

# Amministrazione di Sistemi UNIX

## Compiti del *System Administrator*:

- **Gestione delle risorse**
  - gestione file system (mount/umount)
  - backup (dump/restore)
  - monitoraggio del sistema:
    - occupazione file system
    - controllo servizi di rete
    - controlli per la sicurezza
- **Gestione degli utenti**
  - inserimento/rimozione
  - accounting (quote, numero processi)
- **Configurazione**
  - nuovi dispositivi
  - nuovi filesystem
  - nuovo software

Molte operazioni non per utenti ordinari:

### **superutente** (utente **root**)

- comando *su*:
  - crea uno shell per l'utente *root*
  - `$ su`
  - `Password: ...`
  - `#`

# Gestione File System

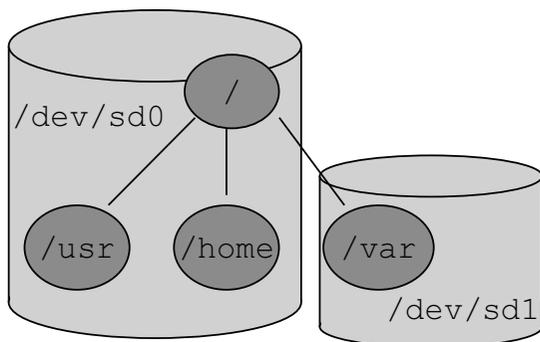
<i>directory</i>	<i>contenuto</i>
<code>/bin</code>	comandi e utilities
<code>/dev</code>	device files
<code>/etc</code>	file config. e startup
<code>/lib</code>	librerie
<code>/tmp</code>	file temporanei
<code>/sys</code>	area kernel, file config.
<code>/usr/bin</code>	file eseguibili
<code>/usr/include</code>	file header del C
<code>/usr/man</code>	pagine del manuale
<code>/usr/local</code>	sw installato
<code>/usr/ucb</code>	Berkeley utilities (BSD)
<code>/var/adm</code>	accounting files
<code>/var/spool</code>	spooling directory
<code>/kernel</code>	file del kernel (Solaris)

<i>file</i>	<i>funzione</i>
<code>/unix, /vmunix</code>	kernel del SO
<code>/etc/passwd</code>	info utenti abilitati
<code>/etc/rc.*</code>	run control script, config. iniziale
<code>/etc/hostname.1e0</code>	nome della macchina
<code>/etc/resolv.conf</code>	indirizzo dei DNS
<code>/etc/hosts</code>	database macchine
<code>/etc/host.equiv</code>	macchine trusted
<code>/etc/exports</code>	file system esportati
<code>/etc/printcap</code>	config. stampanti e spooler

## Gestione File System

- l'insieme delle risorse gestite da un sistema Unix è rappresentato da **un insieme di file system**, organizzati all'interno della stessa struttura logica -> FILESTORE
- un file system corrisponde a un dispositivo fisico (es. un disco oppure una partizione di disco, /dev/sd0...)
- più dispositivi fisici possono essere aggregati in modo omogeneo all'interno della stessa struttura logica (il Filestore)

Filesystem	mounted on
/dev/sd0a	/
/dev/sd0h	/usr
/dev/sd0g	/home
/dev/sd1c	/var



## Comando df:

visualizza la struttura attuale del Filestore

```
$df
Filesystem kbytes used avail cap. mount on
/dev/sd0a 30807 27456 271 99% /
/dev/sd0h 227263 197777 6760 97% /usr
/dev/sd0g 642934 556391 22250 96% /home
/dev/sd1c 1952573 1415225 342091 81% /var
```

## Comando du: du [ -s ] [ filename ... ]

visualizza il numero di kbytes occupati dal filename, o dai file contenuti nella directory filename

```
deis33# du -s ./*
168 ./antonio
5 ./barbara
0 ./boari
76 ./cesare
0 ./evelina
```

## Tipi di file

- regular file
- directory
- device file (character e block)
- UNIX domain socket
- Named pipes
- links (hard e soft)

```
drwxrwxr-x 11 root staff 512 Jun 28 10:51 home
```

