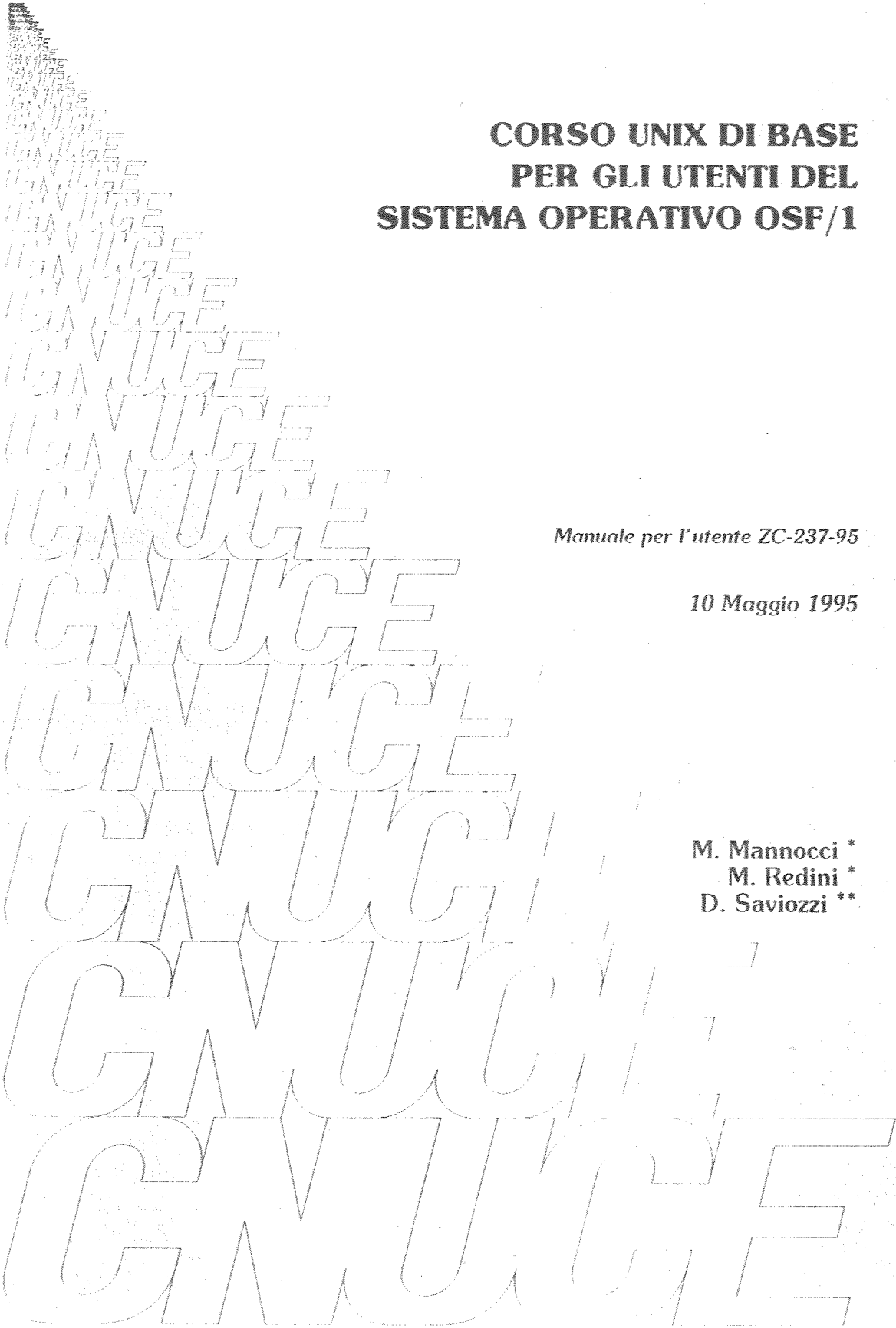


**CORSO UNIX DI BASE  
PER GLI UTENTI DEL  
SISTEMA OPERATIVO OSF/1**

*Manuale per l'utente ZC-237-95*

*10 Maggio 1995*

**M. Mannocci \***  
**M. Redini \***  
**D. Saviozzi \*\***



---

# Corso Unix di base per gli utenti del sistema operativo OSF/1 installato presso l'Autorità di Bacino, sede di Lucca

---

M. Mannocci\* , M. Redini\* e D. Saviozzi\*\*

---

Il corso di base del sistema operativo OSF/1 è stato organizzato per gli utenti dell'Autorità di Bacino, sede di Lucca, in collaborazione con il Consorzio Pisa Ricerche. Il corso è stato tenuto dal Prof. M. Mannocci, che ha tenuto anche il corso di base del sistema operativo OSF/1 per gli utenti dell'Autorità di Bacino, sede di Lucca, in collaborazione con il Consorzio Pisa Ricerche.

