

ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Bologna

INFN/TC-94/25
24 Novembre 1994

A. Bassi, M.L. Luvisetto, E. Ugolini:

ULTRIX: GUIDA ALLA GESTIONE DI SISTEMI COMPLESSI

ULTRIX: GUIDA ALLA GESTIONE DI SISTEMI COMPLESSI

A. Bassi, M.L. Luvisetto, E. Ugolini
INFN-Sezione di Bologna, Via Imerio, 46, I-40126 Bologna, Italy

Abstract

Il presente articolo è concepito come guida alla gestione del sistema operativo Ultrix. Le conoscenze di Unix richieste per la comprensione del materiale sono a livello utente. Sebbene quanto detto si applichi a qualsiasi macchina RISC Ultrix e solo in parte ad altre CPU gestite da Ultrix, l'hardware considerato è la DECstation 5000. L'informazione tratta sistemi operativi stand-alone e sistemi client-server con NFS, sia per quanto riguarda l'installazione sia per quanto riguarda la gestione, il backup, la soluzione dei problemi più frequenti incontrati, la manutenzione dei file system.

1 Introduzione

Lo sviluppo di macchine con architetture molto diversificate che vanno da singola a multipla CPU e specialmente le nuove architetture RISC hanno portato ad uno sviluppo del sistema Unix con *flavours* diversi pur rispettando il concetto basilare del sistema. Dato l'alto rapporto prestazioni/prezzo, la diffusione di tali macchine è in continuo aumento rendendo necessarie cognizioni non solo d'uso ma anche di gestione del sistema operativo.

Il presente articolo parte dal presupposto che il lettore abbia una conoscenza precedente di Unix in generale (i.e. sintassi nome file, directory, protezioni, etc.) e debba affrontare l'installazione e la manutenzione del software di sistema in un ambiente di rete.

Viene fatto un breve riferimento all'hardware per quanto riguarda il boot della macchina e il settaggio dei relativi parametri.

I marchi registrati citati sono referenziati in appendice.

2 GENERALITÀ

2 GENERALITÀ

Nel presente paragrafo verranno indicate le nozioni elementari di Unix (e quindi di Ultrix) necessarie alla comprensione di quanto segue. Tutte le nozioni vengono fornite in modo conciso e con una sequenza che rispecchia i passi che si devono compiere su di una macchina appena installata e priva di software.

In ogni sistema operativo esistono utenti privilegiati che devono curare l'installazione e la manutenzione del sistema. In Unix il sistemista si chiama **administrator**, ha come nome di login **root**, è operativo quando il sistema viene portato in modalità single user o quando un utente che conosca la password di root diventa **superuser** con il comando **su**.

Il compito principale del sistemista e anche quello più complesso è la corretta gestione dei dischi, dato che ogni disco viene suddiviso in partizioni e la scelta delle dimensioni deve essere progettata con cura per un uso ottimale del disco in vista degli aggiornamenti del sistema e del software. Inoltre la variazione della dimensione delle partizioni implica una distruzione parziale o totale del contenuto del disco.

In Unix l'hardware è nascosto all'utente e le varie parti del sistema vengono viste come **file system**, dove un file system ha una directory di radice (la radice principale è /) e può risiedere su una partizione locale o remota senza che l'utente sia costretto a conoscerne i particolari.

Il sistema Unix poggia sul concetto **client-server** che gestisce le funzioni di accesso a dischi remoti e di boot per macchine diskless. La macchina che fornisce il servizio di boot (**server**) possiede il proprio sistema operativo e uno o più sistemi operativi per le macchine **client** che fanno il boot via Ethernet.

Il sistema è formato da un **kernel** di cui possono esistere più versioni che vengono invocate al boot. Inoltre la macchina può lavorare, come già accennato, in modalità single user con il solo **su** attivo da console, oppure in modo multiuser, con selezione del modo a livello di boot, o accesso a livello di halt (shutdown).

In generale, ogni utente ha accesso alla maggior parte dei comandi e dei file di sistema. È quindi buona norma non fare **mai** login come root ma lavorare come utente generico usando **su** solo quando strettamente necessario, tenendo sempre presente che non esistono versioni multiple dei file.

Nella documentazione che segue i comandi sono sempre preceduti dal prompt che ne denota l'ambiente e precisamente:

```
>>      firmware
#       superuser
$       Bourne shell
%       C-shell
!       comando di shell da vi
```

3 HARDWARE DECSTATION 5000

3 Hardware DECstation 5000

3.1 Firmware

La macchina è dotata di firmware che consente di controllare la configurazione, eseguire test e stabilire l'ambiente di boot, cioè specificare quali azioni preferenziali devono essere eseguite alla partenza e al fermo della macchina.

Nelle note seguenti sono illustrati in maniera sintetica i comandi che si possono dare alla macchina con il sistema in halt e cioè quando sul video compare il prompt >>.

Per determinare i controllers e i device configurati sul sistema:

```
>> cnfg
```

Vengono listati dati del tipo:

```
7:KNOZ-AA      . . . . .      (16 MB)
6:PMAD-AA      . . . . .      (enet:08)
5:PMAZ-AA      . . . . .      (SCSI=7)
```

La prima colonna indica i numeri di slot dei controllers che generalmente differiscono in ogni configurazione. In ogni caso quando si fa il boot da un device logico con il numero 0 (default), lo slot di default del boot device SCSI è 3 nei modelli 5000/240, mentre in altri modelli è 5, analogamente il boot di rete è 3 (5000/240) oppure 6 (altri modelli).

Per vedere il boot device (e i device SCSI):

```
>> cnfg 3
```

Per visualizzare le variabili di console:

```
>> printenv
```

Per settare il boot path, cioè da quale disco e sistema deve partire la macchina:

```
>> setenv boot slot_number/device_name_number/unix_kernel [-a]
```

e cioè (ad esempio):

```
>> setenv boot rz(0,1,0)vmunix
```

dove:

0 = SCSI controller (in generale 3 oppure 5)

1 = unita' disco

0 = partizione

in particolare:

```
>> setenv boot "3/rz1/vmunix -a   ambiente multiuser - (3 oppure 5)
>> setenv boot "3/rz1/vmunix     ambiente single user - (3 oppure 5)
>> setenv boot "3/mop -a         multiuser boot dalla rete - (3 oppure 6)
```

3 HARDWARE DECSTATION 5000

Le variabili di `setenv` sono visualizzate dal comando `printenv` indicato sopra. Se si commette un errore battendo il comando `setenv`, l'elemento errato rimane visibile con il comando `printenv`, per eliminarlo usare `unsetenv`.

A seconda del tipo di ripartenza che si desidera all'accensione, occorre settare l'azione di `halt` in uno dei modi indicati.

Per boot automatico:	>> <code>setenev haltaction b</code>
Per boot manuale:	>> <code>setenev haltaction h</code>
Per restart forzato con dump della memoria:	>> <code>setenev haltaction r</code>

Avendo settato l'ambiente come indicato sopra, il comando di boot potrà avere una delle forme seguenti.

Boot da disco di sistema:	>> <code>boot</code>
Boot da disco alternativo:	>> <code>boot 3/rz5/vmunix.new -a</code> dove <code>vmunix.new</code> indica il nuovo file di sistema (default <code>vmunix</code>).
Boot da tk50:	>> <code>boot 5/tz5</code>
Boot da CDR0M:	>> <code>boot 1/rz4/vmunix -a</code> dove 4 indica lo slot del CDR0M unit RRD40
Per boot single user:	>> <code>boot -s</code>
Per boot multi user:	>> <code>auto</code>
Per boot alternativo:	
>> <code>boot -s -f rz(0,#,0)vmunix.new</code>	(single user)
>> <code>boot -f rz(0,#,0)vmunix.new</code>	(multi user)

Per maggiori informazioni sull'uso di console:

>> `help`

oppure:

>> `?`

Per i test dello SCSI:

>> `test 3`

Per una panoramica completa del modo console, consultare il manuale *Guida all'utilizzo - EK-366PA-OP.001*.

3.2 FIRMWARE PER DS 5000/240

Le DECstation modello 240, cioè con processore MIPS con clock a 40 MH, hanno il bus SCSI sull'indirizzo 3 invece che su 5, perciò si dovrà correggere il `setenv` per il boot specificando l'indirizzo 3 invece che 5, ad esempio:

>> `setenv boot "3/rz2/vmunix" -a`

3 HARDWARE DECSTATION 5000

Analogamente nei test hardware si dovrà indicare **test 3** invece di **test 5**. Per il boot da rete si deve specificare **3** invece di **6**.

Nel caso in cui si operi un upgrade da 5000/200 a 5000/240 si deve creare un nuovo kernel in cui compaiono le seguenti modifiche e aggiunte.

Il tipo di cpu è definito come:

```
DS5000-300
```

Viene inserito l'indirizzo SCSI 3 con la definizione di adapter che segue:

```
adapter      ibus3      at nexus?
```

Viene corretto il tipo di interfaccia SCSI modificando il device dc0 come segue:

```
device      scc0      at ibus?      vector sccintr
```

3.3 ERRORI HARDWARE

Per verificare gli errori hardware :

```
>> test
```

Vengono eseguiti tutti i test dell'hardware. Se qualche elemento dà errore, ad esempio l'unità Floating Point, viene scritto un messaggio del tipo:

```
? TFL: 3/fpu (5) [KN03-AA]
```

dove:

```
[KN03-AA]
```

è il codice del modulo hardware difettoso.

Al termine dei test, se si è verificato qualche errore, sul video compare un sommario degli errori.

3.4 INDIRIZZI SCSI

Gli indirizzi SCSI vengono visti da Ultrix nel range 0:7, per cui un disco di tipo RZ con indirizzo SCSI 6, verrà visto da Ultrix come rz6.

Se la macchina è dotata di due controller SCSI, ad esempio CPU VAX con SCSI A e B, che a livello VMS vengono indicati come DKA100, DKA200, etc. per lo SCSI A e come DKB100, DKB200, etc. per lo SCSI B, i device sul primo SCSI (i.e. SCSI A) in Ultrix hanno indirizzo 0:7, i device sul secondo SCSI (i.e. SCSI B) in Ultrix hanno indirizzo 8:15, quindi per creare il device si dovrà usare :

```
# MAKEDEV rz8          (disco)
# MAKEDEV tz11         (tape)
```

3 HARDWARE DECSTATION 5000

Per i dischi si indica il parametro `rz` seguito dall'indirizzo SCSI del disco. Il sistema crea nella directory `/dev` i file `rrz8a`, `rz8a`, `rrz8b`, `rz8b`, etc. dove `8` è l'indirizzo SCSI e `a`, `b`, `c`, etc. sono le partizioni del disco.

Per i nastri si indica il parametro `tz` seguito dal relativo indirizzo SCSI. Il sistema crea nella directory `/dev` i file `nrmt0a nrmt0h nrmt0l nrmt0m rmt0a rmt0h rmt0l rmt0m` dove `a`, `h`, etc. indicano le pseudo partizioni (si usa solo `h` nelle operazioni di accesso ai nastri), il prefisso `n` indica no-rewind, e il numero `0` indica la prima unità nastro collegata, quindi nel caso di nastri, contrariamente a quanto avviene per i dischi, i file creati da `MAKDEV` non sono direttamente connessi all'indirizzo SCSI.

3.5 SHUTDOWN E REBOOT

Le operazioni indicate nel paragrafo precedente si eseguono da console quando non è attivo il sistema operativo. Nel caso in cui la macchina sia funzionante, il `su` può eseguire operazioni di spegnimento e di accensione, di start e di stop secondo le necessità, come descritto di seguito. Come già indicato, il sistema Ultrix può lavorare in modo *single user* cioè come utente privilegiato `su` con accesso da console solamente, oppure in modo multiuser, cioè qualsiasi utente può fare login da terminali generici. Per operazioni di manutenzione si deve fare lo shutdown del sistema da multiuser a single user e cioè:

```
# /etc/shutdown +15 "to install new dev"  
# /etc/shutdown 13:00
```

Nel messaggio di shutdown si possono usare indifferentemente " oppure '. Il sistema scrive il numero di processo dell'operazione che può perciò essere interrotta. Il motivo di shutdown viene registrato nel file `/usr/adm/shutdownlog`. In questo caso viene creato il file `/etc/nologin` che impedisce il login degli utenti. Tale file viene cancellato da `/etc/rc` al reboot.

Comandi per fermare il processore, con e senza reboot:

```
Per fermare il processore:      # /etc/halt  
Shutdown e halt:               # /etc/shutdown -h +10 'maintenance'  
Shutdown e reboot:            # /etc/shutdown -r +20 'upgrade'  
Per shutdown e stop immediato: # shutdown -h now  
Per stop e reboot immediato:  # shutdown -r now
```

4 SISTEMA OPERATIVO - INTERNALS

4 SISTEMA OPERATIVO - INTERNALS

Il sistemista deve essere in grado di installare e configurare il software, gestire gli utenti ed eventualmente le quote, garantire l'integrità dei dischi eseguendo periodici backup, aggiornare il sistema operativo, installare e gestire prodotti software, analizzare il comportamento delle macchine e intervenire in presenza di degrado delle prestazioni.

