

```
linux/kernel/panic.c
Copyright (C) 1991, 1992 Linus Torvalds

/*
 * This function is used through-out the kernel (unless you
 * indicate a major problem.
 */
#include <linux/config.h>
#include <linux/module.h>
#include <linux/sched.h>
#include <linux/delay.h>
#include <linux/reboot.h>
#include <linux/notifier.h>
#include <linux/init.h>
#include <linux/signal.h>
#include <linux/syscalls.h>
#include <linux/interrupt.h>
#include <linux/mm.h>

int panic_timeout;
int panic_on_oops;
int tainted;

EXPORT_SYMBOL(panic_timeout);

struct notifier_block *panic_notifier_list;

EXPORT_SYMBOL(panic_notifier_list);

static int __init panic_setup(char *str)
{
    panic_timeout = simple_strtoul(str, NULL, 10);
    return 0;
}

__setup("panic=", panic_setup);

/**
 * panic - halt the system
 * @fmt: The text string to print
 *
 * Display a message, then perform a hard reboot. Functions
 * in the panic_notifier_list are called after the message is
 * printed. This function never returns.
 */
void panic(const char *fmt, ...)
{
    static char buf[1024];
    va_list args;
    if (defined(CONFIG_MAGIC_SYSRQ) && magic_sysrq)
        unsigned long caller = (unsigned long) __builtin_return_address(0);
    if (caller == 0)
        caller = 0;

    buf_start(buf, 0);
    vsprintf(buf, fmt, args);
    va_end(args);
    printk(KERN_EMERG "Kernel panic - not syncing: %s\n", buf);
    buf_end(buf);

    if (CONFIG_MAGIC_SYSRQ)
        magic_sysrq_send(buf);

    notifier_call_chain(panic_notifier_list, 0, buf);

    if (panic_timeout > 0)
        delay(panic_timeout);
}
```

Chapitre 2

Système d'Exploitation UNIX

Système de Fichiers

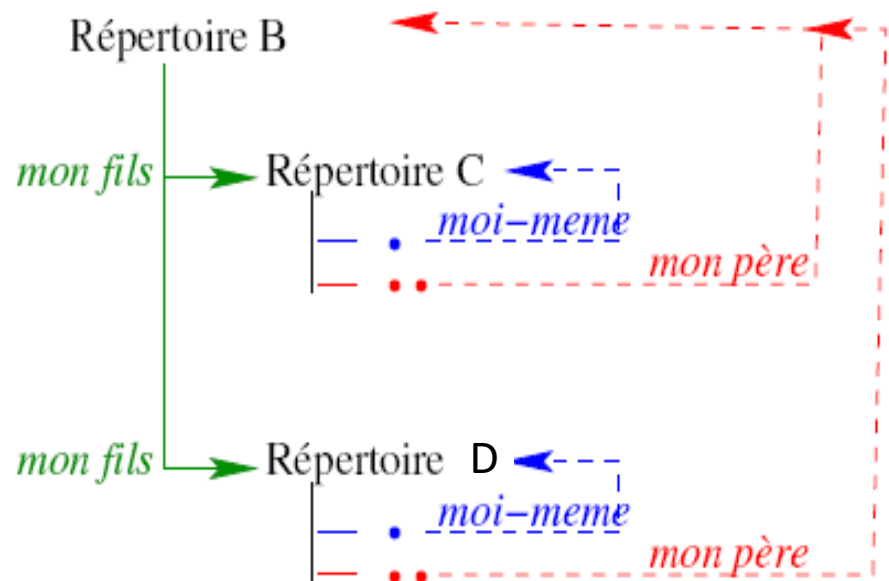
BOUKRI KHALIL

Système de Fichiers: Principe de base

- ❖ Un Système de Fichiers est l'organisation des données sur disque dur définie par le système d'exploitation.
- ❖ les fichiers, sous linux, sont organisés sous forme d'**arborescence** de répertoires et de fichiers.

Chaque répertoire contient un lien vers :

- **lui-même** : désigné par un point `.`
- **son père** : désigné par deux points `..`



