Linux File System

Sistemi Operativi
2018-2019

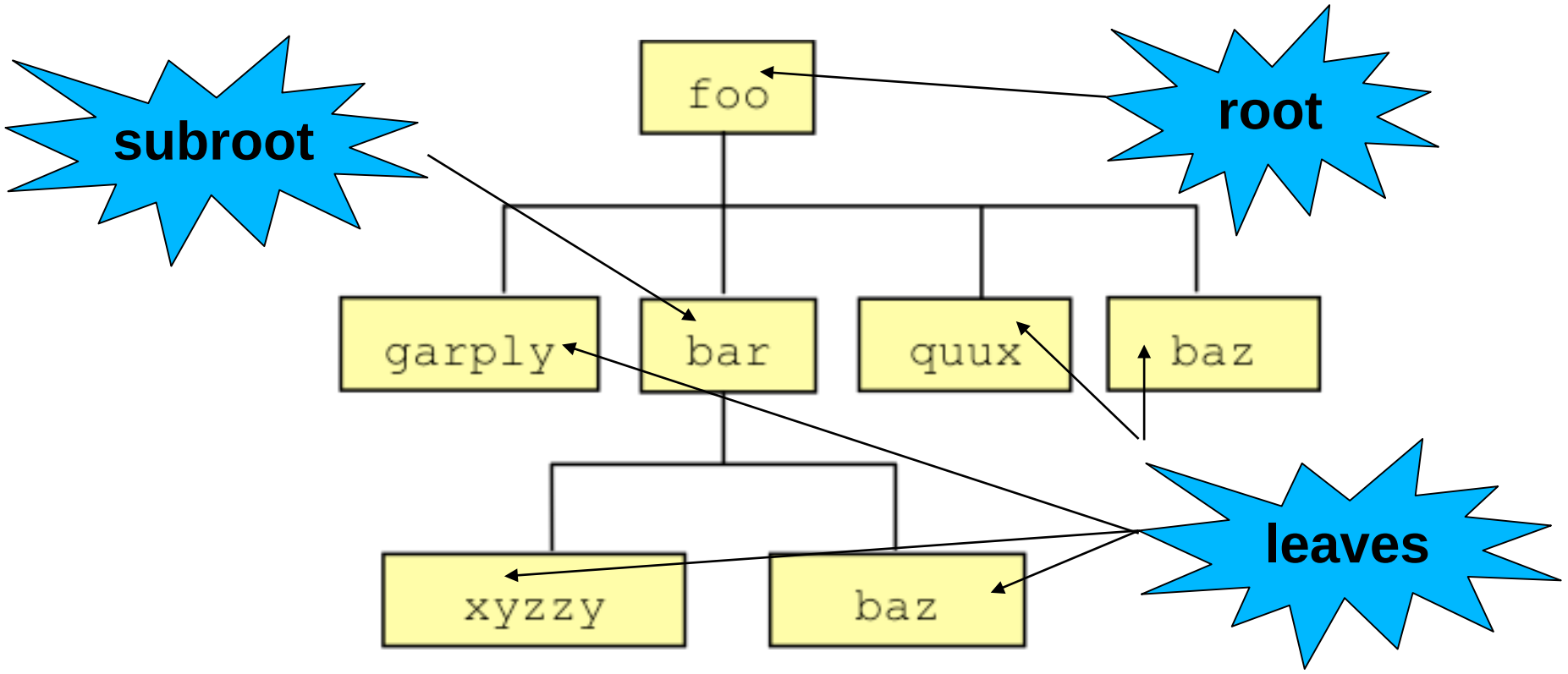
Marco Tessarotto

File system / tree

- Il file system è un modo per organizzare i dati in file e directory al fine di facilitarne l'accesso.
- Il file system è gerarchico ed è organizzato ad albero.
- Gli alberi si compongono di tre componenti: radice (root), sottoradici (subroots) e le foglie (leaves).
- Root ha il «rango» più elevato.
- Subroots sono root di alberi più piccoli.

tree

- Root: no parent (genitore)
- Subroot: has parent & has children (figli)
- Leaves: has parent, no children



