit de lecture pour le propriétaire
chmod u+r toto.txt
# retrait du droit d'écriture pour les autres
chmod o-w tutu.bla
# Modification en octal (ajout de tous les droits à tous)
chmod 777 pomme
```



- ▶ *Globbing* : matching de noms de fichiers
- ▶ *Regexp* : matching de texte

Programmes utilisant les expressions régulières :

- ▶ **expr**
- ▶ **grep**
- ▶ **sed**
- ▶ **find**
- ▶ **AWK**
- ▶ ...

La puissance d'expression des regexp est supérieure au globbing.



Regexp vs. globbing

Regexp		Sens	Globbing
Basic	Extended		
[^a-z05]	[^a-z05]	N'importe quel caractère non dans l'ensemble	Idem
.	.	N'importe quel caractère	.
~\$	~\$	Ancres de début/fin	~\$
\?	?	Répéteur (0 ou 1 fois)	1 caractère
*	*	Répéteur (0 ou n fois)	0 ou n caractères
\+	+	Répéteur (1 ou n fois)	+
\(\)	()	Groupement de patrons	—
\{n\}	{n}	Répéteur (exactement n fois)	Expansion d'accolades
\{n, \}	{n, }	Répéteur (n fois ou plus)	Expansion d'accolades
\{n, m\}	{n, m}	Répéteur (entre n et m fois)	Expansion d'accolades
\		Alternative (un patron ou l'autre)	—

- ▶ Tous autres caractères (+ caractères spéciaux précédés de « \ ») : sens habituel
- ▶ Les trois règles fondamentales du matching (dans l'ordre de considération) :
 1. Matching de la gauche vers la droite
 2. Matching le plus long possible ne remettant pas en cause le matching de la suite
 3. Pas de retour en arrière

Exemples. . .

- ▶ `expr match STRING REGEXP` (retourne le nombre de caractères ou la sous-chaîne matchés)

```
> expr match "Toto est beau" ".*[ \t]e[a-z]\+"
8
> expr match "Toto est beau" ".*[ \t]\(e[a-z]\+\)\)"
est
```

- ▶ `expr substr STRING POS LENGTH`

```
> expr substr 'Toto est beau' 7 4
st b
```

- ▶ `expr length STRING`

```
expr length 'Toto est beau'
13
```

Ancre de début « ^ » implicite

- ▶ Plusieurs programmes (grep, egrep, fgrep)
- ▶ Recherche de l'occurrence d'une chaîne avec une expression régulière
- ▶ Retourne 0 si au moins un matching et 1 si pas de matching
- ▶ Options pour inverser la recherche (**-v**), compter le nombre de matchings (**-c**), ignorer la casse (**-i**), ...

Exemples :

```
> egrep "[Jj](ean|acques)" prenoms1.txt prenoms2.txt
> fmt prog.c | egrep "[a-zA-Z_][a-zA-Z_0-9]*=;"
```

Éditeur de flux de texte

Format général :

> **sed OPTIONS SCRIPT ENTRÉES**

SCRIPT :

filtres/regexp/remplacement flags

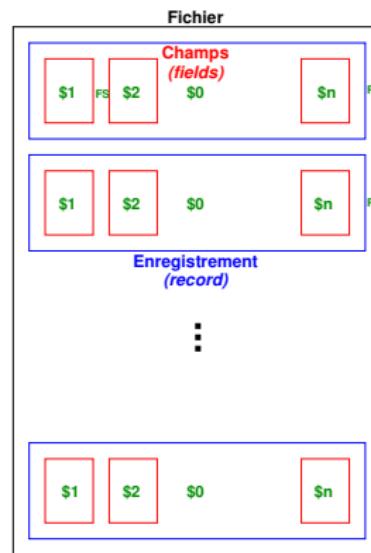
Exemples :

```
# Ajout d'une tabulation 1 ligne sur 2
cat toto.txt | sed 1^2s/\(.*/\t\1/
# Remplacement 'range(5)' par
# 'range(0,5,1)' pour toutes les lignes
# contenant 'for' ou 'foreach'
sed '/for\(each\)\?/s/range(\([0-9]\+\))/range(0,\1,1)/g' prog.py
```

Aho, Weinberger and Kernighan (AWK) : langage de manipulation de fichiers à la *sed*, en plus puissant et plus facile d'utilisation.

Programme AWK :

```
BEGIN { actions }
garde { actions }
...
garde { actions }
END { actions }
```





Exemple AWK

Filtrage du fichier **/etc/passwd** :

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
...
...
```

Programme AWK **filtre.awk** :

```
BEGIN {
    print("** Utilisateurs avec shell");
    FS=":"; RS="\ n";
}
/.*bin.*sh$/ { print($1); }
END { print("** Fin du traitement");
}
```

Appel :

```
> awk -f filtre.awk /etc/passwd
```



- ▶ Gestion par les routines du BIOS :
 - ▶ Avantage : facilement disponible et bien adapté à la configuration initiale
 - ▶ Inconvénient : flashage du BIOS obligatoire pour gérer de nouvelles technologies
 - ▶ Gestion d'un périphérique rare ?
- ▶ Gestion par des pilotes spécialisés :
 - ▶ Interface entre les signaux électriques émis par le périphérique et le système d'exploitation
 - ▶ Un pilote par périphérique ⇒ adaptation facile aux nouvelles technologies
 - ▶ Pilote chargeable par *module*
⇒ changement de configuration aisément
(sans modification du système d'exploitation lui-même)



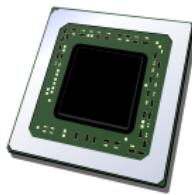
Système d'exploitation multi-tâches :

- ▶ Plusieurs programmes s'exécutent « en parallèle »
- ▶ Concurrence entre les programmes pour utiliser le processeur
- ▶ **Processus** : programme + informations de contexte



Exécution concurrente

Exécution de plusieurs processus sur un monoprocesseur :



CPU



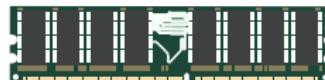
Navigation WEB



Compilation



Édition document

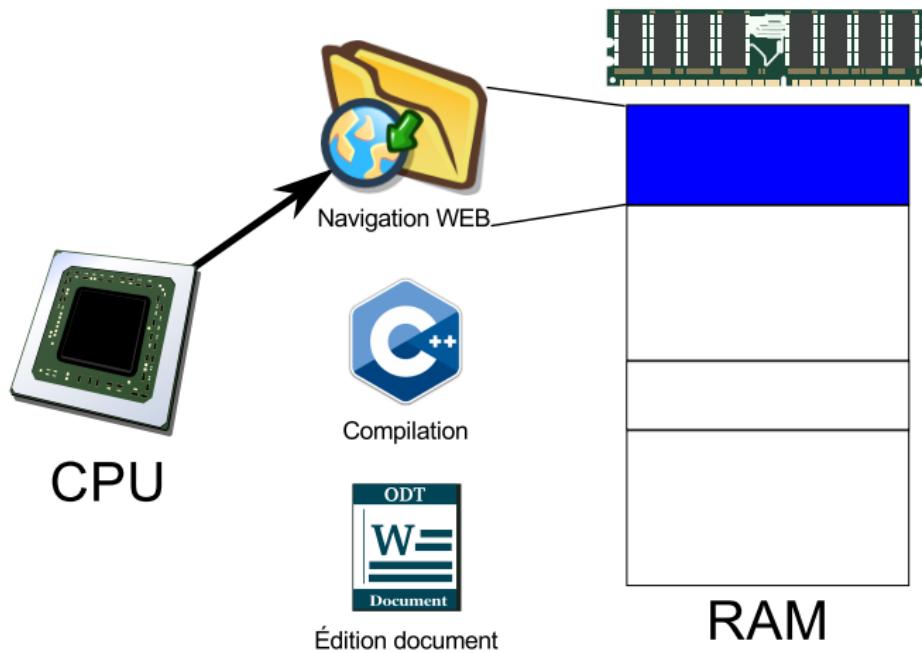


RAM



Exécution concurrente

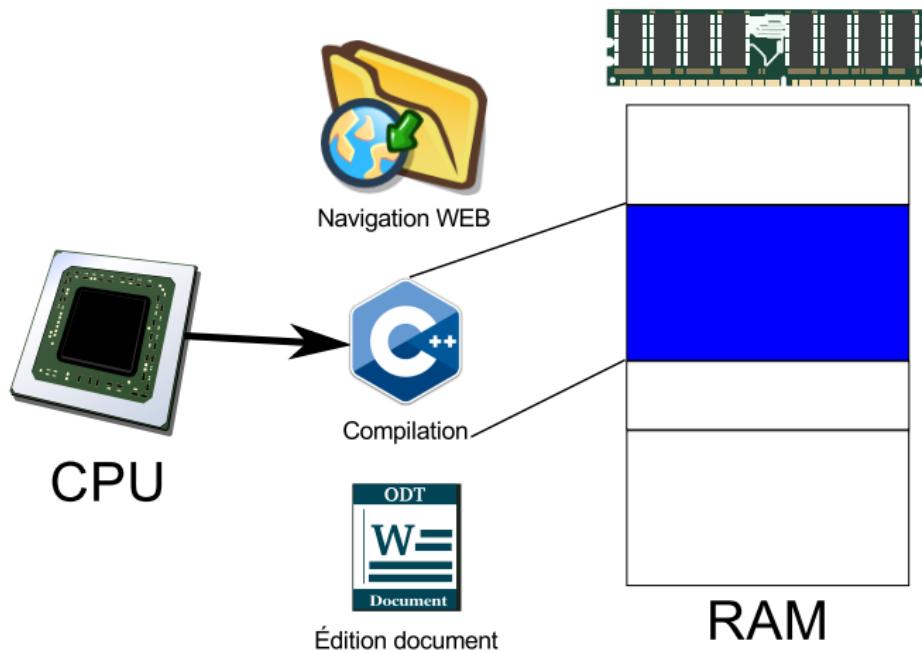
Exécution de plusieurs processus sur un monoprocesseur :





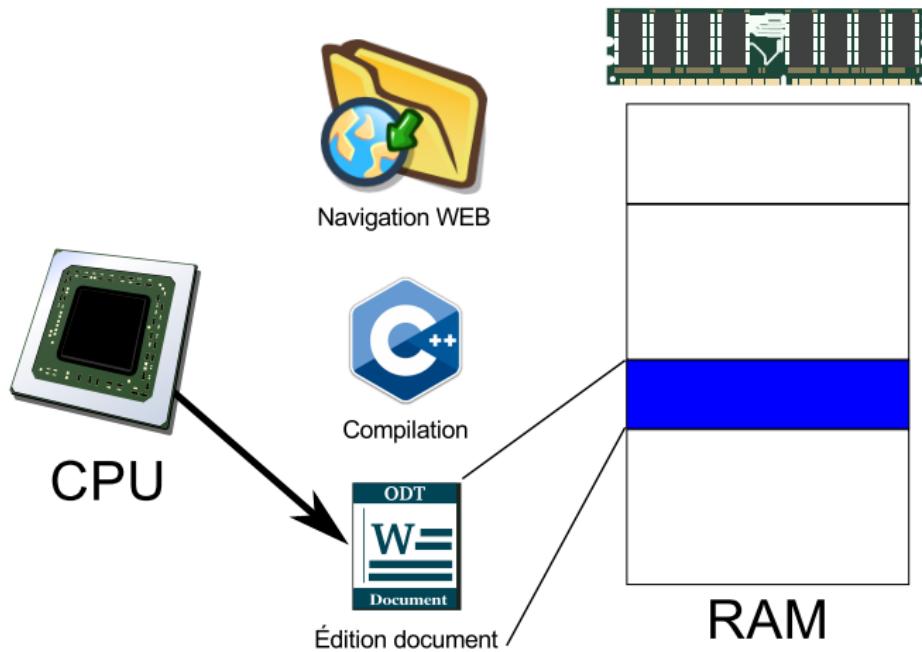
Exécution concurrente

Exécution de plusieurs processus sur un monoprocesseur :