Primo carattere:

```
- regular file
d dir
c character device file
b block device file
s Unix domain socket
p named pipe
l symbolic link
```

```
12 11 10 9 8 7 6 5 4 3 2 1
0 0 0 | 1 1 1 | 1 0 0 | 1 0 0
SUID SGID sticky | R W X | R W X | R W X
                    User   Group   Others
```

Nell'esempio sopra:

**root** è il proprietario (uid)

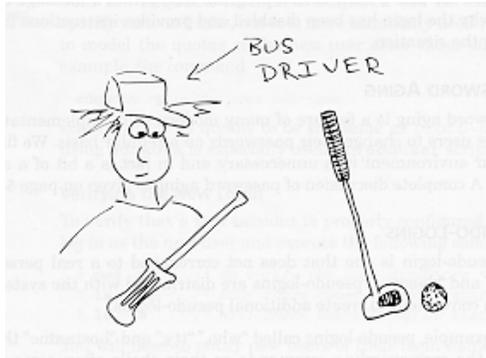
**staff** è il gruppo di appartenenza (gid)

## Pulizia del File System

File **core** : quando un programma va in crash, il kernel scrive un file chiamato **core**, che mantiene l'immagine dello spazio di indirizzamento del programma. Può essere utile a sviluppatori. Per gli utenti non serve, occupa spazio (anche molto) e va perciò eliminato.

File temporanei in **/tmp** o **/var/tmp** : molti programmi usano **/tmp** o **/var/tmp** per la gestione di file temporanei. Spesso accade che, soprattutto le directory, non vengano eliminate dal sistema. Periodicamente **/tmp** e **/var/tmp** vanno svuotate.

## Accesso ai dispositivi e alle periferiche UNIX: i device driver



Un device driver è un programma che gestisce l'interazione del S.O. con una periferica hardware.

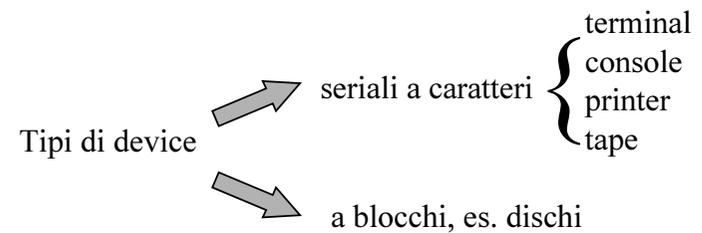
I device driver sono parte del kernel, non sono a livello utente, ma possono essere acceduti:

- direttamente dal **kernel**
- dallo spazio **utente**

## I device driver

Tutti i device driver presentano una interfaccia **standard** verso il kernel, composta di un set di funzioni standard (es. , probe, open, close, read, write, reset, etc.).

I driver implementano tutte o solo alcune di queste funzioni. L'indirizzo delle funzioni è contenuto in jump table nel kernel.



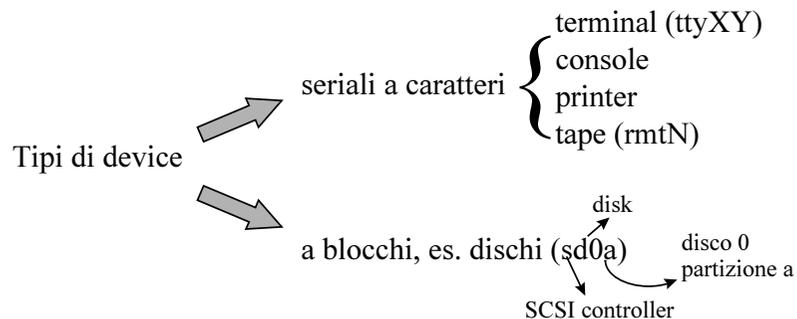
## I device file

Accesso ai device da **spazio utente**:

si utilizzano speciali file, i file device della directory /dev  
Il kernel trasforma le operazioni sui file device (tipiche operazioni sui file, quali open, read, write, etc.) in chiamate al codice del driver.