File system

- Root, subroots: folders o directories (cartelle)
- Una directory contiene altre directories e/o files
- File: non ha children
- (ma anche una directory vuota….)
- PATH (percorso) per un dato file o directory: descrive i parents fino alla root
- Es: /foo/bar/baz

Linux file system

Tutto inizia con la «root directory»: **/** (forward slash)

Windows usa il simbolo «****» (backslash)

La struttura del file system di Linux ha sempre la stessa struttura:

/ root del file system; tutti i file e le directories «ricadono» sotto la radice

/usr : Unix System Resources

/sbin : system binaries (system admin programs, per il «superuser»

Linux file system

/opt : optional software; third party software

/etc : system configuration files (i.e. password file, global system conf, network…)

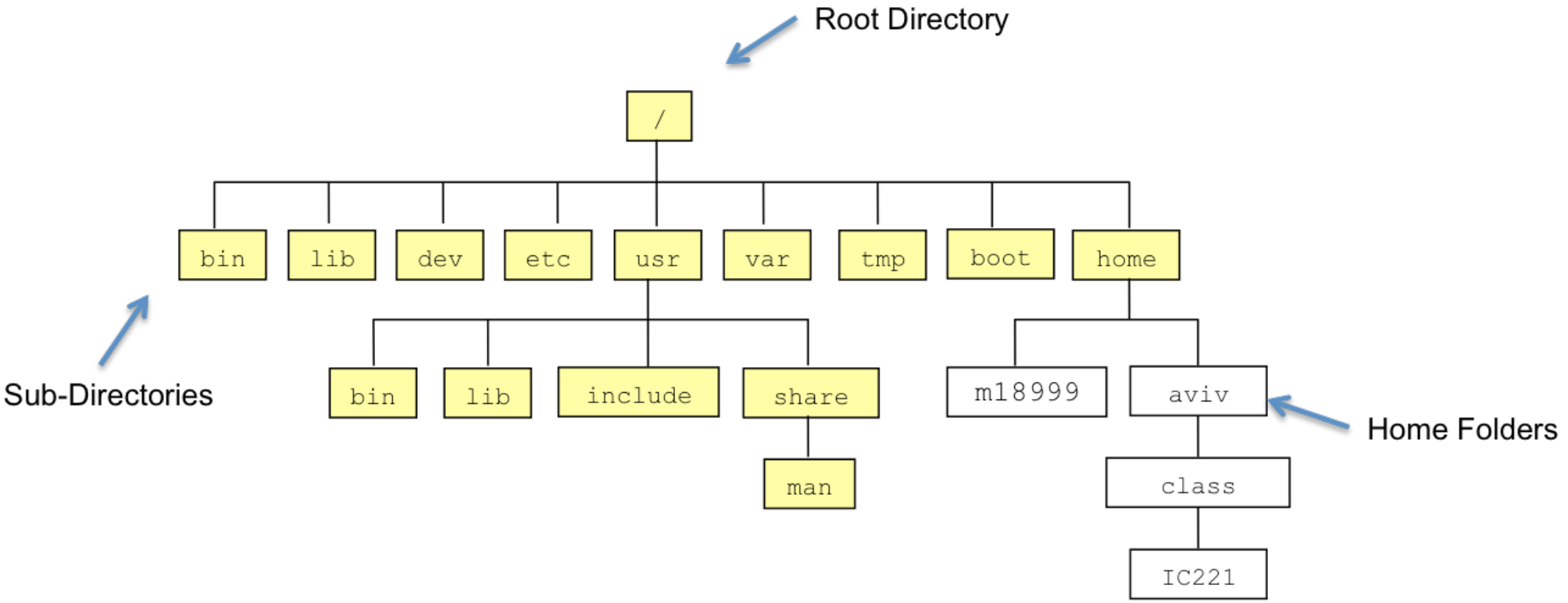
/home : directories degli utenti

/tmp : file temporanei

/boot : core operating system

/usr/lib : librerie binarie (precompilate) per l'uso da parte delle applicazioni

/usr/bin, /usr/local/bin: programmi, utilities…


Root Directory
/
bin
lib
dev
etc
usr
var
tmp
boot
home
Sub-Directories
bin
lib
include
share
man
m18999
aviv
Home Folders
class
IC221