Per poter assolvere a tali compiti, il sistemista deve avere una conoscenza di base sul modo in cui opera Ultrix e sui processi che vengono attivati nei vari stadi.

Il sistemista deve familiarizzarsi con la struttura di directory del sistema, tenendo presente che la radice è / e che i file di gestione si trovano sotto /etc. Nel seguito i file verranno indicati con il path completo e in appendice sono riassunti i file e le directory principali.

Il sistema esegue le istruzioni del file /etc/rc che viene autogenerato e del file di startup /etc/rc.local che contiene le personalizzazioni locali.

Tenere presente che tutte le operazioni temporanee vengono eseguite in /tmp e che tale area viene ripulita ad ogni partenza. È quindi essenziale che il file system su cui risiede /tmp sia dimensionato in modo adeguato alle esigenze del sistema e al numero di utenti attivi contemporaneamente.

4.1 BOOT

Il boot avviene con il caricamento di un kernel, di solito /vmunix, alla locazione zero della memoria. I processi di gestione vengono creati all'inizio del boot con identificazione PPID=0 oppure alla fine dal processo init con PPID=1 (dove PPID è il parent process ID). Tali processi sono ad esempio swapper, pageout, cron, lpd, init. I processi attivi sono visualizzati con:

```
$ ps -aux
```

/etc/init viene invocato come ultimo step della procedura di boot. Viene poi eseguita la procedura di reboot in base al cui esito init inizia l'operazione del sistema in modo *multiuser* o *single user*.

Durante il boot viene invocato il comando /etc/rc con l'argomento *autoboot* che controlla la seguente sequenza di eventi:

1. gira /etc/fsck per controllare il file system
2. fa partire i daemons, come:
 - /etc/cron per la schedulazione dei processi dichiarati in /etc/crontab
 - /etc/update per la scrittura forzata su disco ogni 30 secondi
3. conserva i files di editor
4. ripulisce la directory temporanea /tmp
5. esegue i comandi di /etc/rc.local
6. esegue /etc/init e /etc/getty

4 SISTEMA OPERATIVO - INTERNALS

4.2 LOGIN

Al login hanno luogo i seguenti steps:

- `/etc/init`
se c'è un'entry in `/etc/ttys`, viene creato un processo tramite `/etc/getty`
- `/etc/getty`
esegue login al terminale `ttyn`
scrive la versione del sistema indicata in `/etc/gettytab`
verifica il settaggio del terminale
esegue `login username`
- `login username`
richiede `passwd`
controlla i parametri di processo
esegue lo shell indicato in `/etc/passwd`

4.3 PAGING

Il processo di paging è analogo a quello del VMS, cioè si basa sulle pagine residenti e sulla memoria meno usata per lo swap. La verifica delle pagine da conservare in memoria viene fatta tramite un algoritmo di clock. Lo swap è determinato dalle pagine libere ma l'intervento del clock non può mai avvenire prima che sia trascorso un intervallo di tempo determinato.

Il clock punta sequenzialmente a tutte le pagine di memoria non del kernel. Le pagine che si trovano nel set residente sono dette valide. Se la pagina è valida, viene marcata invalida e fuori del set residente. Se la pagina è invalida e non è stata modificata mentre era nel set residente, viene posta nella lista delle pagine libere. Se è stata modificata, viene scritta nello spazio di swap prima di essere posta fra le pagine libere. Al prossimo ciclo di clock le pagine della lista vengono eliminate (pageout). Se la lista delle pagine libere è troppo piccola o la frequenza di paginazione è troppo elevata, le pagine dei processi a priorità minore vengono scritte su disco sul file di swap, cominciando dai processi inattivi da più di 20 secondi e proseguendo con quelli residenti dal tempo più lungo.

5 FILE SYSTEM

5 FILE SYSTEM

Sotto Unix, i dischi sono suddivisi in aree dette partizioni secondo tabelle predefinite, oppure secondo definizioni assegnate dal sistemista in accordo ad esigenze specifiche. Se il sistema deve essere creato per la prima volta e non si vogliono usare le partizioni predefinite, dopo aver caricato il kernel da TK50, si sceglie dal menu di installazione la voce *system management* e si prosegue come indicato di seguito.

Le dimensioni delle partizioni sono genericamente indicate in settori (blocchi da 512 bytes). Il disco è divisibile in partizioni secondo lo schema:

```

          [ Partizione a                (sistema)
          [ Partizione b                (page-swap-dump area)
          [                               [ Partizione d
Partizione c [ Partizione g            [ Partizione e
          [                               [ Partizione f
          [ Partizione h
```

Se esiste la partizione *h*, le sottopartizioni *d*, *e*, *f*, si sovrappongono ad *h* invece che a *g*. Devono sempre esistere o la partizione *a* o la partizione *c* (tutto il disco), per questo motivo la partizione *a* di default è molto piccola e inadeguata per il sistema operativo. La partizione di default del sistema è la *a*, mentre la *b* è quella destinata al paging. Le partizioni vengono create con *chpt* e formattate con *newfs* in base alle dimensioni indicate in */etc/disktab*. Ad esempio per un disco rz57, si ha:

```
# "@(#)disktab 3.1.1.1 (ULTRIX) 8/1/90"
#
#      disktab from 4.5 4.2 BSD 83/07/30
# Disk geometry and partition layout tables.
# Key:
#      ty      type of disk
#      ns      #sectors/track
#      nt      #tracks/cylinder
#      nc      #cylinders/disk
#      p[a-h]  partition sizes in sectors
#      b[a-h]  partition block sizes in bytes
#      f[a-h]  partition fragment sizes in bytes
#
# All partition sizes contain space for bad sector tables unless
# the device drivers fail to support this.
```

```
rz57|RZ57|DEC RZ57 Winchester:\
      :ty=winchester:ns#71:nt#15:nc#1925:\
      :pa#32768:ba#8192:fa#1024:\
      :pb#184320:bb#4096:fb#1024:\
      :pc#2025788:bc#8192:fc#1024:\
      :pd#299008:bd#8192:fd#1024:\
```

5 FILE SYSTEM

```
:pe#299008:be#8192:fe#1024:\
:pf#596284:bf#8192:ff#1024:\
:pg#614400:bg#8192:fg#1024:\
:ph#1194300:bh#8192:fh#1024:
```

Da osservare che il valore 1024 non indica la dimensione dei blocchi su cui fare il calcolo che è 512 ma indica il raggruppamento dei medesimi, quindi se una partizione non è di dimensioni multiple di 1024 si avrà una perdita di 512 bytes.

Le dimensioni delle partizioni possono essere verificate e modificate con *chpt* (change partition table). Sotto /dev esistono i device drivers che sono stati creati all'installazione del sistema in funzione della configurazione attuale oppure tramite MAKEDEV. Se il disco è di tipo rz ed è connesso al controller 1, i device driver per il disco sono /dev/rrz1a, /dev/rz1a, /dev/rrz1b, /dev/rz1b, etc. uno per ogni partizione. I device /dev/rz* vedono il disco come un *block device*, cioè un device che tratta l'informazione a blocchi, i device /dev/rrz* lo vedono come un *raw device* cioè un device a caratteri. Per visualizzare le informazioni di un disco, occorrono i privilegi di superuser e i comandi: df e chpt.

L'informazione fornita da df è del tipo seguente. Notare che con i dati forniti da df non è possibile risalire alla capacità del disco dato che partizioni non montate (es. la partizione b) non sono indicate.

```
# df
Filesystem      Total    kbytes    kbytes    %
node            kbytes    used      free      used  Mounted on
/dev/rz0a        15343   13416     393      97%   /
/dev/rz0g        227079 159491   44881    78%   /usr
/dev/rz1c        945726 375524  475630   44%   /usr/users
```

L'informazione fornita da chpt per un disco RZ57 (capacità 1Gb) è del tipo seguente. Dato che la partizione *c* è *sempre* tutto il disco in blocchi da 512 byte, dalla sua dimensione si risale alla capacità e quindi al tipo di disco relativo.

```
# chpt -q /dev/rrz1c
/dev/rrz1c
Current partition table:
partition      bottom      top         size        overlap
a              0           32767      32768       c
b             32768      217087     184320       c
c              0          1954049    1954050     a,b,d,e,f,g,h
d             831488     1130495    299008       c,h
e            1130496     1429503    299008       c,h
f            1429504     1954049    524546       c,h
g            217088      831487     614400       c
h            831488     1954049    1122562     c,d,e,f
#
```

5 FILE SYSTEM

Se si vogliono modificare le partizioni sul disco *rz1* prima di installare il sistema, in modo che la partizione *a* cresca di $32768*2=65536$ e la partizione *g* cresca di 204800 a spese della partizione *h*, si dovrà operare come segue:

```
# chpt -a /dev/rrz1a                #creazione superblock
#
(creazione partizione con:
                                v      = verbose
                                pa     = partizione a
                                0      = posizione iniziale partizione
                                98304 = dimensione partizione
)
#
# chpt -v -pa      0 98304 /dev/rrz1a
# chpt -v -pb 98304 184320 /dev/rrz1a
# chpt -v -pg 282624 819200 /dev/rrz1a
#
# chpt -v -ph 1101824 852225 /dev/rrz1a
```

Le partizioni *d*, *e* ed *f* potrebbero essere ridefinite come segue:

```
# chpt -v -pd 1101824 98304 /dev/rrz1a
# chpt -v -pe 1200128 184320 /dev/rrz1a
# chpt -v -pf 1384448 560601 /dev/rrz1a
```

e quindi potrebbero essere destinate a contenere una seconda copia del sistema operativo. Le dimensioni indicate nell'esempio sono state calcolate in base all'esperienza acquisita nella gestione del sistema e corrispondono ad esigenze reali.

N.B. Nel comando `chpt` si possono indicare sia la partizione *a* che la partizione *c*. Se il disco è nuovo le due partizioni sono equivalenti. Se il disco contiene già la partizione *a* e il file system relativo è montato (un caso classico è la modifica dello swap file di un disco sistema esistente), si deve specificare la partizione *c* altrimenti `chpt` fallisce con un messaggio d'errore.

Il file system è agibile dopo essere stato formattato e montato con:

```
# newfs -v /dev/partizione disk_type
# mount device directory
```

oppure:

```
# mount -a
```

per il montaggio dei dischi da `/etc/fstab`.

Il file system può essere verificato con `/etc/fsck`, operazione che viene sempre eseguita a reboot.

Se si dispone di altri dischi oltre al disco di sistema, una volta create le partizioni, il file system viene creato con i comandi:

5 FILE SYSTEM

```
# newfs -v /dev/rrz1x rz57
```

dove *x* indica la partizione interessata (es. *a*) e *rz57* indica il tipo di disco in uso per l'aggancio a `/etc/disktab`. Una volta create le partizioni, si deve avere la directory a cui agganciare il file system (ad es. `/usr/users`), se tale directory non esiste, crearla con `mkdir` nella root da cui dipende e fare il `mount`, come indicato nell'esempio:

```
# cd /usr
# mkdir users
# mkdir students
# mount /dev/rz1a /usr/students
# mount /dev/rz1g /usr/users
```

oppure inserire i nomi in `/etc/fstab` per il `mount` automatico al boot.

5.1 ERRORI DI FSCK

Se un disco è stato partizionato ma non formattato, l'operazione di `mount` dà errore e chiede di eseguire `fsck` manualmente. L'errore viene recuperato ricreando il file system con `newfs`.

Una situazione tipica si presenta, quando si vuole destinare la partizione *b* di un disco locale alle funzioni di page e swap, conservando la partizione *a* per usi generici, ad esempio:

```
partizione  a      grande (uso generico)
partizione  b      page - swap
```

In questo caso si creano le partizioni con `chpt` come indicato sopra, si dichiara la swap area con `dms` se il nodo è un cliente oppure editando il `config` file se il nodo è un server oppure è una macchina stand-alone. Anche se si usa `newfs` per creare il file system relativo alla partizione *a* subito dopo averla creata con `chpt`, questa viene corrotta al primo boot del nodo, quando il sistema crea il file di swap nella partizione *b*. Occorre quindi rifare:

```
# newfs /dev/rz1a
```

prima di poter fare il `mount`. Infatti se un file system non è creato correttamente con `newfs`, l'operazione di `mount` fallisce con il messaggio:

```
"bad argument for mount"
```

A conclusione del paragrafo, segue un esempio di partizioni per page e swap su dischi di varie dimensioni.