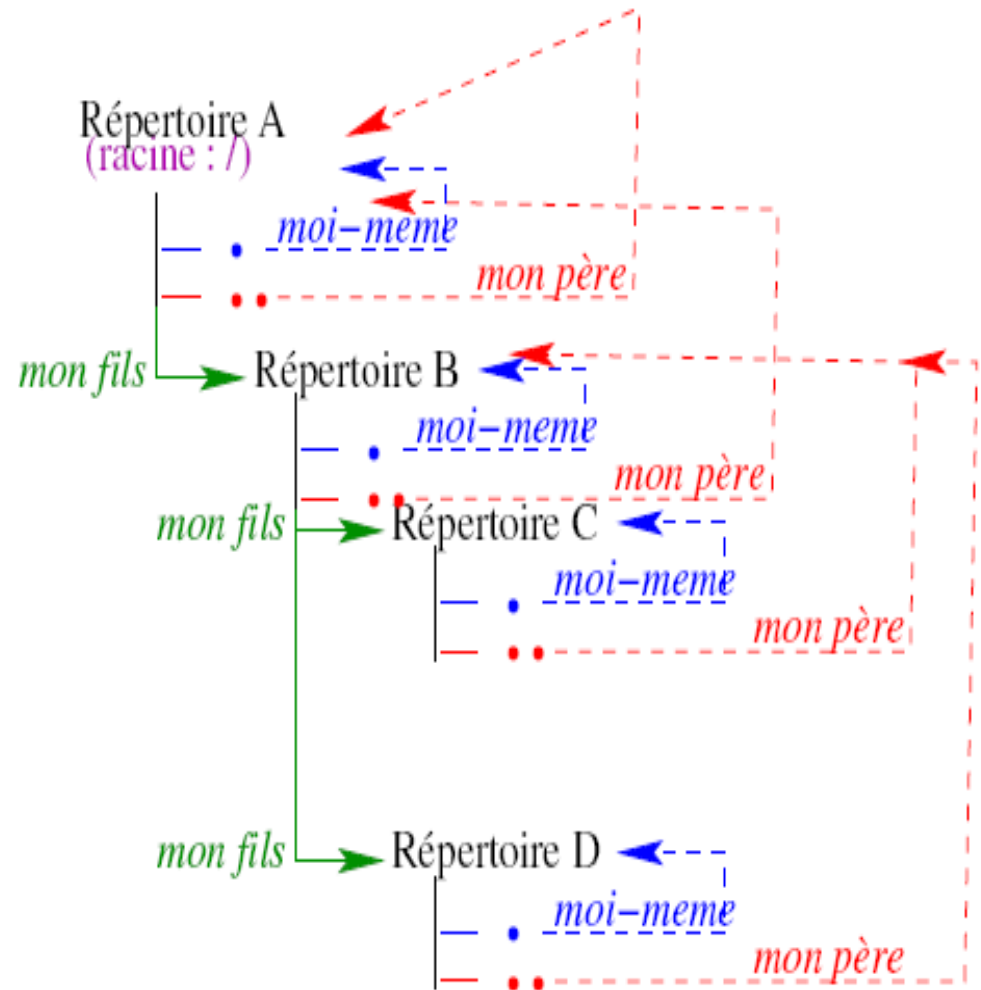
Système de Fichiers: Principe de base

Chaque répertoire contient un lien vers :

lui-même : désigné par un point .

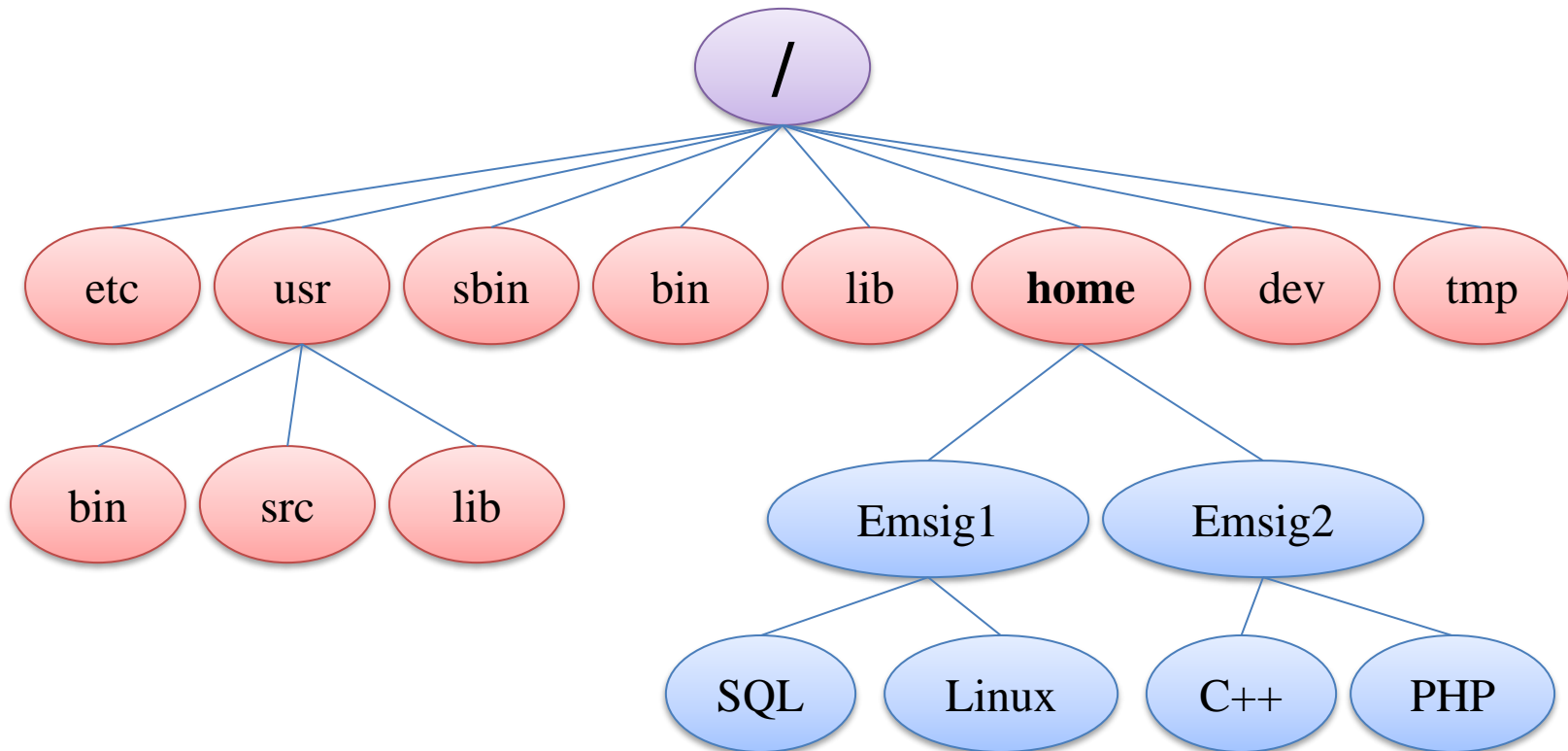
son père : désigné par deux points ..