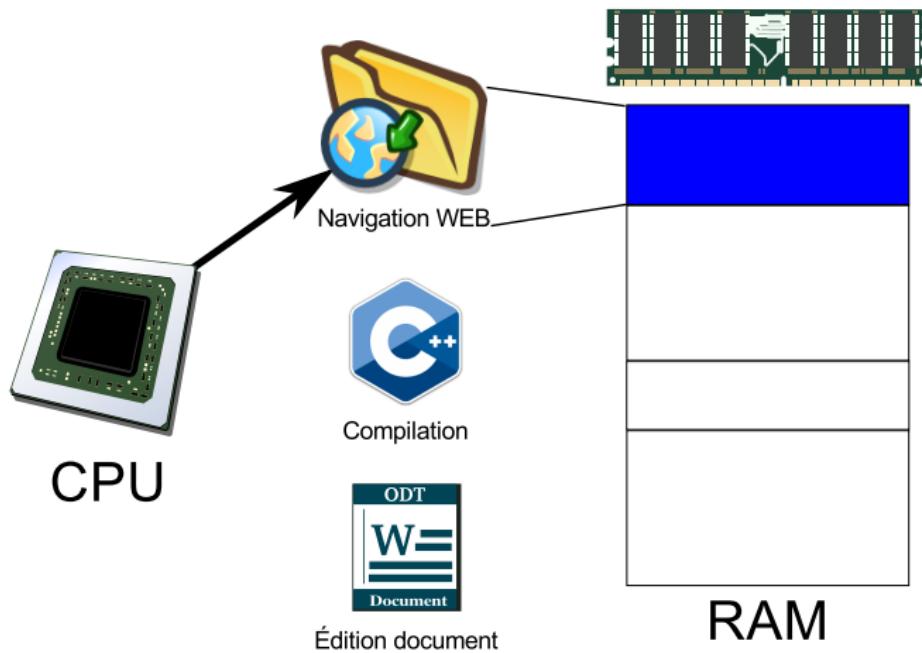
Exécution de plusieurs processus sur un monoprocesseur :





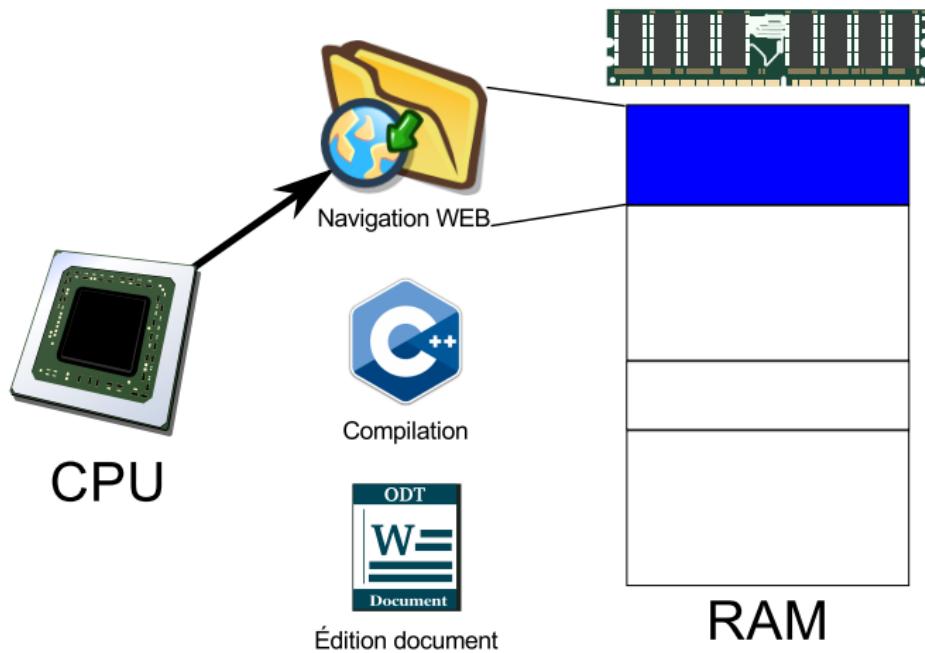
Exécution concurrente

Exécution de plusieurs processus sur un monoprocesseur :





Exécution de plusieurs processus sur un monoprocesseur :



- ▶ Temps partagé



- ▶ Une application \Rightarrow un ou plusieurs processus
- ▶ Informations de contexte :
 - ▶ Identificateur du processus
 - ▶ Valeur de **eip** + contenu des autres registres
 - ▶ État des mots d'état de la machine
 - ▶ Pointeurs sur la portion mémoire contenant le processus
 - ▶ Fichiers ouverts par le processus
 - ▶ Priorité du processus
 - ▶ État des périphériques utilisés par le processus
- ▶ *Temps partagé préemptif* : le noyau décide autoritairement qui peut s'exécuter à tout instant
- ▶ *Temps partagé coopératif* : chaque processus choisit quand passer la main



Les processus (1)

- ▶ Une application \Rightarrow un ou plusieurs processus
- ▶ Informations de contexte :
 - ▶ Identificateur du processus
 - ▶ Valeur de **eip** + contenu des autres registres
 - ▶ État des mots d'état de la machine
 - ▶ Pointeurs sur la portion mémoire contenant le processus
 - ▶ Fichiers ouverts par le processus
 - ▶ Priorité du processus
 - ▶ État des périphériques utilisés par le processus
- ▶ *Temps partagé préemptif* : le noyau décide autoritairement qui peut s'exécuter à tout instant
- ▶ *Temps partagé coopératif* : chaque processus choisit quand passer la main

PCB : *Process Control Block*



- ▶ **Processus *foreground*** (*en avant*) : processus acceptant des entrées au clavier, à la souris, ...
- ▶ **Processus *background*** (*tâche de fond*) : processus non interactif
- ▶ **Démon** : processus *background* effectuant une certaine action à des instants réguliers, ou répondant à certains évènements (e.g. aller rechercher le courrier électronique sur un serveur)

États possibles :

- ▶ *Élu* : en cours d'exécution par le processeur
- ▶ *Bloqué* : en attente d'un évènement extérieur
- ▶ *Prêt* : en attente d'exécution



- ▶ Système d'exploitation multi-tâche :
 - ▶ Plusieurs applications en mémoire en même temps
(butineur web, éditeur, ...)
 - ▶ Requiert beaucoup de mémoire
- ▶ *Mais* : mémoire RAM chère

Remarques :

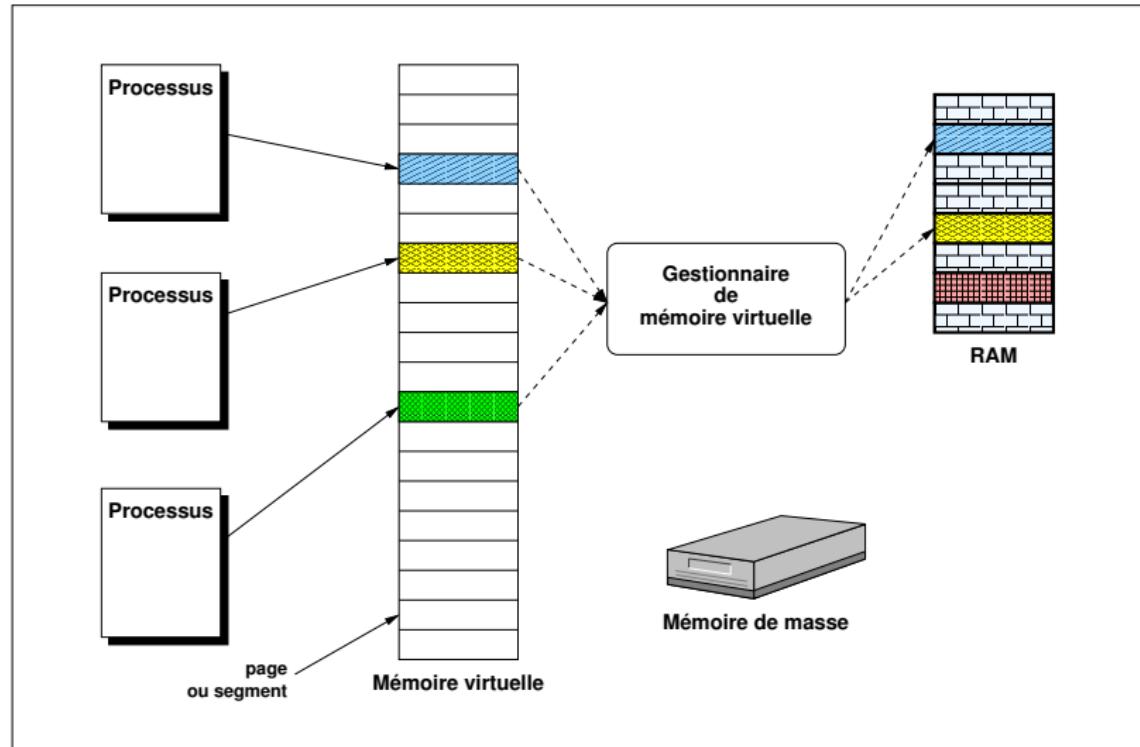
- ▶ On utilise rarement toutes les applications en même temps
- ▶ On n'a pas besoin de tout le code ou toutes les données d'une application en même temps

Idée :

