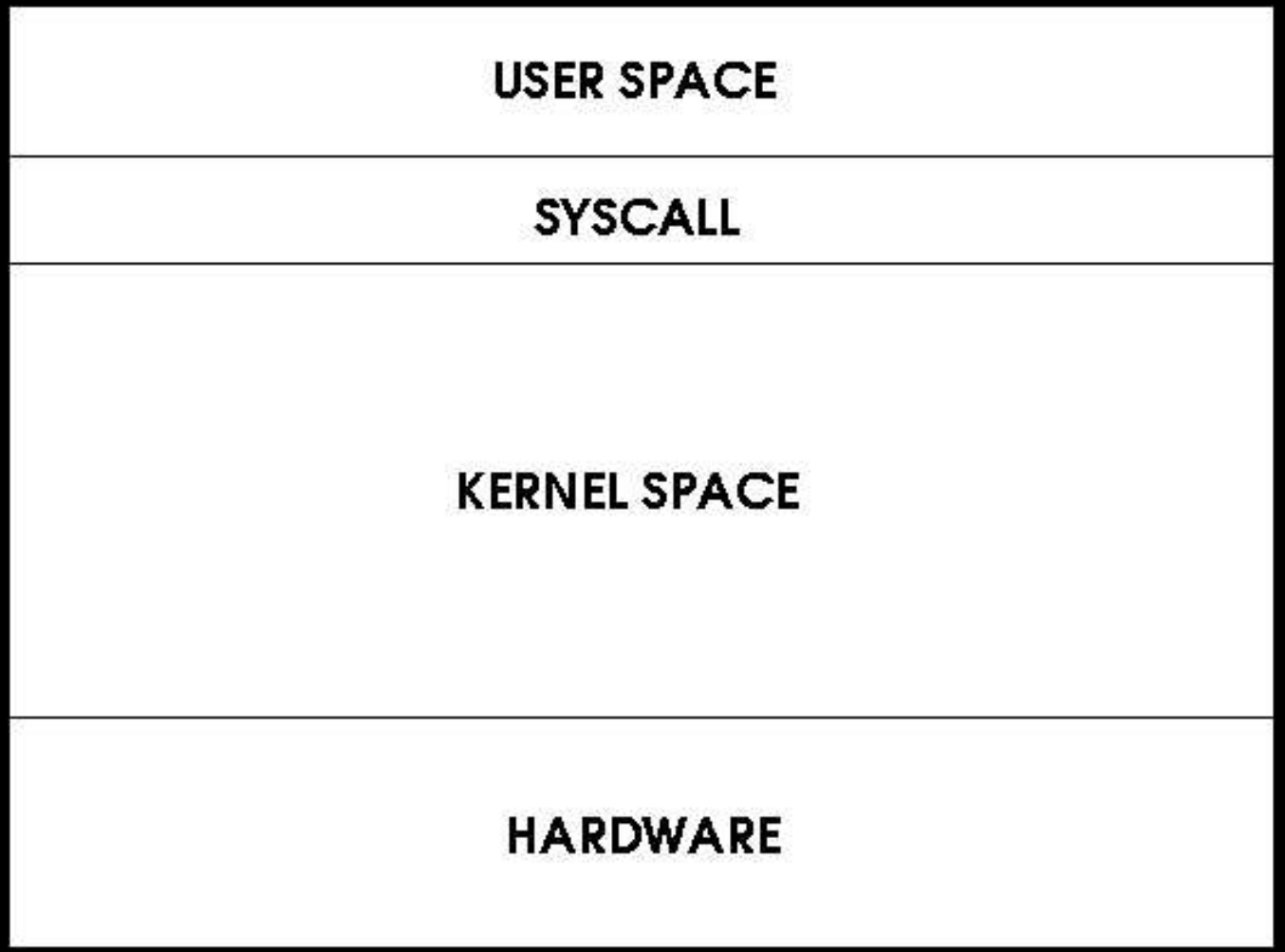
Principi base di un sistema Unix / Linux.