Le répertoire racine / est un répertoire particulier puisque les deux liens: **lui-même** . et **son père** .. le désigne.



Arborescence Standard

❖ Le système UNIX définit une organisation typique des répertoires, l'utilisateur peut créer d'autres répertoires mais un certain nombre sont nécessaires au fonctionnement du système.



Arborescence Standard

/home	Contient répertoires personnels des utilisateurs.
/etc	Contient les fichiers d'administration
/bin	Contient les exécutables essentiels pour le système, directement utilisable par les utilisateurs. (Les fichiers binaires des commandes)
/dev	Contient les fichiers associés aux périphériques (=device).
/boot	Contient les fichiers permettant à Linux de démarrer.
/lib	Contient des bibliothèques partagées essentielles au système lors du démarrage.
/tmp	Contient les fichiers temporaires.
...	

Le répertoire **/home**:

Tous les utilisateurs ont un répertoire personnel (répertoire d'accueil, en anglais home directory) dans le répertoire **/home** ou dans un de ses répertoires subordonnés.

- La variable d'environnement **\$HOME** contient le chemin d'accès à ce répertoire.
- La commande **cd**, (change directory) sans paramètre, permettra très simplement de revenir à ce répertoire après avoir activé un autre répertoire de travail.

Types de Fichiers

On distingue trois types de fichiers:

- Fichiers ordinaires
- Catalogues (répertoires ou directory)
- Fichiers Spéciaux

Fichiers Ordinaires

Ce sont soit des fichiers contenant du texte, soit des exécutables (ou binaires), soit des fichiers de données.

Par défaut, rien ne permet de différencier les uns des autres, sauf à utiliser quelques options de certaines commandes (*ls -F par exemple*) ou la commande **file**.

```
$ file nom_fic  
nom_fic : 32 Bits ELF Executable Binary (stripped)
```

Types de Fichiers

Catalogues

- Les répertoires permettent d'organiser le disque dur en créant une hiérarchie.
- Un répertoire peut contenir des fichiers normaux, des fichiers spéciaux et d'autres répertoires, de manière récursive.

Fichiers Spéciaux

- Ce sont le bien souvent des fichiers servant d'interface pour les divers périphériques. Ils peuvent s'utiliser, suivant le cas, comme des fichiers normaux.
- Un accès en lecture ou écriture sur ces fichiers est directement dirigé vers le périphérique (en passant par le pilote Unix associé s'il existe).

Nomenclature des Fichiers

- Pour nommer un fichier il faut suivre quelques règles simples. Ces règles sont valables pour tous les types de fichiers.
- Sur les anciens systèmes un nom de fichier ne peut pas dépasser 14 caractères.
- Sur les systèmes récents, on peut aller jusqu'à 255 caractères. Il est possible d'utiliser des extensions de fichiers mais cela ne modifie en rien le comportement du système (un exécutable n'a pas besoin d'une extension particulière).

☛ **Unix fait la distinction entre les minuscules et majuscules.**

Emsi, EMSI, EmSi et emsi sont des noms de fichiers différents.