I file device sono mappati sui device attraverso due numeri (informazioni contenuta nell'i-node)

- **major** device number: identifica il tipo di driver associato al file
- **minor** device number: identifica uno specifico device, il numero di istanza di quel device



SCSI: interfaccia dei dischi, standard in ambiente UNIX

## I device file

Alcuni device possono essere acceduti sia a caratteri che a blocchi

esempio:

disco acceduto a blocchi come sd0a

disco acceduto a caratteri come rsd0a

raw

dischi, nastri : sia a blocchi che a carattere  
terminali, stampanti: solo a carattere

Alcuni device file speciali:

/dev/tty il terminale corrente

/dev/mem area di memoria corrente (serial character)

/dev/kmem area di memoria kernel corrente

/dev/pty pseudoterminal (Attenzione: **non** corrisponde a un dispositivo fisico)

## Aggiunta di un device driver

Bisogna aggiungere l'indirizzo delle funzioni del device driver nella corrispondente jump table (1 per serial character e 1 per block device) e il file device nella directory /dev

Modifica del kernel e sua ricostruzione

BSD:

- copiare il driver (file oggetto da linkare nel kernel) nella directory /sys/sunX/conf
- editare il conf.c aggiungendo nella jump table le entries delle funzioni del driver
- uso del config per ricostruzione del kernel
- aggiunta del device file usando comando mknod  
mknod filename type major minor  
(filename è il nome del device, type b o c)

Loadable kernel modules permettono di linkare o unlinkare un driver al kernel durante il suo funzionamento:

- Solaris (SystemV): add\_drv
- SunOS (4.1.2): modload, modinload

## Il sistema di spooling

Gestione dispositivi in un ambiente multiprogrammato

Esempio: lo spooling della stampante

Un processo potrebbe accedere direttamente al device file della stampante

**Problema:** come garantire una equa gestione del dispositivo condiviso (un processo potrebbe aprire il file della stampante e non rilasciarlo, bloccando la stampante)

**Soluzione:**

- creazione di un processo demone (unico processo con i diritti sul file device della stampante)
- directory di spooling

Per stampare un processo si genera il file da stampare e lo si copia nella directory di spooling.

In seguito il demone si incarica di stamparlo (politica FIFO di stampa).

Altro esempio di spooling: USENET

## Implementazione dello spooling della stampante (BSD)

demone di stampa: **lpd**

spooling directory: `/var/spool/nomestampante`

comando per la stampa: **lpr** (`lpr -Pprinter filename`)

comando per cancellare una stampa: **lprm**

(`lprm [-Pprinter][job #...][username]`)

Database di sistema con le info sulle stampanti:

**/etc/printcap** (caratteristiche stampante e dir di spool)

Esempio:

```
lp|colore|oce|color-laser|color laser:\
:lp=/dev/lp3
:sd=/var/spool/lp3:\
:lf=/var/adm/lp3-lat:\
:br#9600
lp|pebbles|pebbles printer:\
:lp=:\
:rm=deis197a:\
:rp=raw\
:sd=/usr/spool/lpd:\
:lf=/usr/adm/lpd-errs:
```

sd → spooler directory

lp → nome della stampante

lf → file di log degli errori

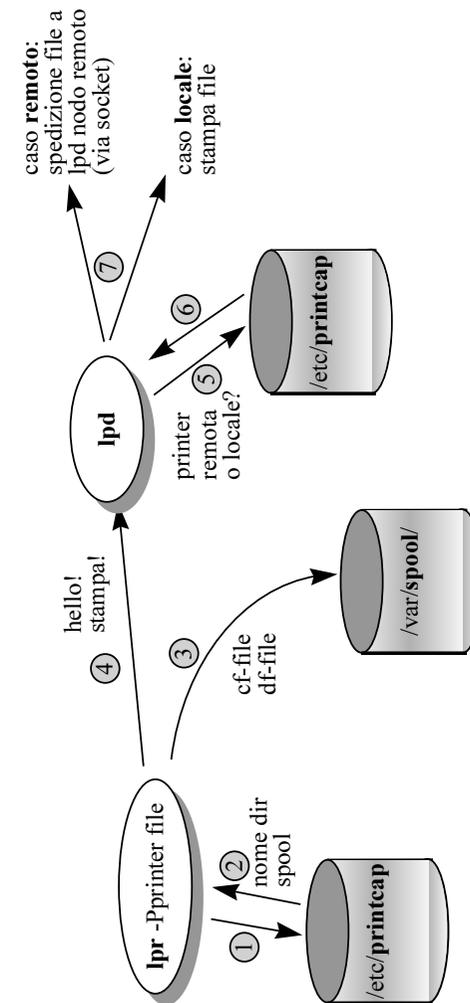
rm → remote machine

rp → nome stampante su macchina remota

`/var/spool/lpd.lock` Lock controllo presenza 1 lpd

Caso di stampante remota: lpd contatta lpd della macchina remota e gli trasmette il file da stampare.

