Programmation WEB

Sébastien MAVROMATIS - sebastien.mavromatis@univ-amu.fr

Peter BANTON - peter.banton@univ-amu.fr

https://mavromatis.org

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Qu'est ce que l'informatique ?

Pour l'Académie Française :

"Science du traitement rationnel, notamment par des machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux"

Pour Philippe DREYFUS (l'inventeur du mot en 1962)

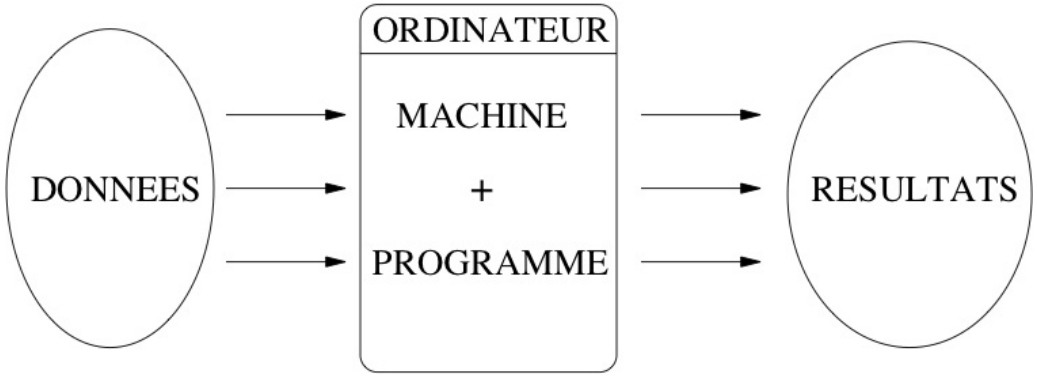
"Informatique = information + automatique"

Pour nous :

"C'est le traitement automatique de l'information à l'aide d'un ordinateur."

Qu'est ce qu'un ordinateur ?

Machine commandée par un programme enregistré qui permet de traiter des informations en exécutant une séquence finie d'instructions.



Différents types d'ordinateur : Les ordinateurs personnels (PC, Macintosh,...), les assistants personnels (PDA), les mainframes, les stations de travail, ...

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Qu'est ce qu'un bit (binary digit) ?

Un bit (BInary digiT) représente un chiffre du système binaire, c'est à dire

$$0 \quad \text{ou} \quad 1.$$

Il est utilisé en informatique pour représenter (ou coder) une information élémentaire qui ne peut prendre que deux valeurs.

Dans l'ordinateur, les informations manipulées ne peuvent prendre que deux valeurs qui dépendent de la nature physique de l'information. Par exemple, une information peut être :

- un signal électrique (les fils) ou magnétique (disque dur), qui, au-delà d'un certain seuil, correspond à la valeur 1 ;
- des aspérités géométriques dans une surface (cdrom) ;
- l'état d'équilibre d'un bistable (mémoire).

Le codage des booléen

$$\text{vrai} \mapsto 1 \quad \text{et} \quad \text{faux} \mapsto 0.$$

À la fin des années 30, Claude Shannon démontra qu'à l'aide de "contacteurs" (interrupteurs) fermés pour vrai et ouverts pour faux il était possible d'effectuer des opérations logiques dans le système binaire en associant le nombre 1 pour vrai et 0 pour faux.

NOT x	$1 - x$
x XOR y	$x + y$
x AND y	$x.y$

Vers un codage universel

En utilisant la même idée que pour les nombres, on peut coder n'importe quelle information. Le principe est le suivant :

Soit x le nombre de symboles différents composant l'information. Soit n tel que $2^{n-1} < x \leq 2^n$. Alors chaque symbole peut être codé sans ambiguïté en utilisant un vecteur de n bits.

La table ASCII (caractères codés sur 8 bits)

Regular ASCII Chart (character codes 0 - 127)

000	(nul)	016 ►	(dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001 ☺	(soh)	017 ◄	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002 ☻	(stx)	018 ↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003 ♥	(etx)	019 ‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004 ♦	(eot)	020 ¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005 ♣	(enq)	021 §	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006 ♠	(ack)	022 ▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007 •	(bel)	023 ↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008 ◘	(bs)	024 ↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009 ○	(tab)	025 ↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	(lf)	026 →	(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011 ♂	(vt)	027 ←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012 ♀	(np)	028 ∟	(fs)	044	,	060	<	076	L	092	\	108	l	124	¦
013	(cr)	029 ↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014 ♫	(so)	030 ▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015 ✲	(si)	031 ▼	(us)	047	/	063	?	079	O	095	_	111	o	127	⌂

Extended ASCII Chart (character codes 128 - 255)

128	Ç	143	Å	158	×	172	¼	186	‖	200	╚	214	Í	228	õ	242	=
129	ü	144	É	159	ƒ	173	¡	187	╗	201	╔	215	Î	229	Õ	243	¾
130	é	145	æ	160	á	174	«	188	╝	202	╩	216	Ï	230	µ	244	¶
131	â	146	Æ	161	í	175	»	189	¢	203	╦	217	┘	231	Þ	245	§
132	ä	147	ô	162	ó	176	▒	190	¥	204	╠	218	┌	232	Þ	246	÷
133	à	148	ö	163	ú	177	▓	191	┐	205	═	219	█	233	Ú	247	¸
134	å	149	ò	164	ñ	178	▓	192	└	206	╬	220	▄	234	Û	248	°
135	ç	150	û	165	Ñ	179	┃	193	┴	207	¤	221	¦	235	Ù	249	¨
136	ê	151	ù	166	ª	180	┤	194	┬	208	ð	222	Ì	236	ý	250	·
137	ë	152	ÿ	167	º	181	Á	195	├	209	Ð	223	▀	237	Ý	251	¹
138	è	153	Ö	168	¿	182	Â	196	─	210	Ê	224	Ó	238	¯	252	³
139	ï	154	Ü	169	®	183	À	197	┼	211	Ë	225	ß	239	´	253	²
140	î	155	ø	170	¬	184	©	198	ã	212	È	226	Ô	240	-	254	■
141	ì	156	£	171	½	185	╣	199	Ã	213	ı	227	ò	241	±	255	
142	Ä	157	Ø														

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Qu'est ce qu'un programme ?

Un programme est une suite d'instructions élémentaires s'enchaînant les unes après les autres pour réaliser un traitement sur les données.

Une instruction est un vecteur de bits qui indique à l'unité centrale une opération à exécuter. L'unité centrale ne comprend généralement que quelques instructions basiques du type :

- "copier le contenu de la case 123 et le placer dans la case 345"
- "ajoute le contenu de la case 145 à celui de la case 132 et placer le résultat dans la case 524"
- "si le contenu de la case 856 vaut 0 alors aller à la case 357"

Remarques :

- chaque unité centrale possède son propre jeu d'instructions.
- la taille d'une instruction est égale à la taille d'un mot de la mémoire.

L'assembleur

Au début de l'informatique :
Écrire un programme = écrire une suite d'instructions (des vecteurs de bits) codant les opérations élémentaires que l'unité centrale devra exécuter pour traiter une information.

De tels programmes sont appelés des programmes exécutables car ils sont directement compris par l'ordinateur. C'est du langage machine.

L'écriture du programme était facilitée par l'utilisation de langages proches du langage machine : les langages d'assemblage.

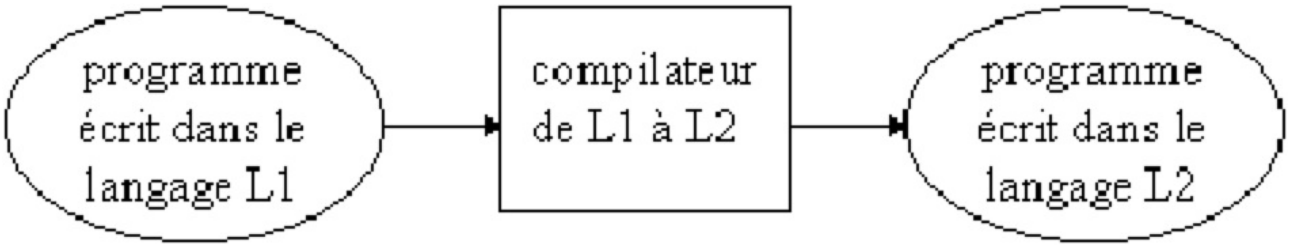
Exemple : ADD R1 R2 R3 =
"ajoute le contenu du registre 1 au registre 2 et stocke le résultat dans le registre 3"

Deux inconvénients :

- il existe (au moins) autant d'assembleurs que de microprocesseurs
- programmer en assembleur reste une tâche très fastidieuse

Les langages de haut niveau

- plus proches du langage humain
- assez puissants pour pouvoir décrire n'importe quels algorithmes
- ne dépendent pas du microprocesseur
- peuvent être automatiquement traduits en langage machine grâce à un compilateur :