Partizioni per swap locale di CPU con 32 Mb su disco RZ56

partizione	bottom	top	size
a	0	971493	971494
b	971499	1299173	327680
c	0	1299173	1299174

5 FILE SYSTEM

Partizioni per swap locale di CPU con 32 Mb su disco RZ73

partizione	bottom	top	size
a	0	3252549	3252550
b	3252550	3907910	655361
c	0	3907910	3907911

5.2 ESEMPIO DI PARTIZIONI PER UN DISCO RZ58

Quando si aggiorna l'hardware di una macchina esistente ci si può trovare nella difficile situazione di avere un modello di disco non ancora rappresentato nella `/etc/disktab` del sistema corrente. In questo caso Ultrix consente di vedere il disco come un unico assieme ma non consente di partizionarlo. Per aggirare l'ostacolo è sufficiente procurarsi una versione del file `/etc/disktab` che preveda il disco in questione e procedere come già visto.

`/etc/disktab` è solo un file di modello per i dischi. Il valore vero delle possibili partizioni viene copiato sul disco stesso, le cui dimensioni e tipo sono stabilite a livello hardware, perciò non è possibile dichiarare un `rz58` come se fosse `rz57`, perchè Ultrix segnala un errore.

Volendo utilizzare il disco `rz58` da 1.3 Gb per il sistema operativo con ampio margine di spazio per applicativi e per evitare di saturare il file system, mantenendo inoltre una piccola partizione di riserva si può operare come indicato di seguito.

```
# chpt -q /dev/rrz2c
```

```
/dev/rrz2c
```

```
Current partition table:
```

partition	bottom	top	size	overlap
a	0	237567	237568	c,d,e,f
b	237568	475135	237568	c
c	0	2698060	2698061	a,b,d,e,f,g,h
d	0	0	0	a,c,e,f
e	0	0	0	a,c,d,f
f	0	0	0	a,c,d,e
g	475136	2285567	1810432	c
h	2285568	2698060	412493	c

Le partizioni create hanno indicativamente le seguenti dimensioni:

a = 115 Mb

b = 115 Mb

g = 880 Mb

h = 270 Mb

Si consiglia di montare ciascuna partizione dopo la creazione eventualmente come `/mnt` e verificare tramite `df` che i Kbytes totali indicati corrispondano veramente alla dimensione voluta.

5 FILE SYSTEM

5.3 DIRECTORY, FILE, LINK

L'informazione su disco è organizzata in *directory* e in *file*. Le *directory* contengono il nome del file, la sua lunghezza e il numero *inode*, cioè un numero che identifica il file e contiene informazioni amministrative quali numero di riferimenti (*link*), *owner*, permessi, dimensione, date di accesso, etc.

Quando il sistema è attivo la tabella di *inode* è gestita in memoria, perciò è necessario sincronizzare i dischi prima di spegnere la macchina, cioè richiedere la scrittura su disco della tabella *inode* aggiornata. Se tale operazione fallisce, si può correggere il file system usando `/etc/fsck` che pone un elenco dei file irraggiungibili nella *directory* `/lost+found`.

Dato che nella *directory* i file sono identificati da un pointer che indica la locazione su disco del file stesso, è possibile porre il pointer ad un file in più di una *directory*, perciò per condividere un file o per creare un accesso a files che non sono raggiungibili dal normale *path*, si creano dei *link* con il comando:

```
# cd /usr/local
# ln -s /cern/cn1201/bin/ypatchy ypatchy
```

In questo modo viene creato un *link* simbolico `ypatchy` nella *directory* `/usr/local` (che è nel *path* di ogni utente) che punta al file fisico `/cern/cn1201/bin/ypatchy` consentendo ad ogni utente l'accesso a `ypatchy` senza dover specificare il *path* completo.

Il *link* creato è visto dal sistema come se fosse un vero file e viene cancellato con il comando `rm` come per qualsiasi altro file. Il *link* può essere di 2 tipi: *hard* o *symbolic*. Se il *link* è di tipo *hard* il file fisico viene rimosso dal sistema solo quando sono stati cancellati tutti i *link* al medesimo. La creazione di un *link* *simbolico* con lo stesso nome di un file fisico nella stessa *directory* determina la cancellazione del file fisico e crea un giro vizioso di *link* che non può essere risolto.

6 INSTALLAZIONE DEL SISTEMA

6 INSTALLAZIONE DEL SISTEMA

Per l'installazione si deve pianificare lo spazio disco in modo che ci sia spazio di swap pari come *minimo assoluto* a 3 volte la memoria, ad es. con 24 MB ci vogliono 72 MB di swap e 20 MB di dump sulla partizione *b* e 20 MB di dump sulla partizione *g* per la *var area* (definita sotto /usr e come link). Inoltre il software completo occupa (come /var/adm/ris) circa 75 MB sulla var area nel caso che il nodo sia abilitato all'installazione remota tramite ris. Il software installato (supported e unsupported) richiede circa 180 MB tra /usr e /var, quindi la partizione *g* non dovrebbe essere inferiore a 400 MB per consentire anche l'installazione di altri prodotti come Fortran, DECnet, etc. Configurare quindi il disco in modo da avere le partizioni adeguate.

L'installazione di un sistema nuovo parte da cassetta o da CDrom. Le cassette da usare sono quelle dei *supported* tools. Per fare il boot da cassetta, battere, ad esempio:

```
>> boot 3/tz5
```

Dopo avere letto la cassetta viene presentato un menu del tipo:

- 1) BASIC installation
- 2) ADVANCED installation
- 3) System management

Normalmente si invoca il menu 3 che carica un kernel minimo per creare le partizioni del disco in modo personalizzato. Se il disco non è già definito si dovrà invocare MAKEDEV per creare i file dei driver:

```
# MAKEDEV rz1
```

dove rz è il disco e 1 è il driver. Si invoca chpt per creare le partizioni e quando si è terminato il lavoro di messa a punto si ritorna al menu principale con ^d. L'installazione da fare è quella *advanced*. Le domande poste riguardano:

- disco
- partizioni
- nome sistema
- data e ora (da correggere)
- password di *root*
- prodotti da installare.

Per l'installazione seguire attentamente le istruzioni del manuale *Installation Guide*. Le domande più ambigue sono quelle relative all'ora. Per quanto riguarda *timezone* si risponde *g*, per *Daylight Savings* si risponde *y*, per l'area geografica, rispondere *c*. In ogni caso il valore corretto che deve comparire sul file di configurazione (file /sys/conf/mips/HOSTNAME) è:

6 INSTALLAZIONE DEL SISTEMA

timezone -1 dst 4

Nel limite del possibile si consiglia di installare tutti i prodotti ed eventualmente di eliminare il materiale che non si desidera in un secondo tempo. Ad esempio se si vuole usare *man*, oltre alle *man pages*, va installato il software di documentazione, altrimenti le informazioni di man non possono essere visualizzate. Alla fine dell'installazione il sistema scrive un elenco dei file creati:

/vmunix	customized kernel
/genvmunix	generic kernel
/usr/adm/install.log	installazione
/usr/adm/install.FS.log	file system
/usr/adm/install.DEV.log	device

e fa il *reboot* automatico dopo circa 1 minuto. Per il contenuto e un esempio di detti file vedi appendice.

Il comando *doconfig* viene richiamato dall'installazione per la costruzione del kernel e riguarda la parte che richiede il nome del nodo, la data, il time zone.

Lo startup del sistema è pilotato da */etc/rc* (generico) e da */etc/rc.local* (locale).

6.1 CONFIGURAZIONE

Dopo l'installazione occorre personalizzare il sistema. Le operazioni di personalizzazione sono indicate nella tabella seguente.

Inserire licenze	<i>lmf, lmfsetup</i>
Inserire utenti	<i>adduser o yellow pages</i>
Aggiungere device	<i>MAKEDEV</i>
Settare stampanti	<i>lprsetup</i>
Settare rete locale	<i>netsetup</i>
Settare nfs	<i>nfsetup</i>
Settare installazione remota	<i>ris</i>
Settare diskless	<i>dms</i>
Settare terminali	<i>ttys -- gettytab</i>
Settare messaggi	<i>/etc/motd</i>

Le operazioni essenziali riguardano la gestione delle licenze e l'installazione dei device e della rete locale. Prima di definire gli utenti, si consiglia di personalizzare il file */etc/motd* che contiene il *message of the day*, messaggio analogo al file *welcome* del sistema VMS.

È bene cominciare a familiarizzarsi con i files di gestione del sistema e precisamente:

<i>/sys/conf/mips/node_name</i>	file di configurazione di sistema
<i>/etc/disktab</i>	tabelle partizioni dischi
<i>/etc/fstab</i>	tabella File system da montare
<i>/etc/rc</i>	startup generica
<i>/etc/rc.local</i>	startup specifica
<i>/etc/ttys</i>	tabella terminali

6 INSTALLAZIONE DEL SISTEMA

Dopo l'installazione, per correggere l'ora:

```
# date 1730
```

N.B. Dopo la *prima* installazione del sistema correggere l'ora con *date solo* se non esiste un sistema automatico di sincronizzazione con un centro specializzato (es. Galileo Ferraris) come quello fornito da DECnet-OSI.

Per verificare i prodotti installati:

```
$ setld -i
```

Se non si è installato un prodotto e si vuole fare un'aggiunta, usare:

```
# setld -l /dev/rmt0h
```

Per maggiori informazioni, consultare *man setld*. Tutte le installazioni fatte tramite *setld* vengono registrate in */etc/setldlog* di cui segue un frammento a titolo di esempio. Da notare che in */etc/setldlog* vengono registrate tutte le chiamate a */etc/setld* sia di caricamento prodotti che di verifica (IVP), comprese quelle di indice dei prodotti. Ogni informazione è accompagnata dal log di successo o di fallimento e relativo errore. Il file viene sempre aggiornato, per cui le installazioni più recenti sono registrate in fondo al file e sono rintracciabili in funzione della data. Mantenere sotto controllo la dimensione del file ed eventualmente cancellare con un editor le informazioni più vecchie.

Nell'esempio che segue, sono riportati frammenti di installazioni corrette e alcuni messaggi di errore. Nel log sono sempre indicate le aree di sistema su cui viene fatta l'installazione (server o client) e il device su cui risiedono i prodotti da installare (nastro, disco, CDrom). Sono evidenziate le informazioni relative a errori o a messaggi relativi ad operazioni di tipo IVP, */etc/setld -d, etc.*

```
SETLD 137 1992.06.29.18:55:03 / -1 0
SETLD 1039 root 1992.06.29.14:11:07 root=/ -1
Loading from . (disk)
DNPBASE420 LOAD, VERIFY, SUCCEEDED
DNPINETGW420 LOAD, VERIFY, SUCCEEDED
```

```
.....
DNPBASE420 SCP POST_L SUCCEEDED
DNPINETGW420 SCP POST_L SUCCEEDED
```

```
....
```

```
SETLD 1039 1992.06.29.14:22:56 / -1 0
SETLD 1418 root 1992.06.29.14:24:40 root=/ -1
Loading from . (disk)
/mnt/RISC/instctrl: no such file or directory
cannot load control information
```

<----

```
SETLD 364 1992.06.30.11:14:13 /dlenv0/root.TMP -1 0
SETLD 2367 root 1992.06.30.11:16:43 root=/dlenv0/root.TMP -1
```

6 INSTALLAZIONE DEL SISTEMA

```
Loading from /mnt/RISC/BASE_UPGRADE (disk)
UDTBASE425 LOAD,
Error installing Base System UPGRADE (UDTBASE425):          <----
Subset UDTBASE425 requires the existence of subset UDTBASE420 to operate
correctly. Please install subset UDTBASE420 before trying to install
subset UDTBASE425.

SETLD 2638 1992.06.30.11:54:07 /dlenv0/root0.mips -l 0
SETLD 4229 root 1992.06.30.11:55:02 root=/dlenv0/root0.mips -l
Loading from /mnt/RISC/UNSUPPORTED (disk)
UDXBASE420 LOAD, VERIFY, SUCCEEDED
UDXEDIT420 LOAD, VERIFY, SUCCEEDED
UDXINET420 LOAD, VERIFY, SUCCEEDED
.....
UDXBASE420 SCP POST_L SUCCEEDED
UDXEDIT420 SCP POST_L SUCCEEDED
UDXINET420 SCP POST_L SUCCEEDED
....

SETLD 7251 1992.06.30.14:07:10 / -l 0
SETLD 9539 root 1992.06.30.14:09:38 root=/dlenv0/root0.mips -l
Loading from . (disk)
/dlenv0/root0.mips/instctrl: no such file or directory      <----
cannot load control information

SETLD 11470 1992.06.30.14:18:46 /dlenv0/root0.mips -l 0
SETLD 205 root 1992.07.01.11:29:41 root=/ -l
Non-Standard Exit                                          <----
Exiting

SETLD 207 root 1992.07.01.11:53:28 root=/ -l
Loading from /dev/nrmt0h (tape)
OTMBASE103 LOAD, VERIFY, SUCCEEDED
DFRDOC310 LOAD, VERIFY, SUCCEEDED
DFRBASE310 LOAD, VERIFY, SUCCEEDED
.....
OTMBASE103 SCP POST_L SUCCEEDED
DFRDOC310 SCP POST_L SUCCEEDED
DFRBASE310 SCP POST_L SUCCEEDED

SETLD 219 1992.11.13.09:48:51 / -i                          (/etc/setld -i) <----

SETLD 1425 root 1992.12.10.10:39:39 root=/ -l
Loading from /cdrom/PCR120/bin (disk)
PCRRNOTE120 LOAD, VERIFY, SUCCEEDED
```

6 INSTALLAZIONE DEL SISTEMA

```
PCRBASE120 LOAD, VERIFY, SUCCEEDED
PCRMAN120 LOAD, VERIFY, SUCCEEDED
PCRRNOTE120 SCP POST_L SUCCEEDED
PCRBASE120 SCP POST_L SUCCEEDED
PCRMAN120 SCP POST_L SUCCEEDED
```

```
SETLD 1425 1992.12.10.10:45:25 / -1 0
SETLD 2829 root 1992.12.10.10:58:59 root=/ -v          (IVP) <----
PCRBASE120, IVP SUCCEEDED
```

```
SETLD 9772 root 1993.02.11.20:38:58 root=/ -d
LGRHELP201
LGRHELP201: not currently installed, cannot delete    <----
```

```
SETLD 10883 root 1993.09.30.11:05:04 root=/dlenv0/root0.vax -d
ULTBASE420
Sorry, You may not delete the Base System (ULTBASE420) subset <----
```

```
SETLD 2873 root 1994.07.04.09:41:20 root=/ -d
CXXMAN10
CXXMAN10: not currently installed, cannot delete
```

L'esempio riportato è relativo al file `/etc/setldlog` del *server*, per i clienti il formato del file differisce in quanto l'installazione non viene fatta localmente ma viene propagata tramite dms.