User & Kernel



Kernel space

Unix/Linux gira in modalita’ kernel:

- Quando e’ in attesa di Input/Output
- Quando e’ necessario accedere direttamente all’hw
- Ogni volta che un processo deve essere creato, distrutto, schedulato
- Quando e’ necessario allocare memoria
- Quando si utilizzano i protocolli di rete (TCP/IP)

User Space

- Tutti gli applicativi degli utenti girano in user-space
- I processi in user-space comunicano tra di loro (IPC)
- I processi in user-space accedono all’hw attraverso il kernel
- Per utilizzare i servizi offerti dal kernel, vengono utilizzate le "chiamate di sistema"

Le chiamate di sistema

- Sono l’interfaccia tra il mondo degli utenti e il kernel
- In Linux sono attualmente 289
- La comunicazione tra processi utente e kernel avviene tramite opportune combinazioni di syscall
- Le syscall sono standard (POSIX)

Concetti base di un sistema Unix/Linux

Gli elementi chiave di un sistema Unix / Linux sono:

- Utenti
- Filesystem
- Processi
- Rete

Utenti

Unix e’ nato come sistema operativo multiutente

- Piu’ utenti possono utilizzare lo stesso sistema (hw/sw) contemporaneamente
- Ciascun utente e’ identificato da un numero (UID) e da un nome (Login)
- Solitamente gli utenti accedono al sistema fornendo la loro Login e una "password"

Root == Potere !!!

In tutti i sistemi Unix esiste sempre un "superutente" che ha UID "0" e login "root".

Root ha il potere di fare TUTTO.

Per questo motivo root "interviene" solo in casi di estrema necessita':

- Configurazione del sistema
- Installazione delle applicazioni
- Gestione degli utenti

Gruppi

- Nei sistemi Unix esiste il concetto di "gruppo di utenti"
- Il sistema dei gruppi consente di "separare" gli ambiti di azione di un determinato insieme di utenti
- Ogni utente puo’ appartenere a diversi gruppi
- Ogni gruppo include, in genere, diversi utenti

Esempi di gruppi

- root
- bin
- sys
- daemon
- users
- www-data
- mail

I Filesystem

Filesystem significa letteralmente "Sistema dei file"

Il termine filesystem si utilizza per indicare:

- L’intero sistema di organizzazione dei file in Unix
- I diversi algoritmi di gestione "fisica" dei file

File e Cartelle

Gli elementi che costituiscono un filesystem possono essere

- File (contenitori di dati)
- Directory/Cartelle (contenitori di file/cartelle)

Tutto e’ un file....

In un sistema Unix vale la massima "All is a file"