---

\* CNUCE C.N.R. Pisa  
\*\* Consorzio Pisa Ricerche

---

**Questa guida si propone di introdurre utenti inesperti all'uso di un sistema DEC OSF/1 e Unix in generale. Esso tende a fornire nozioni di base e rimanda, per le questioni rimaste aperte e per quelle che sorgeranno con l'uso del sistema, all'interazione con il gestore dello stesso.**

# Indice

BREVE STORIA DI UNIX .....	1
ALCUNE CARATTERISTICHE DEL SISTEMA UNIX.....	2
<b>1</b> GENERALITA'.....	<b>5</b>
1.1 I comandi principali	
1.2 Accesso alla documentazione e informazioni sui comandi per l'utente	
1.3 File e Directory.....	6
1.3.1 File	
1.3.2 Directory .....	7
1.3.3 Directory utente	
1.3.4 Il sistema UNIX come organizzazione di files	
1.3.5 Pathnames .....	8
1.3.5.1 Pathnames Assoluti e Relativi	
<b>2</b> UTILIZZO DI FILE E DIRECTORY, COMANDI PRINCIPALI .....	<b>11</b>
2.1 Proprietà dei File	
2.2 Creare/editare un File.....	12
2.3 Creare/concatenare e visualizzare un File .....	13
2.4 Cancellare, spostare e copiare un file	
2.5 Creare e cambiare directory .....	14
<b>3</b> ACCESSO AI FILE .....	<b>15</b>
<b>4</b> ALCUNI ALTRI COMANDI.....	<b>17</b>
<b>5</b> LA SHELL .....	<b>19</b>
<b>6</b> CONTROLLO dei PROCESSI.....	<b>21</b>
<b>GUIDA OPERATIVA.....</b>	<b>22</b>



### Premessa

Un Sistema Operativo è un insieme di strumenti per lo sviluppo e l'esercizio del software e delle comunicazioni per un dato Sistema Hardware, nel rispetto dei seguenti principi: a) visione delle risorse il più possibile indipendente da quelle hardware e dalla loro ubicazione nel sistema hardware; b) gestione corretta, sicura ed efficiente delle risorse. Un sistema operativo è composto da elementi di natura diversa, hardware, firmware<sup>1</sup> e software distribuiti su tutte le unità che costituiscono il sistema hardware. Tuttavia, parlando di sistemi operativi, è consuetudine riferirsi solo alla componente software.

## BREVE STORIA DI UNIX

Nato nel 1969 nei Bell Laboratories Unix deve il suo nome alla contrapposizione che si volle evidenziare con il suo predecessore Multics, sistema operativo innovativo ma troppo complicato nato al M.I.T.<sup>2</sup> negli anni sessanta.

Ken Thompson e Dennis Ritchie crearono un file system<sup>3</sup> di cui quello odierno di Unix mantiene molte caratteristiche e, successivamente, definirono un ambiente a processi<sup>4</sup> con scheduling<sup>5</sup>.

La prima versione fu scritta totalmente in assembly, ma nel tempo la maggior parte del codice è stato riscritto in C<sup>6</sup>, ed oggi solo alcune parti del Kernel<sup>7</sup> sono rimaste nell'originario assembly.

Installato inizialmente solo nelle università americane oggi Unix si può trovare su diverse centinaia di migliaia di sistemi, dai grossi mainframe ai più casalinghi Personal Computer.

1 E' l'insieme dei microprogrammi che risiedono su un'apposita memoria scrivibile di un sistema hardware. I microprogrammi descrivono il comportamento dell'hardware in ogni ciclo di clock.

2 Massachusetts Institute of Technology.

3 Parte del sistema che ha la responsabilità della memorizzazione delle informazioni nelle unità a disco e del loro reperimento ed aggiornamento, in base alle richieste degli utenti e dei programmi.

4 Un processo è un programma in esecuzione.

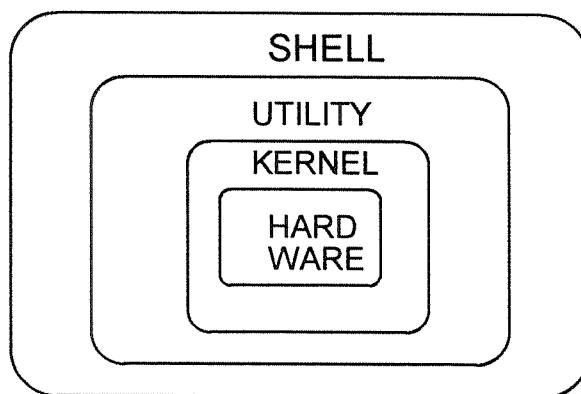
5 E' la strategia che, tenendo conto di diversi obiettivi, consente al sistema operativo di scegliere il prossimo processo a cui assegnare una delle risorse gestite, non appena essa è nuovamente disponibile. Risorsa è qualsiasi componente hardware o software gestita dal sistema operativo.

6 Linguaggio di programmazione, sviluppato ai Bell Laboratories, adatto a programmatori esperti per lo sviluppo del software di base.

7 La parte di sistema operativo, costantemente presente in memoria, che realizza le funzioni base.

**ALCUNE CARATTERISTICHE DEL SISTEMA UNIX**

- UNIX è un sistema multitask<sup>8</sup> multiutente con struttura modulare.



*Fig.1 La struttura*

- La Shell (guscio) è un interprete che esegue i comandi dell'utente e ha la possibilità di lanciare più comandi contemporaneamente concatenandoli tra loro (pipe) oppure ponendoli in background. L'operazione di porre in background ( & ) un comando consente all'utente di continuare ad avere il controllo della tastiera e dello schermo, mentre lo stesso è in esecuzione. La PIPE ( | ) è un mezzo che consente ai processi di scambiarsi dati
- Il linguaggio di comandi di sistema (Shell) è selezionabile da ciascun utente.
- UNIX distingue tra caratteri maiuscoli e minuscoli.
- Molti comandi hanno nomi che sono acronimi o abbreviazioni spesso non intuitive.