6.2 INSTALLAZIONE E GESTIONE LICENZE

Il sistema Ultrix è stato modificato per funzionare solo su prodotti per cui è disponibile la licenza adottando la metodologia in uso per il VMS. Assieme alla macchina vengono consegnati i fogli di licenza (PAK) per Ultrix e per eventuali altri prodotti. Per la registrazione delle licenze, invocare *lmf*:

```
# /etc/lmf register
```

Registrare la licenza usando le informazioni contenute nel PAK e inserire i dati con l'editor *vi* che viene automaticamente richiamato da *lmf*. Terminata la registrazione uscire da *vi* con il comando *:wq* e proseguire secondo il menu indicato da *lmf*. Il prodotto può essere usato anche in modo interattivo con:

```
# lmf
lmf> register
```

Le licenze vengono caricate su file e in memoria, ossia sul kernel volatile. Quando vengono apportate delle modifiche, queste agiscono sui file e devono essere riportate nel kernel volatile perchè diventino operative. La propagazione delle licenze viene fatta con:

6 INSTALLAZIONE DEL SISTEMA

lmf reset

I database e i file di gestione delle licenze si trovano sotto `/usr/var/adm/lmf`, se si desidera cambiare directory si può invocare `lmf` con l'opzione `-d dir`. I comandi di `lmf` sono, in parte, gli stessi della versione VMS e precisamente:

<code>help</code>	informazioni sui comandi
<code>exit</code>	esce da <code>lmf</code>
<code>list</code>	lista dei prodotti registrati
<code>ldb</code>	sommario prodotti registrati
<code>cache</code>	sommario prodotti attivi nel kernel volatile
<code>all</code>	sommario totale
<code>history</code>	storia delle registrazioni, modifiche, etc.
<code>register</code>	registrazione licenza
<code>disable</code>	disabilita la licenza
<code>enable</code>	abilita la licenza
<code>issue</code>	ricostruisce il PAK
<code>cancel</code>	cancella la licenza
<code>delete</code>	cancella fisicamente la licenza
<code>modify</code>	modifica i campi non protetti
<code>amend</code>	modifica i campi protetti
<code>reset</code>	copia le modifiche sul kernel volatile
<code>load</code>	carica la licenza nel kernel volatile
<code>unload</code>	scarica la licenza dal kernel volatile

Da notare che la registrazione può essere fatta anche da file con il comando:

```
# lmf register filename      (usa filename come modello di vi)
# lmf register - < filename  (carica filename nel database)
```

Le licenze appena registrate vengono abilitate automaticamente. Verificare lo stato con:

lmf list

I comandi di lista delle licenze non forniscono le informazioni di *checksum*. Per avere i dati completi della licenza occorre usare il comando *issue* che però cancella la licenza dal database e dal kernel volatile, quindi il file su cui viene scritto il PAK va conservato. Per ripristinare la licenza da un file di archivio:

```
# cd /usr/lic.ark           (spostarsi su una subdir di archivio)
# lmf ult.lic ultrix       (ricreazione PAK Ultrix su ult.lic)
# lmf -register - < ult.lic (ripristino licenza)
```

Il comando `cancel` altera la data di termine della licenza senza eliminare la stessa dal database, quindi è possibile ripristinare la licenza alterando tale data con una nuova operazione di `cancel`. Per rendere `cancel` operativo nel kernel volatile, si deve usare il comando di `load`.

Notare che `delete` cancella la licenza dal database e quindi occorre averne una copia per non perdere i dati. Con `modify` si possono alterare solo il campo commento e il campo `MOD_UNITS`, per gli altri campi occorre usare `amend`.

7 INSTALLAZIONE PRODOTTI DI RETE

7 INSTALLAZIONE PRODOTTI DI RETE

Dopo l'installazione del sistema si devono attivare i prodotti di rete, per le comunicazioni con i terminal server (LAT) e le comunicazioni con protocollo nativo (tcp/ip) ed eventualmente con protocollo DECnet.

7.1 tcp/ip

Dato che molte funzioni di Ultrix si basano su tcp/ip, questo viene installato assieme al sistema operativo e va solo attivato dopo l'installazione. Per attivare tcp/ip:

1. aggiornare `/etc/hosts` e `/etc/networks`
2. correggere `/etc/rc.local` per rendere operativo `/etc/ifconfig`

La personalizzazione di `/etc/rc.local` prevede le istruzioni seguenti:

```
/etc/ifconfig ln0 '/bin/hostname' broadcast 131.154.255.255 \  
                netmask 255.255.0.0  
/etc/ifconfig lo0 localhost
```

Inoltre se la rete prevede un nodo di routing statico, inserire dopo ***routed*** e prima di ***local daemons***:

```
/etc/route add default 131.154.1.9 1
```

dove:

131.154.1.9 = nodo di routing (N.B. valore che dipende dalla rete)
1 = no. hops per raggiungerlo

L'installazione può essere facilitata dalla procedura `/etc/netsetup`.

Per cambiare il no. di nodo e/o il nome di una macchina su cui tcp/ip sia già configurato e funzionante, procedere come segue. Per il ***server***, modificare il dati del nodo nei file:

```
/etc/rc.local  
/etc/hosts  
/etc/networks
```

Per i ***clienti***, si devono eseguire i seguenti step (dal server):

- # `cd /dlclientN/CLIENTNAME.root/etc`
- editare il file `netblk.c` correggendo la mappatura esadecimale degli indirizzi, oppure usare il comando:

```
usr/etc/makpkt SERVERADDRESS CLIENTNAME \  
CLIENTADDRESS BROADCAST NETMASK
```

7 INSTALLAZIONE PRODOTTI DI RETE

- compilare `netblk.c`:
`cc -c netblk.c`
- infine modificare l'indirizzo del cliente nel file:
`/dlclientN/CLIENTNAME.root/etc/dlparam`
correggendo il parametro:
`CLIARP="xxx.xxx.xxx.xxx"`
con l'IP address del nodo, ad esempio:
`CLIARP="131.154.10.73"`

Per ridurre la dimensione del file `/etc/hosts` limitando l'informazione ai soli nomi e indirizzi dei nodi locali, si può accedere ad un *name server*. Il servizio di name server per la connessione al Dominio Internet dei nomi di nodo si chiama BIND, sigla per *Berkeley Internet Name Domain (BIND)/Hesiod service* e viene creato dalla procedura:

```
/usr/etc/bindsetup
```

che crea i file `/etc/hesiod.conf` e `/etc/resolv.conf` con i nomi del dominio locale e dei nodi che svolgono la funzione di *name server*. Segue un esempio di detti file.

```
$ cat /etc/hesiod.conf #File hesiod.conf
rhs=.bo.infn.it
lhs=
$ cat /etc/resolv.conf #File resolv.conf
;
; BIND data file.
;
domain          bo.infn.it
nameserver      131.154.10.60
nameserver      131.154.10.99
```

7.2 DECnet phase IV

DECnet è un prodotto layered che va registrato con `lmf`. Il PAK è disponibile con il `distribution` e la registrazione si può fare con:

```
# lmf register - < /usr/lib/dnet_shared/DECnet-ULTRIX.PAK
# lmf reset
```

Per l'uso di DECnet è richiesto un utente *guest* da aggiungere agli utenti registrati, ad esempio con la procedura *adduser*.

Se la versione del sistema operativo non corrisponde a quella di DECnet, ad esempio il sistema operativo è il 4.2, mentre DECnet è relativo al sistema 4.0, prima dell'installazione, creare i files:

7 INSTALLAZIONE PRODOTTI DI RETE

```
# cd /usr/etc/subsets
# touch UDTBIN400.1k
# touch UDTBASE400.1k
# touch UDTMAN400.1k
# touch UDTCOMM400.1k
```

Per l'installazione:

```
# cd /
# /etc/setld -l /dev/rmt0h
```

Per la gestione, usare *ncp* che funziona in modo analogo alla versione per DECnet-VMS. In particolare, per evitare messaggi di errore sul video grafico porre in off il log degli eventi:

```
# ncp
ncp>show known logging
ncp>set logging console state off
ncp>def logging console state off
ncp>list known logging
```

DECnet consente lo scambio di comunicazioni tra il mondo Ultrix e il mondo DECnet. Funziona come *end node*. Convive con tcp/ip. I comandi consentiti da DECnet sono:

dcp	copia files
dls	directory remota
dcat	visualizzazione file remoto
drm	cancella file remoto
dlogin	login su altro nodo DECnet

Esempi:

```
$ dcp file.ps decnode::'lta4:' (stampa su coda LAT-VMS)
$ dls decnode::'usr$disk:[user.dat]*.dat' (directory su VMS)
$ dcp .profile decnode::'[user]profile.dat' (copia da Ultrix a VMS)
```

Per aggiornare il database dei nodi da un altro nodo che contenga un database completo:

```
# update_nodes -f <node_name>
```

seguito da restart di DECnet, dato che l'aggiornamento viene fatto sul database permanente che risiede su disco, lasciando inalterato il database volatile del kernel. Lo start e stop di DECnet vengono fatti tramite ncp agendo sullo stato dell'*executor*. I database dei nodi DECnet si trovano nella directory /etc/var/dnet e si chiamano *nodes_p* per il database permanente e *nodes_v* per il database volatile.

7 INSTALLAZIONE PRODOTTI DI RETE

7.3 LAT

Tra i prodotti di rete, va considerato anche il LAT per l'accesso a terminali e stampanti connessi tramite terminal server. Il LAT (Local Area Transport) consente l'accesso da parte dei terminali ai nodi via Ethernet. Il LAT viene gestito da `/etc/lcp` per il controllo e la gestione dei terminal servers. Per maggiori informazioni consultare `man`. Per vedere se il LAT è attivo, usare il comando:

```
$ /etc/lcp -d
```

Per attivare il LAT, in `/etc/rc.local` inserire il comando:

```
# Start LAT
/etc/lcp -s; echo ' Starting LAT '
```

Inoltre nel file di configurazione `/usr/sys/conf/mips/NODENAME` devono comparire:

```
options LAT
pseudo-device lat
```

Se l'host è anche abilitato al caricamento dei servers:

```
options DLI
pseudo device DLI
```

8 INSTALLAZIONE DEVICE E DISCHI

8 INSTALLAZIONE DEVICE E DISCHI

Quando il sistema operativo viene installato, crea i driver (device) per ogni device fisico connesso alla macchina e li memorizza in */dev*. Per aggiungere un device alla macchina in seguito, dopo l'installazione dell'hardware e la determinazione del nome fisico del device, eventualmente con `test 5` (oppure `test 3` per 5000/240) a livello di `halt`, supponendo che il device sia `rz2`, fare:

```
# MAKEDEV rz2 (disco)
```

Se si aggiunge un'unità nastro di tipo DAT (TLZ04), a livello di `test hardware` compare un device di tipo `tz` e il relativo slot, es. `tz3`, in questo caso i driver vengono creati con i comandi:

```
# cd /dev
# MAKEDEV tz3 (nastro)
```

In caso di dubbio, se i file di device `rz2*` e `rrz2*` esistono già cancellarli prima di invocare `MAKEDEV`. Creare le partizioni disco consultando `/etc/disktab`, ad esempio:

```
# newfs /dev/rrz1c rz58
```

crea la partizione `C` che occupa tutto il disco. Ripetere l'operazione per tutte le partizioni volute. Se le dimensioni delle partizioni non sono adeguate, modificarle con `/etc/chpt` come illustrato ampiamente nella sezione 5. Creare la directory a cui associare il disco, ad esempio:

```
# mkdir /libdisk
```

Montare il nuovo file system appena creato per verificare che disco, file system e directory di mount siano definiti in modo corretto. Per poter rimontare il nuovo device automaticamente dopo ogni ripartenza, dopo l'installazione e la personalizzazione del file system, aggiornare il file `/etc/fstab` in accordo al file system creato. In generale si avrà un file `/etc/fstab` del tipo:

```
/dev/rz1a/:rw:1:1:ufs::
/dev/rz1g:/usr:rw:1:2:ufs::
/dev/rz1h:/usr/users:rw:1:3:ufs::
```

I campi di `fstab` sono (vedi `man fstab`):

`spec:file:type:freq:passno:name:options`

`con:`

8 INSTALLAZIONE DEVICE E DISCHI

spec	device (per nodo remoto /dev@remote_node)
file	directory file system
type	operazioni ammesse: rw read/write ro read only rq read/write with quota sw swap extension xx ignore
freq	dump frequency
passno	mount order at reboot
name	vale ufs per Ultrix e nfs per SUN o network file system
options	opzioni di mount (vedi man mount)

Una volta aggiornato `fstab`, verificarne la validità con:

```
# /etc/mount -a
```

oppure

```
# /etc/mount /dev/rz2c /mnt
```

nel caso in cui il file system da montare non compaia in `/etc/fstab` e non esista la relativa directory, infatti `/mnt` è una directory di comodo per prove di mount. Per smontare il file system:

```
# umount /mnt
```

L'operazione di `umount` fallisce se il file system è in uso.