## Implementazione dello spooling della stampante (BSD)



## Implementazione dello spooling della stampante

### Unix ATT:

demone di stampa: **lpsched**

spooling directory: **/var/spool/lp/request/dest**

comando per la stampa: **lp** (lp -Ddest filename)

comando per cancellare una stampa: **cancel**

Altri comandi: **lpshut**, **lpadmin**, **lpstat**

**/usr/spool/lp/SCHEDLOCK Lock 1 copia lpsched**

## Network File System (NFS)

NFS permette di condividere file system o parti di file system tra macchine distinte (introdotto da SUN)

Trasparenza per gli **utenti** per quel che riguarda:

- Allocazione file
- Comunicazione tra le varie macchine

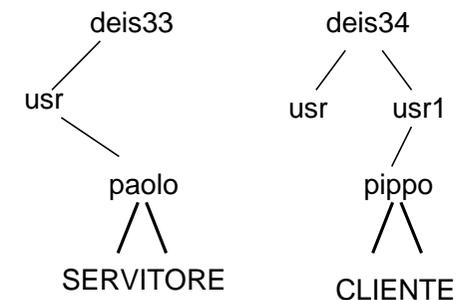
Esempio: si vuole far vedere una parte del file system di deis33 (server) a una macchina deis34 (client)

il superutente del server (deis33) deve **autorizzare l'esportazione** (/etc/exports) del file system richiesto

il superutente del client (deis34) deve **montare** (mount) il file system di deis33

```
mount deis33:/usr/paolo /usr1/pippo
```

A questo punto, tutta la gerarchia /usr/paolo è visibile dal file system come /usr1/pippo



## Network File System

### LATO CLIENT

- **mount** del file system residente sulla macchina server
- uso di tecniche di **caching** per migliorare le prestazioni (es. il demone biod)

### LATO SERVER composto di due demoni

- **mountd** attende le richieste di mount (e controlla l'autorizzazione all'esportazione su /etc/exports) (Attenzione ai possibili problemi di sicurezza)
- **nfsd** serve i file ai clienti (uso di più copie di nfsd per migliorare le prestazioni)

ATTENZIONE: il server NFS è **stateless**

## Mount

Un filesystem deve essere montato prima di essere accessibile. Il punto di mount può essere una qualunque directory. Gli eventuali file locali non saranno più accessibili dopo il mount.

```
mount deis33:/usr/paolo /usr1/pippo
```

Se faccio

```
#ls  
lost+found
```

e' una directory creata automaticamente. E' usata al boot per il check del filesystem. Non deve essere cancellata.

## Automount

Eeguire il mount automaticamente al boot.

Solaris: /etc/vfstab

SunOS BSD: /etc/fstab

Tipica configurazione di /etc/fstab:

```
/dev/sd0a      /          4.2  rw    0 0  
/dev/sd0d      /usr       4.2  rw    0 0  
.....  
.....  
deis33:/home   /rhome    nfs  rw,.. 0 0
```

## Mounting di File Systems

- Operazione che associa ad un file system (cioè un dispositivo fisico), una posizione **arbitraria** all'interno del filestore
- il mounting dei filesystems deve essere fatto **esplicitamente**:

**unica eccezione:** / rappresenta il disco di sistema e viene montato automaticamente nella fase di bootstrap.

- comando **/etc/mount** :  
\$ /etc/mount <nome dispositivo> <nome logico>

### Esempio:

```
$/etc/mount /dev/sd0g /home
```

La nuova situazione viene registrata nel file:  
**/etc/mtab** lista dei file systems montati

### Per effettuare l'operazione inversa:

- comando **/etc/umount** :  
\$ /etc/umount <nome logico>
- Il file system <nome logico> non deve essere in uso  
(open file, processi che han fatto cd al suo interno)

## Come usare il floppy e il CD

Per il floppy e il CD, così come per i file system remoti, occorre la procedura di **mount/umount**.