---

<sup>8</sup> Si dice task un percorso di esecuzione sequenziale all'interno del codice di un programma. Un programma viene eseguito all'interno di un processo. A seconda del sistema operativo un processo può essere costituito da uno o più task. Inoltre, un sistema operativo può essere progettato per eseguire processi multipli. Il progetto del sistema può prevedere che tutti i processi siano assegnati ad un solo utente od applicazione (monoutente). Un'implementazione più complessa del sistema operativo può assegnare i processi a più utenti indipendenti (multiutente).

**CORSO UNIX di BASE PER UTENTI OSF/1<sup>9</sup>**

Per usare un sistema UNIX è necessario che l'amministratore del sistema definisca l'identificatore di utente (userid) e la relativa parola chiave d'accesso (password).

Sul terminale o sulla console di sistema, all'apparire del messaggio di login, fornendo al sistema userid e password si ottiene l'accesso.

L'utente a cui è stato assegnata dall'amministratore di sistema la userid **corso**, ed una password (non visibile durante la digitazione), effettuerà così il login, cioè l'inizio della sua sessione di lavoro:

```
login: corso
Password: <INVISIBILE>
```

Potrà, a questo punto, usare il sistema secondo le possibilità attribuite al suo utente.

Ad esempio verificherà la data e l'ora:

```
date
Tue Aug 30 08:05:15 MET 1994
```

Potrà controllare la presenza di altri utenti:

```
who
atlante      :0          Aug 30 09:45
root         tty2        Aug 30 08:01
archery      vxt2002:0   Aug 30 08:02
atlante      tty5        Aug 30 09:45
renzo        vxt2001:0   Aug 30 10:16
renzo        tty4        Aug 30 10:16
dave         tty3        Aug 30 08:03
dave         tty6        Aug 30 11:46
corso      tty7      Aug 30 11:57
```

---

<sup>9</sup> Sistema della Digital che rispetta gli standard definiti dalla Open Software Foundation.



Terminerà la propria sessione di lavoro:

*exit* (o *logout*, o *CTRL-D*).

Appena effettuato il login compare il prompt<sup>10</sup>, che, di solito, si presenta come un segno dollaro (\$) o un percento (%), comunque modificabile dall'utente. Può, inoltre, essere presente il "message of the day" (solitamente è una segnalazione o un promemoria dell'amministratore del sistema) oppure una segnalazione che è presente della posta da leggere.

Gli errori di battitura possono essere corretti con i caratteri di controllo, i più comuni sono il *backspace* e il *line kill* (di solito CTRL-U).

---

<sup>10</sup> Messaggio inviato all'operatore dal sistema operativo; di norma, richiede una particolare informazione, necessaria per proseguire l'esecuzione di un programma

## 1 - GENERALITÀ

### 1.1 - I comandi principali

Il sistema legge i caratteri man mano che sono inseriti a terminale ma il carattere, o la sequenza di caratteri, deve terminare con *Invio* o *Return* (in dipendenza della tastiera disponibile) affinché possa essere interpretato. Può capitare di non vedere subito sul video quello che si batte sulla tastiera (ad esempio quando si da un comando prima che il precedente sia terminato), ma il sistema accetta tutto quello che viene scritto, anche prima che ritorni il prompt dal comando precedente.

L'esecuzione della maggior parte dei comandi può essere interrotta utilizzando contemporaneamente i tasti **CTRL-C**

Il carattere **CTRL-S** sospende l'output al terminale, mentre il carattere **CTRL-Q** lo ripristina.

Il comando **exit** o **logout** termina la sessione ed altrettanto fa la combinazione di tasti **CTRL-D** (talvolta la Shell è personalizzata per ignorare il tasto CTRL-D).

La seguente tabella mostra la situazione di default, cioè quella relativa all'impostazione standard dei comandi; tuttavia l'assegnazione delle varie funzioni ai caratteri della tastiera può essere modificata dall'utente, anche se ciò non è consigliabile per mantenere la compatibilità tra sistemi Unix.

Chiave	Funzione
<b>BACKSPACE</b>	Cancella l'ultimo carattere sulla linea di comando
<b>CTRL - W</b>	Cancella l'ultima parola sulla linea di comando
<b>CTRL - U</b>	Cancella l'intera linea di comando
<b>CTRL - D</b>	Carattere usato anche per terminare l'input di un file
<b>CTRL - S</b>	Sospende momentaneamente l'esecuzione di un programma
<b>CTRL - Q</b>	Riprende l'esecuzione del programma interrotto con CTRL - S
<b>CTRL - C</b>	Interrompe un programma in esecuzione

Tab. 1 : Funzioni

### 1.2 - Accesso alla documentazione e informazioni sui comandi per l'utente

La documentazione di UNIX è normalmente disponibile in linea, tramite il comando *man*. Se si desiderano informazioni circa l'utilizzo del comando *man*, basta digitare *man man*.

Il sistema DEC OSF/1 dispone dei comando *man* e di *xman*; quest'ultimo consente una modalità di consultazione più interattiva.

## 1.3 - File e Directory

Noto come *file system* nei sistemi Unix, il sottosistema dei servizi di accesso ai file/dati offre funzioni per la memorizzazione dei dati sui dispositivi di massa. Il sistema Unix tratta i file come file testo o file binari. I file testo sono sequenze di caratteri che usano il carattere ASCII<sup>11</sup> newline come separatore delle varie linee del testo. Un file binario è una sequenza di parole binarie<sup>12</sup>. I file eseguibili sono creati come file binari dal processo di compilazione<sup>13</sup> ed includono strutture aggiuntive per caricare il programma nella memoria principale.