Tutti i dispositivi e i meccanismi di input/output

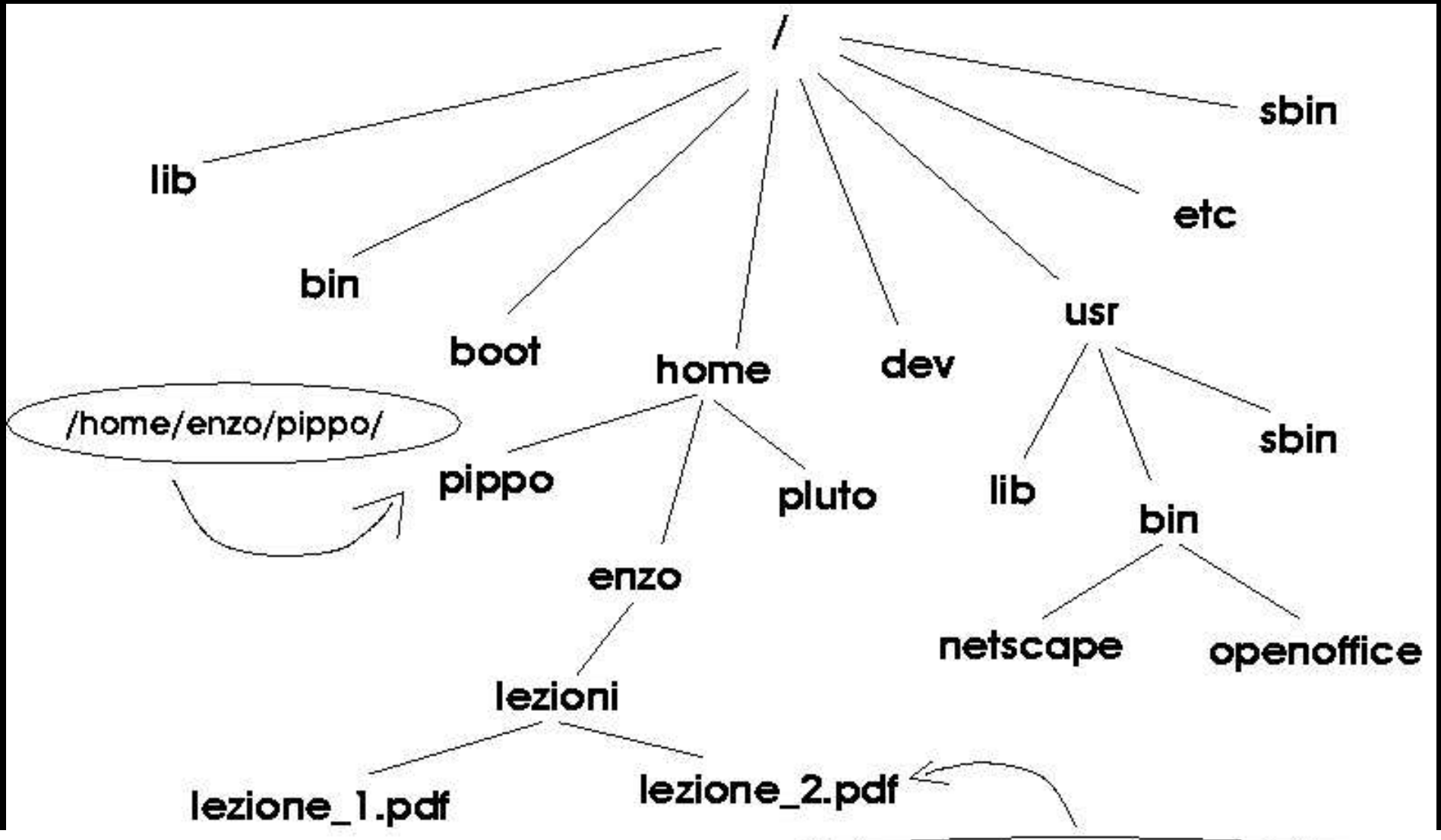
sono rappresentati mediante file

- File dati
- File di dispositivo
- Link
- Directory
- File speciali (pipe)

Il filesystem

- Il filesystem di Unix e’ di tipo gerarchico
- La directory principale si chiama "/" (root)
- Ciascuna directory puo’ contenere dei file o delle directory
- Ciascun file/directory appartiene ad un utente e ad un gruppo

Aspetto di un filesystem Unix



./ e ../

In ciascuna directory esistono sempre almeno due "directory"

- ./ : fa riferimento alla stessa directory
- ../: fa riferimento alla directory superiore

I percorsi (Path)

Per identificare un file o una directory del filesystem e’ necessario conoscerne la posizione

Il percorso per raggiugere un file o una directory si chiama PATH

Un path puo’ essere:

- Assoluto (/home/enzo/index.html)
- Relativo (./../../pippo.java)

I permessi....

In Unix ciascun file ha associati dei "permessi"

I permessi di un file indicano le operazioni ammesse sul file

Un file puo’ essere:

- Leggibile (r)
- Scrivibile/Modificabile (w)
- Eseguibile (x)

Permessi utente, gruppo e "resto-del-mondo"

- I permessi di un file vengono rappresentati mediante 9 caratteri
- I caratteri sono raggruppati a gruppi di tre
- Ciascun gruppo indica, da sinistra a destra:
  - I permessi per l’utente possessore
  - I permessi per il gruppo possessore
  - I permessi per tutti gli altri utenti

Permessi: un esempio

```
enzo@bagheera:~/$ ls -l ./esempio
total 0
-rw-r--r--  1 enzo users    0 Jan 18 23:20 index.html
drwxr-xr-x  2 enzo users 4096 Jan 18 23:22 paperino
-rw-rw-r--  1 enzo users    0 Jan 18 23:20 pippo.java
-rwxr-----  1 enzo users    0 Jan 18 23:21 pluto.sh
```

Le directory di sistema

E’ possibile creare directory a proprio piacimenento

Esistono pero’ un certo numero di directory "di sistema", presenti solitamente su tutti i sistemi unix

FHS: Filesystem Hierarchy Standard

Stabilisce uno standard per l’organizzazione delle directory di sistema

FHS (1)