### Floppy Disk

Piattaforma	File device
Solaris	/dev/diskette
HP-UX	/dev/rdisk/cntnd0
SunOS	/dev/rfloppy

```
mount /dev/diskette /floppy
```

Se in /etc/fstab c'è ;

```
/dev/diskette /floppy pcfs rw,noauto 0 0
```

allora basta fare

```
mount /floppy
```

### CDROM

#### Solaris

```
mount -F hsfs /dev/dsk/c0t6d0s0 /CDROM
```

#### BSD

```
mount -t iso9660 /dev/sd1a /CDROM
```

## Come usare un floppy DOS su un sistema UNIX

### SunOS

*'solo se si deve formattare il dischetto'*

```
# fdformat -d -b label
```

*'solo se non esiste già la directory di mount'*

```
# mkdir /floppy
```

*'mount del floppy'*

```
# mount -t pcfs /dev/fd0 /floppy
```

*'operazioni di lettura e scrittura'*

```
# cp *.dat /floppy
```

```
# cp file2.txt /floppy/file1.txt
```

```
# cp /floppy/*.txt /file1.txt
```

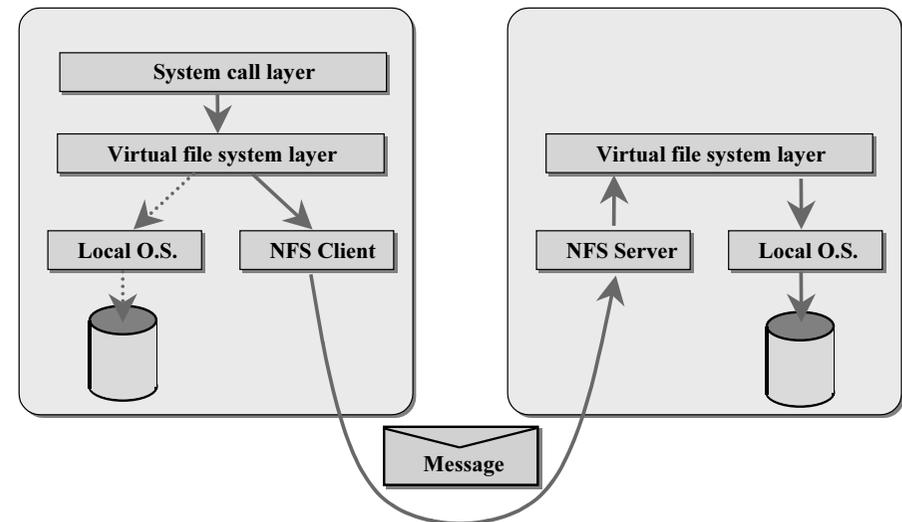
*'unmount del floppy'*

```
# umount /floppy
```

*'estrazione del floppy'*

```
# eject
```

## Implementazione NFS



## Network File System

- Un utente sulla macchina client NFS deve avere un login sulla macchina server
- Solitamente si mette come nome della dir montata il nome del nodo server
- NFS usa UDP, che dà problemi nel caso di reti estese o che attraversano router, poiché UDP manca del controllo di congestione dei pacchetti. Esistono varianti di NFS basate su TCP per ovviare a questi problemi.
- Comandi per il controllo del funzionamento di NFS: `nfsstat`
- Attenzione ai problemi di coerenza dei dati (NFS è stateless)
- Automounting, possibili replicazioni (fault tolerance) maggiore efficienza

## Backup dei file system

- Necessità di salvare periodicamente le informazioni contenute nel Filestore.
- Necessità di supporti ad alta capacità (es. nastri)

### **Comando `/etc/dump` :**

effettua il backup di un archivio in un dispositivo seriale.