Il file system è caratterizzato da una struttura di directory a più livelli (gerarchica), strutturata come un albero capovolto con un singolo nodo radice. Questo è detto radice (root), ed è convenzionalmente identificato dal carattere "/".

### 1.3.1 - File

Dal punto di vista del sistema, un file è una sequenza di bytes e può contenere qualunque tipo di informazione.

In UNIX tutto è rappresentato da files (file, directory, dischi, nastri, ecc.) ed il file system è gerarchico (come per l'Ms/Dos).

Ad ogni file viene assegnato un nome, lo spazio disco sufficiente a ospitare i dati e le informazioni che ne permettono l'amministrazione. Il nome è costituito da una stringa (sequenza) di caratteri qualsiasi; sono da evitare caratteri non visibili a terminale (es: ESC) e metacaratteri (es: \*, \$, -, ecc.); il nome del file può essere anche molto lungo, ma si consiglia di non superare gli 8 caratteri più 3 di estensione<sup>14</sup>, se si vuole interagire con il Dos, o comunque di non creare files con nome più lungo di 10/20 caratteri.

Il file system è organizzato in modo che ogni utente possa avere i suoi file personali senza interferire con quelli degli altri utenti.

Un file di UNIX è caratterizzato dalle seguenti specifiche:

- nome
- dimensione
- permessi di accesso
- data e ora di creazione o dell'ultima modifica, ed altro ancora

In un ambiente multiutente è utile, e spesso necessario, che i files di ogni utente siano separati da quelli degli altri; UNIX risolve questo tipo di problema raggruppando i file in directory.

---

11 Acronimo di American Standard Code for Information Interchange. Metodo di rappresentazione di un insieme di simboli mediante un altro insieme di simboli.

12 Si noti che la distinzione tra file testo e binari non è in alcun modo visibile al sistema operativo, ma soltanto alle applicazioni ed ai programmi di utilità che leggono e scrivono questi file.

13 Processo di traduzione del programma nel linguaggio della macchina.

14 Il sistema operativo MS-DOS prevede per il nome di un file un formato composto da un prefisso (il nome) di 8 caratteri ed un suffisso (estensione) di 3 caratteri, separati da un punto; l'estensione qualifica il tipo del file.

### 1.3.2 - Directory

Un file è collocato in una directory. Una directory è semplicemente un tipo di file che ha la capacità di contenere altri file ed altre directory (sotto-directory). E' possibile avere una estensione gerarchica illimitata di sotto-directory tutte contenute nella prima, cioè nella directory principale.

La struttura del file system può essere immaginata come un albero rovesciato.

E' possibile spostarsi attraverso questo albero ed individuare ogni file del sistema partendo dalla directory principale (radice o root) e muoversi lungo diversi rami. Non esiste nessuna limitazione di livello.

### 1.3.3 - Directory utente

Ogni utente ha una directory personale, la home directory (chiamata anche login directory) che contiene files che appartengono solo a lui. Al login l'utente si trova nella propria home directory. Può cambiare la directory in cui lavora (chiamata current directory) ma la home directory rimane sempre la stessa.

### 1.3.4 - Il sistema UNIX come organizzazione di files

Il sistema operativo considera files, directory e devices tutti alla stessa stregua, e li manipola come tali. La struttura ad albero del file system Unix può essere rappresentata secondo il seguente schema :

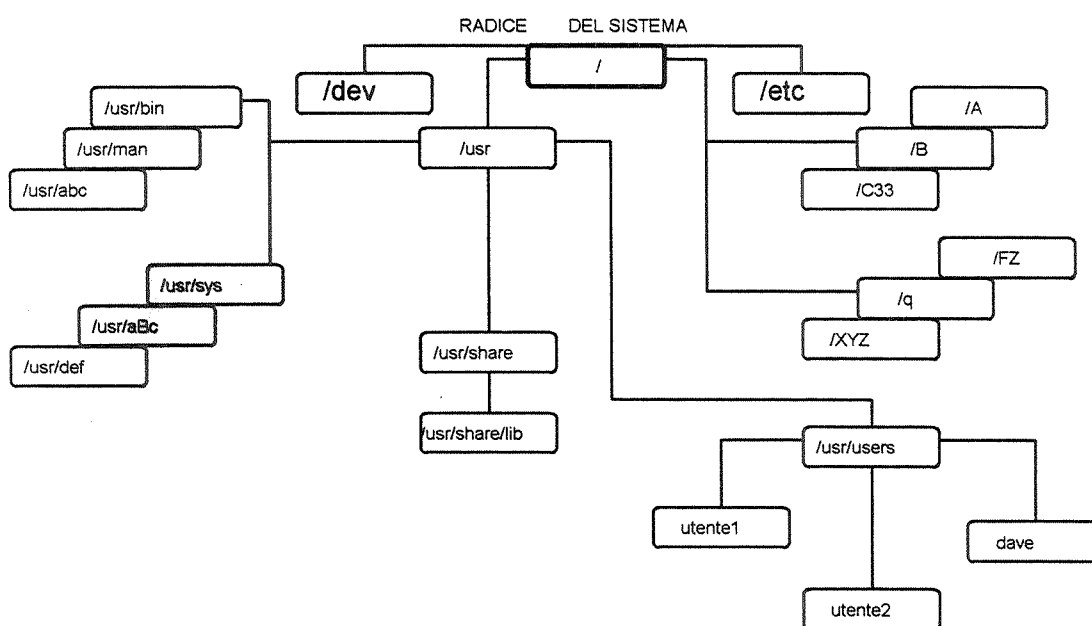


Fig. 2 : Struttura del File System

Il sistema operativo usa le directory /dev, /users, /etc, /bin e /usr come insieme base di directory.