9 INSTALLAZIONE TERMINALI E STAMPANTI

9 INSTALLAZIONE TERMINALI E STAMPANTI

Per la gestione dei terminali esistono due tabelle descrittive delle capacità dei terminali disponibili: `/etc/termcap` (per BSD) e `/usr/lib/terminfo` (per SYS V) e che vengono usate da programmi come `vi`, `clear`, `more`, etc.

`/etc/termcap` è una tabella editabile, mentre `/usr/lib/terminfo` è una directory che contiene delle subdirectory (ad es. `v` per i terminali DEC) nelle quali sono memorizzati i files binari di descrizione dei terminali. Per creare un nuovo terminale occorre usare il programma `tic` di compilazione. Le tabelle di gestione dei terminali per i due sistemi sono:

```
BSD:      /etc/ttys          /etc/gettytab
SYS V:    /etc/inittab     /etc/gettydefs
```

Inoltre esiste il programma `stty`, di cui parleremo in seguito, per la gestione di terminali su linee seriali. Dato che Ultrix è orientato maggiormente verso BSD, in questa sede non verranno prese in considerazione le informazioni relative a SYS V. Il tipo di terminale può essere settato in maniera esplicita con i comandi:

```
$ TERM=VT100; export TERM    (per sh)
% setenv TERM vt100         (per csh)
```

Se si definiscono nuovi terminali non conviene modificare `/etc/termcap` fino a che le nuove definizioni non sono corrette, è consigliabile creare una nuova versione di prova e definire:

```
$ TERMCAP=/fullpath/newtermcap; export TERMCAP    (per sh)
% setenv TERMCAP /fullpath/newtermcap             (per csh)
```

Per eliminare la definizione:

```
$ TERMCAP=                                         (per sh)
% unsetenv TERMCAP                                 (per csh)
```

Il tipo di terminale viene settato automaticamente al login dai file di personalizzazione `.profile` o `.login` a seconda dello shell di default, con i comandi:

```
% cat .login
stty dec new cr0
tset -I -Q
.....

$ cat .profile
tty -s
if test $? = 0
then
    stty dec crt
fi
.....
```

9 INSTALLAZIONE TERMINALI E STAMPANTI

Il settaggio del terminale viene gestito da *tset* in funzione del tipo di terminale ricavando le informazioni dal database */etc/ttys* che è un file editabile e va personalizzato a seconda delle necessità. Il file è formato da una lista di ogni file */dev/tty** con campi di personalizzazione separati da *tab* o da *blank*. Se un campo è formato da più parole deve essere incluso in *"*, i campi commento sono preceduti da *#*. Il formato di *ttys* è:

nome comando tipo flags

dove:

nome	nome del file di <i>/dev/</i>
comando	comando da eseguire all'inizializzazione del terminale, in generale il comando <i>getty</i> controlla baud-rate, legge il nome di login e chiama <i>login</i> ossia un qualsiasi comando di inizializzazione.
tipo	tipo di terminale comunemente associato al <i>/dev/</i> file
flags	sono flag di status per la routine <i>getttyent</i> .

I possibili valori dei flags sono:

<i>on</i>	abilita login per il terminale
<i>off</i>	disabilita login (default)
<i>secure</i>	consente il login di root sul terminale
<i>su</i>	consente di collegarsi con <i>su</i>
<i>nomodem</i>	ignora segnali di modem (default)
<i>modem</i>	ricosce segnali di modem
<i>window</i>	sistema di window

Per accedere con un terminal server alla macchina inserire un certo numero di definizioni di terminale con la qualifica *on modem* come segue:

```
tty00  "/etc/getty std.9600" vt100    on modem      # lat
tty01  "/etc/getty std.9600" vt100    on modem      # lat
```

In genere in */etc/ttys* vengono designati solo terminali *vt100*, per gestire anche altri modelli di terminali occorre modificare */etc/ttys*. Le modifiche diventano attive dopo un *reboot* oppure con il comando:

```
# kill -HUP 1
```

Segue un esempio del file */etc/ttys*, da notare che *window* dipende dalla versione del sistema.

```
# @(#)ttys      3.1 (ULTRIX) 4/20/90
#
#
#
# name  getty           type           status         comments
#
```

9 INSTALLAZIONE TERMINALI E STAMPANTI

```

console "/etc/getty std.9600" vt100      off      secure # console term
tty00   "/etc/getty std.9600" vt100      on modem # lat
tty01   "/etc/getty std.9600" vt100      on modem # lat
tty02   "/etc/getty std.9600" vt100      off nomodem # laser
tty03   "/etc/getty std.9600" vt100      off nomodem # laser_ps
tty04   "/etc/getty std.9600" vt100      on modem # lat
tty05   "/etc/getty std.9600" vt100      on modem # lat
tty06   "/etc/getty std.9600" vt100      on modem # lat
tty07   "/etc/getty std.9600" vt100      on modem # lat
ttyd0   "/etc/getty std.9600" vt100      off shared secure # modem line
ttyp0   none                network
ttyp1   none                network
ttyp2   none                network
ttyp3   none                network
.....
.....
ttyqf   none                network
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession -e" none
on secure window="/usr/bin/Xcbf" # Con Ultrix 4.2a Xcbf diventa XWS bc

```

La voce :0 ... di /etc/ttys compare solo per stazioni grafiche.

/etc/getty è uno fra i vari processi con cui l'utente accede ad Ultrix via terminale. /etc/getty inizializza la linea, legge il nome di login e invoca **login**, tentando di adattare il sistema alla velocità e al tipo di terminale specificato. Viene quindi aperto il corretto /dev/tty e creati i descriptor 0, 1, 2 per **stdio**, **stdout**, **stderr** rispettivamente. Inoltre /etc/getty fa controlli d'errore sulla linea e li segnala alla console.

Dopo il login, tset controlla lo stato fisico del terminale, setta i caratteri di **erase** (cancellazione carattere battuto) e **kill** (cancellazione linea battuta), i ritardi, sequenze di inizializzazione, etc. derivando il tipo di terminale da /etc/ttys. Per le opzioni previste da tset consultare man. Le più comuni sono:

- I sopprime l'invio delle stringhe di inizializzazione
- Q sopprime il messaggio di erase e kill
- s scrive i comandi per TERM
- S analogo a -s ma completato con i comandi di set per csh
- m mappa i terminali per login in dialup
- il nome del terminale viene scritto su stdout, catturato dallo shell e posto nella variabile TERM

Altri comandi correlati al terminale sono **stty** e **tty**. **stty** fornisce informazioni sul settaggio del terminale oppure altera le caratteristiche indicate a seconda degli argomenti di chiamata. I terminali possono essere settati secondo due tipi di collegamento (**disciplina della linea**, in terminologia unix) NTTYDISC per terminali generici, TERMIODISC per terminali SYS V o terminali IEEE-POSIX. I comandi di visualizzazione sono:

Per informazione generica:

9 INSTALLAZIONE TERMINALI E STAMPANTI

`$ stty`

Per informazione sulla disciplina:

`$ stty disc`

Per informazioni su tutti i settaggi usati normalmente non-termio:

`$ stty all`

Per tutti i settaggi non-termio:

`$ stty everything`

Per tutti i settaggi SYS V:

`$ stty -a`

Per tutti i settaggi IEEE-POSIX:

`$ stty -p`

Esempi di output da `stty`:

```
$ stty disc  
NTTYDISC
```

```
$ stty  
new tty, speed 38400 baud ; tabs crt  
decctlq
```

```
$ stty all  
new tty, speed 38400 baud , 0 rows, 0 columns; tabs  
crt  
decctlq  
erase kill werase rprnt flush lnext susp intr quit stop eof  
^? ^U ^W ^R ^O ^V ^Z/^Y ^C ^\ ^S/^Q ^D
```

```
$ stty everything  
new tty, speed 38400 baud , 0 rows, 0 columns  
even odd -raw -nl echo -lcase -tandem tabs -cbreak  
crt: (crtbs crterase crtkill ctlecho) -tostop  
-tilde -flusho -litout -pass8 -nohang -autoflow  
-pendin decctlq -noflsh  
erase kill werase rprnt flush lnext susp intr quit stop eof  
^? ^U ^W ^R ^O ^V ^Z/^Y ^C ^\ ^S/^Q ^D
```

9 INSTALLAZIONE TERMINALI E STAMPANTI

```
$ stty -a
line = NTTYDISC; speed 38400 baud
erase = DEL; kill = ^u; min = 6; time = 1; intr = ^c; quit = ^|;
eof = ^d; eol <undef>; start = ^q; stop = ^s;
parenb -parodd cs7 -cstopb -hupcl cread -clocal -loblk
-ignbrk brkint -ignpar -parmrk -inpck istrip -inlcr -igncr icrnl -iucLC
ixon -ixany -ixoff
isig icanon -xcase echo echoe -echok -echonl -noflsh
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel
```

```
$ stty -p
line = NTTYDISC; speed 38400 baud
erase = DEL; kill = ^u; min = 6; time = 1; intr = ^c; quit = ^|;
eof = ^d; eol <undef>; start = ^q; stop = ^s;
susp = ^z; dsusp = ^y; rprnt = ^r; flush = ^o; werase = ^w;
lnext = ^v;
quote = parenb -parodd cs7 -cstopb -hupcl cread -clocal -aflow
-ignbrk brkint -ignpar -parmrk -inpck istrip -inlcr -igncr icrnl
-iucLC ixon -ixany -ixoff
isig icanon -xcase echo echoe -echok -echonl -noflsh
-ixten -tostop ctlech -prtera crtbs crtera crtkil
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel -tilde
```

Infine `tty` stampa il nome corrente del terminale a meno che non sia specificato (`-s`). Se `stdio` è un terminale `l'exit status` è 0, altrimenti è 1. Esempio:

```
$ tty
/dev/ttyp1
```

9.1 GESTIONE TERMINALI UTENTI

Di seguito diamo alcune informazioni e un riepilogo dei comandi a livello utilizzo. Per vedere i valori assegnati a `termcap` sia in `sh` che in `csh`:

```
$ tset -Q -s vt100
```

Per il set automatico di `TERMCAP` da Bourne shell:

```
$ eval 'tset -Q -s vt100'
```

in questo modo `TERMCAP` contiene i valori essenziali del file `/etc/termcap` riducendo il tempo di esecuzione delle operazioni dato che il sistema non deve esaminare il file `ASCII /etc/termcap`.

Per il reset del terminale:

```
$ tset
```

9 INSTALLAZIONE TERMINALI E STAMPANTI

Per una mappatura dinamica di terminali:

```
$ eval 'tset -s -Q -m 'dialup@1200:vt100' vt400'
```

in cui il terminale viene settato (-s) senza comunicazione di erase e kill (-Q) con la mappatura (-m) a vt400 per terminali diretti oppure a vt100 a 1200 baud per collegamenti dialup. La velocità può essere indicata con la sintassi seguente:

<1200	minore di 1200
>1200	maggiore di 1200
@1200	uguale a 1200
!@1200	diversa da 1200, ossia tutte le velocità eccetto 1200.