La plupart des caractères (chiffres, lettres, majuscules, minuscules, certains signes, caractères accentués) sont acceptés, y compris l'espace (très déconseillé). Cependant quelques caractères sont à éviter :

& ; () ~ <espace> \ | ` ? - (en début de nom)

Quelques noms valides :

Fichier1

Paie.txt

123traitement.sh

Paie_juin_2002.xls

8

Quelques noms pouvant poser problème :

Fichier*

Paie(decembre)

Ben&Nuts

Paie juin 2002.xls

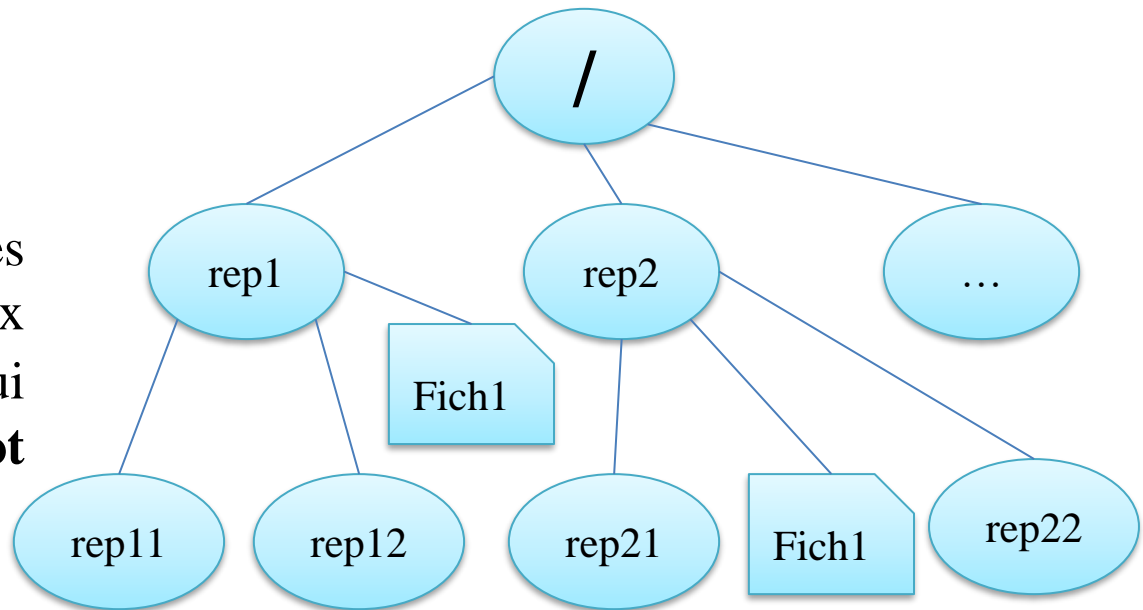
-f

...

Les Chemins

- Les chemins permettent de se déplacer dans le système de fichiers.
- Un nom de fichier est ainsi généralement complété de son chemin d'accès.
- C'est ce qui fait que le fichier « Fich1 » du répertoire « rep1 » est différent du fichier « Fich1 » du répertoire « rep2 ».
- Le FS d'Unix étant hiérarchique, il décrit une arborescence.

❖ Une Arborescence des fichiers sous Unix commence par « / » qui est la **racine** ou **root directory**.