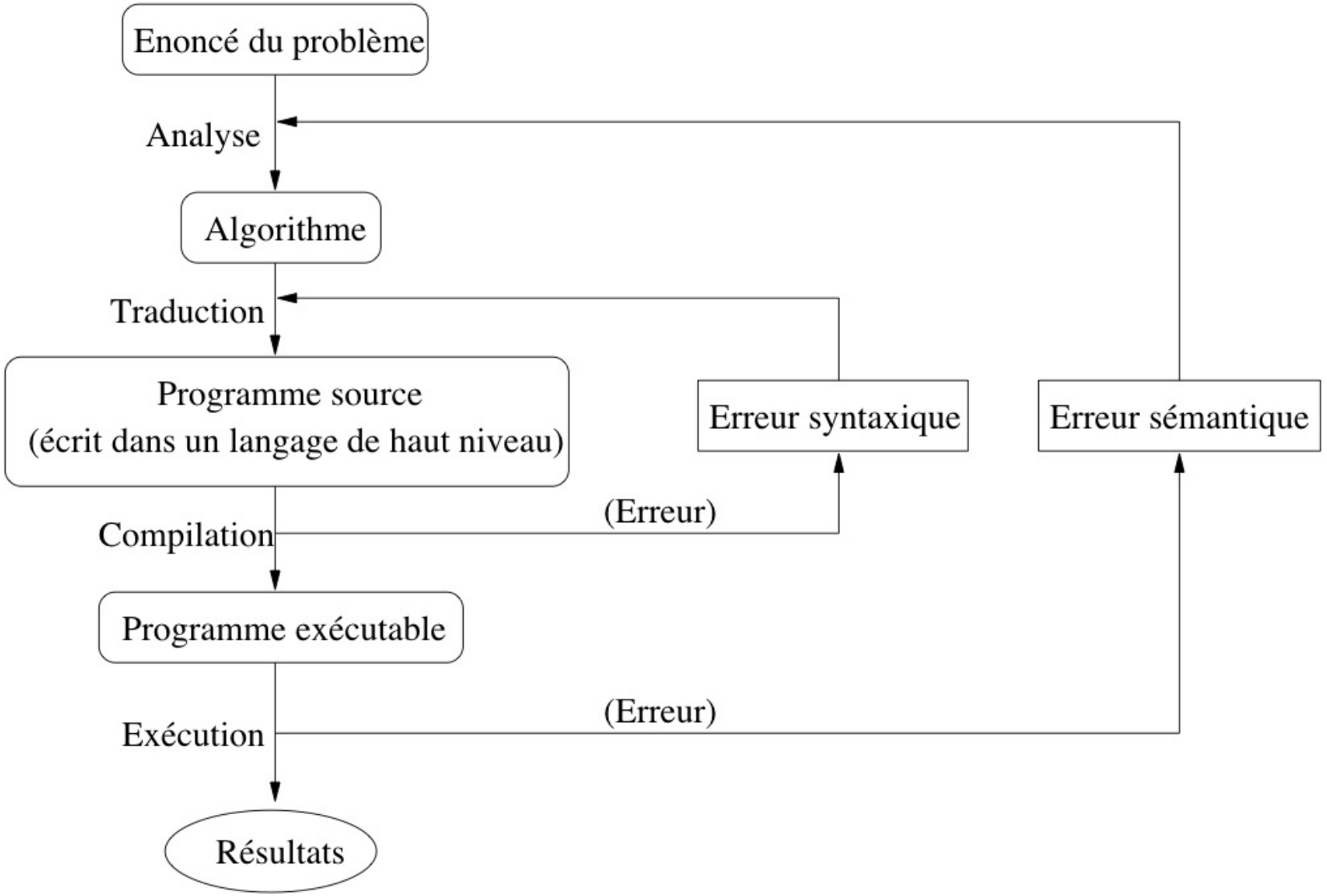


Remarque : Pour chaque langage de haut niveau, il existe (au moins) un compilateur pour chaque microprocesseur.

Exemple : ADD R1 R2 R3 peut s'écrire en (pseudo-)pascal :

```
VAR integer nombre1, nombre2, nombre3 ;
BEGIN
        nombre3 := nombre1 + nombre2 ;
END
```

Comment concevoir un programme ?



Terminologie

- Algorithme : méthode de calcul qui indique la démarche à suivre pour résoudre un ou plusieurs problèmes, en appliquant dans un ordre précis une suite finie de règles (Académie Française).
  Ex : l'algorithme de multiplication de nombres à plusieurs chiffres.
- Programme source : ce qui résulte de la traduction d'un algorithme dans un langage de programmation.
- Programme exécutable : une suite d'instruction machine.
- Erreur syntaxique : une "faute d'orthographe" dans le programme source.
- Erreur sémantique : le programme ne rend pas le résultat attendu. Il y a donc une erreur dans l'algorithme.
- Processus : un programme en cours d'exécution (i.e. le compteur ordinal contient une adresse où se trouve une instruction de ce programme).

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

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

Le système d'exploitation d'un ordinateur est en quelque sorte son gestionnaire central, son chef d'orchestre. En tant que programme, il peut être écrit en Assembleur (sinon, il est écrit dans un autre langage et compilé) et est en permanence actif quand l'ordinateur est sous tension.

Rôles principaux :

- fournir une interface (texte ou graphique) entre l'ordinateur et l'utilisateur pour permettre à ce dernier de donner des ordres à la machine.
- gérer les ressources de l'ordinateur, à savoir ses mémoires, son microprocesseur et ses périphériques.
- être indépendant du matériel, masquer les particularités de la machine en substituant aux ressources physiques des abstractions.

Composition d'un système d'exploitation

Le noyau : il comprend

- des gestionnaires de périphériques (device driver) ;
- un ordonnanceur (scheduler) ;
- un gestionnaire de mémoire ;
- un système de fichier.

Les programmes systèmes : ils permettent à l'utilisateur d'interagir avec le noyau, et donc le matériel. Exemples :

- interpréteurs de commande ou shells ;
- commandes systèmes (décrites tout au long de ce cours) ;
- éditeurs de texte ;
- compilateurs ;
- ...

Qu'est ce qu'un programmme utilisateur ?

Tout logiciel installé sur une machine prend place au dessus du système d'exploitation, avec qui il échange les données nécessaires à son fonctionnement.

Par exemple, quand un logiciel prévoit une fonction impression, la demande d'exécution de cette instruction par l'utilisateur est transmise par le logiciel au système d'exploitation, seul habilité à lancer une telle commande.

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Linux et les autres

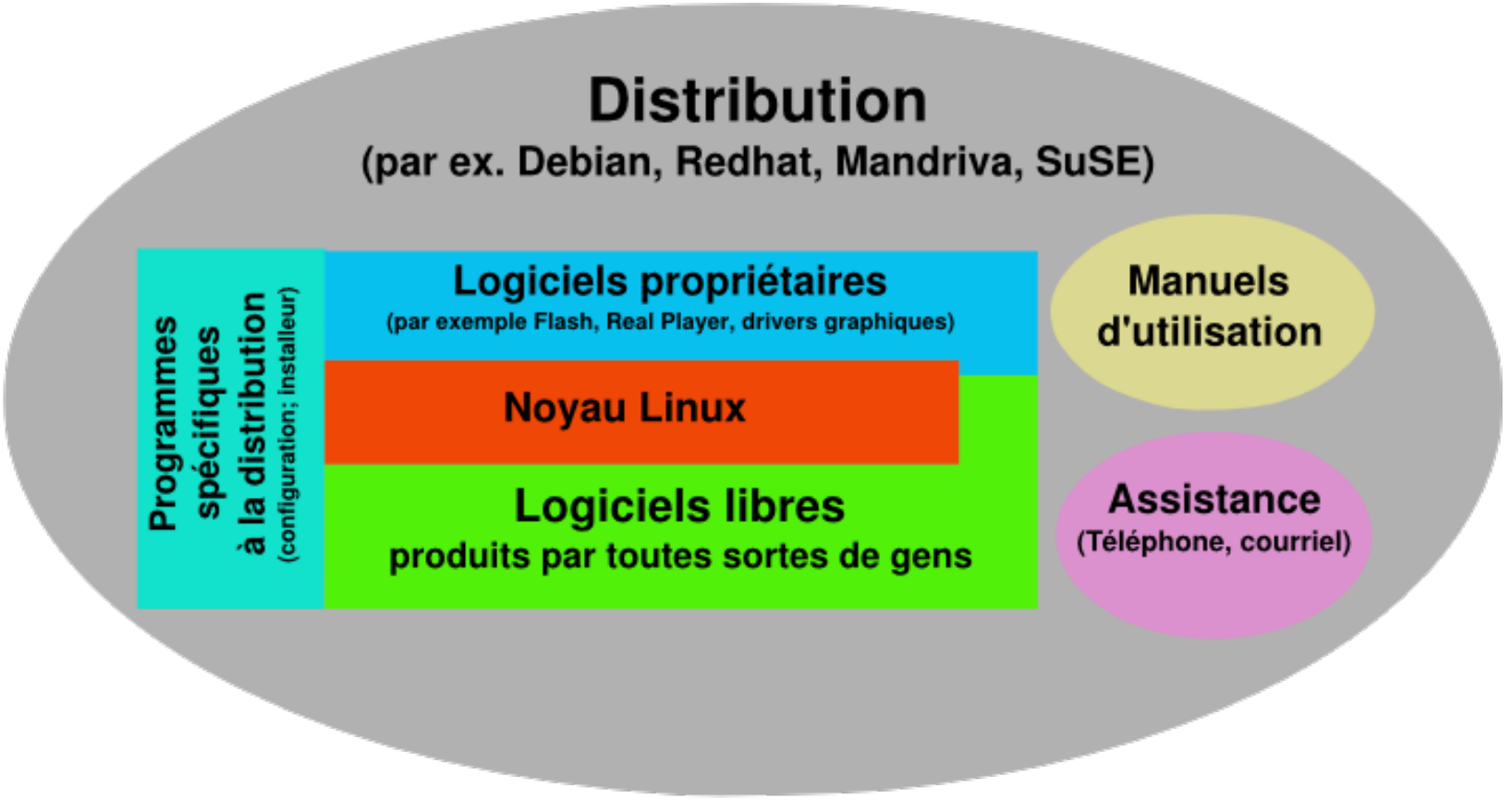
Quelques SE :

- MS DOS : mono-utilisateur, mono-tâche ;
- Windows 2000 : mono-utilisateur, multi-tâches coopératif ;
- Mac OS 9 : multi-utilisateurs, multi-tâches coopératif ;
- Windows XP, vista : multi-utilisateurs, multi-tâches préemptif ;
- Mac OS X ; multi-utilisateurs, multi-tâches préemptif ;
- SE du type Unix (BSD, Linux, ...) : multi-utilisateurs, multi-tâches préemptif.

Historique de Linux

- 1969-1971 : Ken Thompson et Dennis Ritchie (laboratoires de Bell) écrivent les premières versions d'Unix. Il résulte de cette expérience le premier compilateur C.
- 1974-1977 : le code source d'Unix est distribué librement aux universités.
- 1978 : UNIX, 7ème édition. AT&T annonce que les sources du système deviennent payantes.
- 1979 : l'université de Californie à Berkeley crée sa propre variante appelée BSD UNIX.
- 1991 : les clones d'Unix sous licence libre (projet GNU) comme Mimix (Andrew Tannenbaum), Linux (Linus Thorvald) et FreeBSD commencent à émerger.
- mars 1992 : diffusion de la première version Linux comportant quasiment aucun bug.

Les distributions Linux



Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Notion de compte utilisateur

Par défaut, Linux offre un environnement sécurisé aux utilisateurs.

Pour y parvenir, chaque utilisateur Linux est identifié par :

- un identificateur (login), représentant son "nom" dans le SE ;
- un mot de passe confidentiel (password) ;
- un répertoire principal (noté $HOME) dans lequel l'utilisateur gère ses fichiers, sous-repertoires, configurations personnelles, ...

Le triplet (login,password,$HOME) constitue un compte utilisateur.

Seul le propriétaire d'un compte peut modifier son mot de passe, créer ou supprimer des fichiers à l'intérieur de son répertoire principal, ...

Remarque : l'administrateur du SE fait exception à cette règle

Principales règles sur les mots de passe

Un mot de passe ne doit pas être divulgué.

Un mot de passe ne s'oublie pas.