Tipicamente esse contengono:

/dev file speciali per l'I/O sui dispositivi  
/usr comandi di sistema e programmi  
/etc file contenenti informazioni sugli utenti e programmi per la manutenzione del sistema  
/bin programmi di utilità  
/users file system montabili, contenenti i files utente

### 1.3.5 - Pathnames

Il comando *pwd*, visualizza il nome della directory in cui vi trovate (la current directory)

```
pwd  
/home/corso
```

Il risultato del comando vi informa che siete nella directory corso che si trova nella directory home che a sua volta è nella directory root.

La root directory viene chiamata "/". Il carattere "/" separa anche i componenti del pathname.

Il pathname è quindi l'identificazione della posizione di un file o di una directory all'interno della struttura ad albero del file system.

#### 1.3.5.1 - Pathnames Assoluti e Relativi

Il pathname si dice assoluto quando, indipendentemente dalla directory corrente, esso mostra l'intero cammino per l'identificazione del file partendo dalla radice; diversamente si dice relativo quando tiene conto della directory corrente. Poniamo di trovarci nella directory

```
/utenti/programmi
```

e di voler copiare il file *prova.prg* nella directory

```
/software/utility/prog
```

useremo un pathname relativo impartendo il comando

```
cp prova.prg /software/utility/prog
```

useremo un pathname assoluto invece con

```
cp /utenti/programmi/prova.prg /software/utility/prog
```

Se invece ci fossimo trovati nella directory

```
/software/utility/prog
```

avremmo fatto

```
cp /utenti/programmi/prova.prg .
```

dove il punto sta ad indicare la directory corrente.



## 2 - UTILIZZO DI FILE E DIRECTORY, COMANDI PRINCIPALI

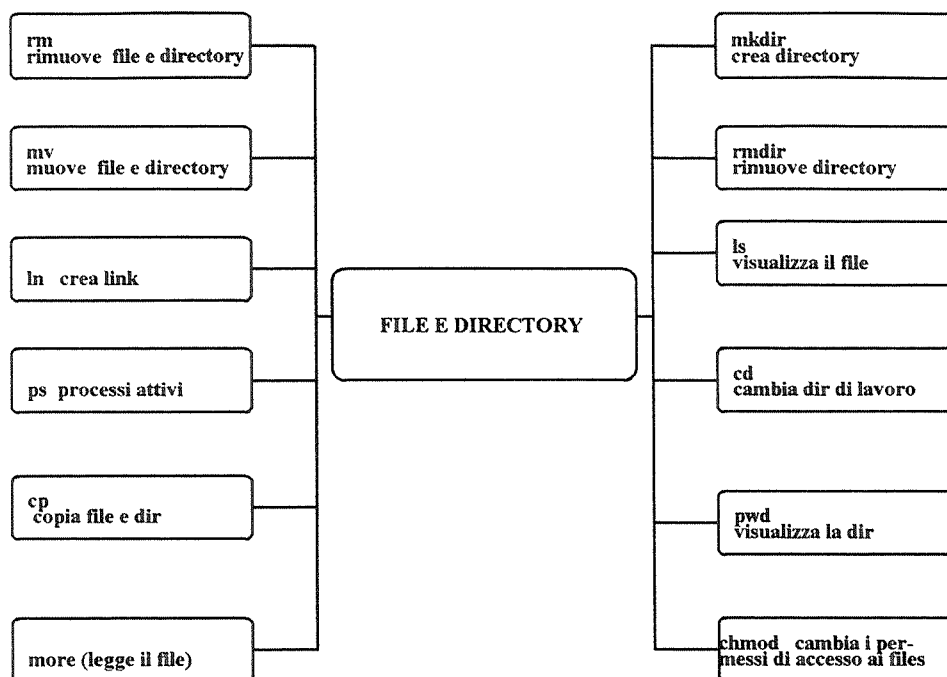


Fig. 3 Alcuni dei principali comandi

### 2.1 - Proprietà dei File

Oltre ad avere un nome ed un contenuto, un file di UNIX ha altre importanti proprietà che si possono esaminare con il comando *ls*.

Come la maggior parte dei comandi, *ls* ha delle opzioni che permettono di modificarne il comportamento di default<sup>15</sup>. Le opzioni seguono il nome del comando e sono normalmente precedute dal segno "-" (meno).

Esempio:

```
ls
DXMail  DXterm  Mail  bin  ds  tmp
```

<sup>15</sup> Mancanza. Assunzione di un determinato tipo o valore per un parametro in caso di mancata specificazione dello stesso



```
ls -l
-rw-r--r--  archery  gis  375  Aug 30 08:53 DXMail
-rw-r--r--  archery  gis  5547 May 6 11:02 DXterm
drwx----- archery  gis  512  Aug 30 08:53 Mail
drwxr-xr-x  archery  gis  512  Apr 7 16:39 bin
-rw-----  dave     users 367  Aug 12 13:48 ds
drwxr-xr-x  archery  gis  512  Jun 29 11:57 tmp

permessi  utente      gruppo  dimensione  data  ora  nome file
```

Il comando *ls* non fa vedere i files con il nome che inizia con un punto<sup>16</sup>. Per vedere anche questi files utilizzare l'opzione *-a*.

Più in generale un comando è seguito da due tipi di stringhe:

OPZIONI che ne modificano il comportamento e sono solitamente precedute dal segno *-* (meno).