Se un terminale è bloccato:

```
$ reset; tset
```

se non accetta neanche questi comandi, probabilmente è stato alterato anche <RET>, per il ripristino:

```
$ ^Jreset^J
```

Se si vuole vedere la codifica di caratteri speciali come **backspace**, usare la sequenza:

```
$ stty -echo; cat -v; stty echo
```

che svolge le seguenti funzioni: elimina l'eco, invoca **cat** con visualizzazione dei caratteri non stampabili, a questo punto si batte il carattere desiderato seguito da <RET> e si termina con ^D o ^C; ripristina l'eco.

9.2 FORMATO DI TERMCPAP

Per ogni terminale descritto in termcap deve comparire una riga contenente il nome seguita dalle righe che contengono la descrizione dei campi, ogni campo è terminato da : e, dato che la descrizione del terminale deve comparire su di una unica riga logica, le righe intermedie devono terminare con \. I campi descrivono le capacità del terminale, ogni capacità è descritta con un codice di 2 lettere. Le capacità possono essere delle qualifiche, dei valori o delle sequenze. Il nome è formato da due caratteri di cui il primo indica il costruttore (es. d = DEC, I = IBM, etc.) e il secondo identifica il modello del terminale, (es. dO = VT100, dF = VT400, etc.), seguiti da | e altri nomi alias, l'ultimo dei quali dà la descrizione completa del modello (es. dec vt400 series, vt420-am 132 cols). Le qualifiche sono indicate da due caratteri come **am** per auto margin, i valori sono formati da due caratteri seguiti da # e dal valore numerico, es. co#80 per 80 colonne, le qualifiche sono formate da due caratteri seguiti da = e dalla stringa che specifica la sequenza, ad esempio cr=~M per carriage return, do=~J per cursor down, c1=50\E[;H\E[2J per il clear dello schermo, dove \E indica <escape> e 50 è il ritardo in con cui il terminale esegue il comando.

Il tempo di ritardo è chiamato **padding** e può essere espresso come un intero o come un intero seguito da * , l'intero indica il ritardo in millisecondi, * indica invece che il

9 INSTALLAZIONE TERMINALI E STAMPANTI

ritardo è proporzionale al numero di linee interessate all'operazione (es. 5dd in vi, cioè cancellare 5 righe con l'editor).

Le definizioni di termcap hanno la forma indicata nel frammento seguente che si riferisce al vt100.

```
d0|vt100|vt100-am|dec vt100:\
:cr=^M:do=^J:nl=^J:bl=^G:co#80:li#24:cl=50\E[;H\E[2J:\
:le=^H:bs:am:cm=5\E[%i%d;%dH:nd=2\E[C:up=2\E[A:\
:ce=3\E[K:cd=50\E[J:so=2\E[7m:se=2\E[m:us=2\E[4m:ue=2\E[m:\
:md=2\E[1m:mr=2\E[7m:mb=2\E[5m:me=2\E[m:\
:is=\E[1;24r\E[24;1H:\
:ct=2\E[3g:st=2\EH:\
:rf=/usr/lib/tabset/vt100:\
:rs=\E>\E[?31\E[?41\E[?51\E[?7h\E[?8h:\
:ks=\E[?1h\E=:ke=\E[?1l\E>:\
:ku=\EOA:kd=\EOB:kr=\EOC:kl=\EOD:kb=^H:\
:ho=\E[H:k1=\EOP:k2=\EOQ:k3=\EOR:k4=\EOS:ta=^I:pt:sr=5\EM\
:vt#3:xn:\
:sc=\E7:rc=\E8:cs=\E[%i%d;%dr:
```

#-----

9.3 GESTIONE STAMPANTI

Per usare una stampante su linea seriale o su terminal server verificare che sia corretta la definizione `ttys` corrispondente ed usare `lprsetup` che consente la gestione delle stampanti in modo interattivo. Analogamente ai terminali le stampanti sono definite dal file `/etc/printcap` che descrive le capacità delle stampanti e che viene modificato dalla procedura `/etc/lprsetup`. Il formato del file è descritto in modo completo da *man printcap*. La stampa è gestita in modo *spooled*, cioè su code di stampa, dai processi e dai file:

<code>/usr/lib/lpd</code>	gestione stampante (sempre attivo)
<code>/usr/ucb/lpq</code>	esame della coda di stampa
<code>/usr/ucb/lprm</code>	cancellazione dalla coda di stampa
<code>/etc/lpc</code>	amministrazione delle code
<code>/usr/ucb/lpr</code>	inserimento stampa nella coda
<code>/etc/printcap</code>	descrizione stampanti
<code>/dev/printer</code>	driver associato a lpd

Dato che `/usr/ucb` è nel `PATH`, non occorre specificarlo per eseguire i programmi di stampa. Se si è definita una stampante di nome `ln05`, per stampare si avrà:

```
$ lpr -Pln05 .profile
```

I file di stampa vengono creati temporaneamente sotto `/var/spool/lpd` dove si trovano anche i file:

9 INSTALLAZIONE TERMINALI E STAMPANTI

lock ID del daemon di stampa
status stato della stampante

lock e status sono file ASCII stampabili da qualsiasi utente. Per ogni stampa vengono creati il file cf* che contiene le istruzioni di stampa e il file df* che contiene il testo formattato per la stampa. Se la stampante, ad esempio, è di tipo PostScript, si indicherà, a livello setup /usr/lib/lpfilters/ln03rof, il filtro che traduce un file ASCII in un formato PostScript.

Per la gestione delle code di stampa si usa lpc che è accessibile solo da *su* e funziona in modo interattivo con help.

Per l'installazione delle stampanti si usa lprsetup da *su* che consente i comandi:

```
Command < add modify delete exit view quit help >: quit
```

Di seguito compare un esempio di setup per due stampanti, lp0 e lp1, di cui una (lp1) PostScript.

```
# lprsetup
ULTRIX Printer Setup Program

# @(#)printcap 4.1 (ULTRIX) 7/2/90
lp0|lp|0|local line printer:\
    :lp=/dev/lp:\
    :of=/usr/lib/lpfilters/lpf:\
    :sd=/usr/spool/lpd:\
    :lf=/usr/adm/lpd-errs:

# lp|lp1|1|psjet_gri:\
    :af=/usr/adm/lpacct:\
    :br#9600:\
    :ct=lat:\
    :fc#0177777:\
    :fs#03:\
    :if=/usr/lib/lpfilters/ln03rof:\
    :lf=/usr/adm/lperr:\
    :lp=/dev/tty02:\
    :mc#20:\
    :mx#0:\
    :of=/usr/lib/lpfilters/ln03rof:\
    :op=PSJET_GRI:\
    :os=\
    :pl#66:\
    :pw#80:\
    :rw:\
    :sd=/usr/spool/lpd:\
    :ts=GRI2:\
```

9 INSTALLAZIONE TERMINALI E STAMPANTI

```
:uv=4.0:\
:xc#0177777:\
```

Per installare una stampante su terminal server via *LAT*:

1. controllare in `/etc/ttys` che ci sia una porta `tty` per ogni stampante disponibile da settare con `off nomodem`. Esempio:

```
console "/etc/getty std.9600" vt100 on secure # console terminal
tty00   "/etc/getty std.9600" vt100 on nomodem # direct connect tty
tty01   "/etc/getty std.9600" vt100 off nomodem # direct connect tty
tty02   "/etc/getty std.9600" vt100 off nomodem # laser psjet_gri
tty03   "/etc/getty std.9600" vt100 off nomodem # laser laserps_gri
```

2. modificare in `rc.local` lo start del `lat`, inserendo `-h /dev/ttyx` dove `x` e' il `tty` della stampante. Esempio:

```
#####
# %LatSTART% #
#####

if [ -f /etc/lcp ]; then
    echo 'Starting Lat '                                >/dev/console
    /etc/lcp -s -h /dev/tty02,/dev/tty03
fi
# %LatEND
```

3. eseguire `lprsetup` e rispondere adeguatamente alle domande rispettando i default riguardanti la creazione dei file di controllo e di spool. L'operazione crea/modifica il file `/etc/printcap`.

4. per stampare

```
lpr -Pname_printcap_printer
```

Ad esempio, i seguenti comandi sono equivalenti e producono l'uscita sulla stampante di nome `laserps_gri` corrispondente a `lp1`:

```
lpr -P1 file_name
lpr -Plp1 file_name
lpr -Plaserps_gri file_name
```

Ricordiamo infine alcuni comandi di gestione del terminal server da parte di `lcp`:

```
lcp -t          stop LAT
lcp -d          show LAT
lcp -z          zero error counters LAT
lcp -p /dev/tty15  indica i servizi connessi a tty15
```

10 GESTIONE UTENTI

10 GESTIONE UTENTI

10.1 GESTIONE UTENTI IN AMBIENTE STAND-ALONE

In un ambiente stand-alone, gli utenti vengono gestiti tramite il file `/etc/passwd`, un file editabile formato da 8 campi separati da `:` con il nome dell'utente, la password, etc. e precisamente:

1	nome utente	
2	password	crittografata
3	User ID	valore numerico che indica il gruppo di appartenenza dell'utente per gestire i permessi di accesso ai file. Valore generalmente diverso per ogni utente. I valori bassi da 0 a 99 sono riservati agli utenti di management (<code>root</code> , <code>operator</code> , etc.), valori superiori (generalmente da 267 in su) vengono assegnati agli utenti generici
4	Group ID	valore numerico di appartenenza ad un gruppo per la determinazione dell'accesso ai file, registrato nel file <code>/etc/group</code>
5	Personal	nome utente, usato come commento
6	Directory	path completo per la directory di login
7	shell	path assoluto per lo shell utente (es. <code>/bin/sh</code> , <code>/bin/csh</code> , etc.)

I campi del file `/etc/group` sono 4 separati da `:` e precisamente:

1	nome gruppo	i.e. <code>system</code> , <code>users</code> , etc.
2	password	crittografata, di solito ignorata e sostituita da <code>*</code> per eliminare il test
3	Group ID	
4	nomi utenti	eventuale elenco dei componenti del gruppo separati da virgola.

Per il management degli utenti si può procedere manualmente con l'editor ***vipw*** che è equivalente a ***vi*** ma opera dei controlli di protezione ed evita l'accesso di più utenti contemporanei. In questo modo il campo ***password*** deve essere lasciato ***vuoto***. Occorre poi creare la ***home directory*** dell'utente:

```
# mkdir /usr/users/newone
```

copiare i login file per i vari shell dal template:

```
# cp /usr/skel/.??* /usr/users/newone
```

cambiare l'owner della directory:

```
# /etc/chown newone /usr/users/newone /usr/users/newone/.??
```

cambiare il gruppo:

10 GESTIONE UTENTI

```
# chgrp group /usr/users/newone /usr/users/newone/.??*
```

verificare la protezione dei file che deve essere almeno 700.

Per un management più comodo e sicuro conviene usare la facility *adduser* che esegue le operazioni indicate in modo interattivo con verifica dei campi. Per cancellare un utente usare *removeuser*.

10.2 GESTIONE UTENTI IN AMBIENTE YELLOW PAGES

In un ambiente client-server gli account degli utenti vengono gestiti tramite un database condiviso che prende il nome di *YELLOW PAGES* (Pagine Gialle). Il file delle password si trova nella directory */var/yp/src* del nodo che gestisce *YP*, cioè le Yellow Pages. L'aggiunta di un utente deve quindi essere propagata dal database ai singoli nodi e le password non sono più gestite da */etc/passwd* ma da *yppasswd*.

Per aggiungere un utente in *ambiente YP*, occorre quindi modificare il file di account */var/yp/src/passwd*, aggiornare il database tramite *make* e creare la directory utente come illustrato nel precedente paragrafo. In sintesi i passi sono i seguenti:

```
# cd /var/yp/src
# vi passwd
# cd /var/yp
# make DIR=/var/yp/src passwd
# mkdir directory
# chown newone /usr/users/newone /usr/users/newone/.??
# chgrp group /usr/users/newone /usr/users/newone/.??
# cp /usr/skel/.??* /usr/users/newone
```

Dato che *DIR=/var/yp/src* è la directory di default non occorre indicarla nel comando *make*.

Per cambiare la password usare *yppasswd*.

11 GESTIONE DISCHI: UFS E NFS

11 GESTIONE DISCHI: UFS E NFS

Il sistema vede due tipi di file system UFS (locale) e NFS(remoto). I dischi locali fanno parte di UFS, i dischi remoti fanno parte di NFS. La gestione dello spazio non differisce fra disco locale e remoto. I file system dei due sistemi sono visibili in maniera trasparente per l'utente.