- ▶ Utiliser le disque dur comme « prolongement » de la mémoire

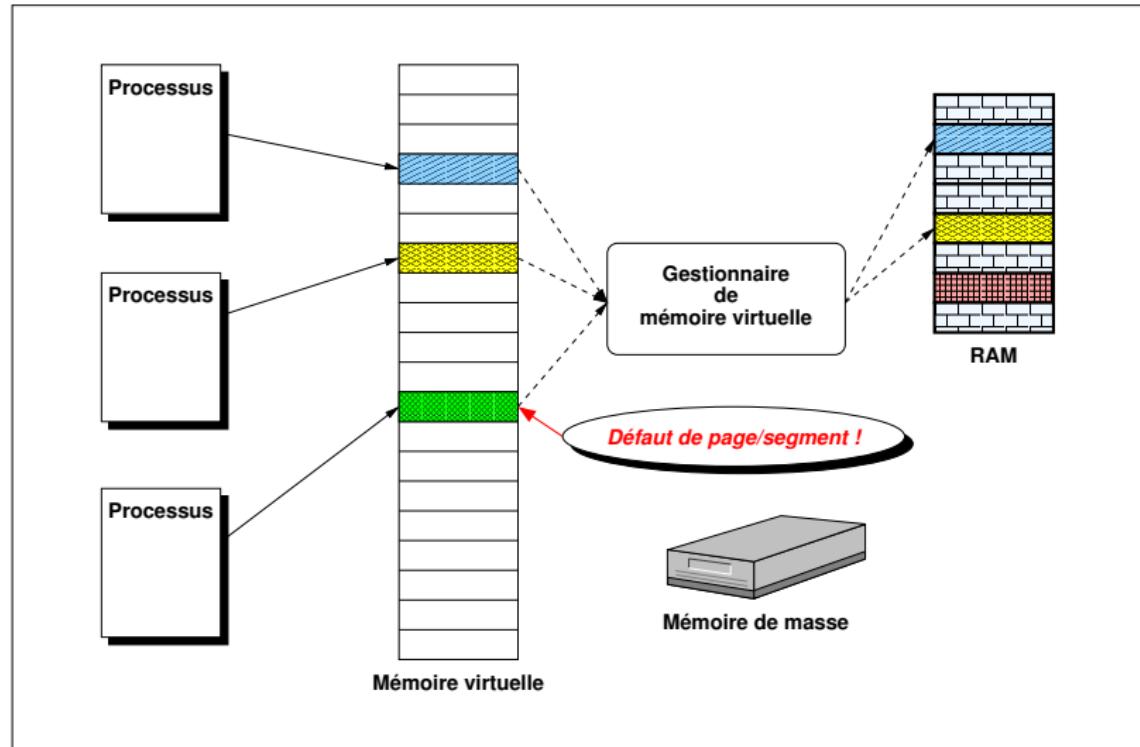


La mémoire virtuelle



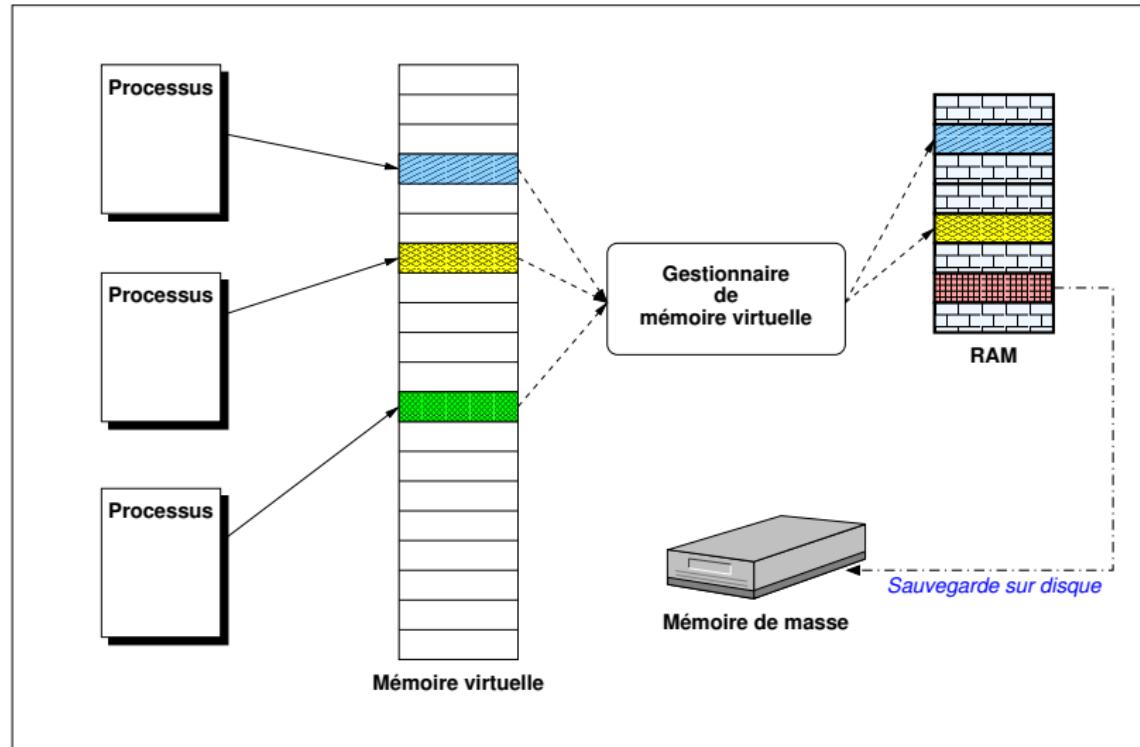


La mémoire virtuelle



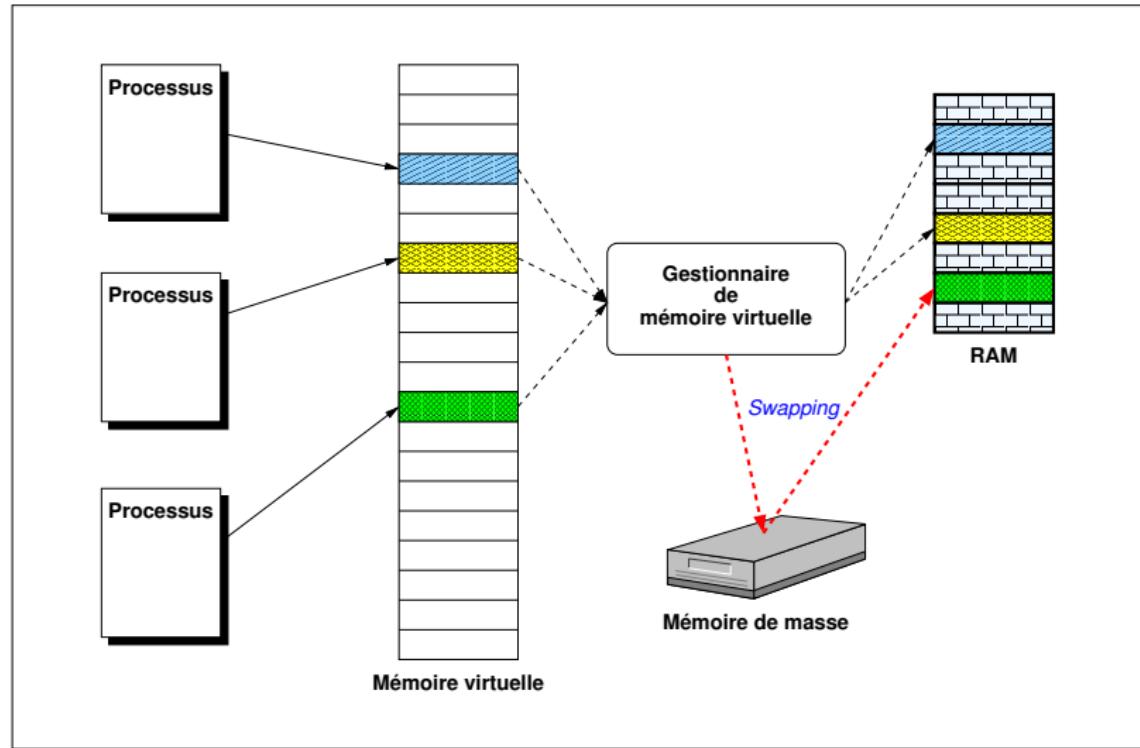


La mémoire virtuelle





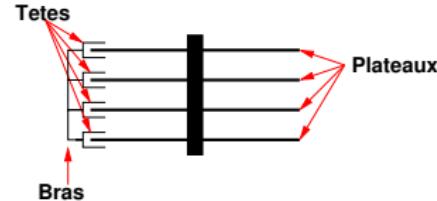
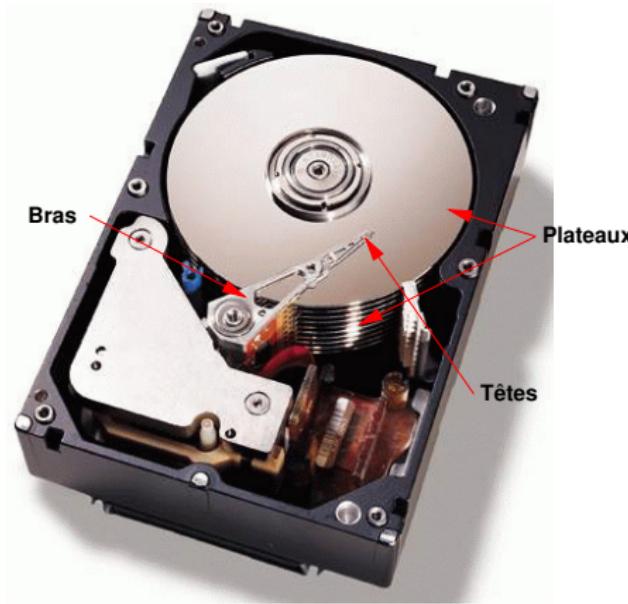
La mémoire virtuelle





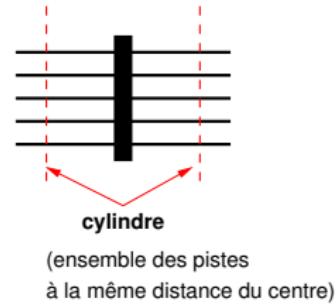
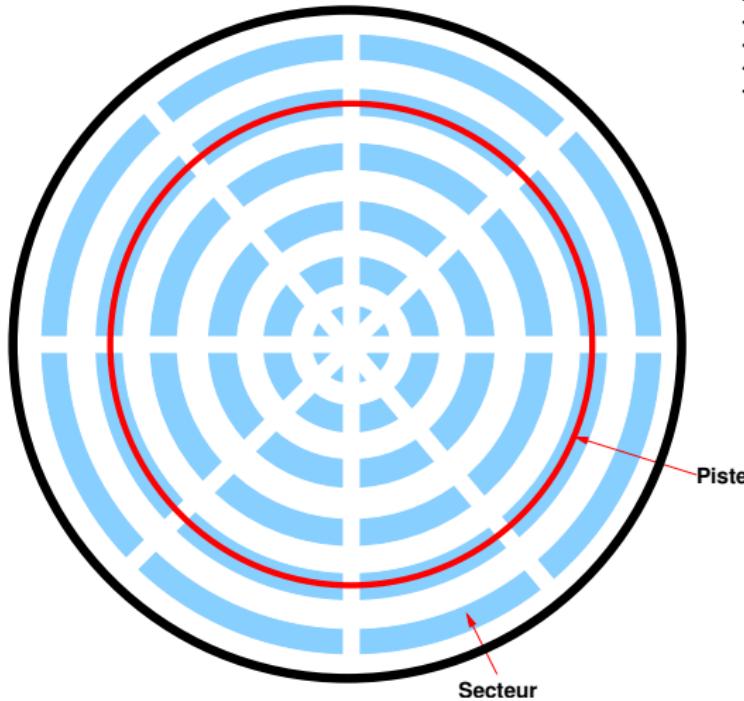
Le disque dur

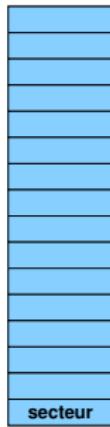
- ▶ Rotation rapides des plateaux (7500–15000 tours/mn)



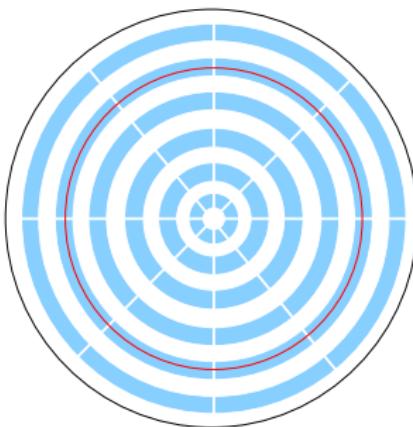


Les plateaux

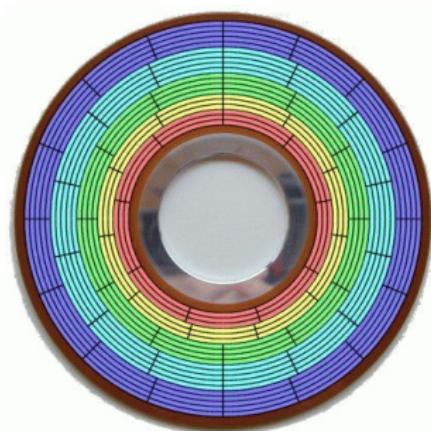




Vision logique
du système d'exploitation



Vision logique
du BIOS



Vision physique

- ▶ Nombre de bits par secteur constant + taille des secteurs croissante de l'intérieur vers l'extérieur \Rightarrow densité linéaire décroît.
- ▶ *Zoning* : augmentation du nombre de secteurs/piste vers l'extérieur pour conserver une densité linéaire maximale



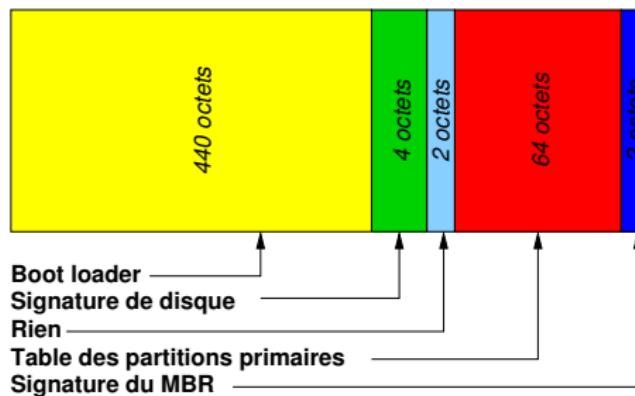
- ▶ *Formatage bas niveau* :
 - ▶ Création des pistes et des secteurs sur un plateau vierge
 - ▶ Ajout des informations de positionnement des secteurs/pistes
 - ▶ Enregistrement des secteurs défectueux
 - ▶ Indépendant du S.E.
- ▶ *Partitionnement* : découpage du disque en volumes physiques (e.g. C:, D: sous Windows)
- ▶ *Formatage haut niveau* : écriture du système de fichiers (dépendant du S.E.)