### **Caratteristiche:**

- backup su nastro magnetico anche su più cassette
- possibilità di scelta dei file da copiare
- in un dump si può copiare al massimo un intero file system (un device con più file system richiede diverse operazioni di dump)
- conserva tutte le permission dei file
- può essere usato solo dal super-user (root)
- possibilità di ridirigere il dump su un qualunque dispositivo: opzione *f*
- operazione costosa in termini di tempo: possibilità di back up incrementale

## dump a livelli

- **livello 0:** copia integrale -> viene copiato tutto

```
$dump 0uf /dev/rst0 /usr
```

l'opzione *u* provoca la registrazione della data e del tipo del dump nel file */etc/dumpdates*

- **livelli [1,..9]:** copia parziale "alle differenze"

```
$ dump 3u /usr
```

vengono copiati soltanto i files ed i direttori la cui data è posteriore a quella del dump di livello massimo *N* tale che  $N < 3$

## Esempio:

- Dump giornaliero

Lunedì:	<i>\$ dump 0u /usr</i>
Martedì:	<i>\$ dump 1u /usr</i>
Mercoledì:	<i>\$ dump 2u /usr</i>
Giovedì:	<i>\$ dump 3u /usr</i>
Venerdì:	<i>\$ dump 4u /usr</i>
Sabato:	<i>\$ dump 5u /usr</i>

## Versione remota del dump

```
/usr/etc/rdump 0uf hostname:device /dir
```

## Ripristino di Files Restore

Operazione di ripristino di files salvati mediante dump.

### Comando */etc/restore*:

- */etc/restore t* (oppure *dumpdir*)  
l'opzione *t* provoca la visualizzazione della lista dei file e direttori contenuti nel dump
- */etc/restore r*  
l'opzione *r* indica che l'intero contenuto del dump va ripristinato
- */etc/restore x <nomefile>*  
l'opzione *x* consente di estrarre in modo selettivo i files/direttori contenuti nel dump
- */etc/restore i*  
modalità interattiva (utenti meno esperti).

## Alternativa al Dumping: *tar*

### Comando *tar* (tape archive):

- è disponibile a tutti gli utenti
- possibilità di effettuare copie selettive in base al nome
- minore compressione dei dati

### Opzioni:

- *c* indica che si vuole creare un archivio  
Es: `$ tar c /usr/anna/miofile`
- *x* indica che si vuole estrarre dei file dall'archivio  
Es: `$ tar x /usr/pippo/file1 .`
- *t* visualizza il contenuto del nastro
- *f* *<nomedev>* indica un dispositivo diverso da quello di default (anche un file ordinario!)

```
Es: $ tar cf tarfile f*
     $ tar tf tarfile
     fil.c
     fileris
     filetemp1
     filetemp2
     $ tar xf tarfile fil.c
```

## Alternativa al Dumping: *cpio*

*cpio* è un altro comando per l'archiviazione, simile al *tar*, molto diffuso in ambito System V.

*cpio* copia dei file in un archivio o li estrae da esso.

Copy in mode: (estrae file da un archivio)

- *cpio -i* estrae dei file dallo standard input, che si assume essere il prodotto di un precedente *cpio -o*
- pattern matching per scegliere file da estrarre
- owner e group sono quelli dell'utente che esegue l'operazione, a meno che non sia superuser (che preserva owner e group originari)

Copy out mode: (inserisce file in un archivio)

- *cpio -o* legge dallo standard input una lista di nomi di file da archiviare
- questi file sono stampati sullo standard output (ridirezionabile)

Altre opzioni:

-E *file* specifica un file che contiene una lista di file da estrarre dall'archivio (opzione non sempre disponibile)

-O *archivio* specifica il nome di un archivio in cui inserire i file

```
$ ls a* p* | cpio -oc > ../archivio
$ ls a* p* | cpio -ocO ../archivio
$ cat ../archivio | cpio -ic
$ cat ../archivio | cpio -iE nomifile
```

## Gestione degli Utenti

### File `/etc/passwd`:

registra le informazioni associate ad ogni utente abilitato ad utilizzare il sistema

```
root:L1ps76dnmDdlo:0:1:Operator:/:usr/bin/tcsh
cesare:Eq66j976v/eug:230:30:Cesare Stefanelli:/home/deis33/cesare:/bin/tcsh
anna:N/ESBxb3fEJFU:210:30:Anna Ciampolini:/home/deis33/anna:/bin/sh
paola:*:290:30:Paola Mello:/home/deis33/paola:/bin/csh
evelina:*:250:30:Evelina Lamma:/home/deis33/evelina:/bin/csh
.....
```