PARAMETRI che indicano su cosa deve operare il comando (il file *ds* nell'esempio che segue)

```
ls -l ds
-rw-----  dave users 367 Aug 12 13:48 ds
```

## 2.2 - Creare/editare un File

Normalmente, per creare un file, si possono usare alcuni editor (come *vi*, *emacs*,...), oppure si può utilizzare il comando per ridirigere l'output<sup>17</sup> (il carattere ">") insieme a *cat*.

L'editore *vi* è normalmente presente in ogni sistema UNIX. Se per esempio si volesse creare il file *prova* supponendo che non esista, faremo:

```
vi prova
```

A questo punto ogni riga del video contiene a colonna 1 il carattere *~*, l'ultima riga in basso contiene *prova* [new file] ed il cursore viene posizionato all'inizio della prima riga.

Battere il carattere *i* (che non viene scritto sul terminale) per entrare in modalità di input. Tutto quello che viene scritto di qui in poi farà parte del file.

Battere *esc* per uscire dalla modalità di input e scrivere *:wq* per salvare il file ed uscire da *vi*, oppure *:q!* per uscire senza salvare il lavoro.

Per una trattazione esaustiva di *vi* si rimanda al comando *man* o ad un manuale Unix. Nel Sistema Operativo DEC OSF/1 è compreso un editor piuttosto semplice e di immediata comprensione e se ne consiglia l'uso.

<sup>16</sup> E' una convenzione Unix di non mostrare i file che iniziano con un punto, in quanto a questa categoria appartengono alcuni file che è opportuno rimangano più nascosti possibile all'utente generico.

<sup>17</sup> Il comando *cat* mostra il contenuto di un file ed ha, quindi, un suo dispositivo di uscita (output) di default, lo schermo. L'operazione di cambiare il dispositivo di uscita si dice redirezione.

## 2.3 - Creare/concatenare e visualizzare un File

Il comando *cat* permette di visualizzare sul terminale il contenuto dei files specificati come argomento e di concatenare più files.

Creiamo un file di nome uno contenente la stringa ciao, terminiamo l'input con la combinazione di tasti CTRL-C

```
cat > uno  
ciao  
CTRL-C
```

Creiamo un file di nome due contenente la stringa a tutti

```
cat > due  
a tutti  
CTRL-C
```

Concateniamo ora i due files in un terzo di nome tre

```
cat uno due > tre
```

Esaminiamo il contenuto di *tre*

```
cat tre  
ciao  
a tutti
```

Per visualizzare files più lunghi si può usare il comando *more* che propone l'output a schermate ponendo sull'ultima riga dello schermo la scritta *more*; premendo la barra spaziatrice si visualizzerà la schermata successiva.

## 2.4 - Cancellare, spostare e copiare un file

```
rm [-ir] file...
```

Cancella il file dalla directory: il contenuto del file verrà irrevocabilmente perso.

L'opzione *-r*, (ricorsiva) cancella i contenuti delle directory specificate (e delle relative sotto directory), nonché la directory stessa (da usare con molta attenzione).

L'opzione *-i* fa sì che venga chiesta conferma prima della cancellazione.

```
mv [-i] file1 file2  
mv [-i] file directory
```

Cambia nome a file1 con file2, oppure lo sposta in un'altra directory.

```
cp [-i] [-r] file1 file2
cp [-i] [-r] file directory
```

Copia il file1 sul file2 nella stessa o in un'altra directory

## 2.5 - Creare e cambiare directory

Il comando *mkdir* crea una nuova directory

Il comando *rmdir* cancella una directory purché sia vuota

Il comando *cd* permette di cambiare la working directory

Esempio

```
pwd
/home/corso
```

```
mkdir prog
```

```
pwd
/home/corso
```

```
cd prog
pwd
/home/corso/prog
```

```
cd ..
pwd
/home/corso
```

Nell'esempio precedente si crea una nuova directory con *mkdir* poi, con *cd*, ci spostiamo in essa e, successivamente, ritorniamo alla sottostante.

I caratteri ".." indicano sempre la directory padre, ovvero la directory che è di un livello più vicina alla radice.

Il carattere "." indica sempre la directory corrente.

### 3 - ACCESSO AI FILE

Ricordando che Unix è un sistema operativo multiutente, occorrerà allora che esso dia la possibilità ad ogni utilizzatore di proteggersi da intrusioni o di condividere l'uso di file. Ogni file dispone, quindi, di informazioni di controllo che indicano chi è autorizzato ad accedere al file. Esistono 3 tipi di autorizzazioni:

- per il proprietario del file
- per gli appartenenti allo stesso gruppo del proprietario
- per tutti gli altri

Ogni autorizzazione è composta da tre bit che indicano:

- l'autorizzazione a leggere il file
- l'autorizzazione a scrivere
- l'autorizzazione ad eseguire (se il file è una directory indica la possibilità di visualizzarne il contenuto)

Vediamo come `ls -l` ci mostra le autorizzazioni del file `ds` :

```
ls -l ds
-rw-----   dave users 367 Aug 12 13:48 ds
```

Il primo carattere indica il tipo di file e può avere diversi valori, ad uso degli utenti OSF/1 indichiamo i seguenti:

- indica un file normale
- d** indica una directory
- l** indica un link simbolico

I successivi 9 caratteri sono divisi in tre gruppi di 3 caratteri ciascuno. Il primo gruppo di 3 indica i permessi del proprietario. I successivi 3 caratteri indicano i permessi degli altri utenti del gruppo e gli ultimi 3 indicano i permessi degli altri utenti.