Le MBR (*Master Boot Record*)

- ▶ MBR : secteur 1, cylindre 0, tête 0 du disque dur
- ▶ Contient :
 - ▶ Le code de *bootstrap* donnant la main à un S.E.
Le boot loader
 - ▶ Un identifiant du disque
 - ▶ La liste et le type des *partitions primaires*

Le MBR





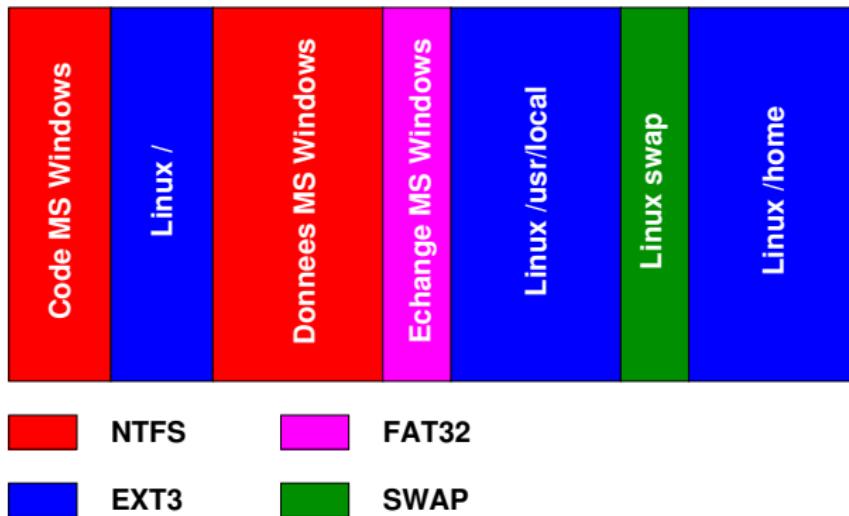
- ▶ Partition primaire : au plus 4 par disque
- ▶ Partition étendue (ou secondaire)
 - ▶ Une seule par disque
 - ▶ Peut contenir un nombre indéfini de disques logiques
- ▶ Partition active : partition *bootable* (avec un système d'exploitation)
- ▶ Droits associés aux partitions (lecture seule, ...)



Utilisation des partitions

- ▶ Un système de fichiers différent par partition
⇒ Un S.E. possible par partition

Exemple typique de partitionnement





Le bootloader de 2^e niveau

- ▶ GRUB, BOOTMGR, LILO, ...
- ▶ Programme dont le début se trouve en général dans le MBR
- ▶ Lance le système d'exploitation
- ▶ *Multiboot* : menu avec choix du S.E. à lancer



http://linux.softpedia.com/screenshots/GNU-GRUB_3.png



Démarrage de l'ordinateur :

- ▶ POST (*Power on Self Test*)
- ▶ Saut automatique à la mise sous tension à l'adresse 0xFFFF0 en mémoire (cases occupées par le BIOS—*Basic Input/Output System*)
- ▶ Exécution du BIOS de la carte video (adresse fixe : 0xC000)
- ▶ Recherche des autres périphériques à des adresses pré-déterminées
- ▶ Recherche d'un disque contenant un *système d'exploitation* à charger :
 - ▶ Sur disque dur : lecture du MBR *Master Boot Record*, cylindre 0, tête 0, secteur 1
 - ▶ Le BIOS passe à la main au code du MBR
 - ➡ **Chargement du système d'exploitation**
- ▶ BIOS remplacé par UEFI (*Unified Extensible Firmware Interface*)



- ▶ Vision « bas niveau » d'un périphérique de stockage : tableau de secteurs (1 secteur = n octets)
- ▶ Un fichier utilisateur = plusieurs secteurs

Système de fichiers :

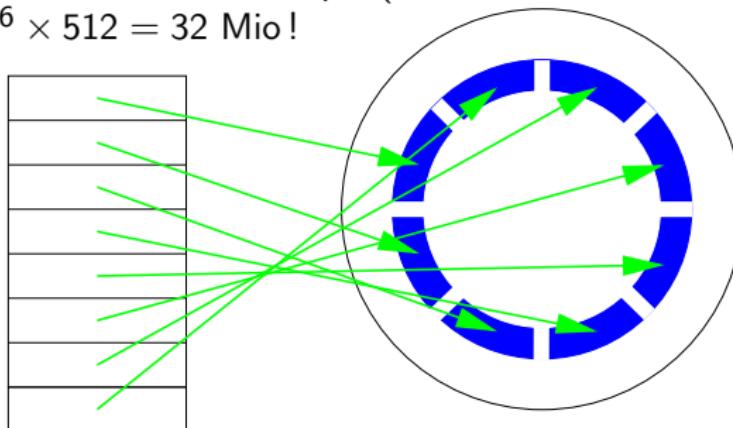
- ▶ Gestion des secteurs occupés (par quel fichier ?), libres, défectueux, ...
- ▶ Décomposition hiérarchique en répertoires et fichiers
- ▶ Types de permissions des fichiers (lisible, modifiable, archivable, caché, ...)



Exemple : le système FAT (1)

FAT (*File Allocation Table*)

- ▶ Système de fichiers utilisé par MS-DOS
- ▶ FAT = table de correspondances entre des « régions » du disque et les fichiers
- ▶ Une entrée de la FAT : entier sur 12/16/32 bits (FAT 12/16/32)
- ▶ Correspondance entrée/secteur ?
 - taille maximum du disque (avec secteurs de 512 octets) = $2^{16} \times 512 = 32 \text{ Mio} !$

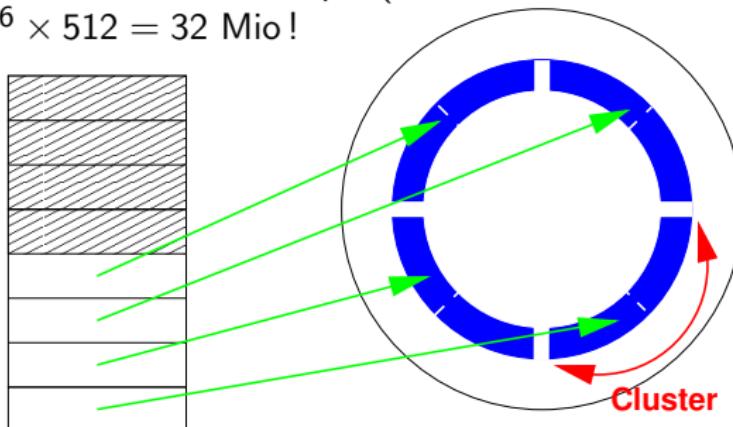




Exemple : le système FAT (1)

FAT (*File Allocation Table*)

- ▶ Système de fichiers utilisé par MS-DOS
- ▶ FAT = table de correspondances entre des « régions » du disque et les fichiers
- ▶ Une entrée de la FAT : entier sur 12/16/32 bits (FAT 12/16/32)
- ▶ Correspondance entrée/secteur ?
 - taille maximum du disque (avec secteurs de 512 octets) = $2^{16} \times 512 = 32 \text{ Mio} !$



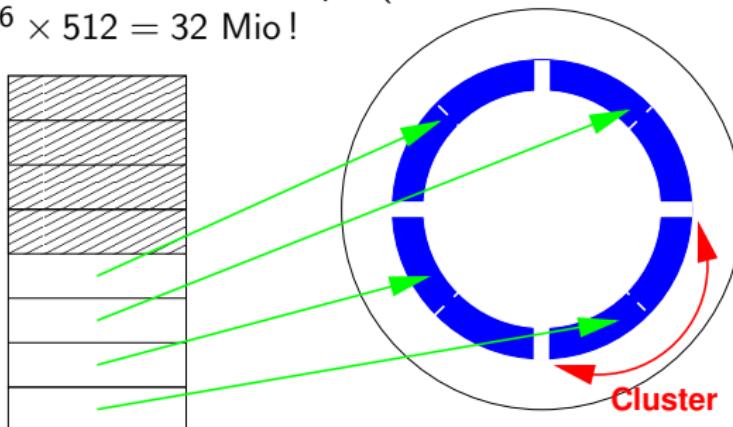


Exemple : le système FAT (1)

FAT (*File Allocation Table*)

- ▶ Système de fichiers utilisé par MS-DOS
- ▶ FAT = table de correspondances entre des « régions » du disque et les fichiers
- ▶ Une entrée de la FAT : 12/16/32)
- ▶ Correspondance entrée/secteur ?
 - taille maximum du disque (avec secteurs de 512 octets) = $2^{16} \times 512 = 32 \text{ Mio !}$

Cluster. *Groupe de secteurs. Taille choisie en fonction de celle de la partition.*





- ▶ *Secteur*. Plus petite unité lue ou écrite en une fois sur le disque dur
- ▶ *Cluster*. Plus petite unité allouée à un fichier (un fichier est représenté par un nombre entier de clusters)

Gaspillage de place (cluster de 2048 octets) :

- ▶ Un fichier de 1024 octets occupe 1 cluster (perte = 1024 octets)
- ▶ Un fichier de 4097 octets occupe 3 clusters (perte = 2047 octets)

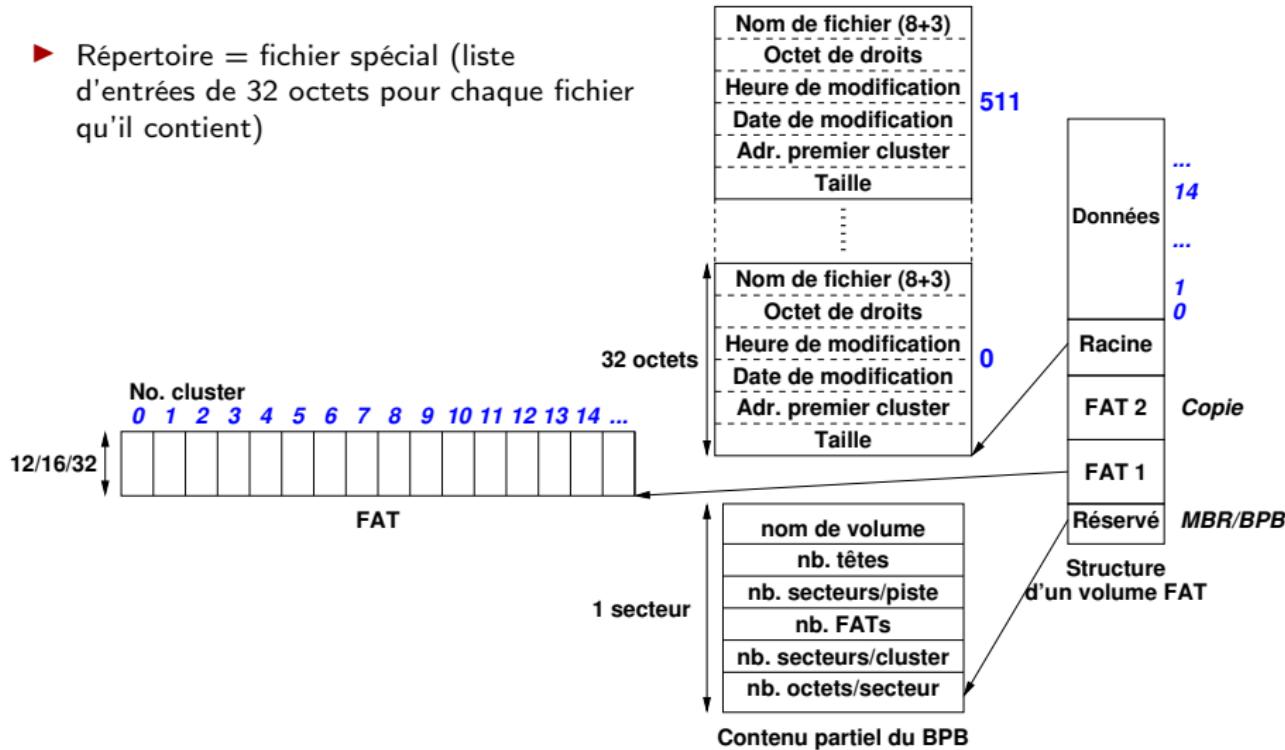
Pour un gros disque :