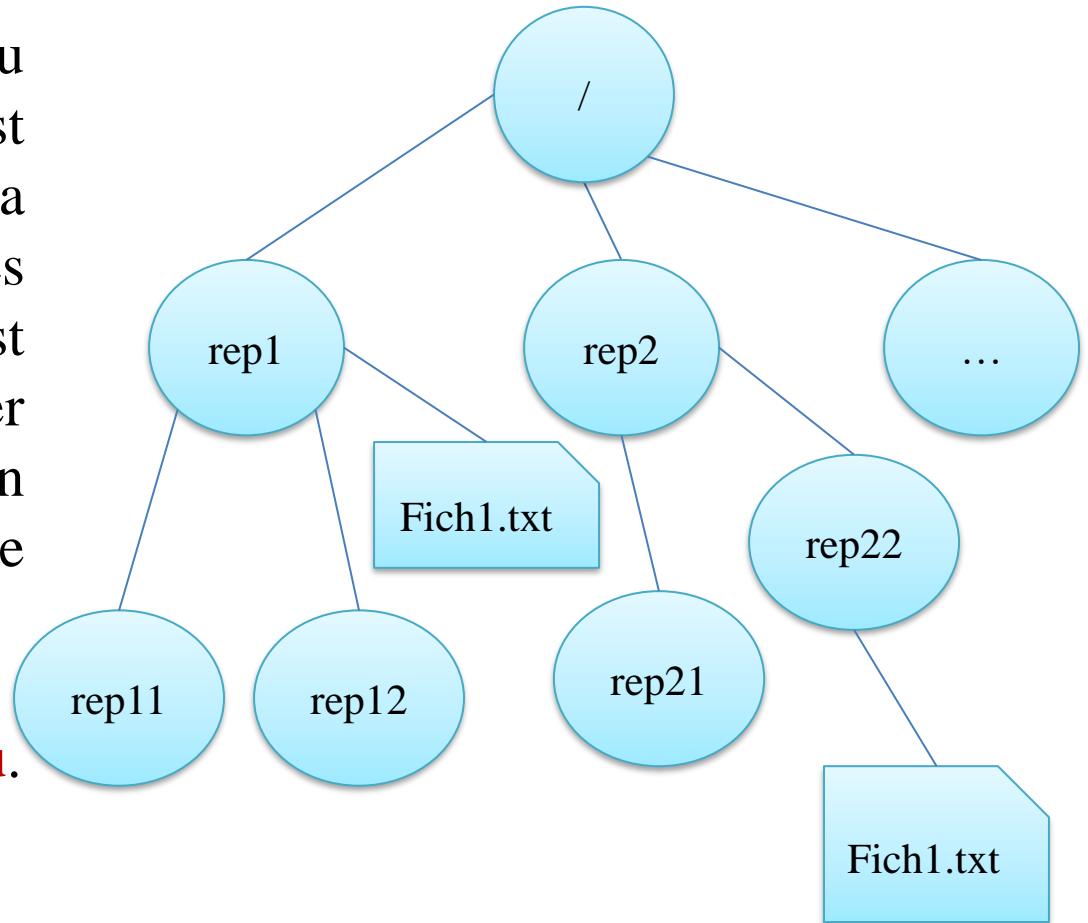


Les Chemins

Le nom de chemin ou **path name** d'un fichier est la concaténation, depuis la racine, de tous les répertoires qu'il est nécessaire de traverser pour y accéder, chacun étant séparé par le caractère « / ».

☛ C'est un **chemin absolu**.

Exemple: */rep2/rep22/Fich1.txt*

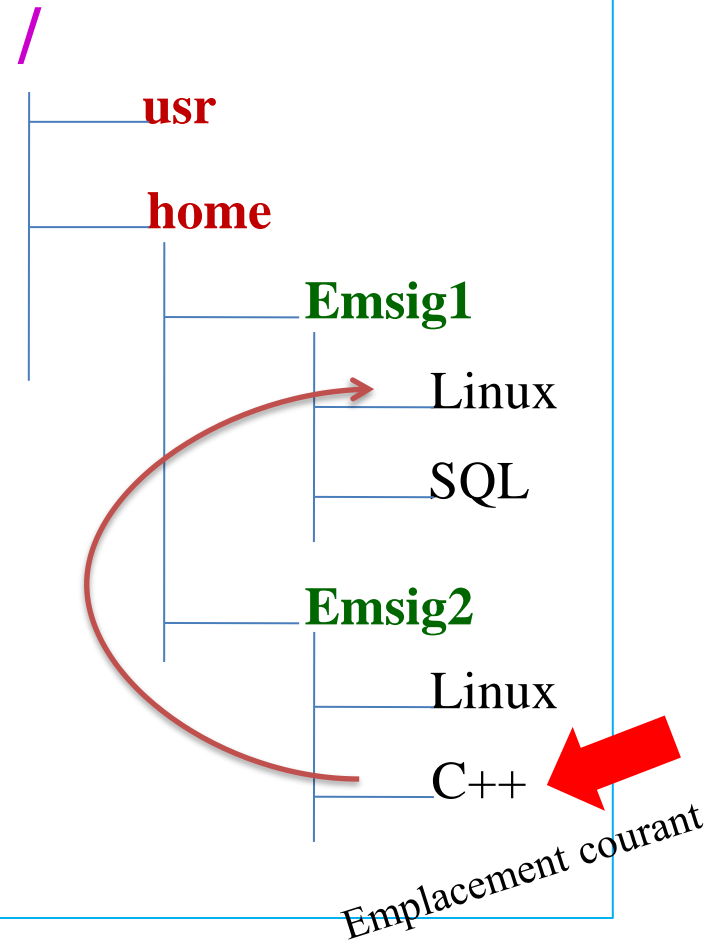


Les Chemins

Chemin relatif:

- Un nom de chemin peut aussi être relatif à sa position courante dans le répertoire.
- Le système (ou le shell) mémorise la position actuelle d'un utilisateur dans le système de fichier, le répertoire actif. On peut donc accéder à un autre répertoire de l'arborescence depuis l'emplacement actuel sans taper le chemin complet.
- Le « .. » permet d'accéder au répertoire de niveau supérieur.
- Le « . » définit le répertoire actif (répertoire courant).