Il comando *chmod* permette di modificare le autorizzazioni dei files

Tipo	Proprietario (u)			Gruppo (g)			Altri (o)		
1	2	3	4	5	6	7	8	9	10

1 - mostra le caratteristiche del file: "-" "d" "l"

2, 3, 4; 5, 6, 7; 8, 9, 10; visualizzano rispettivamente il modo di accesso per proprietario, gruppo e altri utenti. Con questa serie di triplette, si ha la possibilità rispettivamente di leggere, scrivere ed eseguire un programma.

Gli accessi sono indicati come di seguito:

- r il file è leggibile
- w il file può essere scritto
- x il file può essere eseguito
- accesso negato

e fanno riferimento a :

- u utente proprietario (user)
- g gruppo di appartenenza del proprietario (group)
- o altri (other)

```
ls -l ds
-rw----- dave users 367 Aug 12 13:48 ds
```

```
chmod g+r ds
```

```
ls -l ds
-rw-r----- dave users 367 Aug 12 13:48 ds
```

Ora anche gli altri utenti del gruppo users potranno leggere il file ds.

## 4 - ALCUNI ALTRI COMANDI

*whoami*

Visualizza il nome dell'utente.

*groups*

Mostra il gruppo di appartenenza dell'utente (può essere anche più di uno).

*users*

Mostra una lista degli utenti attivi al momento. Lo stesso risultato si ottiene digitando semplicemente *u* o *w*; quest'ultimo comando offre anche maggiori informazioni.

*who*

Propone una lista: dei nomi dell'operatore, del terminale e dell'orario di inizio di ogni sessione Unix.

*ps*

Visualizza alcune informazioni relative ai processi; dispone di molte opzioni.

*wc*

Conta il numero di righe, parole e caratteri in uno o più file.

*grep*

Ricerca in uno o più file le linee che contengono un pattern.

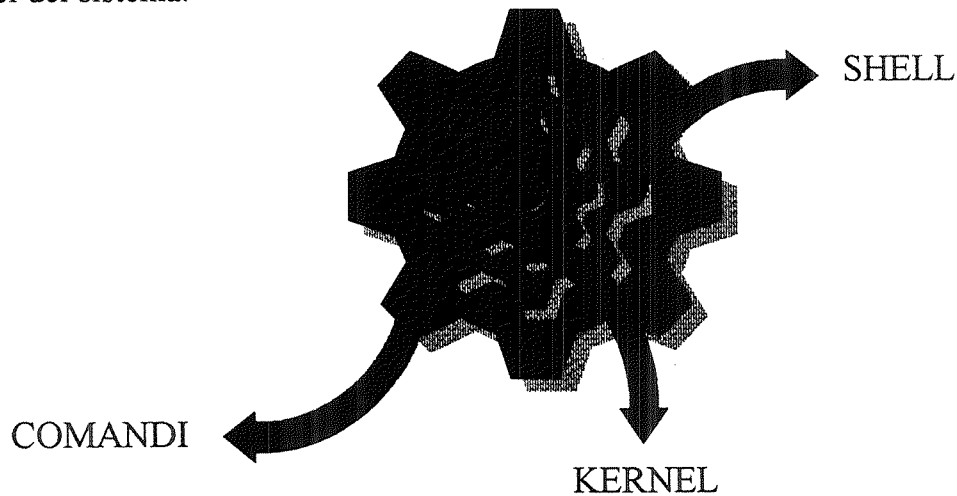
L'opzione *-v* rovescia il senso della ricerca; ossia il risultato saranno le righe che non contengono il pattern.



## 5 - LA SHELL

La Shell è un "interprete di comandi", cioè è il programma di interfacciamento tra l'utente ed il sistema, in grado di interpretare le richieste dell'utente e di avviarne l'esecuzione.

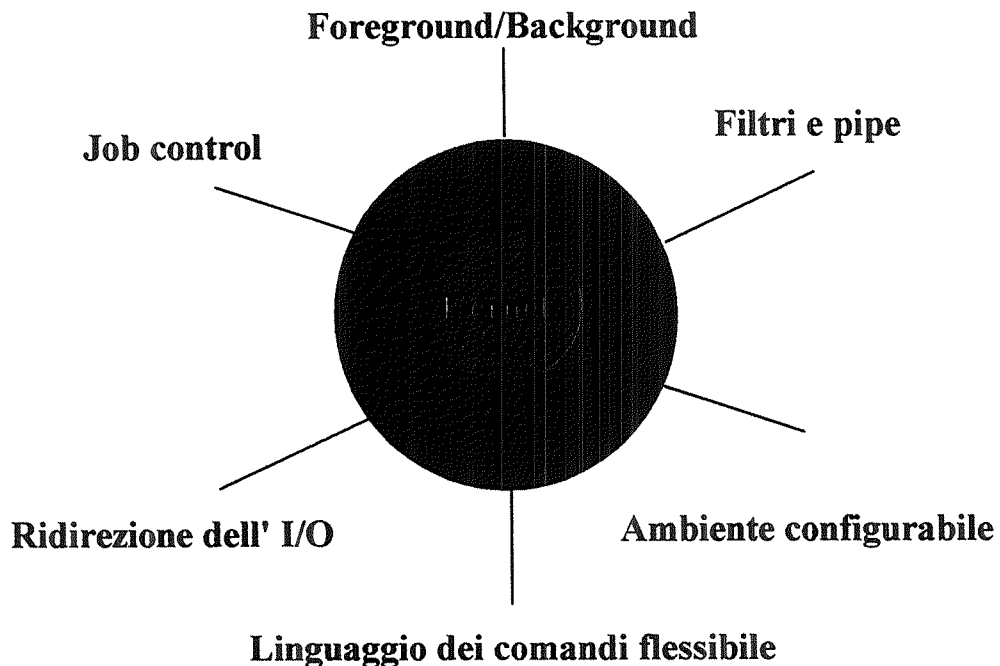
Si può pensare la Shell come uno "strato" di software collocato tra l'utente e il kernel del sistema.