- ▶ Petit cluster \Rightarrow grosse FAT, peu de gaspillage
- ▶ Gros cluster \Rightarrow petite FAT, beaucoup de gaspillage



Le système FAT (2)

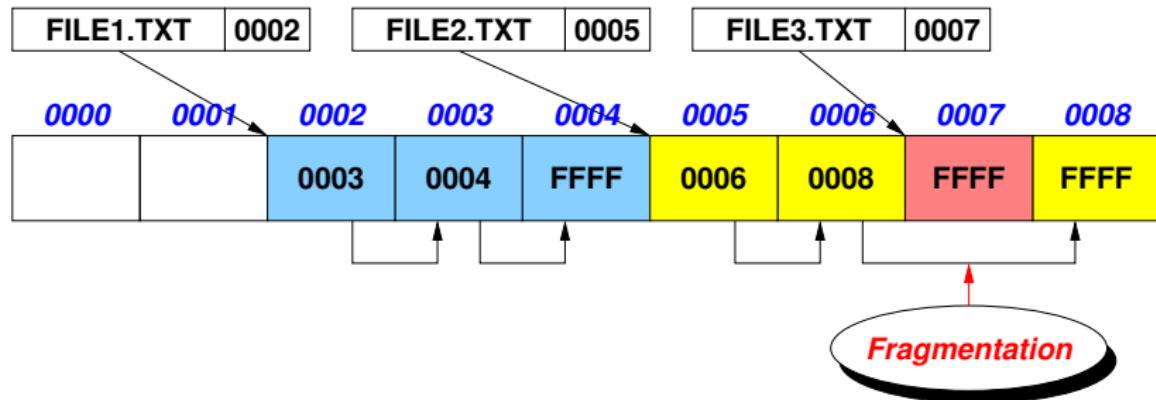
- ▶ Répertoire = fichier spécial (liste d'entrées de 32 octets pour chaque fichier qu'il contient)





Le système FAT (3)

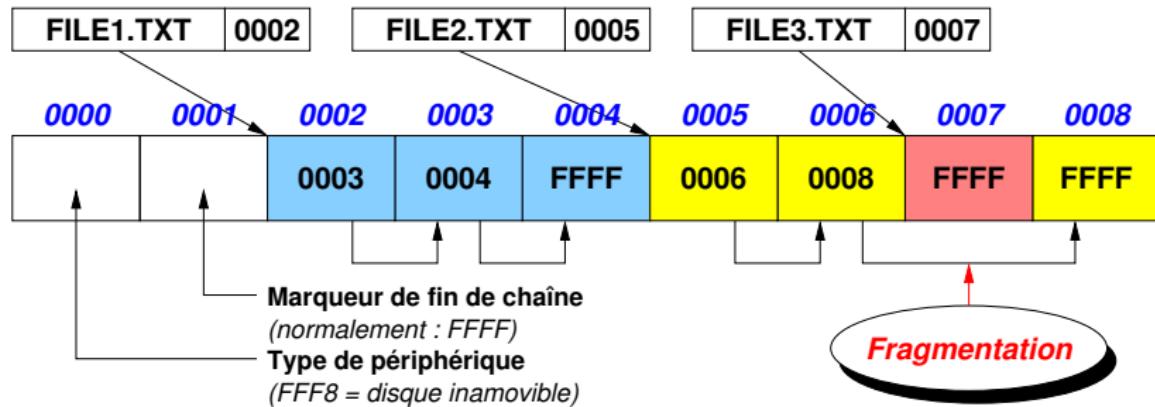
Partie « données » :





Le système FAT (3)

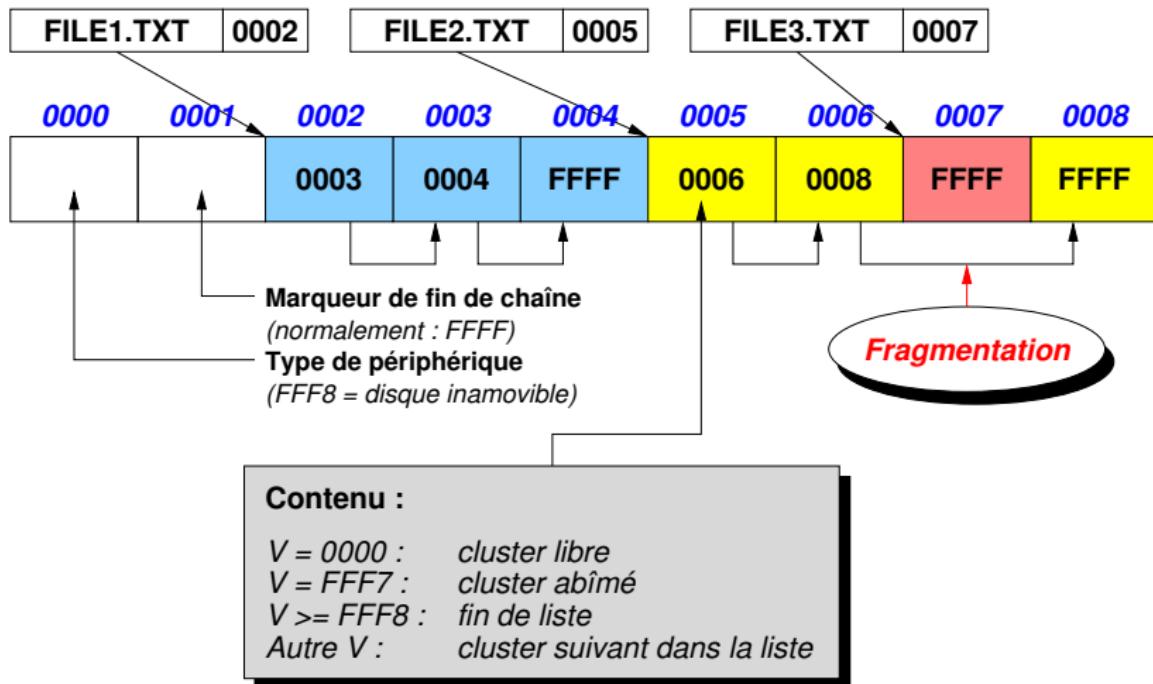
Partie « données » :





Le système FAT (3)

Partie « données » :

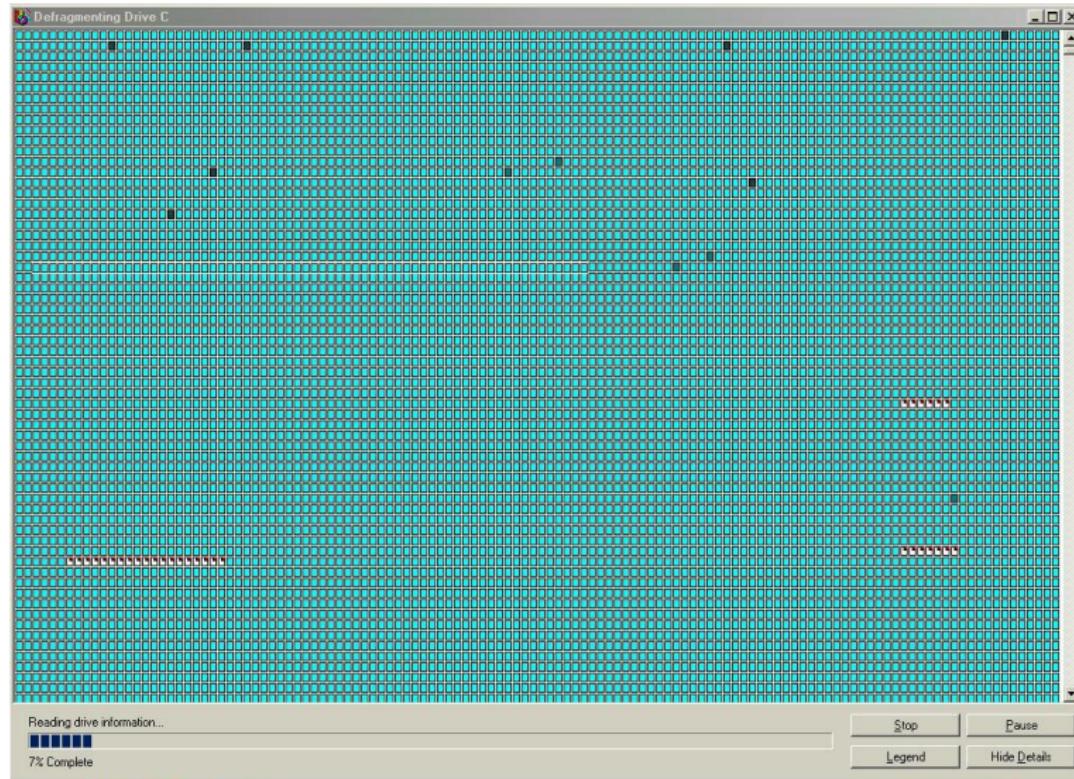




- ▶ Stockage des fichiers dans des clusters non consécutifs
- ▶ Plus le disque est utilisé (effacements/ajouts de fichiers), plus la fragmentation augmente
- ▶ Fragmentation = diminution des performances (augmentation déplacement des têtes du disque)
- ▶ Défragmentation : nécessaire pour des systèmes FAT



Défragmentation

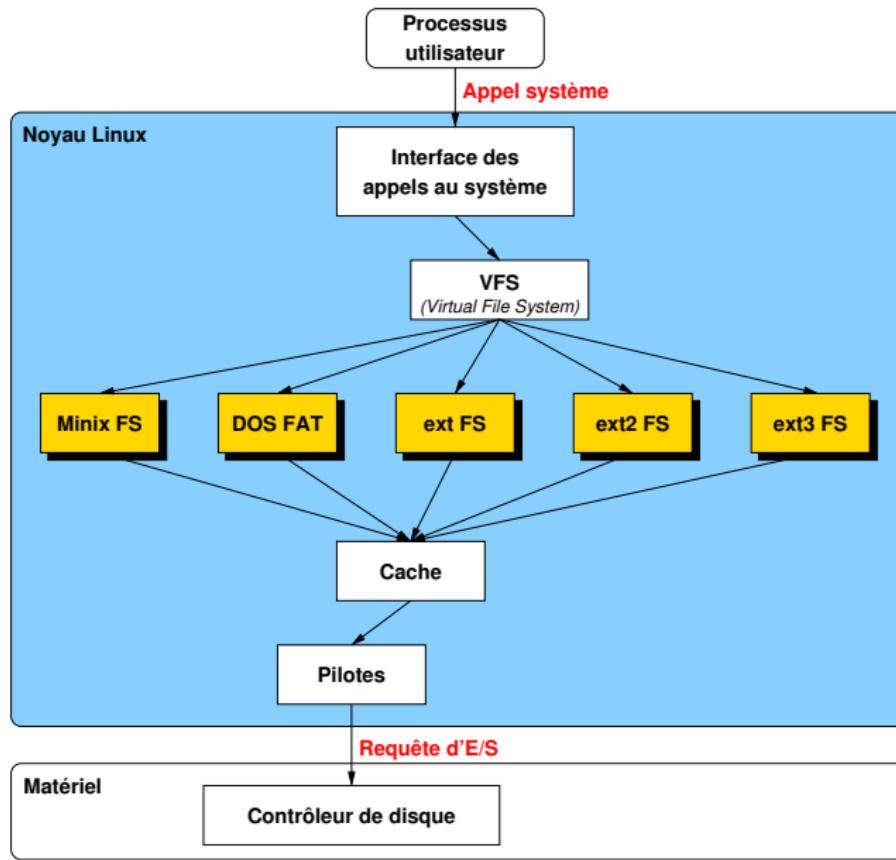




- ▶ **FAT 32** (Windows 9x) :
 - ▶ Pointeurs de clusters sur 32 bits
 - ▶ Noms longs en plus du format 8+3
- ▶ **NTFS** : très différent du système FAT (mais compatible)
 - ▶ Support de noms de fichiers en Unicode
 - ▶ Attributs supplémentaires pour les fichiers (droits d'accès)
 - ▶ Compression/décompression à la volée des fichiers peu utilisés