Un mot de passe doit être difficile à trouver :

- évitez qu'il ne se rapporte à vous ;
- évitez les mots du dictionnaire ;
- utilisez au moins 6 caractères ;
- utilisez des majuscules et des miniuscules ;
- utilisez des chiffres et des caractères spéciaux (Ex : &, ?).

La commande pour changer son mot de passe est passwd

Lancer une session de travail

Pour vous connecter au système Linux, et ainsi accéder à votre compte utilisateur, vous devez ouvrir une session de travail.

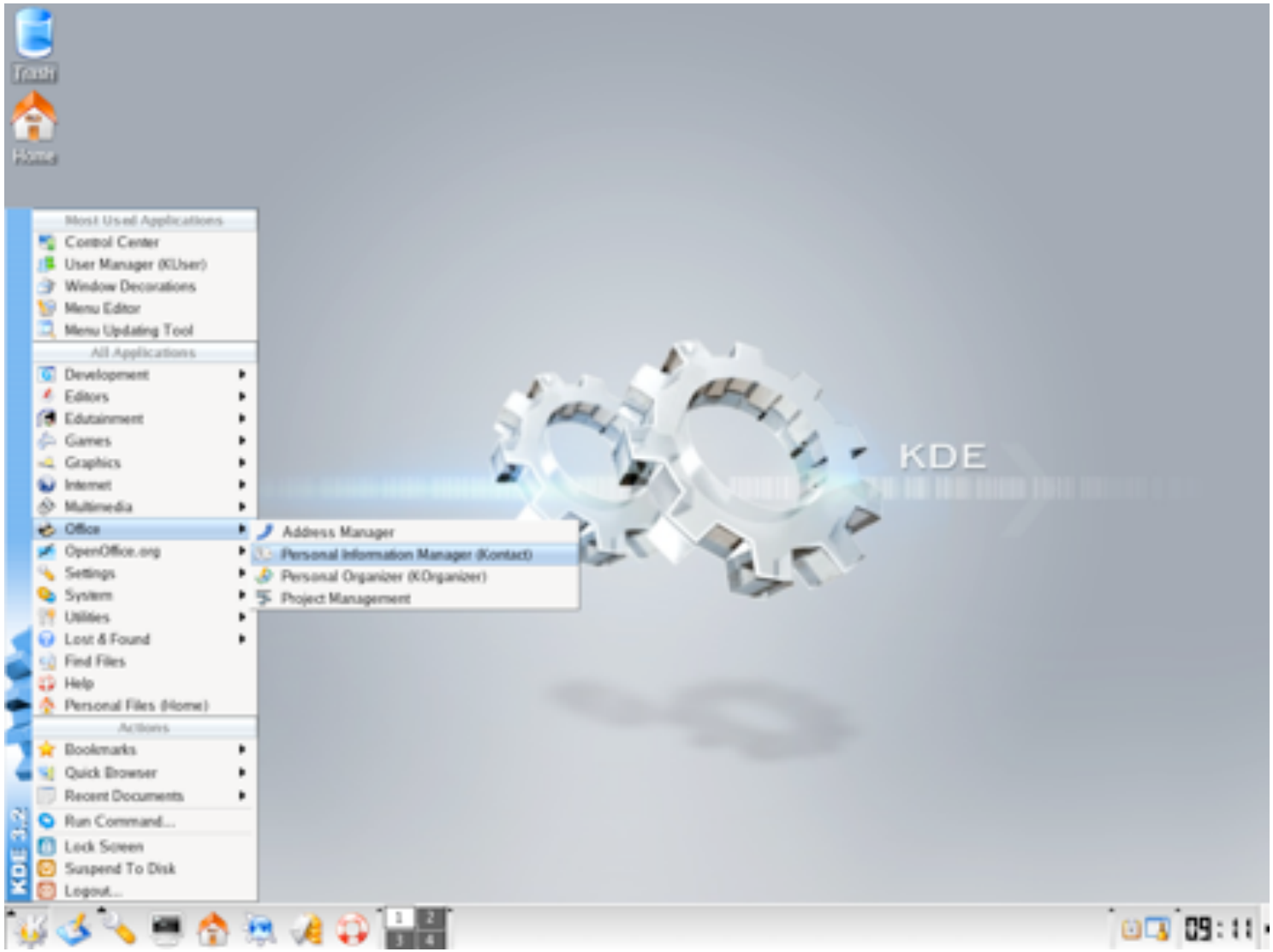
L'ouverture de session consiste simplement à s'identifier, c'est-à-dire donner votre login et votre mot de passe.

Interfaces graphiques

Linux dispose de nombreuses interfaces graphiques comparables à celles de Windows ou Mac OS. Citons par exemple KDE, Gnome, CDE, xfce, ...

☞ Remarque Les interfaces graphiques sont des programmes utilisateurs et ne font donc pas partie du SE Linux à proprement parlé. Elles sont néanmoins couramment fournies avec toutes distributions Linux.

Exemples d'interface garphique : KDE



Exemples d'interface graphique : Gnome

Linux : un environnement de travail complet

De nombreux logiciels devenus indispensables sont présents dans la plupart des distributions Linux. Ils sont accessibles via les différentes interfaces graphiques.

Exemples :

- éditeurs de textes (emacs, xemacs, nedit, ...) ;
- navigateurs (Konqueror, mozilla, firefox, ...) ;
- clients de messagerie (Kmail, mozilla, thunderbird, ...) ;
- applications bureautiques (open office) ;
- lecteurs de divX, de mp3, ...
- jeux vidéos, logiciels d'astronomies, ...

La plupart de ces logiciels sont sous licence libre

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Nature des shells

Un shell (ou interpréteur de commande) est un programme système qui permet à un utilisateur de donner des ordres (appelés commandes) au SE de manière efficace. Comme pour les interfaces graphiques, les shells permettent d'interagir avec le noyau du SE.

Le shell peut aussi être utilisé comme langage de programmation interprété (scripts).

Raison d'être des shells

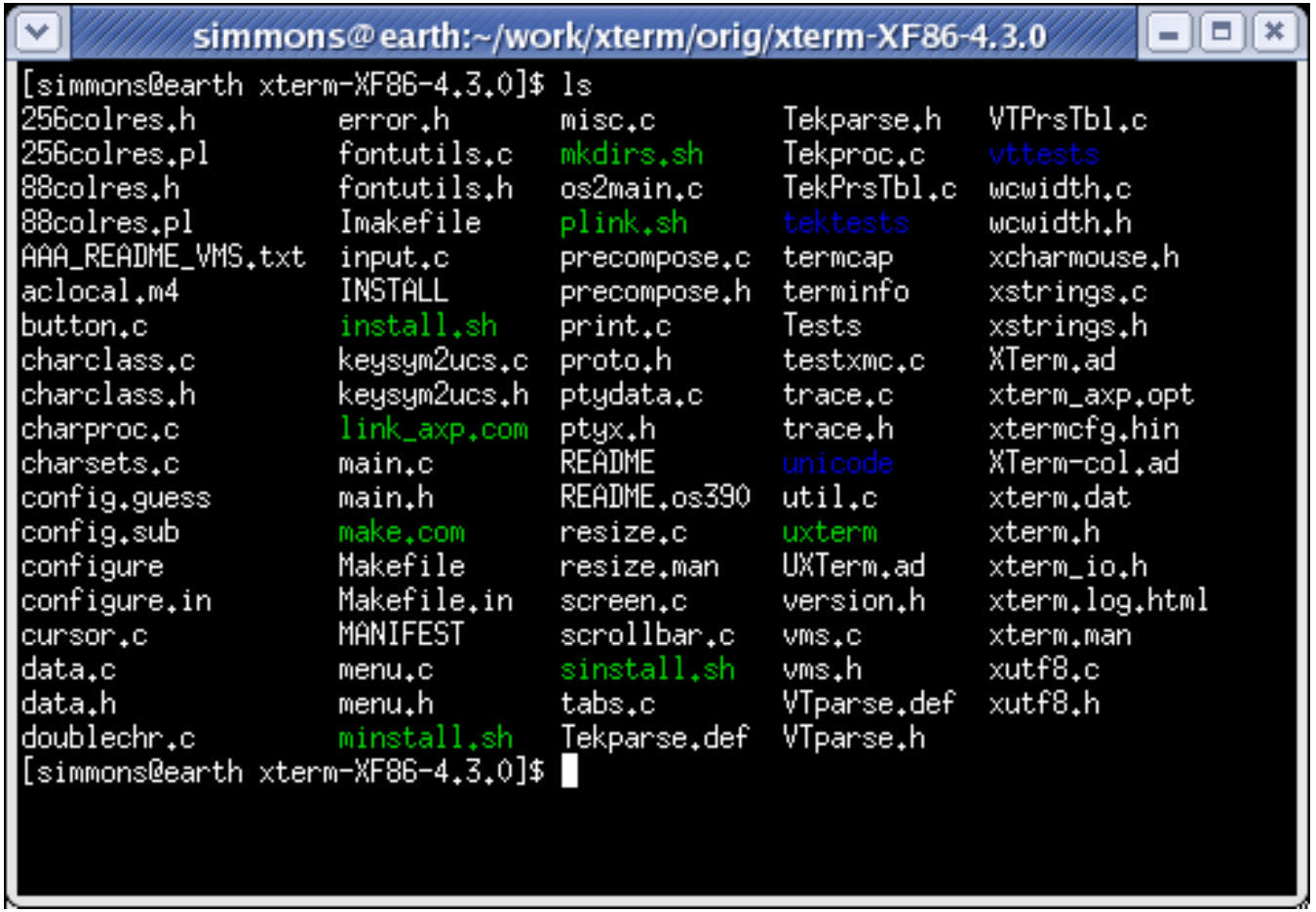
- le travail en "ligne de commande" est souvent beaucoup plus efficace qu'à travers une interface graphique ;
- il existe de nombreux contextes ne disposant pas d'interface graphique ;
- un shell permet une automatisation des tâches répétitives (scripts) ;
- de nombreuses parties du système Linux sont écrites en shell.

Les shells les plus répendus

nom des shells	mot-clé associé	prompt par défaut	commentaires
Bourne shell	sh	$	disponible sur toute plateforme Unix/Linux
C shell	csh	%	shell développé pour BSD
Korn shell	ksh	$	Bourne shell amélioré par AT&T
Bourn again shell	bash	$	version améliorée de sh et csh ; distribué avec Linux

Démarrer un shell

Les shells s'executent dans un terminal texte. Un terminal peut être lancé à partir de l'interface graphique via le programme nommé "xterm".
Dès le lancement du terminal, un shell est démarré.



L'invite de commande (prompt)

Dès le lancement d'un shell, un message se terminant par "$" ou "#" apparaît : c'est l'invite de commande (ou prompt).