- /bin: Comandi essenziali
- /boot: File necessari per il boot del sistema
- /dev: File di dispositivo
- /etc: File di configurazione
- /home: Home-directory degli utenti
- /lib: librerie essenziali e moduli kernel
- /root: Home di root

FHS (2)

- /sbin: comandi essenziali per root
- /tmp: file temporanei
- /usr: Programmi e librerie condivisi
- /var: dati variabili

Tipi di filesystem

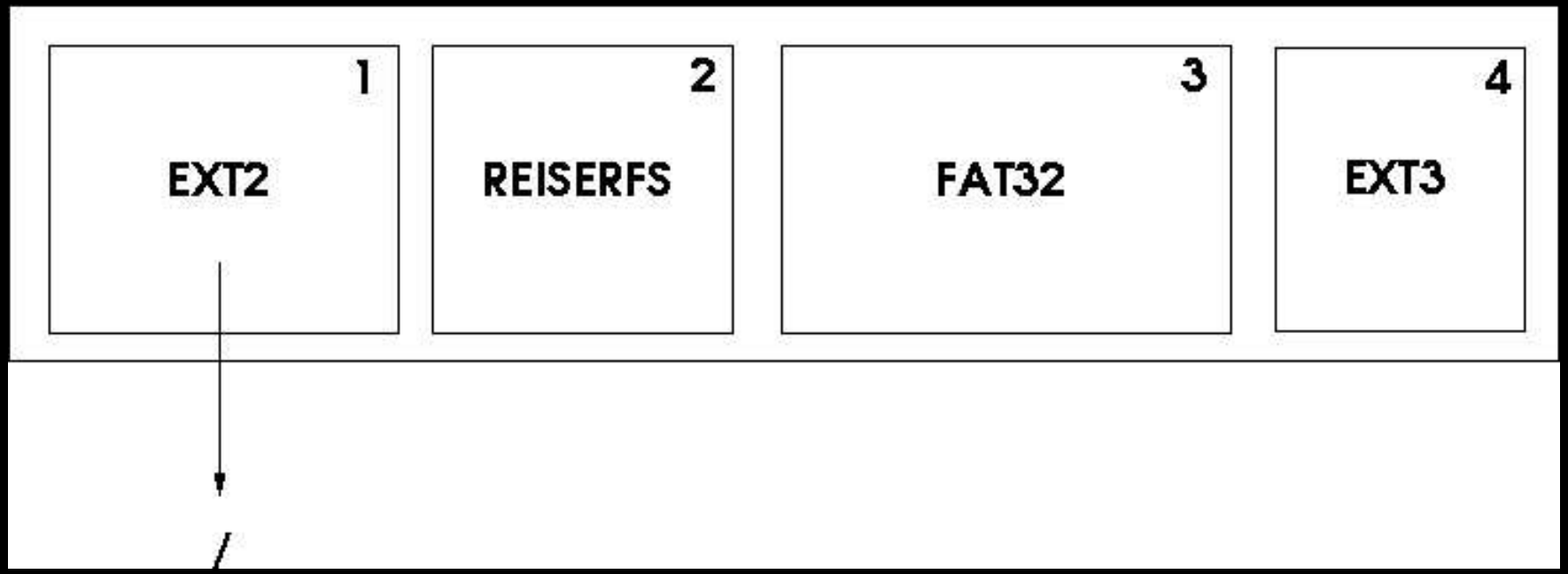
- I sistemi Unix permettono l’utilizzo di diversi tipi di FS
- Linux supporta quasi tutti i FS esistenti:
  - ext2
  - ext3
  - reiserfs
  - FAT (16, 32, VFAT) (DOS/Windows)
  - NTFS (Windows)
  - iso9660 (CDROM)