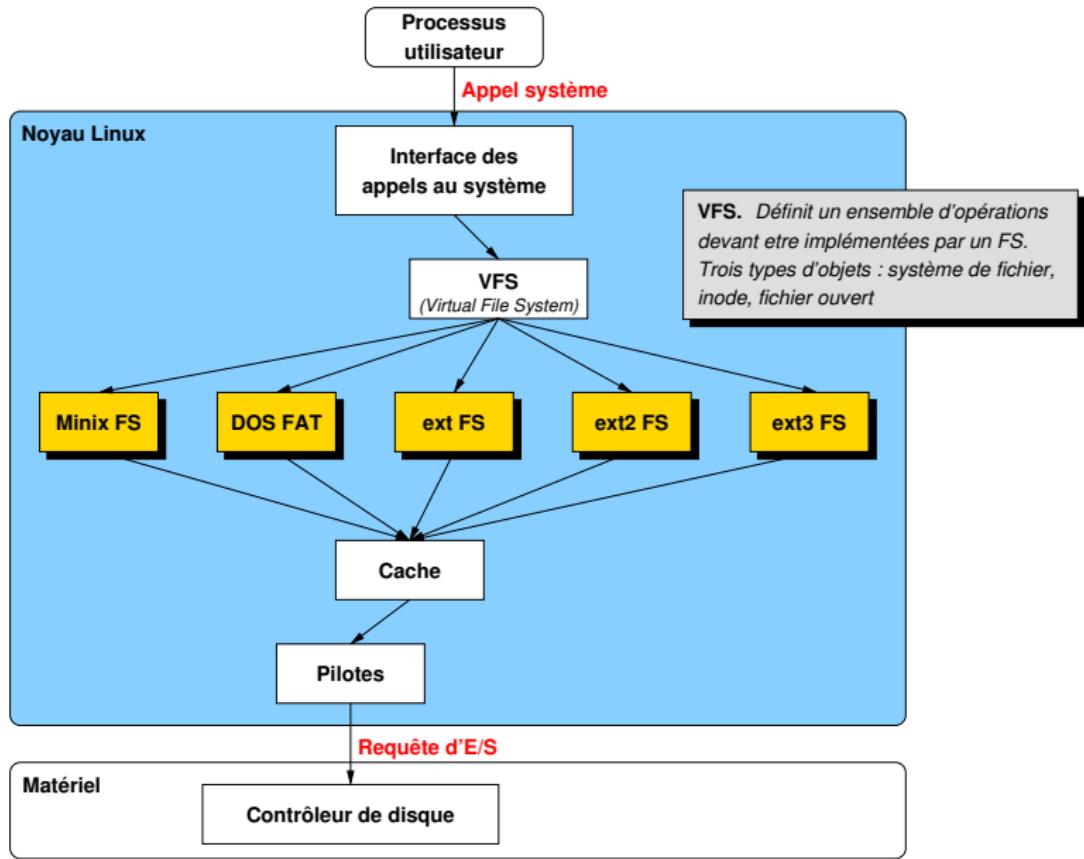


Le système de fichiers sous Linux





Le système de fichiers sous Linux





DOS/Windows :

- ▶ Répertoires : fichiers particuliers
- ▶ Périphériques de stockage : lettre associée (**C:**, **D:**, ...)
- ▶ Autres périphériques : pas reliés au système de fichier

Unix/Linux :

- ▶ Répertoires : fichiers particuliers
- ▶ *Tous* les périphériques : fichiers spéciaux dans l'arborescence globale
 - ▶ Souris (**/dev/mouse**), CDROM (**/dev/cdrom**), ...
 - ▶ Mémoire (**/proc/meminfo**)



Caractéristiques des syst. de fichiers

Nom	Taille nom de fichier	Taille max. d'un fichier	Taille max. d'une partition
Minix FS	30 car.	64 Mio	64 Mio
FAT 16	8+3 car.	2 Gio	2 Gio
FAT 32	255 car.	4 Gio	8 Tio
NTFS	255 car.	16 Tio	256 Tio
Ext2/Ext3 FS	255 car.	2 Tio	32 Tio
Ext4 FS	255 car.	16 Tio	1 Eio
JFS	255 car.	4 Pio	32 Pio

- ▶ 1 Eio (*Exbiocet*) = 2^{60} octets = 1024 Pio
- ▶ 1 Pio (*Pebiocet*) = 2^{50} octets = 1024 Tio
- ▶ 1 Tio (*Tebiocet*) = 2^{40} octets = 1024 Gio
- ▶ 1 Gio (*Gibiocet*) = 2^{30} octets = 1024 Mio



File Hierarchy Standard : Organisation recommandée des répertoires (et partitions) sous Unix/Linux + commandes essentielles devant être présentes

Résumé :

/bin	Commandes système essentielles
/boot	Fichiers statiques du bootloader
/dev	Fichiers <i>device</i>
/etc	Fichiers de configuration
/home	Comptes des utilisateurs
/lib	Librairies système
/media	point de montage pour media déconnectable
/mnt	Point de montage temporaire
/root	Compte du <i>root</i>
/tmp	Fichiers temporaires
/usr	Hiérarchie secondaire
/var	Fichiers de données amenés à changer



File Hierarchy Standard : Organisation recommandée des répertoires (et partitions) sous Unix/Linux + commandes essentielles devant être présentes

Résumé :

/usr/bin	Exécutables des applications
/usr/lib	Librairies des applications
/usr/include	Fichiers d'en-têtes
/usr/local/bin	Applications locales
/usr/local/lib	Librairies locales
/usr/X11R6	X Windows
...	



Distribution Linux = noyau Linux
+ packaging d'installation
+ applications

- ▶ (K)Ubuntu
- ▶ Debian
- ▶ Mandriva
- ▶ RedHat
- ▶ Suse
- ▶ ...



Exemple : Kubuntu

Language		
Amharic	Hebrew	Polski
Arabic	Hindi	Português do Brasil
Беларуская	Hrvatski	Português
Български	Magyar	Română
Bengali	Bahasa Indonesia	Русский
Bosanski	Italiano	Sāmegillii
Català	日本語	Slovenčina
Čeština	ქართული	Slovenščina
Dansk	Khmer	Shqip
Deutsch	한국어	Svenska
Dzongkha	Kurdî	Tamil
Ελληνικά	Lietuviškai	Thai
English	Latviski	Tagalog
Esperanto	Македонски	Türkçe
Español	Malayalam	Українська
Eesti	Marathi	Tiếng Việt
Euskaraz	Norsk bokmål	Wolof
Suomi	Nepali	中文(简体)
Français	Nederlands	中文(繁體)
Galego	Norsk nynorsk	
Gujarati	Punjabi (Gurmukhi)	

26 s

F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options



Exemple : Kubuntu



Try Kubuntu without any change to your computer

Install Kubuntu

Check CD for defects

Test memory

Boot from first hard disk

Press F4 to select alternative start-up and installation modes.