Il disco viene visto come un numero di settori da 512 bytes indirizzati da LBN (logical block number) con valori da 0 a *max*, dove *max* dipende dalla capacità del disco. Come abbiamo già visto, i dischi sono divisi in partizioni con un massimo di 8, designate con lettere da *a* ad *h*. Un file system è descritto dal **superblock** con parametri quali il numero di blocchi di dati, il numero massimo di file, un pointer alla lista dei blocchi liberi, etc. L'informazione sulle partizioni dell'intero disco si trova nella partizione *a*. Il superblock è costruito da *newfs*, si trova in una posizione specifica della partizione e ne esistono copie in altre posizioni ai fini di backup. Inoltre, se le partizioni sono diverse da quelle di default (*/etc/disktab*), il superblock contiene anche la **partition table**. Tale tabella viene copiata in memoria (in sostituzione della tabella di default) nella tabella attiva a cui fa riferimento il driver del disco. Per informazioni sui superblock consultare il file creato con l'installazione */usr/adm/install.FS.log*. Ogni file è rappresentato sul disco da un *inode* che contiene informazioni quali owner, date di accesso e modifica, indici ai dati, etc.

I dischi vengono montati secondo le indicazioni contenute in */etc/fstab*. Si deve rispettare l'ordine di montaggio in modo che vengano montati prima i file systems le cui directory sono le radici di altri file systems. Operazioni abituali sui dischi sono gestite da */etc/mount*, */etc/umount*, */etc/fsck*. Tali operazioni consentono di verificare lo stato dei dischi e di eliminare inconsistenze del file system quali *inode* senza riferimento, blocchi mancanti nella free list, conteggi errati nel superblock, etc. **Cause di corruzione** sono: errori hardware, cadute di tensione, mancanza di sync prima di shutdown. Da notare che */etc/fsck* gira su file system smontati.

Per convenzione la partizione *a* contiene la root directory, la partizione *b* contiene i file di page, swap e dump e non viene montata, la partizione *g* contiene */usr* (ossia la *var area*). Se l'area di swap risulta insufficiente si può assegnare un'ulteriore area all'installazione del sistema oppure in seguito agendo sul file di configurazione. Le directory primarie del sistema sono root (*/*), */usr* e */var* suddivise come segue:

<i>/</i>	root
<i>/etc</i>	manutenzione, accounting, management
<i>/bin</i>	utilities di shell (es. <i>sh</i> , <i>csh</i> , etc.)
<i>/tmp</i>	temporanea
<i>/dev</i>	device
<i>/usr</i>	file system <i>usr</i>
<i>/usr/adm</i>	funzioni amministrative
<i>/usr/lib</i>	librerie
<i>/usr/ucb</i>	utilities (es. <i>grep</i> , <i>edit</i> , <i>ftp</i> , etc.)
<i>/usr/hosts</i>	informazioni di LAN
<i>/usr/man</i>	documentazione on-line

11 GESTIONE DISCHI: UFS E NFS

<code>/usr/users</code>	utenti (usualmente file system su altro disco, obbligatorio nei sistemi client-server, dato che <code>/usr</code> deve essere privato per ogni cliente)
<code>/var</code>	log file (su file system separato per client-server)
<code>/var/adm</code>	account, error log, crash dump
<code>/var/spool</code>	per printer, mail, etc.
<code>/var/adm/ris</code>	installazione remota

11.1 GESTIONE SPAZIO

La verifica dello spazio occupato può essere fatta *globalmente* (`df`) o a partire dalla *working directory* (`du`) o *per partizione* (`/etc/quot`). Per l'occupazione globale:

```
$ df
```

Per avere anche l'informazione su inode:

```
$ df -i
```

Per un sommario:

```
$ du -s
```

Per directory, a partire dalla working directory `wd`:

```
$ cd /any
```

```
$ du
```

Per singolo file a partire dalla working directory `wd`:

```
$ du -a
```

Per tutti gli utenti:

```
$ du -s /usr/users/*
```

Mentre `du` e `df` sono accessibili a tutti, il comando `/etc/quot` può essere eseguito solo da *su*, agisce sul device fisico e fornisce il numero di blocchi e di file per utente:

```
# /etc/quot -f /dev/rz1f
```

Ad esempio:

```
/dev/rz1f:
```

94611	1523	zeus
92593	395	cern
3291	17	jones
1670	23	bos
1506	27	julian

11 GESTIONE DISCHI: UFS E NFS

1060	51	testjobs
359	5	proofs
32	21	root
12	8	bachque
12	8	online
5	5	guest

Per la ricerca di file critici si può usare `find` per il cui uso si consiglia di consultare `man`. Ad esempio per visualizzare tutti i file che sono cambiati negli ultimi 7 giorni:

```
# find / -ctime -7 -print
```

11.2 QUOTE

Sebbene non sia previsto in generale da Unix, Ultrix prevede una gestione dello spazio disco a livello utente tramite un sistema di quote in parte simile a quanto avviene nel VMS. Per attivare il meccanismo di quota occorre che nel `config file` esista l'opzione:

```
options QUOTA
```

Se la quota non è mai stata attivata, si deve creare il file di quota (`quotas`) sul file system su cui la quota deve essere attivata:

```
# /etc/quotacheck -f file_system
```

Si deve poi editare `/etc/fstab` e modificare `rw` in `rq` per abilitare read-write con quota. Si deve invocare `/etc/edquota` per creare un entry per ogni utente e ogni file system; `edquota` invoca `vi`. Normalmente si crea un utente prototipo per propagare le informazioni di quota, il comando:

```
# /etc/edquota -p proto-user james
```

aggiunge i dati di quota per l'utente `james`. I dati di quota sono del tipo:

```
fs /usr/student blocks (soft=2000, hard=2500) inodes (soft=30,hard=50)
```

Perché la quota sia attiva, occorre invocare `/etc/quotactl` da `/etc/rc.local` e fare `reboot`. Per disattivare la quota usare `/etc/quotactl`. Verifiche sulla quota vengono eseguite da:

<code>quotacheck</code>	verifica
<code>quota</code>	valore attuale
<code>repquota</code>	sommario

La quota può essere attivata *solo* su file system locali, l'informazione viene propagata quando il file system viene visto via `nfs` da altri sistemi consentendo comunque il controllo dello spazio disco.

11 GESTIONE DISCHI: UFS E NFS

11.3 NETWORK FILE SYSTEM

Il Network File System (NFS) consente di creare dei file system su di un nodo e condividerli con altri nodi di una LAN in modo che un utente non veda differenza tra i dischi locali e i dischi remoti. La condivisione dei file avviene in un ambiente eterogeneo, cioè tra processori, sistemi operativi e reti differenti, quindi un sistema Ultrix può esportare i suoi dati ad un sistema VAX-VMS dotato di tcp/ip con funzionalità NFS e viceversa. In questo modo l'utente può leggere e scrivere i dati su di un qualsiasi nodo o sistema, purchè la rete sia efficiente e robusta.

Il sistema di sharing è basato sul concetto *client-server*, in cui un cliente richiede risorse fornite da altri sistemi detti server. Un *server* è un qualsiasi nodo o processo che fornisce un servizio di rete. Un *cliente* è un qualsiasi nodo o processo che usa il servizio. Il ruolo client-server non è rigido ed un nodo può agire come server per alcune funzioni e come cliente per altre. Nel caso di NFS i rapporti client-server sono indicati nella tabella seguente.

Client	Server
Richiesta di mount remoto	Risposta alla richiesta
Lettura di <code>/etc/fstab</code>	Lettura di <code>/etc/exports</code>
Verifica server conosciuto	Verifica client conosciuto

Il mount remoto inizia sempre dal cliente. Il server completa il collegamento in base alle regole di NFS. Il mount remoto può essere fatto sia con `mount` che con `automount`. Se i file system devono essere montati in permanenza con uso frequente, vengono dichiarati in `/etc/fstab` e montati alla partenza o con il comando `mount -a`. Se i nodi remoti non sono attivi, viene prodotto il messaggio "`server not responding`" e il mount viene sottomesso in `background`, purchè nelle opzioni di `/etc/fstab` sia specificato il parametro `bg`. Tenere presente che per smontare i file system locali senza interferenze da parte di nodi non attivi si deve specificare il parametro `-f` in `umount`, cioè:

```
# umount -fa
```

Se si usa `automount`, il montaggio viene fatto solo quando il file system viene richiesto. La gestione di `automount` è più complessa perchè richiede la creazione e la manutenzione di tabelle gerarchiche con la descrizione dei file system destinati al montaggio. La terminologia di NFS è indicata di seguito.

Server	macchina che fornisce risorse ai clienti
Client	macchina che utilizza le risorse fornite dal server
User	utente loggato sul cliente
Application	programma che gira sul cliente
Export	metodo di comunicazione dei file system dal server ai clienti che possono montare detti file system
RPC	metodo di Remote Procedure Call per gestire le comunicazioni tra server e cliente con semantica di subroutine
XDR	metodi di External Data Representation per la descrizione dei dati remoti

11 GESTIONE DISCHI: UFS E NFS

I processi connessi all'uso e la gestione di NFS sono indicati nel seguente prospetto.

Programmi	Operazioni
<code>nfs</code>	Network File System
<code>biod</code>	Start dei daemon di I/O asincrono
<code>exports</code>	definizione sistemi NFS da esportare
<code>fsirand</code>	installazione generatore random di inode
<code>mount, umount</code>	monta, smonta NFS
<code>mountd</code>	server per le richieste di mount NFS
<code>nfsasynddaemon</code>	invoca daemon NFS
<code>nfsmount</code>	monta Network File System
<code>nfssvc</code>	invoca daemon NFS
<code>nfsu mount</code>	smonta NFS
<code>nfsd</code>	daemon server di NFS
<code>nfssetup</code>	setup di NFS
<code>nfsstat</code>	statistica di NFS
<code>portmap</code>	mapper per DARPA INTERNET
<code>mtab</code>	tabelle dei sistemi locali montati da clienti NFS remoti
<code>rpcinfo</code>	informazione su RPC
<code>showmount</code>	elenco file system montati remoti

La personalizzazione di NFS può essere agevolata dall'uso di `nfssetup`, una procedura interattiva in cui sono stabiliti i valori di default dei parametri e delle opzioni che consente di creare NFS anche al neofita.

11.4 CREAZIONE DI NETWORK FILE SYSTEM

Per creare un file system di rete, bisogna installare NFS e verificare che nel file di configurazione `/usr/sys/conf/mips/HOST` sia presente:

```
options NFS
```

Verificare che il software di NFS sia stato caricato sul sistema con il comando:

```
$ setld -i
```

Se non è stato caricato, installarlo con il comando:

```
# setld -l /dev/rmt0h
```

dove `/dev/rmt0h` indica l'unità nastro contenente il kit di installazione di NFS.

Per l'operazione di NFS devono essere attivi i daemons `portmap` (mappatura programmi), `mountd` (remote mount), `biod` (block I/O) e `nfsd` (server). Per un carico normale di lavoro sono *necessari* 4 daemons `nfsd` e 4 daemons `biod`.

11 GESTIONE DISCHI: UFS E NFS

11.5 FORMATO DEL FILE /ETC/FSTAB

Il mount di NFS da parte del cliente avviene tramite le specifiche di `/etc/fstab` in cui vanno inseriti gli eventuali mount di file system che si vogliono *importare* dai server. La sintassi di `fstab` per NFS è la seguente:

`spec:file:type:freq:passno:name:options`

dove:

<code>spec</code>	path del file system remoto e nome server: <code>path/@remote_node</code> es: <code>/usr/users/cern@ds5ze1</code>
<code>file</code>	nome del file system del cliente (il file system del cliente viene creato con le stesse regole dei file system locali)
<code>type</code>	operazioni ammesse, cioè read-write, read-only, etc. <code>rw</code> - read-write, <code>ro</code> - read only, etc.
<code>freq</code>	frequenza di dump
<code>passno</code>	ordine di <code>fsck</code> al reboot
<code>name</code>	tipo di file system = <code>nfs</code> (locale= <code>ufs</code>)
<code>options</code>	opzioni di mount

Ad esempio per il mount remoto di `/usr/man`:

```
/usr/man@ds5ze2:/usr/man:ro:0:0:nfs:soft,bg:
```

Ricordare che `/etc/fstab` *non accetta commenti*. Se non si vuole montare un file system al prossimo rebot, bisogna copiare `fstab` con altro nome e cancellare i file system che non interessano. Per un file system remoto la frequenza di dump e l'ordine del check al reboot vengono posti a zero. Indicare sempre `bg` perché il reboot non fallisca andando in *hang* nel caso in cui un server sia down. Usare l'opzione *hard* per mantenere attivi file system su server lenti. Sebbene i nomi dei file system locali (client) non debbano essere uguali a quelli remoti (server), è prassi corrente porre i due nomi uguali.