### File `/etc/group`:

- registra le informazioni associate ad ogni gruppo definito nel sistema

```
wheel:*:0:
staff:*:30:
guests:*:40:
users:*:50:
print:*:100:andrea,cesare,michela,enrico,franco
studenti:*:210:
```

Possibilità di usare **shadow** password (crittate):  
`/etc/shadow`

Per cambiare password:

da **root**: `passwd nome_utente`

da **user** (solo la propria) : `passwd`

## Aggiunta di un nuovo utente

- editing del file `/etc/passwd`  
(attenzione ai possibili conflitti !)

```
paolo::230:30:P. Rossi:/home/paolo:/bin/tcsh
```

- eventuale editing di `/etc/group` (in caso di un nuovo gruppo)

```
staff:*:30:
print:*:100:andrea,cesare,michela, paolo
```

- creazione della home directory dell'utente e relativa attribuzione di username e gruppo:

```
#mkdir /home/paolo
#chown paolo /home/paolo
#chgrp staff /home/paolo
#chmod 700 /home/paolo
```

- copia di files di configurazione standard relativi alla shell prescelta nella home directory:

### C Shell (`/bin/csh`)

`.login` (terminal type, variabili d'ambiente)

`.cshrc` (alias di comandi, path del command search)

### Bourne Shell (`/bin/sh`)

`.profile` (circa come `.login` + `.cshrc`)

## Eliminazione di un Utente

```
paolo::230:30:P. Rossi:/home/paolo:/bin/tcsh
```

- eliminazione della home directory dell'utente:  

```
#rm -r /home/paolo
```
- eliminazione di eventuali altri files/direttori di proprietà dell'utente:  

```
# find / -user 230 -print
```

=> ottengo la lista dei files di proprietà dell'utente con UID = 230.

```
# rm...
```
- eliminazione della riga corrispondente nel file  

```
/etc/passwd
```
- eliminazione di eventuali riferimenti all'utente nel file  

```
/etc/group
```

## Strumenti avanzati di gestione utenti

### Solaris system V e Linux :

```
useradd (userdel, usermod)
useradd [ -c comment ] [ -d dir ] [ -e
  expire ][ -f inactive ] [ -g group ]
  [ -G group [, group...]] [ -m [ -k
  skel_dir ]][ -u uid [ -o]] [ -s shell
  ] login
```

**SunOS BSD :** adduser (script in shell Bourne)

### *Srtrumenti ad interfaccia grafica*

**Solaris 2.4 :** admintool

**HP-UX :** SAM

## Accounting

- Necessità di registrare i tempi di utilizzo del sistema per ogni utente accreditato
- Necessità di conoscere lo spazio allocato complessivamente nel filestore per ciascun utente

### Registrazione dei login:

- `/usr/adm/wtmp` è un file di sistema in cui vengono registrati tutti i login
- comando `ac`: fornisce i tempi complessivi di collegamento a partire dalla data di creazione (o inizializzazione del file `/usr/adm/wtmp`)

```
# ac -p (ottengo i totali individuali)
antonio 719.63
paola 800.66
cesare 1679.91
evelina 857.97
michela 1154.04
franco 918.23
...
total 7753.36
```

### Calcolo dell'area occupata:

- comando `quot -f` restituisce il numero totale di blocchi per ogni utente

## Bootstrap/Shutdown

- **Bootstrap**: è la fase transitoria di inizializzazione del sistema
- **Shutdown**: è la fase che **deve** precedere lo spegnimento della macchina

### Cosa succede nella fase di bootstrap:

- 1) viene caricato ed eseguito il codice contenuto nel Boot-block del disco di sistema
- 2) compare un prompt (ad esempio: `>`):  
`> boot sd(0,0) vmunix`  
viene eseguito il programma boot che :
  - monta il file system associato al disco specificato `sd(0,0)` sul direttorio `/`.
  - carica il **kernel**:
    - check della memoria principale
    - creazione del primo processo: *init* (PID=1)
- 3) *init* crea un singolo shell di super-user (*single-user mode*): l'amministratore può compiere alcune azioni preliminari agendo sul sistema in modo esclusivo. Al termine dello shell (`^D`):