F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

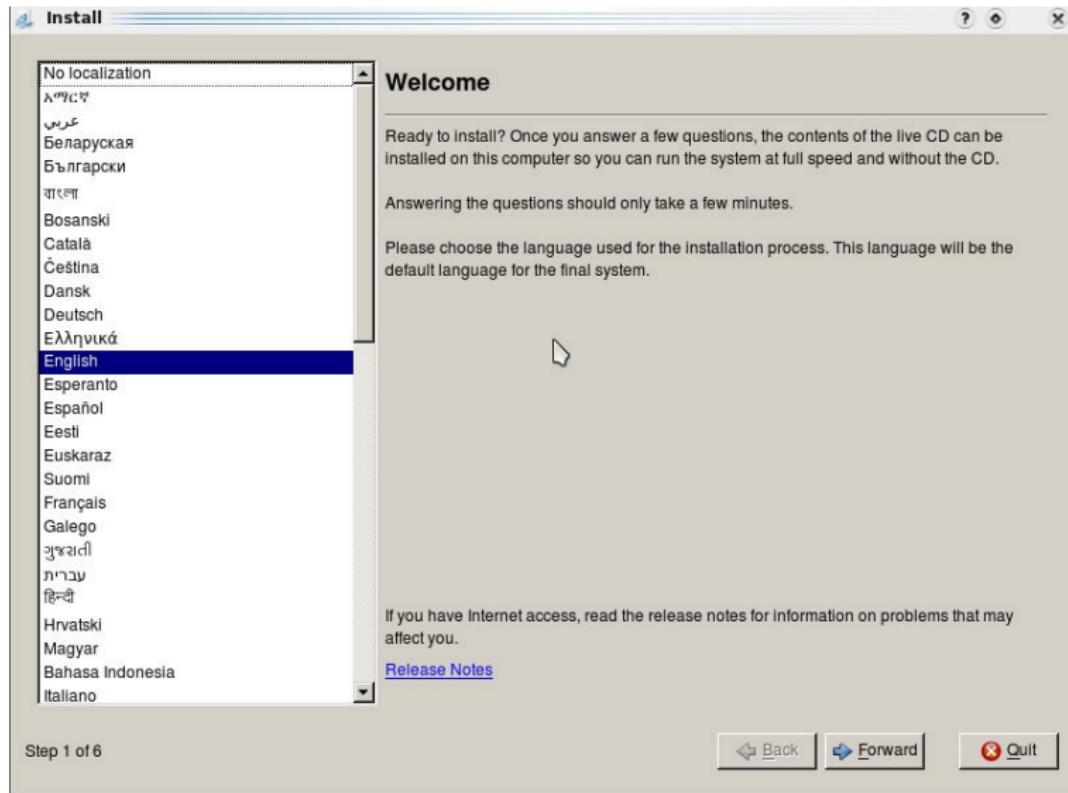


Exemple : Kubuntu



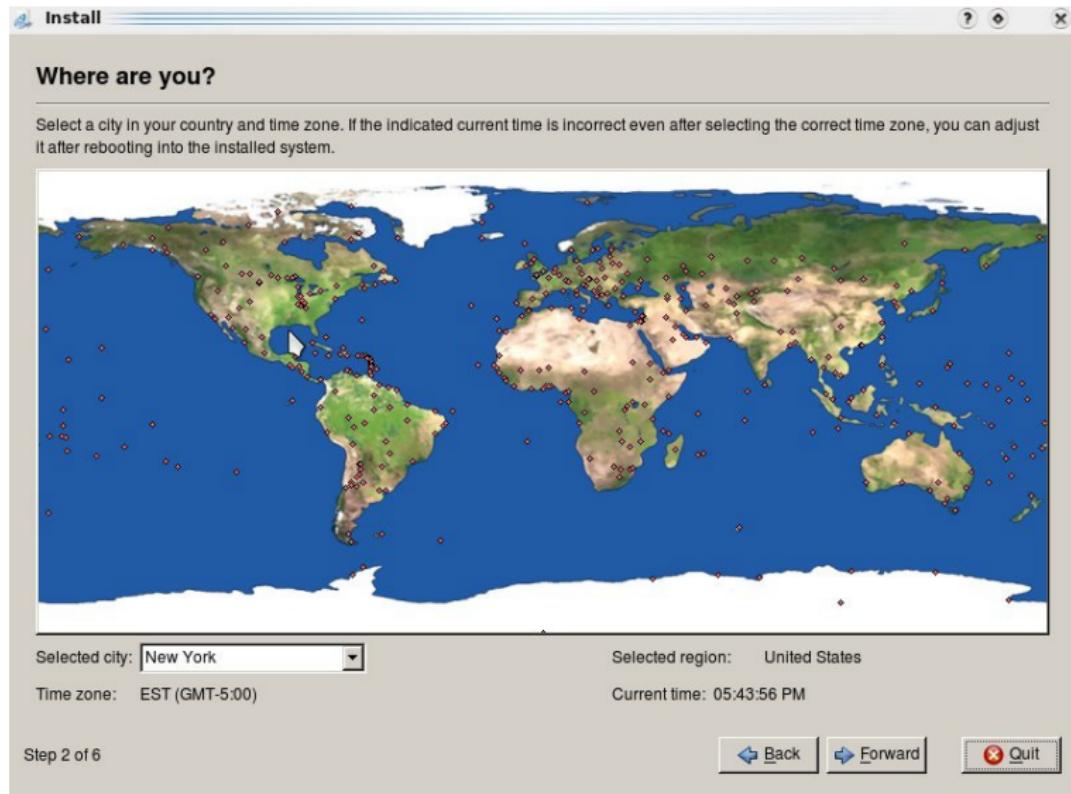


Exemple : Kubuntu





Exemple : Kubuntu





Exemple : Kubuntu

Install

Keyboard layout

Which layout is most similar to your keyboard?

Country	Layout
Canada	France
China	France - (Legacy) Alternative
Congo, Democratic Republic of the	France - (Legacy) Alternative, Sun dead keys
Croatia	France - (Legacy) Alternative, eliminate dead keys
Czechia	France - (Legacy) Dvorak
Denmark	France - Alternative
Esperanto	France - Alternative, Sun dead keys
Estonia	France - Alternative, eliminate dead keys
Ethiopia	France - Alternative, latin-9 only
Faroe Islands	France - Bépo, ergonomic, Dvorak way
Finland	France - Bépo, ergonomic, Dvorak way, latin-9 only
France	France - Eliminate dead keys
Georgia	France - Georgian AZERTY Tskapo
Germany	France - Macintosh
Ghana	France - Sun dead keys
Greece	
Guinea	
Hungary	
Iceland	
India	
Iran	

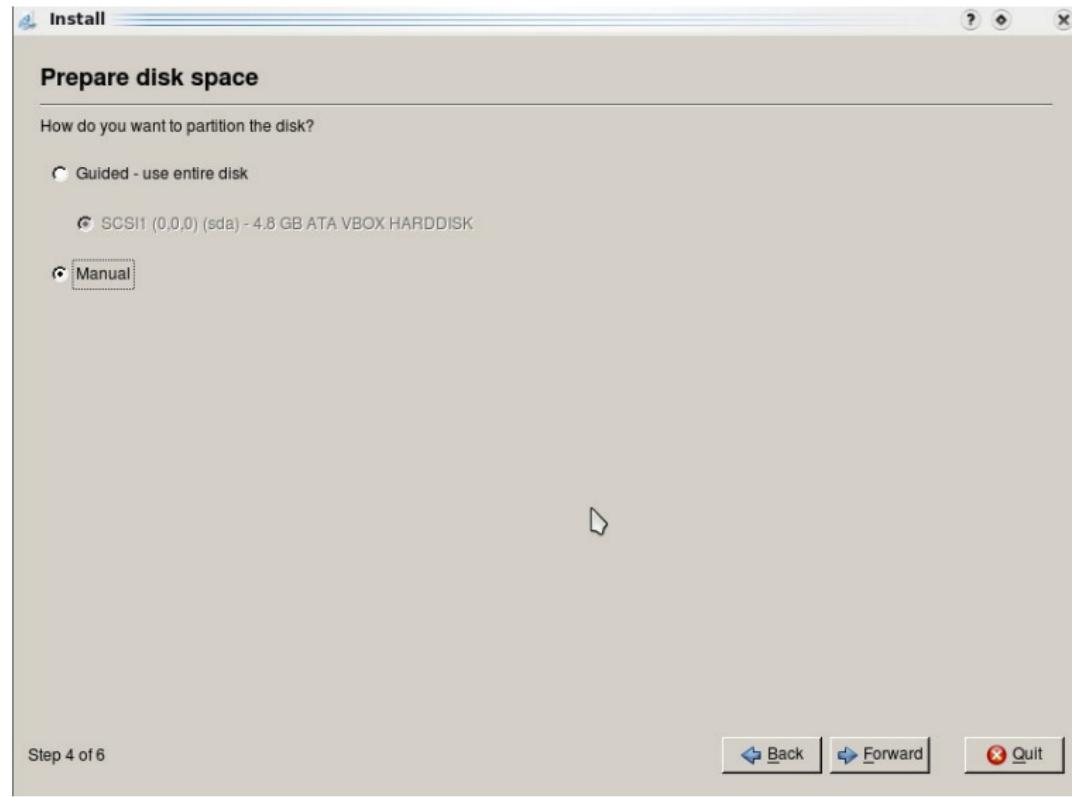
You can type into this box to test your new keyboard layout.