Exemple:

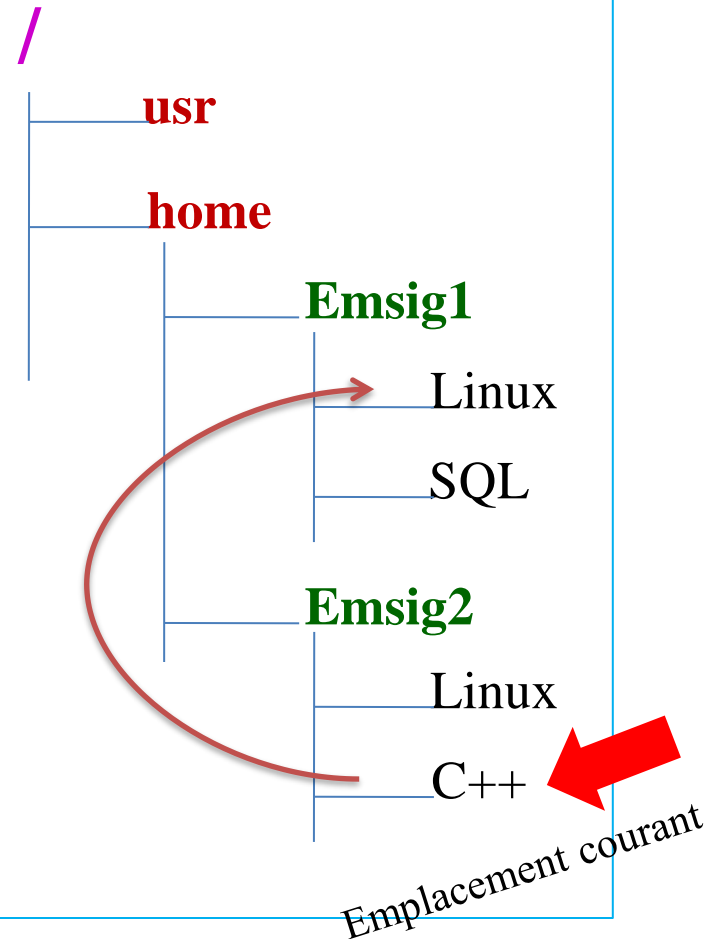


Les Chemins

Chemin relatif:

- Un nom de chemin peut aussi être relatif à sa position courante dans le répertoire.
- Le système (ou le shell) mémorise la position actuelle d'un utilisateur dans le système de fichier, le répertoire actif. On peut donc accéder à un autre répertoire de l'arborescence depuis l'emplacement actuel sans taper le chemin complet.
- Le « .. » permet d'accéder au répertoire de niveau supérieur.
- Le « . » définit le répertoire actif (répertoire courant).

Exemple: *../../Emsig1/Linux*



Le Répertoire personnel:

Lors de la création d'un utilisateur, l'administrateur lui alloue un répertoire utilisateur.

Après une connexion, l'utilisateur arrive directement dans ce répertoire, qui est son répertoire personnel.

C'est dans ce répertoire que l'utilisateur pourra créer ses propres fichiers et répertoires.

☛ Le caractère tilde « ~ » est une référence au répertoire personnel de l'utilisateur.

Commandes de Déplacement

PWD

PWD= Path of Working Directory

☛ Affichage du nom du répertoire de travail en cours.