*Fig.4 : Localizzazione della Shell*

Vi sono diverse Shell, le più diffuse sono:

**C-Shell      Bourne-Shell      Korn-Shell**



*Fig.5 : Servizi offerti dalla Shell*





## 6 - CONTROLLO DEI PROCESSI

Unix è un sistema operativo time-sharing, all'interno del quale i processi competono per le risorse del sistema. Ad ogni processo viene allocata la Central Processing Unit (CPU) per un periodo di tempo prefissato, detto time slice. Al completamento della slice, od in caso di blocco per una richiesta di I/O, la CPU viene allocata ad un altro processo pronto. I criteri di selezione del processo sono basati sulla priorità e lo stato del processo. I processi possono operare in foreground o in background.

La Shell esegue solo un comando in foreground e attende che esso sia terminato prima di eseguirne altri.

La Shell non attende invece, che il comando lanciato in background sia terminato, e ritorna subito il prompt permettendo così di lanciare altri comandi mentre esegue i precedenti.

Sia che siano in background che in foreground ci sono diverse soluzioni per ripetere in toto od in parte i comandi precedenti, vediamo alcuni :

Chiave	Funzione
!!	il comando precedente
!n	l'n-esimo comando
!-n	l'n-esimo comando a ritroso.
! <i>string</i>	il più recente comando che comincia con <i>string</i>
!? <i>string</i>	il più recente comando che contiene <i>string</i>
^ <i>prima</i> ^ <i>dopo</i>	sostituisce <i>prima</i> con <i>dopo</i> nel comando precedente

E' infine possibile definire una serie di *alias* per i comandi più lunghi o ricorrenti, o per aumentare la sicurezza dei dati; l'esempio più tipico di quest'ultima possibilità è quello di definire un alias per *rm*, (*alias rm rm -i*) facendo sì che quando si cancellano dei files venga automaticamente richiesta la conferma.

## Guida operativa

<i>ls</i>	elenca i nomi dei files nella directory corrente
<i>ls filename</i>	elenca i files dati come argomento
<i>ls -t</i>	elenca in ordine di data: prima il più recente
<i>ls -l</i>	elenco dettagliato
<i>ls -u</i>	elenco in ordine di accesso
<i>ls -r</i>	elenco in ordine inverso
<i>vi filename</i>	edit del file
<i>cp file1 file2</i>	copia <i>file1</i> su <i>file2</i> , sostituisce <i>file2</i> se esiste già
<i>mv file1 file2</i>	muove o rinomina <i>file1</i> su <i>file</i> , sostituisce <i>file2</i> se esiste già
<i>rm filename</i>	rimuove i files dati come argomento
<i>more filename</i>	visualizza il file
<i>cat filename</i>	visualizza i files dati come argomento
<i>cd dirname</i>	cambia directory
<i>pwd</i>	visualizza la directory corrente
<i>ps</i>	mostra i processi attivi
<i>lpr -PprinterID filename</i>	stampa il file sulla stampante specificata <sup>18</sup>
<i>wc filename</i>	conta le righe, le parole e i caratteri per ogni file
<i>wc -l filename</i>	conta le linee per ogni file
<i>grep pattern filename</i>	mostra ogni riga in cui compare pattern
<i>grep -v pattern filename</i>	mostra ogni riga in cui non compare pattern
<i>sort filename</i>	ordina le righe del file in argomento
<i>tail filename</i>	mostra le ultime 10 righe del file
<i>tail -n filename</i>	mostra le ultime n righe del file
<i>tail +n filename</i>	mostra il file a partire dalla riga n
<i>cmp file1 file2</i>	mostra la posizione della prima differenza tra i files
<i>diff file1 file2</i>	mostra tutte le differenze tra i files
<i>man</i>	manuale

<b>BACKSPACE</b>	Cancella l'ultimo carattere sulla linea di comando
<b>CTRL - W</b>	Cancella l'ultima parola sulla linea di comando
<b>CTRL - U</b>	Cancella l'intera linea di comando
<b>CTRL - D</b>	Carattere usato anche per terminare l'input di un file
<b>CTRL - S</b>	Sospende momentaneamente l'esecuzione di un programma
<b>CTRL - Q</b>	Riprende l'esecuzione del programma interrotto con CTRL - S
<b>CTRL - C</b>	Interrompe un programma in esecuzione

<b>!!</b>	il comando precedente
<b>!n</b>	l'n-esimo comando
<b>!-n</b>	l'n-esimo comando a ritroso,
<b>! string</b>	il più recente comando che comincia con string
<b>!?string</b>	il più recente comando che contiene string
<b>^prima^dopo</b>	sostituisce <i>prima</i> con <i>dopo</i> nel comando precedente

18 La classica operazione di stampa dell'utente del Sistema Informativo Territoriale dell'Autorità sarà quella per la produzione di una carta (sul plotter, supponiamo dei limiti di Bacino e che il file di plottaggio si chiami *lim\_adb.color*), egli dunque userà il seguente comando : *lpr -Pccp981 lim\_adb.color*.