Le prompt varie d'un shell à l'autre et d'un système à l'autre. De plus, il peut être personnalisé.

Quelques exemples de prompt :

- sh-2.05b$ (c'est celui utilisé dans ce cours)
- Laptop :~/Document nomLogin$
- [Laptop :~/Document] nomLogin%

La présence du prompt à l'écran signifie que le shell est prêt à traîter une commande. L'utilisateur peut alors taper au clavier cette commmande.

Attention : le prompt ne doit pas être tapé au clavier

Quelques commandes simples (1)

☞ date affiche la date et l'heure.

```
sh-2.05b$ date
Sat Sep  2 10:51:49 CEST 2006
sh-2.05b$
```

☞ whoami affiche le nom de l'utilisateur.

```
sh-2.05b$ whoami
nicolas
sh-2.05b$
```

☞ clear efface le contenu du terminal.

Quelques commandes simples (2)

☞ passwd change le mot de passe.

```
sh-2.05b$ passwd
Changing password for nicolas.
Old password:
New password:
Retype new password:
sh-2.05b$
```

☞ cat affiche à l'écran les lignes tapées au clavier.

```
sh-2.05b$ cat
tout ce que je tape au clavier
tout ce que je tape au clavier
est de nouveau ecrit a l'ecran
est de nouveau ecrit a l'ecran
sh-2.05b$
```

Quelques commandes simples (3)

☞ echo "ch" affiche à l'écran la chaîne de caractères ch.

```
sh-2.05b$ echo "bienvenue dans le monde de Linux"
bienvenue dans le monde de Linux
sh-2.05b$
```

☞ cal n m affiche le calendrier du n-ième mois de l'année m.

```
sh-2.05b$ cal 02 2002
   February 2002
 S  M Tu  W Th  F  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28

sh-2.05b$
```

Forme générale des commandes Linux

Une commande Linux est une suite de mots séparés par des espaces ou des tabulations. Le premier mot décrit le nom de la commande. Les autres mots constituent les arguments (ou paramètres) de la commande.

Les arguments peuvent être :

- des données (fichiers, expressions, ...) utilisées par la commande ;
- des options modifiant le comportement "normal" de la commande.

Basiquement, une commande Linux peut prendre les formes suivantes :

- nomCommande
- nomCommande option1 option2 ... optionN
- nomCommande donnée1 donnée2 ... donnéeM
- nomCommande option1 ... optionN donnée1 ... donnéeM

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Notion de fichiers

Un fichier Linux (ou Unix) est une suite éventuellement vide d'octets contenant une information quelconque. Aucune distinction n'est faite entre fichiers ASCII, binaires ou de toute autre catégorie.

Il existe 6 types de fichiers :

- ordinaire (fichier texte, programme, image, vidéo, son, ...) ;
- répertoire ;
- lien symbolique ;
- bloc (pour les périphériques disque dur, cdrom, ...)
- caractère (pour les périphériques souris, clavier, ...)
- socket (pour les réseaux).

Syntaxe des noms de fichiers

Le nom d'un fichier Linux est composé au maximum de 256 caractères.

Quelques caractères particuliers :

- les caractères "espace", *, $, %, ', ?, { },[ ], <, >,. . ., sont à éviter ;
- le point "." est un caractère comme les autres hormis s'il est le premier caractère du nom. Dans ce cas le fichier est "caché" ;

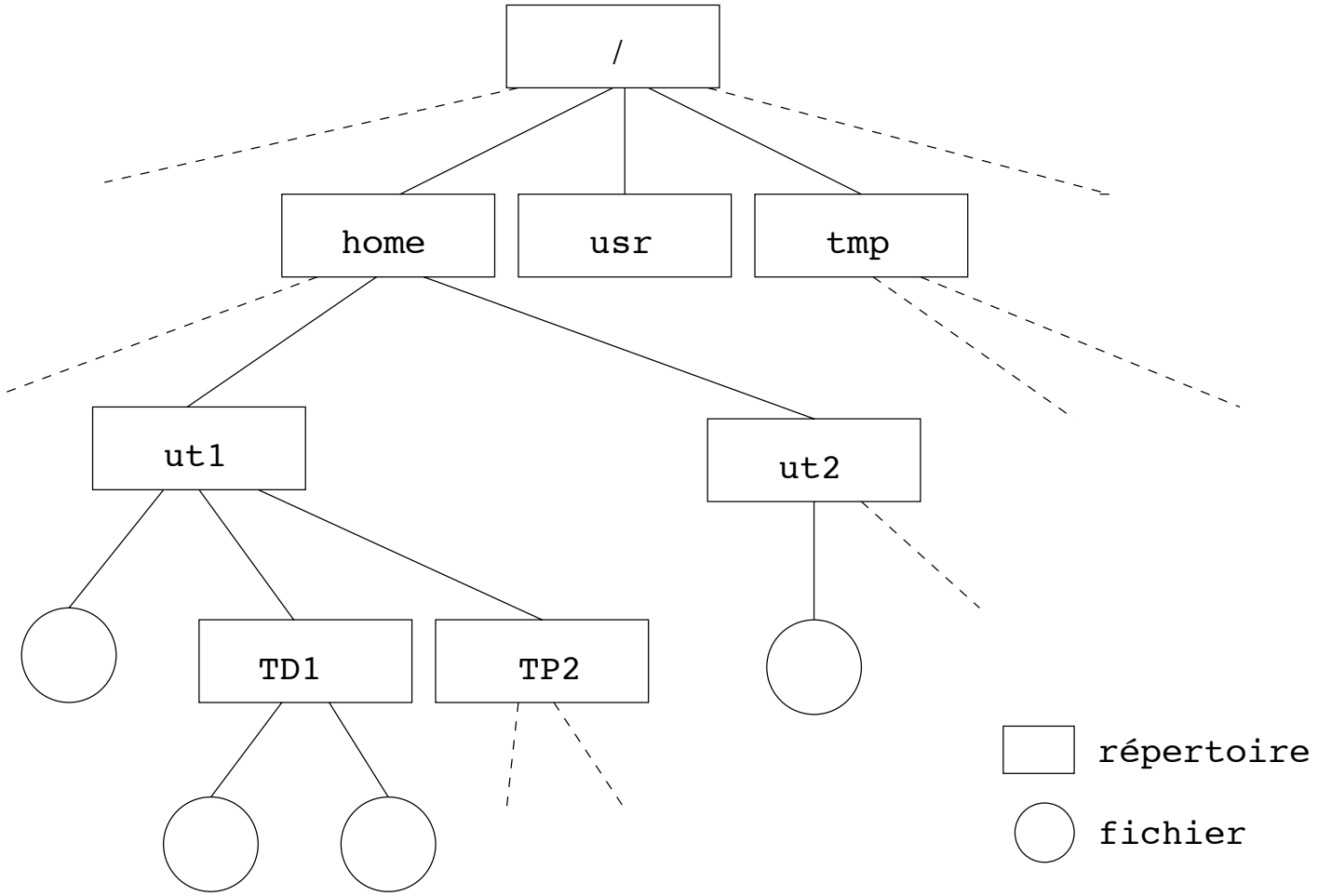
Attention : Linux différencie les minuscules des majuscules !

Exemples : Fic34.Rg, Introduction_a_Linux, mon fichier, test, TeSt, . . .

Que produit la commande touch Introduction au systeme Linux ?
Et la commande touch Introduction\ au\ systeme\ Linux ?

Notion d'arborescence

Dans Linux, tous les fichiers sont "rangés" en mémoire à l'aide d'une structure particulière appelée arborescence.



Racine, père et fils

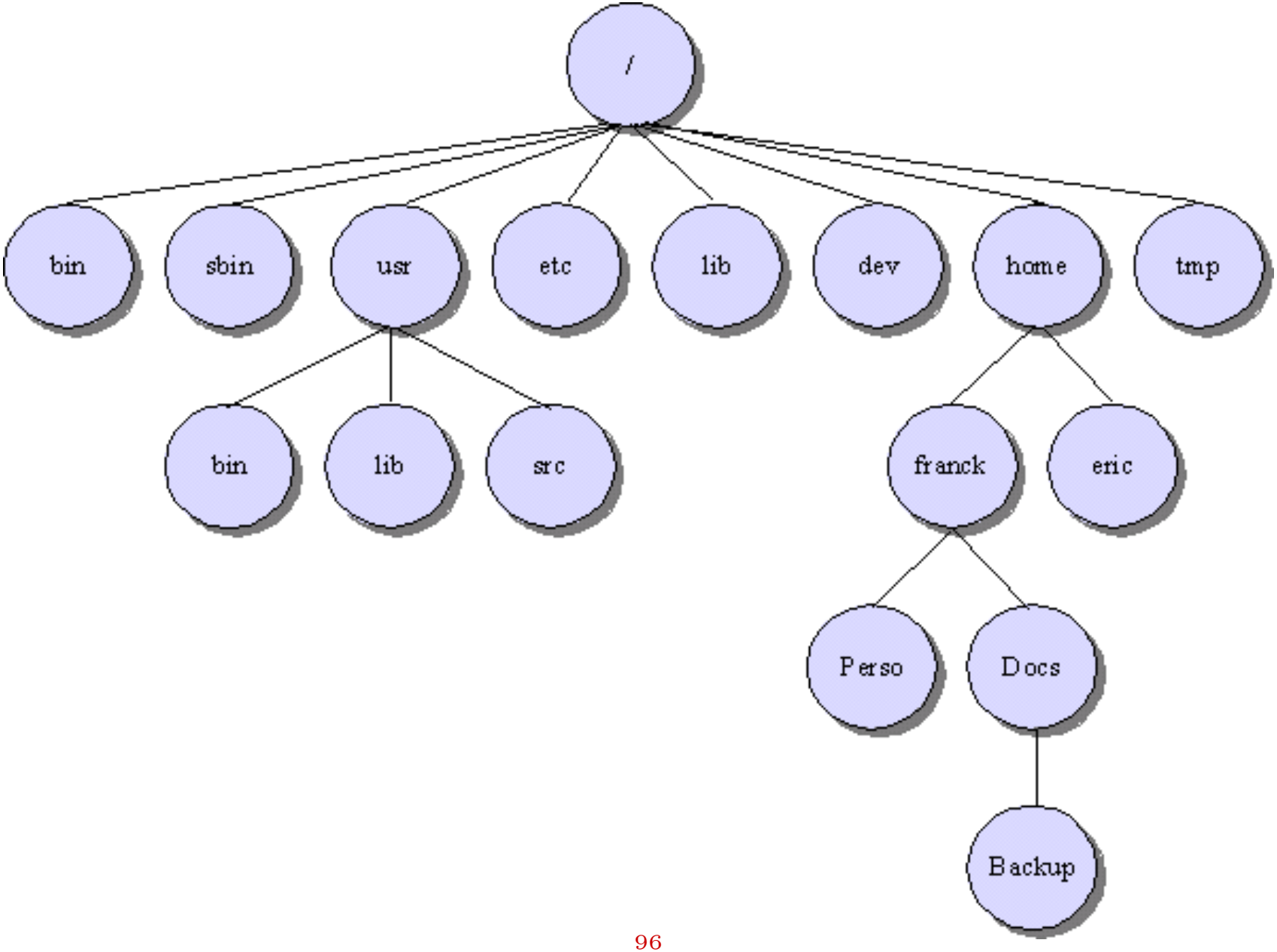
Dans une arborescence, il existe un répertoire qui est rattaché à aucun répertoire de niveau supérieur : c'est la racine (ou root), notée "/".