Montare i filesystem

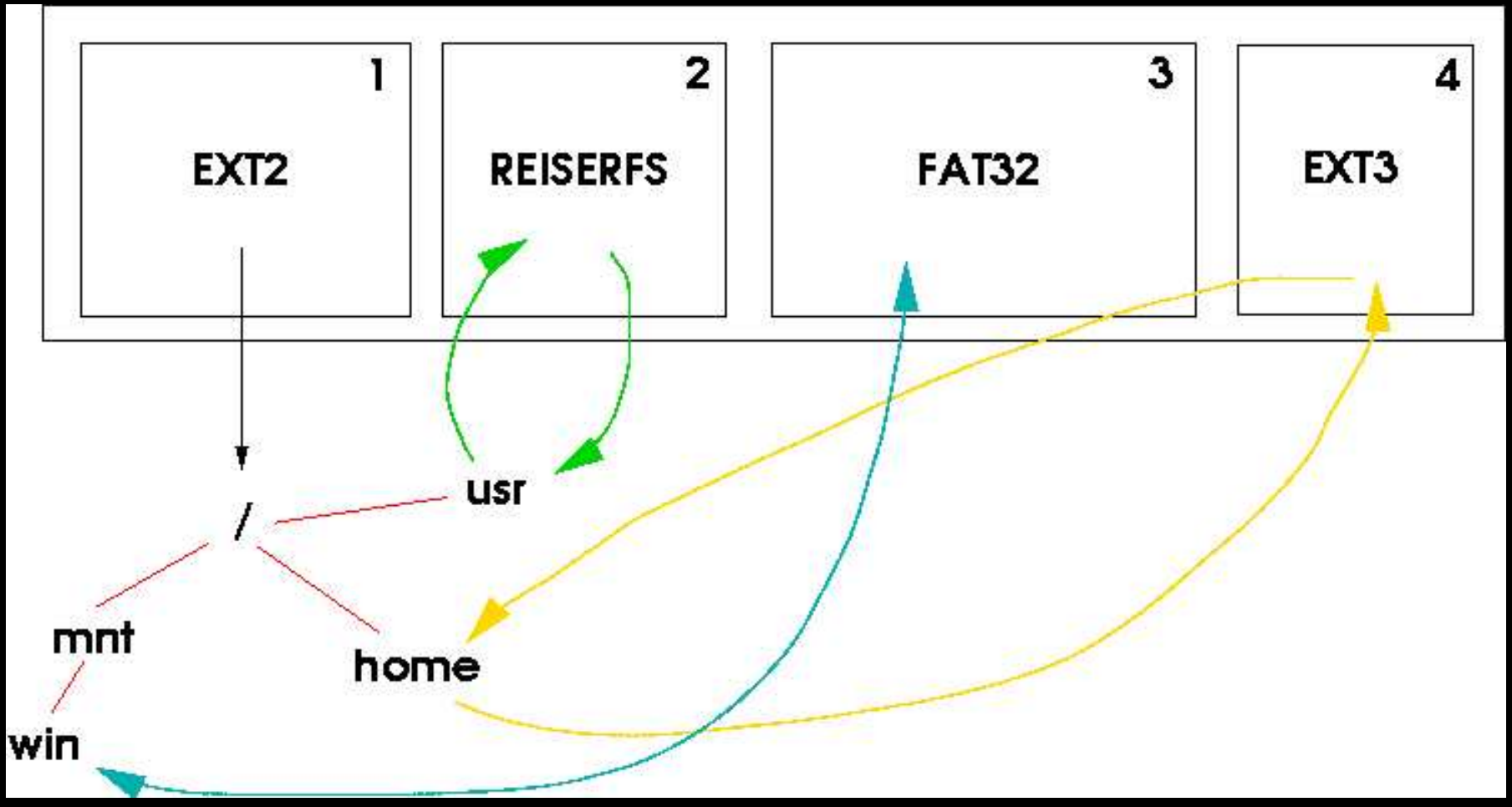
- In genere in uno stesso disco esistono diverse partizioni
- Ciascuna partizione contiene un FS differente
- Per rendere un FS "visibile" in Unix, e’ necessario "montarlo"
- L’operazione di montaggio "aggancia" il FS ad una subdir di "/"
- Se un FS e’ montato, e’ possibile accedervi, leggere e scrivere dati
- Quando un FS non serve piu’, si "smonta"

Esempio: Montaggio di un FS

Ammettiamo di avere un disco con 4 partizioni:



Esempio: Montaggio di un FS



Processi (1)