4) il sistema passa nello stato *multi-user*:

- *init* esegue il file comandi */etc/rc*:  
una serie di operazioni preliminari vengono eseguite
  - check dei filesystems (*/etc/fsck*)
  - mounting dei filesystems
  - creazione di processi in background (demoni):  
*update*  
*cron*
- *init* crea tanti processi *getty* quanti sono i terminali collegati al sistema (informazione contenuta nel file */etc/tty*) -> ogni processo *getty* realizza una macchina virtuale dedicata a un utente:
  - esecuzione di *login*

## Shutdown

### Necessità di lasciare i filesystems in uno stato consistente:

- il comando *sync* fissa sui dispositivi di memorizzazione secondaria la situazione attuale

### Uccisione di tutti i processi:

- *kill -1 1* uccide il processo *init*

### In molti sistemi:

- */etc/shutdown* è un file comandi che realizza le operazioni relative allo shutdown

## Reboot e ShutDown

Operazione del *root*

PC: spesso risolve i problemi

UNIX: quasi mai risolve problemi e spesso causa danni al sistema (tabelle inconsistenti, scollega gli utenti, ...)

### Quando farlo:

- installazione nuovo hardware;
- componenti che non si riesce a resettare;
- modifiche a file di configurazione che vengono letti solo al boot.

### Metodi per lo ShutDown/Reboot

#### Togliere l'alimentazione

- Perdita di dati;
- Tabelle di sistema in stato inconsistente;
- Danni a driver dei dischi.

#### Comando *shutdown*

Sintassi differente per le diverse piattaforme;

waiting period: invia un messaggio a tutti gli utenti collegati dando il tempo per il logoff;

piattaf.	comando	tempo	reboot	halt	sin.
Solaris	/usr/sbin/shutdown	-gsecs	-i6	-i0	}
HP-UX	/etc/shutdown	secs	-r	-h	
SunOS	/usr/etc/shutdown	+mins	-r	-h	

sin. : single user mode

## Altri strumenti (BSD)

Comandi remoti: iniziali tutte con r

**r<sub>cp</sub>** copia dei file da remoto a locale e viceversa

**r<sub>ls</sub>**

**r<sub>exec</sub>** esecuzione su un nodo remoto

uso di un demone **r<sub>excd</sub>**

...

### System Management a livello TCP

**hostname** identità del nodo corrente

**netstat** stato locale TCP

-a comunicazione

-i statistica

-r tabelle di routing

-s

-m buffer

-I monitoring

**ifconfig** esamina l'interfaccia con la rete

**ping** invia pacchetti ICMP al nodo remoto

**inetd** fa partire un server unico per i servizi locali

#### strumento di monitoring di basso livello

**etherfind** estrae informazioni riguardo ai pacchetti sulla rete (invocabile solo da root)

**traceroute** invio di messaggi fornendo informazioni sul routing

## Linux UMSDOS

Condivide la stessa partizione del DOS. E' stato adottato dalla SlackWare (free distribution).

### Come configurare la tastiera italiana

**KeyCode:** eventi associati dal kernel alla pressione di un tasto

**KeySim:** caratteri o stringhe (standard VT100) che Linux adotta per i terminali virtuali.

**KeyMap:** tabella di conversione (file di testo) che associa : *KeyCode* → *KeySim*  
Ossia, crea l'associazione tra tasto premuto (o combinazione di tasti) e carattere di output al terminale.

Quindi, modificando la KeyMap si puo' configurare a piacere la tastiera.

```
/etc/rc.d/rc.keymap
```

### Soluzione più semplice per caricare la tastiera italiana:

```
loadkeys it.map
```

oppure

```
loadkeys  
/usr/lib/kbd/keytables/it.map
```

### Conviene automatizzarlo al boot, inserendolo in:

```
/etc/rc.d/rc.local
```

### Per vedere la tabella di conversione corrente:

```
dumpkey
```