Chaque répertoire (ou fichier) est rattaché à un répertoire de niveau supérieur : c'est son répertoire père, noté "..".

Chaque répertoire peut être rattaché à zéro, un ou plusieurs répertoires (ou fichiers) de niveau inférieur : ce sont ses fils.

Attention : seuls les répertoires peuvent avoir des fils !

Une partie de l'arborescence Linux



Description des principaux répertoires Linux

/bin	Commandes système
/boot	Fichiers de démarrage
/dev	Fichiers utilisés par les périphériques
/etc	Fichiers de configuration locale
/lib	Bibliothèques partagées entre les fonctionnalités du système
/mnt	Points de montages de périphériques temporaires
/sbin	Commandes spécifiques pour gérer le système
/tmp	Fichiers temporaires
/usr	Données partagées, en lecture seule
/usr/bin	La plupart des commandes de l'utilisateur
/usr/lib	Bibliothèques pour la programmation

Répertoire courant (ou répertoire de travail)

Un utilisateur connecté se situe toujours dans un répertoire de l'arborescence : c'est le répertoire courant.

Dans Linux, le répertoire courant est symbolisé par un point ".".

Lors de l'ouverture d'une session, un utilisateur est toujours placé dans son répertoire principal (noté ~), qui est généralement un répertoire portant son nom de login.

Il est possible de changer de répertoire courant grace à la commande cd

La commande pwd affiche le répertoire courant

Lister le contenu du répertoire courant

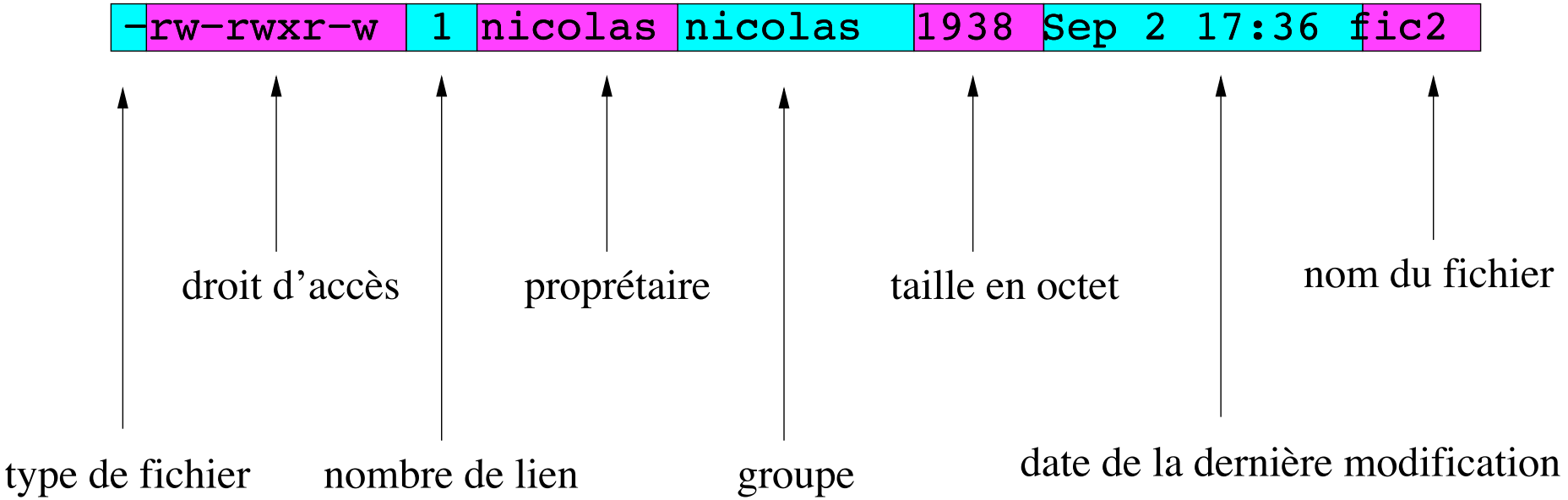
☞ ls Affiche les noms des fichiers contenus dans le répertoire courant.

```
sh-2.05b$ ls
affiche          fic3             image.tiff
fic1             fic4             program.java
fic2             hello.c          test
sh-2.05b$
```

☞ ls -l Affiche la liste des fichiers avec de nombreuses informations utiles.

```
sh-2.05b$ ls -l
total 16
-r-xr-xr-x   1 nicolas  nicolas     0 Sep  2 18:44 affiche
-rw-rwxr-x   1 nicolas  nicolas  1938 Sep  2 17:36 fic2
-rw-r--r--   1 nicolas  nicolas  1938 Sep  2 17:35 fic8
-rw-r--r--   1 nicolas  nicolas     0 Sep  2 18:43 hello.c
-rw-r--r--   1 nicolas  nicolas     0 Sep  2 18:44 image.tiff
-rw-r--r--   1 nicolas  nicolas     0 Sep  2 18:44 program.java
drwxr-xr-x   4 nicolas  nicolas   136 Sep  5 10:18 rep1
drwxr-xr-x   4 nicolas  nicolas   136 Sep  5 10:45 rep2
-rw-r--r--   1 nicolas  nicolas     0 Sep  2 18:42 test
sh-2.05b$
```

Description du format long



Le type de fichier peut être :

- \- : fichier ordinaire ;
- d : un répertoire ;
- l : lien symbolique ;
- b : bloc ;
- c : caractère ;
- s : socket.

Lister les fichiers "cachés"

☞ ls -a Affiche les fichiers d'un répertoire, y compris les fichiers "cachés", i.e. ceux commençant par ".".

```
sh-2.05b$ ls -a
.               affiche         hello.c         rep1
..              fic2            image.tiff      test
.rep2           fic8            program.java
sh-2.05b$ ls -a ./rep1/
.       ..      fic3    fic4
sh-2.05b$
```

Rappel :

- ".." représente le répertoire père ;
- "." représente le répertoire courant en train d'être affiché.

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Chemin d'accès

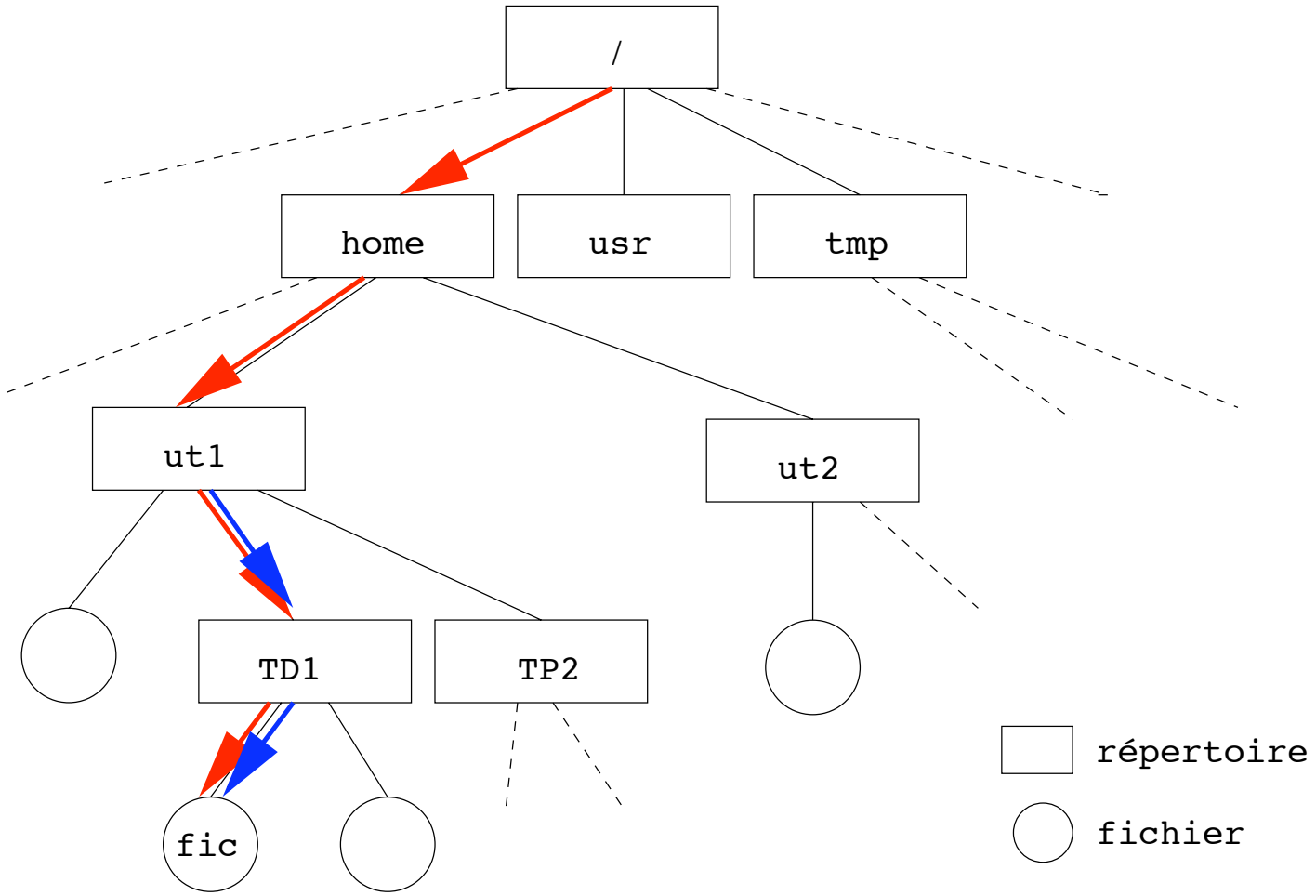
Pour manipuler un fichier, il faut indiquer au SE le chemin à parcourir dans l'arborescence pour y accéder : c'est le chemin d'accès. Il existe deux façons de décrire un chemin :

☞ Chemin absolu Démarrer de la racine "/", puis décrire tous les répertoires rencontrés en chemin en les séparant par le symbole "/", jusqu'à atteindre le fichier à manipuler. Le chemin obtenu commence par "/" : c'est le chemin d'accès absolu.