Step 3 of 6

Back Forward Quit

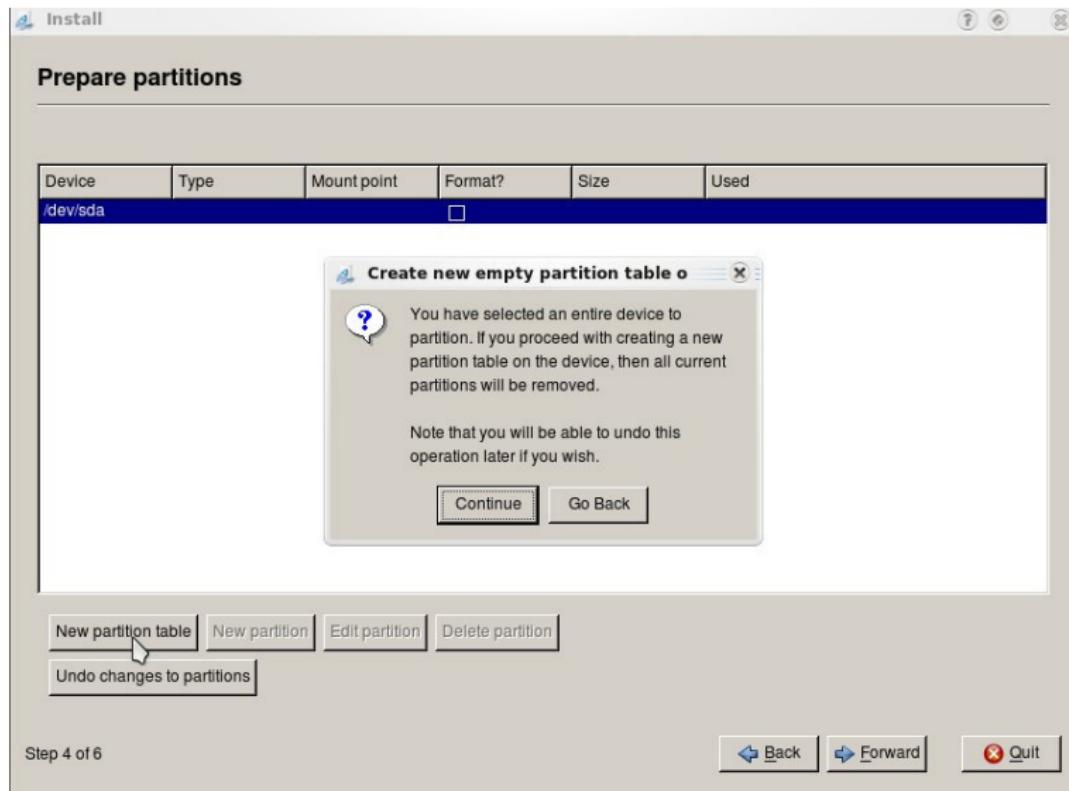


Exemple : Kubuntu



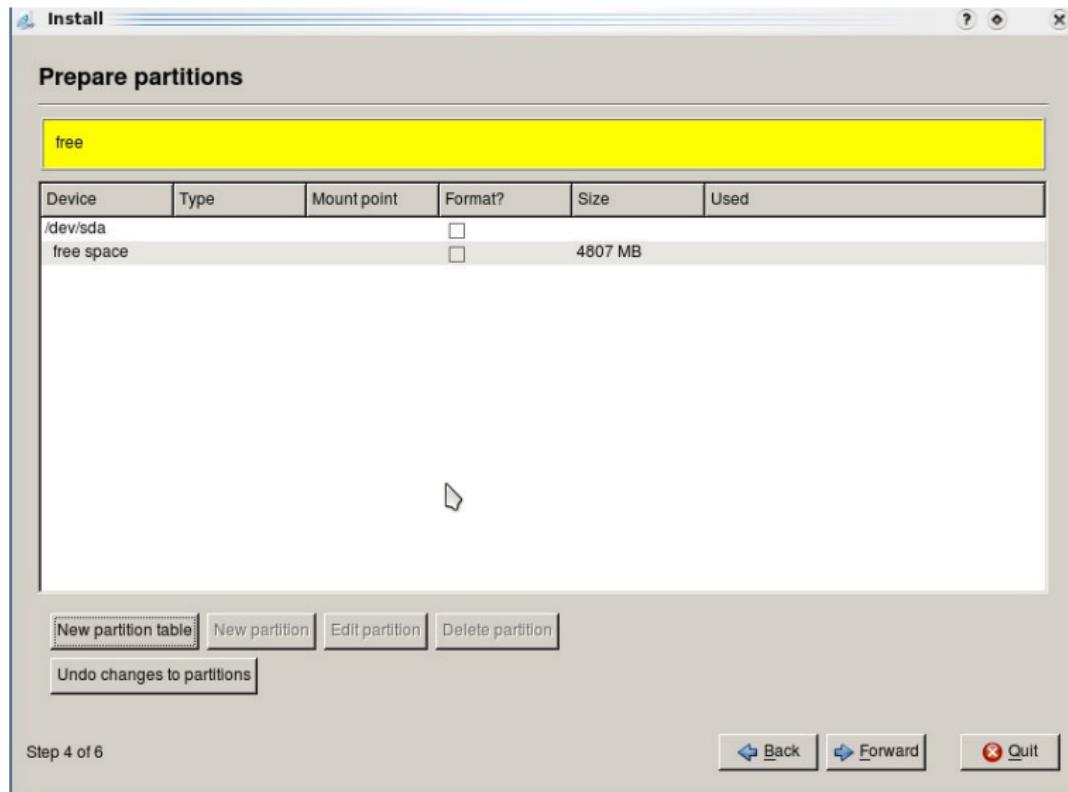


Exemple : Kubuntu



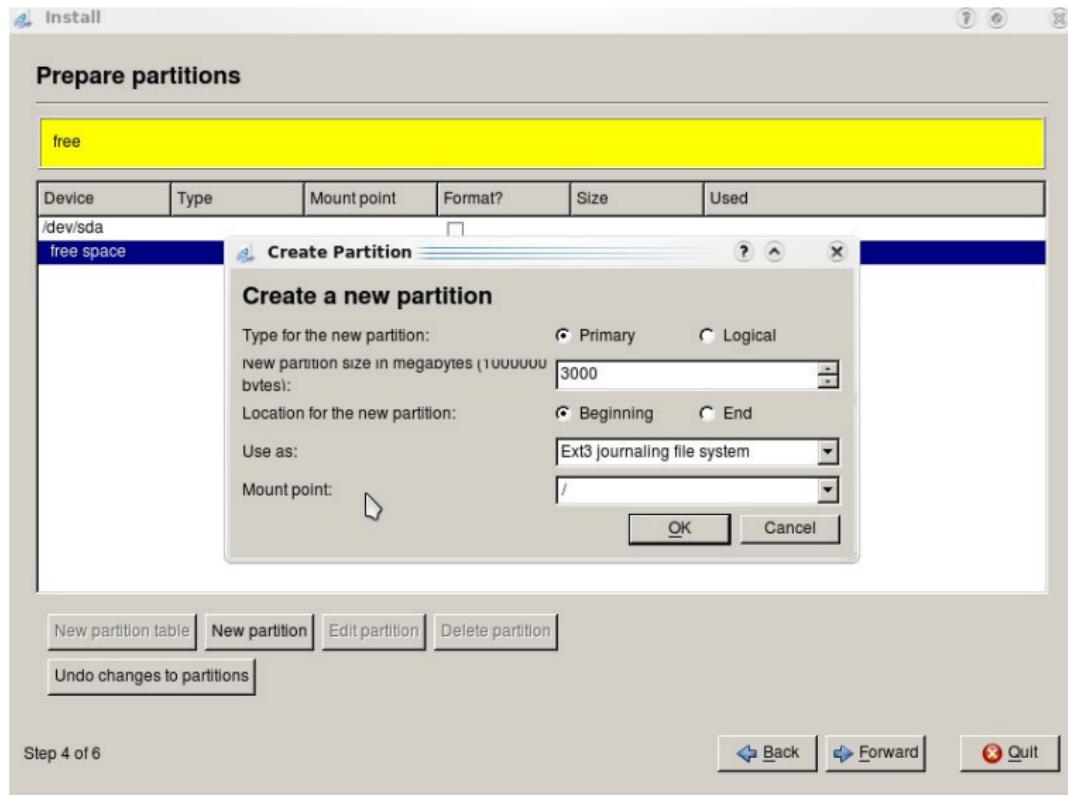


Exemple : Kubuntu



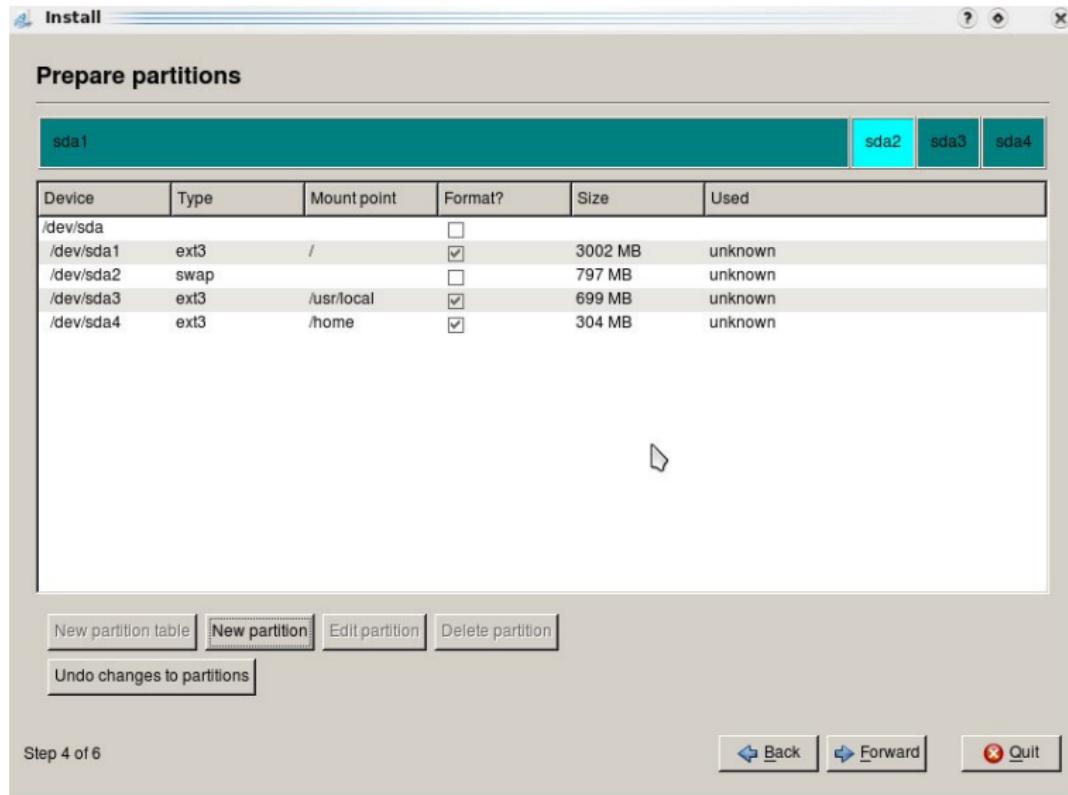


Exemple : Kubuntu





Exemple : Kubuntu





Exemple : Kubuntu

Install ? ×

Who are you?

What is your name?

What name do you want to use to log in?

If more than one person will use this computer, you can set up multiple accounts after installation.

Choose a password to keep your account safe.

Enter the same password twice, so that it can be checked for typing errors.

What is the name of this computer?

This name will be used if you make the computer visible to others on a network.

Step 5 of 6

Back Forward Quit



Exemple : Kubuntu

Install

Ready to install

Your new operating system will now be installed with the following settings:

Details

Language: English
Keyboard layout: France
Name: Frederic Goualard
Login name: goualard
Location: Europe/Paris
Migration Assistant:

If you continue, the changes listed below will be w...
Otherwise, you will be able to make further change...

WARNING: This will destroy all data on any partic...
well as on the partitions that are going to be formatted.

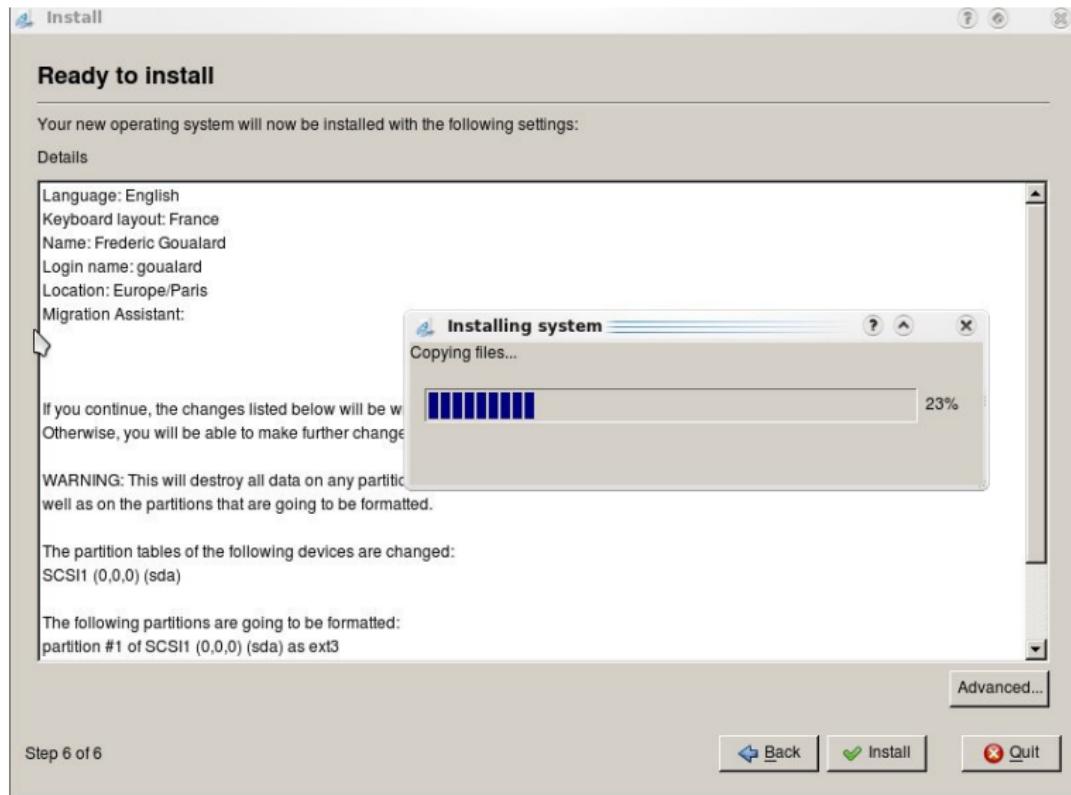
The partition tables of the following devices are changed:
SCSI1 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI1 (0,0,0) (sda) as ext3

Advanced...

Step 6 of 6

Back Install Quit





Exemple : Kubuntu

Ready to install

Your new operating system will now be installed with the following settings:

Details

Language: English
Keyboard layout: France
Name: Frederic Goualard
Login name: goualard
Location: Europe/Paris
Migration Assistant:

If you continue, the changes listed below will be made to your computer. If you do not want to make these changes, you can cancel the installation now.

WARNING: This will destroy all data on the selected device as well as on the partitions that are going to be formatted.

The partition tables of the following devices are changed:
SCSI1 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI1 (0,0,0) (sda) as ext3

Installation complete

Installation is complete. You need to restart the computer in order to use the new installation.

Restart now

Step 6 of 6

Advanced...

Back **Install** **Quit**



- ▶ Installation/désinstallation propre d'applications
 - ⇒ *paquetages*
 - ▶ Système de paquetages dépend de la distribution :
 - ▶ RPM : RedHat, Mandriva
 - ▶ DEB : Debian, (K)Ubuntu
 - ▶ ...
 - ▶ Liste de dépendances pour (dés-)installation correcte
 - ▶ Installation d'applications suivant le *GNU build System*



Installation par paquetage

```
goualard : apt-get
File Edit View Scrollback Bookmarks Settings Help
goualard@goualard-laptop:~$ sudo apt-get install dia
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dia-common dia-libs gconf2 gconf2-common gsfonts-x11 libgconf2-4 libidl0
  liborbit2 python-cairo python-gtk2 python-numeric xutils-dev
Suggested packages:
  python-gtk2-doc python-gtk2-dbg python-numeric-tutorial python-numeric-ext
  python-numeric-dbg
The following NEW packages will be installed:
  dia dia-common dia-libs gconf2 gconf2-common gsfonts-x11 libgconf2-4 libidl0
  liborbit2 python-cairo python-gtk2 python-numeric xutils-dev
0 upgraded, 13 newly installed, 0 to remove and 222 not upgraded.
Need to get 5748kB of archives.
After this operation, 37.3MB of additional disk space will be used.
Do you want to continue [Y/n]? █
```



Installation à partir du source

```
goualard@almighty> tar -zxf hello-2.4.tar.gz
goualard@almighty> cd hello-2.4
goualard@almighty> ls
ABOUT-NLS  ChangeLog  configure.ac  gnlub      man      src
aclocal.m4  ChangeLog.O  contrib     INSTALL    NEWS     tests
AUTHORS    config.in   COPYING      Makefile.am  po       THANKS
build-aux   configure  doc         Makefile.in README  TODO
goualard@almighty> ./configure --prefix=/usr/local
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
[...]
config.status: executing depfiles commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
goualard@almighty> make
make all-recursive
make[1]: Entering directory '/home/goualard/temp/hello-2.4'
[...]
goualard@almighty> sudo make install
Password:
Making install in contrib
make[1]: Entering directory '/home/goualard/temp/hello-2.4/contrib'
make[2]: Entering directory '/home/goualard/temp/hello-2.4/contrib'
[...]
```

Fin du cours