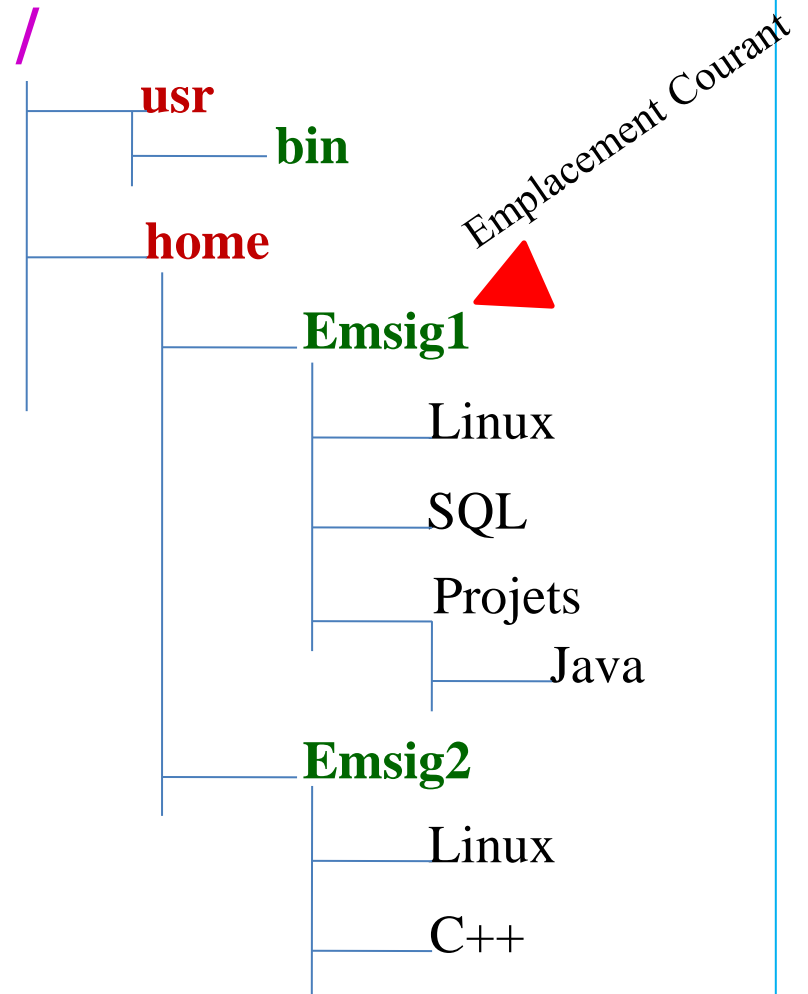
CD

CD= Change Directory

☛ Changement de répertoire courant.

Commandes de Déplacement

```
$ pwd
/home/Emsig1
$ cd projets/Java
$ pwd
/home/Emsig1/projets/Java
$ cd ..
$ pwd
/home/Emsig1/projets
$ cd ../../Emsig2
$ pwd
/home/Emsig2
$ cd /usr/bin
$ pwd
/usr/bin
$ cd ../../../../home/./Emsig1/../../
$ pwd
/home
```



ls

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.

- R :** Afficher récursivement le contenu des sous-répertoires.
- a :** Afficher tous les fichiers des répertoires, y compris les fichiers commençant par un '.'
- i :** Afficher le numéro d'index (i-noeud) de chaque fichier à gauche de son nom.
- l :** En plus du nom, afficher le type du fichier, les permissions d'accès, le nombre de liens physiques, le nom du propriétaire et du groupe, la taille en octets, et l'horodatage

Visualisation de l'arborescence

Information détaillée sur les fichiers : ls -ial

```
emsicentreg11@localhost:/home/emsicentreg11
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost emsicentreg11]# ls -ial
total 84
143972 drwx-----. 15 emsicentreg11 emsicentreg11 4096 21 oct. 11:50 .
131074 drwxr-xr-x.  5 root            root          4096 18 oct. 09:45 ..
144077 -rw-----.  1 emsicentreg11 emsicentreg11   58 18 oct. 09:56 .bash_history
143978 -rw-r--r--.  1 emsicentreg11 emsicentreg11   18 15 mars 2018 .bash_logout
143977 -rw-r--r--.  1 emsicentreg11 emsicentreg11  193 15 mars 2018 .bash_profile
143976 -rw-r--r--.  1 emsicentreg11 emsicentreg11  231 15 mars 2018 .bashrc
144042 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Bureau
144003 drwx-----. 15 emsicentreg11 emsicentreg11 4096 18 oct. 09:42 .cache
143979 drwx-----. 14 emsicentreg11 emsicentreg11 4096 18 oct. 09:42 .config
272739 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Documents
143984 -rw-----.  1 emsicentreg11 emsicentreg11   16 18 oct. 09:39 .esd_auth
144005 -rw-----.  1 emsicentreg11 emsicentreg11  620 21 oct. 11:44 .ICEauthority
272741 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Images
144009 drwx-----.  3 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 .local
272737 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Modèles
143973 drwxr-xr-x.  4 emsicentreg11 emsicentreg11 4096 25 avril 07:36 .mozilla
272740 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Musique
272745 drwxrw----.  3 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 .pki
272738 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Public
272736 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Téléchargements
272742 drwxr-xr-x.  2 emsicentreg11 emsicentreg11 4096 18 oct. 09:40 Vidéos
[root@localhost emsicentreg11]#
```