☞ Chemin relatif Similaire, mais on démarre du répertoire courant ".". Le chemin décrit commence donc par "./" : c'est le chemin d'accès relatif.

☞ Remarque Un chemin relatif peut comporter des "../", ce qui signifie "remonter au répertoire père".

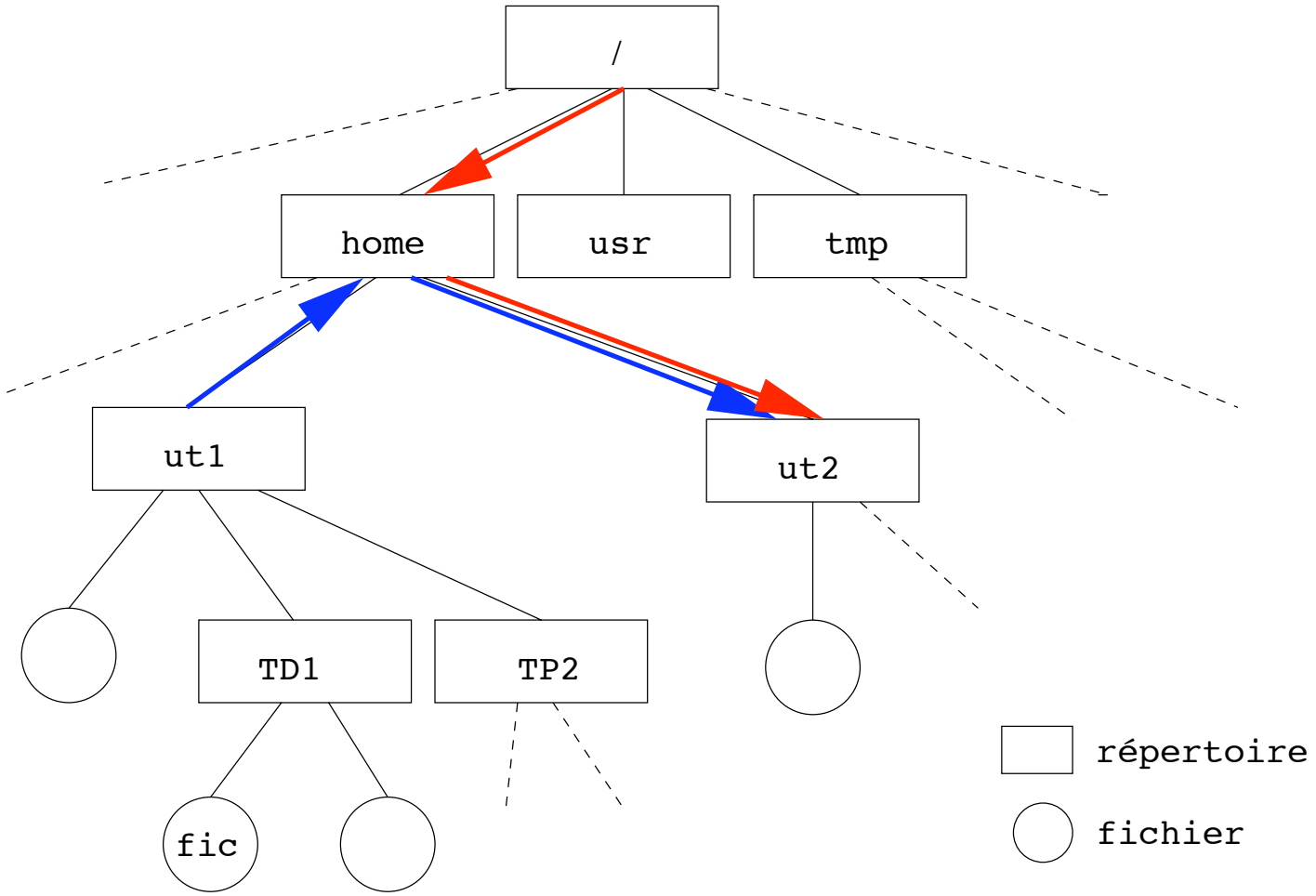
Exemple 1



☞ Répertoire courant : ut1 ; on veut atteindre fic.

- Chemin d'accès absolu : /home/ut1/TD1/fic (flèche rouge)
- Chemin d'accès relatif : ./TD1/fic (flèche bleu)

Exemple 2



☞ Répertoire courant : ut1 ; on veut atteindre usr.

- Chemin d'accès absolu : /home/ut2 (flèche rouge)
- Chemin d'accès relatif : ./../ut2 (flèche bleu)

Se déplacer dans l'arborescence

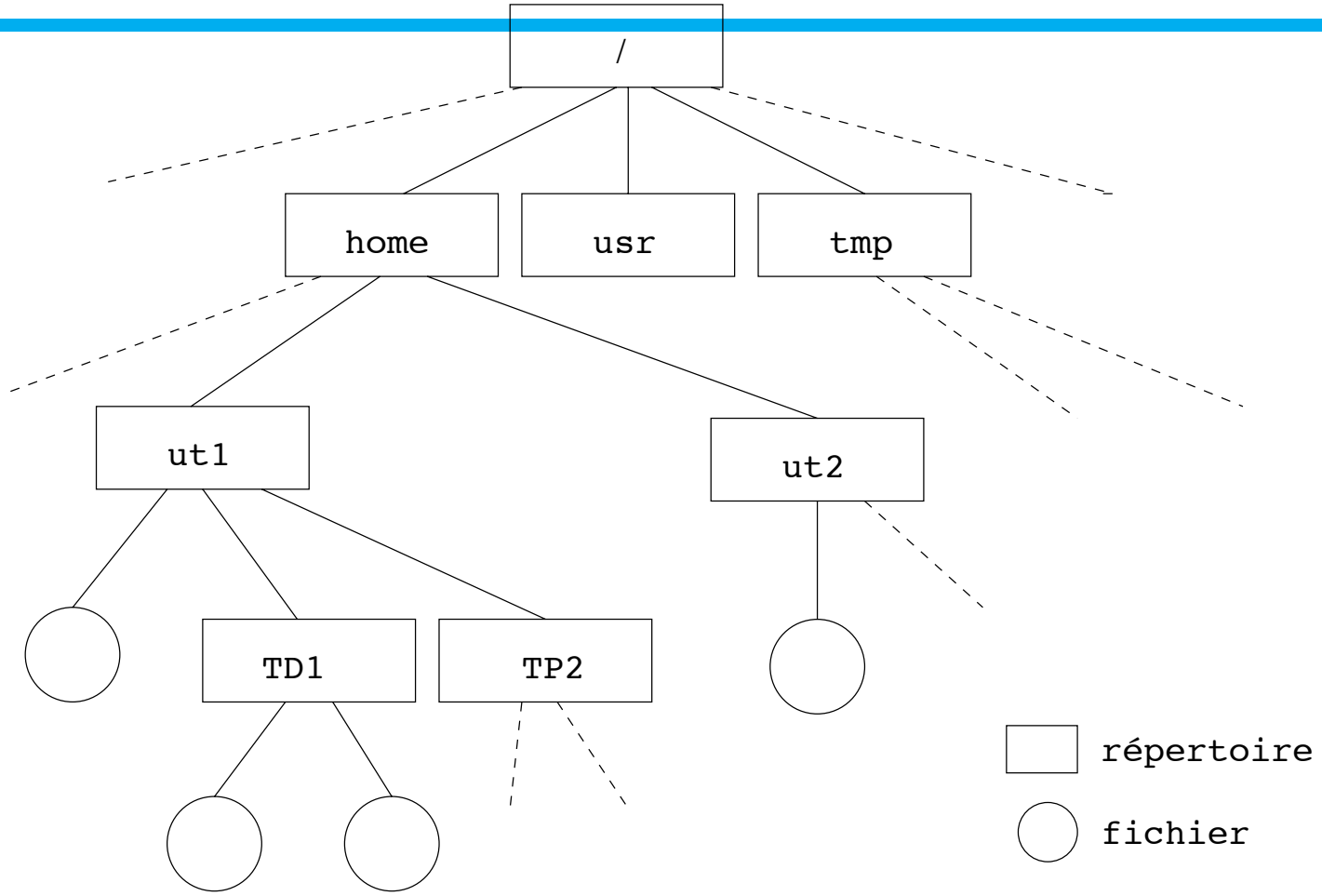
☞ cd chemin_accès_répertoire Modifie le répertoire courant. Le nouveau répertoire courant est celui désigné par chemin_accès_répertoire.

```
sh-2.05b$ pwd
/Users/nicolas/TD1
sh-2.05b$ ls
affiche          fic8             image.tiff       rep1
fic2             hello.c          program.java     test
sh-2.05b$ cd ./rep1/
sh-2.05b$ pwd
/Users/nicolas/TD1/rep1
sh-2.05b$ ls
fic3    fic4
sh-2.05b$
```

Remarque : la commande cd sans argument fait du répertoire principal de l'utilisateur son répertoire courant.

Que signifie cd . ? cd .. ? cd ~ ?

Exercice



Le répertoire courant est TD1.

- Comment atteindre tmp ? (donnez le chemin relatif et absolu)
- Quel fait la commande cd /home/ut2/./../ut1/TD1/../../././/../.. ?
- Quel fait la commande cd /home/ ut2 ?

Lister le contenu d'un répertoire quelconque

☞ ls chemin_accès_répertoire Affiche le contenu du répertoire spécifié sans modification du répertoire courant.

```
sh-2.05b$ pwd
/Users/nicolas/TD1
sh-2.05b$ ls
affiche          fic8             image.tiff       rep1             test
fic2             hello.c          program.java     rep2
sh-2.05b$ ls ./rep2
fic3    fic4
sh-2.05b$ pwd
/Users/nicolas/TD1
sh-2.05b$
```

Cette commande est équivalente aux trois commandes suivantes :
cd chemin_accès_répertoire puis ls puis cd chemin_accès_répertoire_de_départ

Que signifie ls . ? ls .. ? ls ~ ?

Lister une arborescence

☞ ls -R Liste le contenu d'un répertoire, et récursivement de ses sous-répertoires.

```
sh-2.05b$ ls -R
affiche          fic8             image.tiff       rep1             test
fic2             hello.c          program.java     rep2

./rep1:
fic12            fic3             fic4             sous-rep1

./rep1/sous-rep1:
fic20

./rep2:
fic3    fic4
sh-2.05b$
```

Créer, copier, dépacer et supprimer un répertoire

☞ mkdir chemin_accès_répertoire_à_créer Crée un répertoire à l'endroit spécifié.