Un processo e’ rappresentato semplicemente da un programma utente

Un utente puo’ eseguire "contemporaneamente" piu’ processi

Ogni processo puo’ essere suddiviso in "thread" autonomi

Processi in Unix

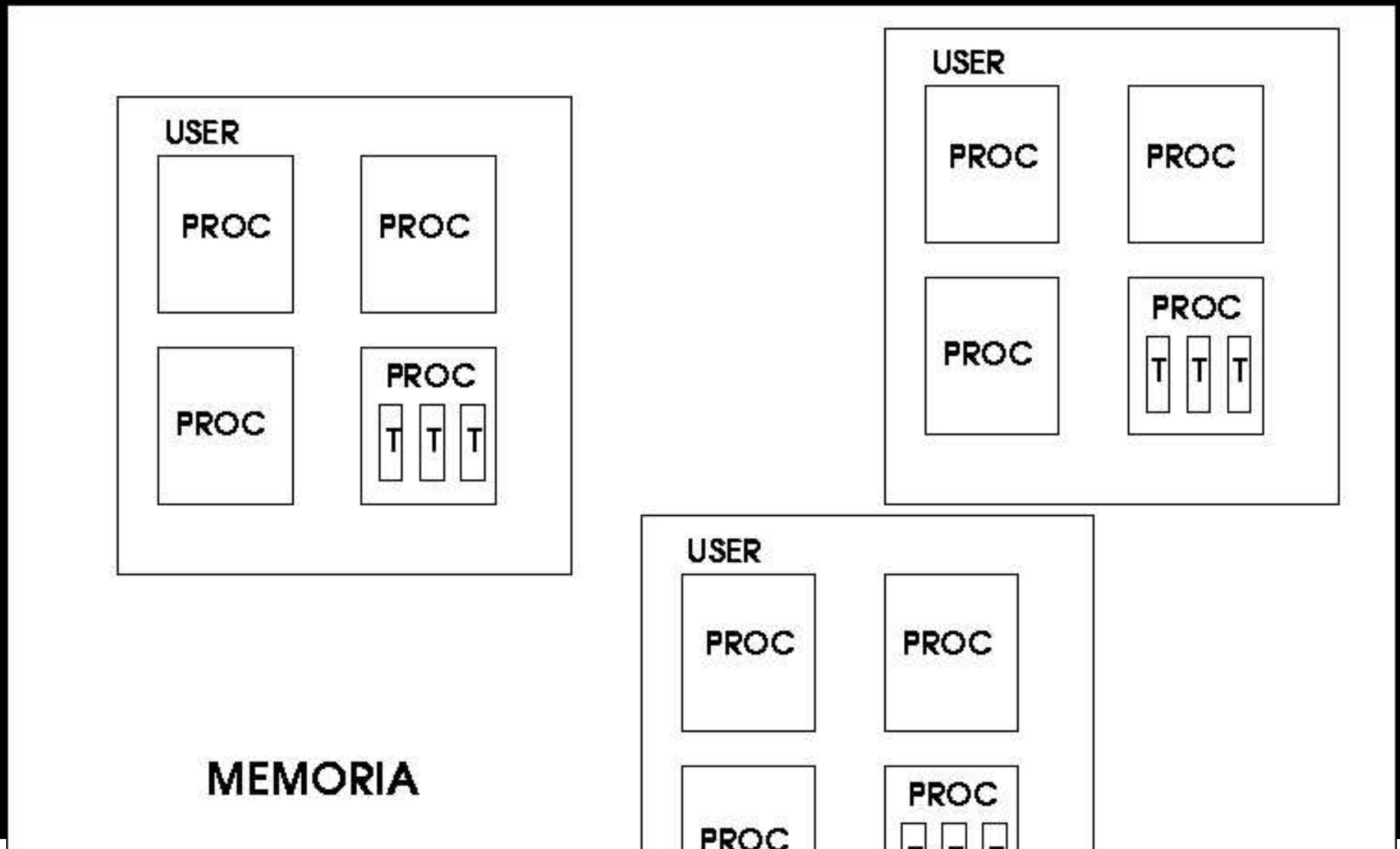
In Unix ogni processo e’ identificato da un numero (PID)

Ogni processo Unix appartiene in genere all’utente che lo ha lanciato

Ogni processo Unix ha una certa "priorita’ di esecuzione"

Un processo Unix puo’ essere creato, bloccato, sbloccato, distrutto

Processi (2)


USER
PROC
PROC
PROC
PROC
T
T
T
USER
PROC
PROC
PROC
PROC
T
T
T
USER
PROC
PROC
PROC
MEMORIA