Visualisation de l'arborescence

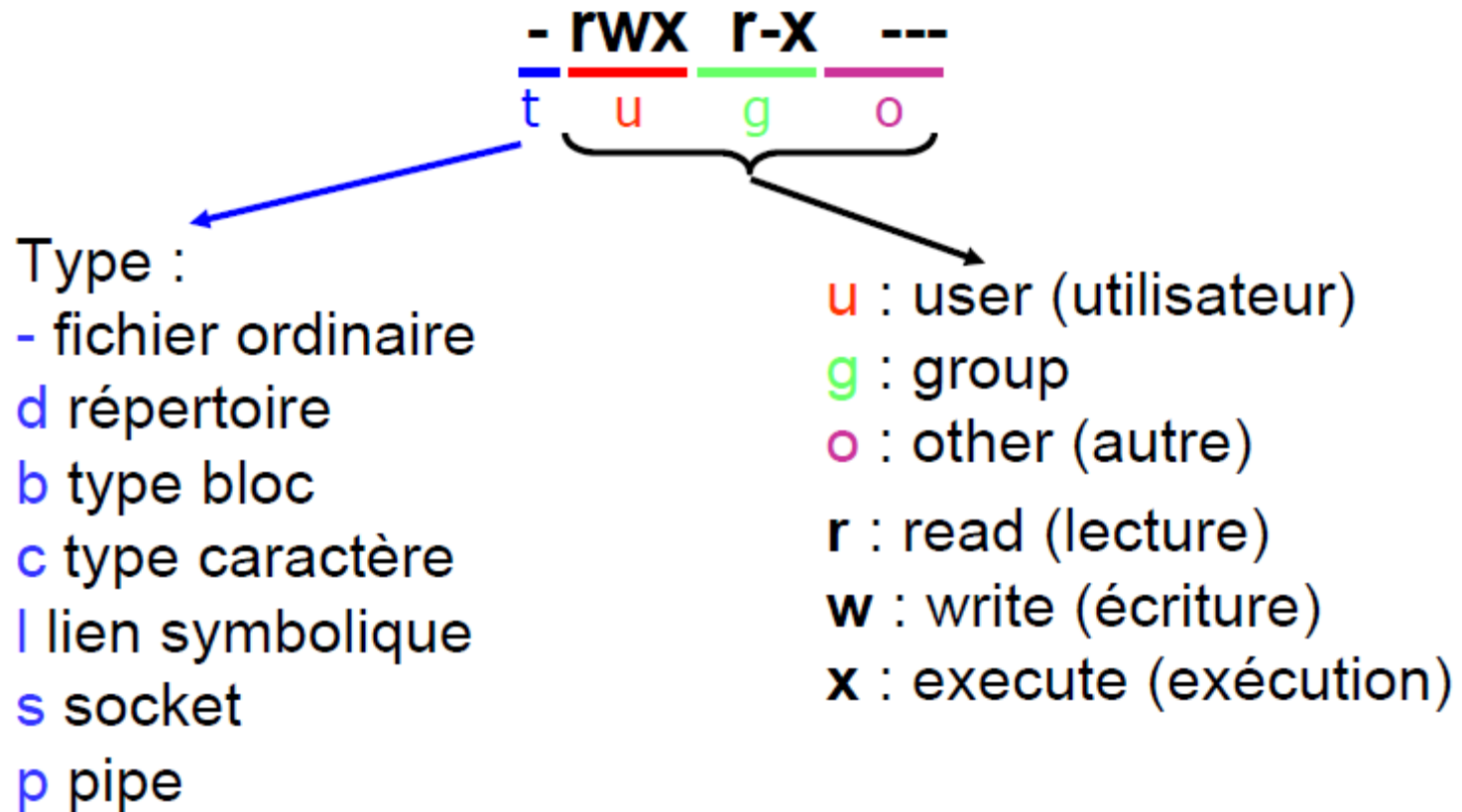
Information détaillée sur les fichiers : ls -ail

2769	drwxr-xr-x	3	Emsiirg1	Emsiir	4087	2018-10-21 21:13	Linux
2896	-rwxr-xr-x	22	Emsiirg1	Emsiir	238	2018-10-20 08:13	Fich1
71452	drwxr-xr-x	4	Emsiirg1	Emsiir	6700	2018-10-20 08:13	SQL

Numéro d'inode	Type du Fichier	Les Droits	Nombre de liens physique sur le fichier	Propriétaire	Groupe du propriétaire	La taille du fichier	Date de dernière modification	Le nom du fichier
-------------------	--------------------	------------	---	--------------	---------------------------	-------------------------	-------------------------------------	----------------------

Visualisation de l'arborescence

Types et modes d'accès au fichier



Manipulation des noms de fichiers : basename, dirname

basename

☛ La commande *basename* élimine les répertoires en tête du chemin d'accès du fichier.

Exemple:

```
$ basename /usr/local/bin/lynx  
lynx
```

dirname

☛ *dirname* ne conserve que les répertoires en tête du chemin d'accès du fichier.

Exemple:

```
$ dirname /usr/local/bin/lynx  
/usr/local/bin
```

Type des fichiers : file

file

- ☛ Tente de deviner le type du fichier (répertoire, exécutable, binaire ...)

```
emsicentreg11@localhost:~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
[emsicentreg11@localhost ~]$ ls  
Bureau      Images      Musique     Téléchargements  test2  
Documents  Modèles     Public      test1             Vidéos  
[emsicentreg11@localhost ~]$ file Musique  
Musique: directory  
[emsicentreg11@localhost ~]$ file test2  
test2: empty  
[emsicentreg11@localhost ~]$
```

Exercices