☞ cp -R chemin_accès_répertoire_source chemin_accès_répertoire_destination/nouveau_nom Copie un répertoire et son contenu vers la destination spécifiée. Le répertoire copié est renomé.

☞ mv chemin_accès_répertoire chemin_accès_répertoire_destination/nouveau_nom Déplace un répertoire et son contenu dans l'arborescence. Le répertoire déplacé sera en plus renommé.

☞ rm -R chemin_accès_répertoire Supprime un répertoire et son contenu de l'arborescence.

Remarque : Pour pouvoir supprimer un répertoire, son répertoire courant ne doit pas être situé dans un des sous-répertoires du répertoire à effacer.

Créer des fichiers ordinaires

☞ touch fichier Créer un fichier vide.

☞ echo "ch" > fichier Met la chaîne de caractères ch dans un fichier. Si le fichier spécifié existe déjà, son contenu est remplacé par ch. Sinon, le fichier est créé.

☞ cat > fichier Suivant le même principe que précédemment, cat copie dans le fichier spécifié tout ce qui est tapé au clavier, jusqu'à ce que la séquence CTRL-D soit tapée.

Afficher et imprimer un fichier

☞ cat fichier_texte Affiche le contenu du fichier texte spécifié

☞ more fichier Affiche le contenu du fichier texte spécifié page par page.

- barre [espace] : page suivante ;
- touche [entrée] : ligne suivante ;
- touche [b] : page précédente ;
- touche [Suppr] : arrêt de la commande.

☞ La commande less permet aussi un affichage page par page d'un fichier texte (voir le man pour plus d'information).

☞ lp -d nom_imprimante fichier_1 ...fichier_N Imprime le(s) fichier(s) sur l'imprimante définie après l'option -d.

Copier, supprimer, déplacer des fichiers

☞ cp chemin_acces_fichier_source chemin_acces_fichier_destination Copie le contenu d'un fichier dans un autre. Le fichier source est donc dupliqué.

> Attention : si le fichier destination existe déjà, son contenu est écrasé

☞ rm chemin_acces_fichier_1 ...chemin_acces_fichier_N Supprime le(s) fichier(s). Leur contenu est perdu.

☞ mv chemin_acces_fichier_source chemin_acces_répertoire_destination/nouveau_nom Déplace et renome un fichier.
Si aucun nouveau nom de fichier n'est spécifié, le fichier source est simplement déplacé.

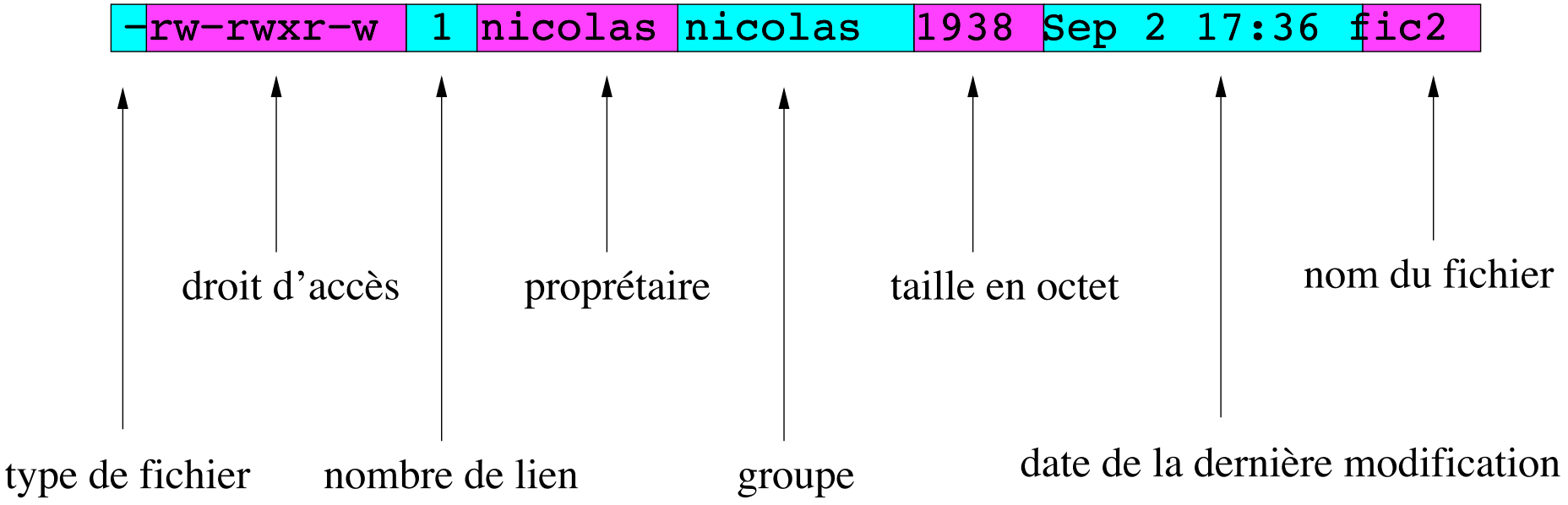
> Attention : si le fichier destination existe déjà, son contenu est écrasé

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Rappel

☞ ls -l Permet d'afficher les droits d'accès d'un fichier.



Les droits d'accès

user			group			others		
r/-	w/-	x/-	r/-	w/-	x/-	r/-	w/-	x/-

Il existe trois droits d'accès associés à chaque fichier :

- droits du propriétaire du fichier ("u" user) ;
- droits des membres du groupe associé au fichier ("g" group) ;
- droits des autres utilisateurs ("o" others).

Il existe trois permissions :

- lecture (Read) : autorisé (a=r) / interdit (a=-) ;
- écriture (Write) : autorisé (b=w) / interdit (b=-) ;
- execution (eXecute) : autorisé (c=x) / interdit (c=-) ;.

Changer les droits d'accès

☞ chmod qui action permission fichier Modifie les droits d'accès.

qui	action	permission
une combinaison de u, g et o	+ (ajoute) ou - (enlève) ou = (définit)	une combinaison de r, w et x

```
sh-2.05b$ ls -l
total 8
-rwxr-xr--   1 nicolas  nicolas  1938 Sep 12 17:33 fic3
-rwxr-xr--   1 nicolas  nicolas     0 Sep 12 17:33 fic4
sh-2.05b$ chmod u-w fic3
sh-2.05b$ chmod go=wx fic4
sh-2.05b$ ls -l
total 8
-r-xr-xr--   1 nicolas  nicolas  1938 Sep 12 17:33 fic3
-rwx-wx-wx   1 nicolas  nicolas     0 Sep 12 17:33 fic4
sh-2.05b$
```

Changer les droits d'accès

☞ chmod nombre fichier Modifie les droits d'accès. Les permissions sont exprimées en base octale.

droits	valeur octale	droits	valeur octale
- - -	0	r - -	4
- - x	1	r - x	5
- w -	2	r w -	6
- w x	3	r w x	7

```
sh-2.05b$ ls -l
total 8
-r-xr-xr--   1 nicolas  nicolas  1938 Sep 12 17:33 fic3
-rwx-wx-wx   1 nicolas  nicolas     0 Sep 12 17:33 fic4
sh-2.05b$ chmod 037 fic3
sh-2.05b$ chmod 642 fic4
sh-2.05b$ ls -l
total 8
-----wxrwx   1 nicolas  nicolas  1938 Sep 12 17:33 fic3
-rw-r---w-   1 nicolas  nicolas     0 Sep 12 17:33 fic4
sh-2.05b$
```

Changer de propriétaire/de groupe

☞ chown nouveau_propriétaire fichier Change de propriétaire.

☞ chgrp nouveau groupe fichier Change le groupe d'utilisateurs.

Remarque : le nouveau propriétaire ou le nouveau groupe doivent être des noms existant respectivement dans les fichiers /etc/passwd et /etc/group.

Seul l'administrateur et le propriétaire peuvent utiliser chown et chgrp.

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Documentation Linux en ligne

Il existe une documentation électronique accessible pendant le fonctionnement du système : c'est l'aide en ligne.

La commande donnant l'aide est man. Elle donne accès aux pages du manuel des commandes Linux qui sont réparties selon des sections comme suit :

- Section 1 : Commandes destinées aux utilisateurs
- Section 2 : Appels systèmes
- Section 3 : Fonctions des bibliothèques
- Section 4 : Fichiers spéciaux (dans /dev/)
- Section 5 : Formats de fichiers
- Section 6 : Jeux
- Section 7 : Divers et conventions
- Section 8 : Outils d'administration système

La commande man

Les numéros de section apparaissent dans de nombreux exemples. Lorsque on lit "printf (3)" dans une page man, il faut se reporter à la commande "printf" de la section 3 du manuel.

Syntaxe de la commande man :

man [numero de section] nom_commande

Les crochets indiquent que le numéro de section n'est pas obligatoire. Il ne faut pas taper les crochets !

L'aide est là pour se rappeler les nombreuses options des commandes et leurs syntaxes particulières.

Exemple : man ls

```
LS(1)                    BSD General Commands Manual                    LS(1)

NAME
     ls -- list directory contents

SYNOPSIS
     ls [-ABCFGHLPRTWZabcdefghiklmnopqrstuwx1] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls
     displays its name as well as any requested, associated information.  For
     each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested, asso-
     ciated information.

     If no operands are given, the contents of the current directory are dis-
     played.  If more than one operand is given, non-directory operands are
     displayed first; directory and non-directory operands are sorted sepa-
     rately and in lexicographical order.
:
```

Exemple : man ls

-g This option is only available for compatibility with POSIX; it is used to display the group name in the long (-l) format output.

-h When used with the -l option, use unit suffixes: Byte, Kilobyte, Megabyte, Gigabyte, Terabyte and Petabyte in order to reduce the number of digits to three or less using base 2 for sizes.

-i For each file, print the file's file serial number (inode number).

-k If the -s option is specified, print the file size allocation in kilobytes, not blocks. This option overrides the environment variable BLOCKSIZE.

-l (The lowercase letter ``ell''.) List in long format. (See below.) If the output is to a terminal, a total sum for all the file sizes is output on a line before the long listing.