Linux file system

Ogni utente ha una sua directory sotto /home

/home/utente/

~ (tilda) : scorciatoia per la propria home directory

«dot» . : current directory

«dot-dot» .. : parent directory

/home/utente/../../etc/

Da root, vado in home, poi in utente, poi al parent di utente (che è home), poi al parent di home (che è root), poi vado in etc

Linux file system

Nel file system di Linux, sono «montati» anche altri file system….

Da terminale, provate questo comando:

cat /proc/cpuinfo

File

- file: “archivio”, contenitore di informazioni in formato digitale, conservato su “file system”
- Il file può essere aperto, chiuso, scritto, letto
- Il file ha delle proprietà: nome, estensione, flag
- In Linux (Unix) “tutto è file”, cioè può essere aperto. Chiuso, scritto, letto
- Uno dei compiti del OS è rendere trasparente alla applicazione il reale “immagazzinamento” del file su dischi e/o altri tipi di dispositivi

Nome e posizione

- Ciascun file è identificato da un nome, associato ad un percorso (path) che ne individua la posizione (directory o cartella) nel file system
- Il file può avere una “estensione” (che suggerisce il tipo di contenuto)
- Il contenuto di un file è un array unidimensionale di bytes
- Il file ha una dimensione espressa in bytes

esempio

marco@asusi7:~$ pwd

/home/marco

marco@asusi7:~$ ls -l eclipse

lrwxrwxrwx 1 marco marco 20 Oct 13 2017 eclipse -> java/eclipse/eclipse

marco@asusi7:~$ readlink -f java/eclipse/eclipse

/home/marco/java/eclipse/eclipse

marco@asusi7:~$ ls -l /home/marco/java/eclipse/eclipse

-rwxr-xr-x 1 marco marco 73064 Sep 6 15:10 /home/marco/java/eclipse/eclipse

Operazioni base su files

- Creare un nuovo file
- Cambiare i permessi di accesso e gli attributi di un file
- Aprire un file e rendere i contenuti del file disponibile al programma
- Leggere dati da un file
- Scrivere dati su un file
- Chiudere un file, terminando l’associazione tra il file stesso ed il programma che lo ha aperto
- Il sistema operativo fornisce un livello di astrazione, ovvero le interazioni con un file dallo user space avvengono semplicemente attraverso il percorso del file.

Operazioni base su files

- I file possono essere spostati, copiati, cancellati, ridotti in dimensioni, aumentati…
- In Linux, I programmi in user space non operano direttamente (a basso livello) sui file. È il kernel a gestire i files ed occuparsi di tutte le richieste di operazioni sui file provenienti dallo user space in modalità completamente trasparente alle applicazioni.

java.io.File

An abstract representation of file and directory pathnames.

public File(String pathname)

Creates a new File instance by converting the given pathname string into an abstract pathname

public File(String parent, String child)

Creates a new File instance from a parent pathname string and a child pathname string

java.io.File

public boolean exists()

Tests whether the file or directory denoted by this abstract pathname exists.

public boolean isDirectory()

public boolean isFile()

public long length()

public boolean delete()

java.io.File

public String getName()

Returns the name of the file or directory denoted by this abstract pathname. This is just the last name in the pathname's name sequence.

public File getParentFile()

Returns the abstract pathname of this abstract pathname's parent

java.io.File

public String getAbsolutePath()

Returns the absolute pathname string of this abstract pathname. (relative path -> absolute)

public String getCanonicalPath()

A canonical pathname is both absolute and unique. The precise definition of canonical form is system-dependent. Utilizza *getAbsolutePath*.

Risolve . .. e “symbolic links”…

java.io.File

public File[] listFiles()

Returns an array of abstract pathnames denoting the files in the directory denoted by this abstract pathname.

public boolean mkdir()

Creates the directory named by this abstract pathname.