-m Stream output format; list files across the page, separated by
:

Exemple : man ls

```
       File information is displayed with one or more <blank>s separating the
       information associated with the -i, -s, and -l options.

   The Long Format
       If the -l option is given, the following information is displayed for
       each file: file mode, number of links, owner name, group name, number of
       bytes in the file, abbreviated month, day-of-month file was last modi-
       fied, hour file last modified, minute file last modified, and the path-
       name.  In addition, for each directory whose contents are displayed, the
       total number of 512-byte blocks used by the files in the directory is
       displayed on a line by itself immediately before the information for the
       files in the directory.  If the file or directory has extended security
       information, the permissions field printed by the -l option is followed
       by a '+' character.

       If the modification time of the file is more than 6 months in the past or
       future, then the year of the last modification is displayed in place of
       the hour and minute fields.

:
```

Exemple : man ls

```
EXAMPLES
     The following is how to do an ls listing sorted by size (and shows why ls
     does not need a separate option for this):

           ls -l | sort -n +4

     Additionally, the -r flag to sort(1) may be used to get the results
     sorted from largest to smallest (a reverse sort).

DIAGNOSTICS
     The ls utility exits 0 on success, and >0 if an error occurs.

ENVIRONMENT
     The following environment variables affect the execution of ls:

     BLOCKSIZE       If the environment variable BLOCKSIZE is set, the block
                     counts (see -s) will be displayed in units of that size
                     block.

:
```

Exemple : man ls

```
     The group field is now automatically included in the long listing for
     files in order to be compatible with the IEEE Std 1003.2 (``POSIX.2'')
     specification.

SEE ALSO
     chflags(1), chmod(1), sort(1), xterm(1), compat(5), termcap(5),
     symlink(7), sticky(8)

STANDARDS
     The ls utility conforms to IEEE Std 1003.1-2001 (``POSIX.1'').

HISTORY
     An ls command appeared in Version 1 AT&T UNIX.

BUGS
     To maintain backward compatibility, the relationships between the many
     options are quite complex.

BSD                                May 19, 2002                                BSD
(END)
```

Plan

6 Les systèmes d'exploitations au travers de Linux
- Définition et rôle des systèmes d'exploitation
- Présentation du système Linux
- Prise de contact
- Introduction aux shells
- Arborescence de fichiers
- Chemins d'accès, manipulation des fichiers et répertoires
- Protection des fichiers
- L'aide en ligne : la commande man
- Quelques commandes utiles
- Les redirections

Gérer l'espace disque utilisé par son compte

Chaque utilisateur de Linux possède un quota, c'est-à-dire une limitation de l'utilisation du disque dur.

Pour connaître son quota, il faut utiliser la commande quota.

Pour connaître la place que prend un répertoire sur le disque dur, il faut utiliser la commande du -s nom_répertoire.

Régulièrement, vous devez vous assurer que la place utilisée par votre répertoire principal (noté ~) sur le disque dur est inférieure à votre quota. Si ce n'est pas le cas, vous devez faire de la place en effaçant plusieurs de vos fichiers.

Gérer ses processus

A chaque fois que vous lancez une nouvelle application, un processus est créé. Lorsque vous arrêtez une application, le processus doit normalement être "tué".

Vous pouvez lister la liste de vos processus en cours grâce à la commande ps. Les principales informations obtenues sont :

- PID : c'est l'identifiant du processus
- TTY : le terminal à partir duquel a été créé le processus
- COM : le nom de la commande

Si vous désirez arrêter un processus de manière forcé, il faut utiliser la commande kill -9 PID où PID est l'identifiant du processus à tuer.

Plan

1 Introduction

2 Codage de l'information dans l'ordinateur

3 Autopsie d'un ordinateur

4 Représentation mathématique d'un ordinateur

5 Les programmes

6 Les systèmes d'exploitations au travers de Linux

7 Internet, web et réseaux

Les réseaux informatiques

Un réseau informatique est un ensemble d'ordinateurs connectés par des lignes physiques permettant l'échange de données entre des entités distantes.

Les réseaux dans la société :

- partager des données (fichiers, applications)
- communiquer entre personnes (msm, mail, blog, ...)
- communiquer entre processus (machines industrielles, calculs distribués)
- garantir l'unicité de l'information (bases de données)
- le multimédia (ex : jeux vidéos, radios, vidéoconférence)
- le commerce électronique
- etc

Nom	Modèle commercial	Exemples
B2C	entreprise à client	achat de livres, CDs
B2B	entreprise à entreprise	commande de pneus par Renault
G2C	gouvernement à client	distribution de formulaires fiscaux
C2C	client à client	vente aux enchères
P2P	égalitaire	partage de fichiers

Comment les machines communiquent ?

Pour qu'un ensemble d'ordinateurs puissent communiquer, il faut :

- des liaisons physiques entre les machines : des fils (signaux électriques, 1Gb), de l'air (signaux électromagnétiques ou radioélectriques, WiFi 54Mb), des supports optiques (100Mb).

- un protocole de communication (FTP, UDP, TCP, IRC,...) : c'est un ensemble de règles à respecter pour émettre et recevoir des données de façon disciplinée.

Différents types de réseaux

Selon leur taille :

- Local Area Network (LAN) : situé dans une petite zone géographique ;
- Metropolitan Area Network (MAN) ou Wide Area Network (WAN) : situé dans une grande zone géographique, interconnection de LAN au moyen de routeur (matériel informatique “intelligent” guidant les informations émises dans le réseau vers les bons destinataires).

Selon leur architecture :

- architecture client/serveur : des machines puissantes au centre du réseau (les serveurs) fournissent des services (programmes, informations, données) à d’autres machines (les clients).
- architecture égalitaire (peer to peer) : pas de serveur dédié, chaque machine du réseau joue à la fois le rôle du client et du serveur.

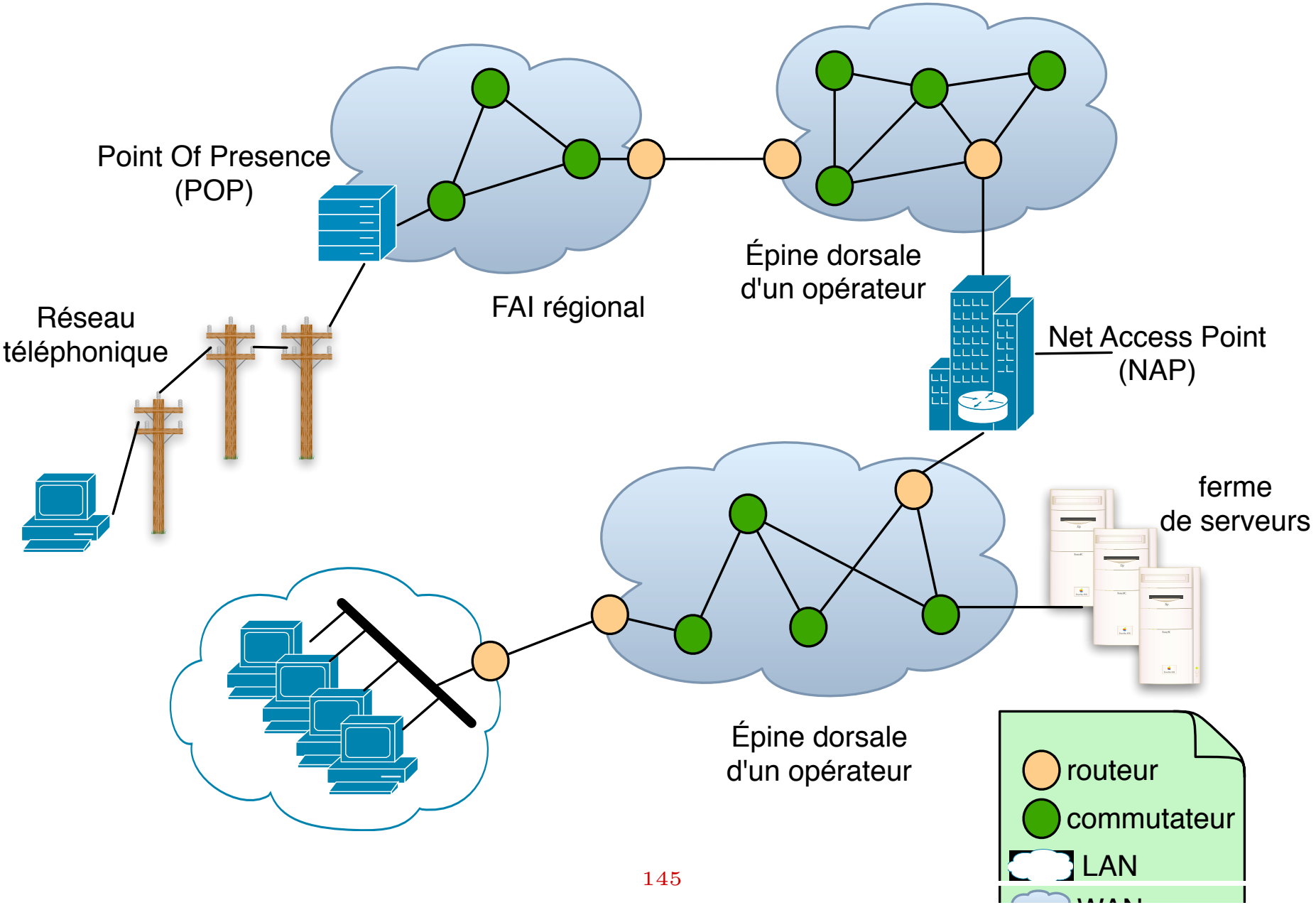
Internet

Internet est un réseau de réseaux hétérogènes (nature, vitesse et topologie différentes).

Internet est basé sur le protocole TCP/IP (1972) :

- chaque ordinateur possède un identifiant sur le réseau appelé adresse IP (ex : 195.124.24.123), qui peut être associée à un nom de domaine (ex : esil.univ-mrs.fr) ;
- sur la machine émettrice : le protocole IP fragmente les informations à envoyer en petits paquets, rajoute l'adresse de l'émetteur et du destinataire à chaque paquet créé, envoie tous les paquets sur le réseau ;
- sur la machine réceptrice : le protocole réceptionne et réordonne tous les paquets envoyés puis reconstitue le message initial ;
- pendant le transport : chaque paquet envoyé est guidé dans le réseau par les routeurs (après examen de l'adresse IP du destinataire contenue dans le paquet) jusqu'au destinataire.

Exemple d'une connection à Internet



Web (World Wide Web)

35 millions de serveurs Web
5 milliars de pages web

C'est un système d'information planétaire constituant une gigantesque toile d'araignée inventé par Tim Berners-Lee. Il est basé sur :

- l'hypertexte, pour "naviguer" ou "surfer" ;
- Internet, pour le transport des informations ;
- deux logiciels :
  - les serveurs web qui stockent et rendent accessibles des documents HTML ;
  - les navigateurs ou browsers qui permettent de demander et de visualiser les documents stockés sur les serveurs ;
- HTTP qui est le protocole de communication entre serveurs et navigateurs ;
- les moteurs de recherche pour rechercher une information ;
- le World Wild Web consortium W3C qui guide le développement du Web.