11.6 SETUP MANUALE DI UN SERVER NFS

Si deve creare o aggiornare il file `/etc/exports` con l'informazione del file system o della directory da esportare e dei nodi ai quali si concede l'accesso. Ad esempio:

Esportare `/usr/users/cern` solo sul nodo `testlib`:

```
/usr/users/cern testlib
```

Esportare `/usr/users/cern` come `read only` sul nodo `prod`:

```
/usr/users/cern -o prod
```

Esportare `/usr/users/cern` su tutti i nodi:

```
/usr/users/cern
```

Esportare `/usr/users/cern` sul nodo `manager` con accesso di *su*:

11 GESTIONE DISCHI: UFS E NFS

```
/usr/users/cern -r=0 manager
```

L'esportazione non è vincolata ad un intero file system ma può essere limitata ad una sola directory e relative subdirectories. I nodi specificati possono anche essere definiti in un database di Yellow Pages e quindi appartenere ad un netgroup di YP. I database di Unix sono descritti nel paragrafo dedicato alle Pagine Gialle. Le radici che si vogliono esportare devono essere indicate esplicitamente. Esportare / non implica l'esportazione di tutto il sistema, se si vuole esportare anche /usr si deve inserire l'entry in /etc/exports, altrettanto per /usr/users se questo è un file system separato da /usr. In caso di specifiche multiple, è valida solo la prima. Si può esportare un intero file system read-only ed esportare separatamente una sua subdirectory read-write. Per motivi di sicurezza montare i file system ufs con l'opzione nodev e quelli esportati con -r=0, montarli ufs con l'opzione nosuid.

Una volta modificato /etc/exports, lo si rende attivo con:

```
# /usr/etc/showmount -e
```

L'informazione dei file system montati e dei relativi nodi sono mantenute da mountd e scritte sul file /etc/rmtab, showmount lista dette informazioni e contemporaneamente (opzione -e) aggiorna il database in memoria ricavando le informazioni statiche dal file /etc/exports. Oltre a preparare il file /etc/exports, per attivare NFS ad ogni boot editare il file /etc/rc.local verificando che siano presenti le chiamate per attivare le seguenti funzioni.

1. Configurazione ethernet:

```
/etc/ifconfig lo0 localhost
```

2. Attivazione del daemon portmap:

```
if [ -f /etc/portmap ]; then
    /etc/portmap ; echo ' portmap.' >/dev/console
fi
```

3. Attivazione del daemon mountd:

```
if [ -f /etc/mountd -a -f /etc/portmap -a -s /etc/exports ]; then
    /etc/mountd -i ; echo -n ' mountd -i' >/dev/console
fi
```

4. Attivazione il daemon nfsd:

```
if [ -f /etc/nfsd -a -f /etc/portmap ]; then
    /etc/nfsd 4 ; echo -n ' nfsd' >/dev/console
fi
```

11 GESTIONE DISCHI: UFS E NFS

Per attivare NFS in ambiente single-user, fare reboot del sistema:

```
# /etc/shutdown -r now
```

In ambiente multiuser dare i comandi:

```
# /etc/portmap
# /etc/mountd
# /etc/nfsd 4 &
```

11.7 SETUP MANUALE DI UN CLIENTE NFS

Per un cliente va corretto il file `/etc/fstab` come descritto sopra. Inoltre per attivare NFS ad ogni boot editare il file `/etc/rc.local` verificando che siano presenti le chiamate per attivare le seguenti funzioni.

1. Configurare ethernet:

```
/etc/ifconfig lo0 localhost
```

2. Attivare il daemon portmap:

```
if [ -f /etc/portmap ]; then
    /etc/portmap ; echo ' portmap.' >/dev/console
fi
```

3. Attivare il daemon biod:

```
if [ -f /etc/biod ]; then
    /etc/biod 4 ; echo ' biod' >/dev/console
fi
```

4. Attivare il daemon rwalld:

```
if [ -f /usr/etc/rwalld -a -f /etc/portmap ]; then
    /usr/etc/rwalld ; echo 'rwall daemon: rwalld' >/dev/console
fi
```

Per attivare NFS in ambiente single-user, fare reboot del sistema:

```
#/etc/shutdown -r now
```

in ambiente multiuser dare i comandi:

```
# /etc/portmap
# /usr/etc/rwalld &
# /etc/biod 4 &
```

11 GESTIONE DISCHI: UFS E NFS

Per riattivare NFS dopo il boot, cancellare i processi portmap e mountd con i comandi:

```
# ps -aux | egrep "biold|mountd"
# kill -9 portmap-pid
# kill -9 mountd-pid
```

e riattivare NFS manualmente con i comandi:

```
# /etc/portmap
# /etc/mountd
# /etc/nfsd 4 &
```

Per verificare lo stato dei processi sui nodi:

```
# /etc/rpcinfo -p HOST
```

Le informazioni stampate dal sistema per un *cliente* sono del tipo:

```
$ /etc/rpcinfo -p ds5ze2
  program vers proto  port
  100007    2    tcp   1024  ypbind
  100007    2    udp   1043  ypbind
  100007    1    tcp   1024  ypbind
  100007    1    udp   1043  ypbind
  100005    1    udp   1061  mountd
  100005    1    tcp   1027  mountd
  100003    2    udp   2049  nfs
```

Le informazioni stampate dal sistema per un *server* sono del tipo:

```
$ /etc/rpcinfo -p ds5ze1
  program vers proto  port
  100004    2    udp   1027  ypserv
  100004    2    tcp   1024  ypserv
  100004    1    udp   1027  ypserv
  100004    1    tcp   1024  ypserv
  100007    2    tcp   1025  ypbind
  100007    2    udp   1036  ypbind
  100007    1    tcp   1025  ypbind
  100007    1    udp   1036  ypbind
  100009    1    udp   1023  yppasswdd
  100005    1    udp   1058  mountd
  100005    1    tcp   1028  mountd
  100003    2    udp   2049  nfs
```

Dalla lista globale dei processi fornita da `/etc/rpcinfo`, si può verificare lo stato di ogni singolo processo usando il numero del processo listato nella colonna `program`:

```
# /etc/rpcinfo -u ds5ze2 100005
```

11 GESTIONE DISCHI: UFS E NFS

Se il programma è in funzione si ottiene un messaggio del tipo:

```
program 100005 version 1 ready and waiting
```

Il mount può essere fatto anche manualmente (valido solo fino al prossimo boot) con la seguente sintassi:

```
# mount -t nfs -o soft,bg ds5ze2:/usr/users/cern /mnt
```

dove:

-t **nfs** indica file system di tipo network
-o **soft** error code (soft) per nodo remoto down
-o **bg** bg indica di ritentare il mount in background se il primo tentativo fallisce

Segue il nome del file system fisico sul nodo remoto e il nome del file system logico sul nodo locale. In pratica il comando precedente indica che il file system /mnt è un file system remoto appartenente fisicamente al nodo ds5ze2. Le opzioni di mount che possono essere specificate con -o sono indicate di seguito. Per maggiori informazioni, consultare man con il comando:

```
$ man 8nfs mount
```

Opzioni di mount NFS:

rw	consente accesso read-write al file system remoto
ro	file system remoto con accesso read only
bg	se il primo tentativo fallisce, riprova in background
retry=n	pone=n il numero di tentativi prima di fallire
rsize=n	pone la dimensione del read buffer a n byte
wsiz=n	pone la dimensione del write buffer a n byte
timeo=n	pone il timeout di NFS a n decimi di secondo
retrans=n	pone il numero di ritrasmissioni =n
port=n	setta il valore del port IP
soft	setta un errore se il server non risponde
hard	ritenta fino a che il server non risponde o il processo viene terminato
intr	consente di interrompere operazioni di mount hard
nosuid	programmi di su non possono essere eseguiti dal file system
noexec	immagini binarie non possono essere eseguite dal file system

Per smontare un file system remoto, verificare che il file system non sia in uso ed usare il comando:

```
# /etc/umount ds5ze4:./usr/man
```

Il comando /etc/mount senza parametri fornisce l'informazione completa sui file system in uso:

11 GESTIONE DISCHI: UFS E NFS

```
$ /etc/mount
/dev/rz0a on / type ufs
/dev/rz0g on /usr type ufs
/dev/rz1f on /usr/users type ufs
/dev/rz1d on /dlclient0 type ufs
/dev/rz1e on /dlenv0 type ufs
ds5ze2:/ds5ze2 on /ds5ze2 type nfs (rw,hard,intr,bg)
ds5bo1:/usr/utenti on /ds5bo1 type nfs (rw,hard,bg)
```

Da ricordare che un file system può essere ereditato solo se designato esportabile da parte del nodo proprietario. Tutti i file system di carattere locale possono essere esportati. File system di sistema **comunemente** esportati sono:

/usr/man	per centralizzare l'informazione di man
/usr/src	file sorgenti da usare con <i>sccs</i> per centralizzare gli update
/usr/sys	file di costruzione del sistema per centralizzare il kernel
/usr/local	utility locali (es. librerie CERN)

File system da **NON** esportare:

/etc	versione locale di <i>ttys</i> , <i>fstab</i> , <i>passwd</i>
/bin	routine critiche per crash di sistema
/dev	accesso remoto ai device NON supportato
/usr/spool	le directory di spool sono specifiche del sistema

File system da esportare con **cautela**:

/tmp	area di lavoro temporaneo, per NFS va partizionata tra i clienti creando opportune subdirectory
/usr/users	in NFS potrebbe aprire la strada ad accessi non autorizzati
/usr/spool/mail	crea problemi di sicurezza

Quando si opera su un file system remoto, notificare *shutdown* e simili con *rwall*d.

11.8 NFS SECURITY

Controlli su NFS e sull'accesso ai file possono essere fatti attivando *nfsportmon*. Si può migliorare la sicurezza del sistema con *fsirand* che randomizza la generazione dei numeri *inode*. Norme generali di sicurezza implicano un controllo del file */etc/exports* per limitare l'accesso a file system non critici, un controllo di tutti i file *.rhosts* per limitare il numero di nodi che possono eseguire *rsh*, utilizzare *sccs*, cioè il meccanismo di Source Code Control System.

12 MANUTENZIONE FILE SYSTEM

12 MANUTENZIONE FILE SYSTEM

12.1 CONTROLLO FILE

Per una buona manutenzione del sistema, occorre tenere sotto controllo l'occupazione disco e l'integrità dei file. Inoltre è necessario sapere cosa fare quando il sistema dà il messaggio:

File system full

I file possono avere i modi **rw**x per read-write-execute e i modi speciali, che si applicano solo agli eseguibili:

- 1000 sticky bit per programmi molto usati - migliora le prestazioni conservando swap space
- 2000 setta group ID uguale all'ID dell'owner (sistema)
- 4000 setta user ID uguale all'ID dell'owner (sistema)

Maggiori informazioni sulla nomenclatura dei file sono reperibili nei manuali di Unix (vedi note bibliografiche).

12.2 LOG FILE

Ultrix è in grado di registrare errori locali e remoti. Per generare un core dump in caso di **crash** del sistema inserire in **/etc/rc.local**:

```
/etc/savecore /usr/adm/crash >/dev/console
```

Il core dump **vmcore** e **vmunix** sono memorizzati in **/usr/adm/crash**. Per memorizzare il solo error log in caso di spazio disco insufficiente:

```
/etc/savecore -e /usr/adm/crash.
```

Verificare periodicamente l'occupazione dei file di log per contenerne la dimensione. Usare il comando **df** per verificare lo spazio disco. I file di log creati dal sistema sono posti in:

/usr/adm/syserr/	errori sistema
/usr/adm/	dati di accounting
/usr/spool/mqueue/	errori critici di sistema

Da notare che la directory è **/var/adm** con link **/usr/adm** e **/usr/var/adm**. I file di log di uso corrette sono:

/var/adm/fverifylog	verifica file system, directory, gid
/var/adm/sulog	log del comando su
/var/adm/shutdownlog	log del comando shutdown

I file vengono creati dai daemon:

12 MANUTENZIONE FILE SYSTEM

/etc/accton	accounting comandi - disabilitabile
/etc/elcsd	messaggi d'errore
/etc/syslog	errori critici

I daemon `accton` e `elcsd` vengono attivati da `/etc/rc`, mentre `syslog` è attivato da `/etc/rc.local`. Il daemon `elcsd` trasferisce i log da memoria a file ed è pilotato dal file `/etc/elcsd.conf` che può essere modificato solo da `/etc/eli`.

I daemon `elcsd` e `syslog` sono pilotati dai file di configurazione:

```
/etc/elcsd.conf
/etc/syslog.conf
```

che definiscono i file di log e l'eventuale ambiente. La configurazione specifica se il log è locale o remoto e il path dell'error log. Tali dati sono indicati esplicitamente dai commenti contenuti in `elcsd.conf`. I parametri di `elcsd.conf` sono:

<p>stato dimensione del file di log path del file di log path della directory di backup per il file di log path del file di log in single-user path per il log dei nodi remoti nome del nodo remoto nodi di cui fare il log</p>
--

Lo stato indica gli errori che devono essere conteggiati. Gli errori possibili sono:

1	local	log messaggi locali
2	logrem	log messaggi dei sistemi remoti
3		log dei messaggi locali e remoti
4	remlog	log dei messaggi locali su nodo remoto
5	remlog+priloglocal	log dei messaggi locali su nodo remoto e log locale dei messaggi di high priority

I messaggi possono essere: **severe**, ad alta priorità (**high**), a bassa priorità (**low**). Un errore **severe** indica errori fatali che implicano il fermo del sistema. Un messaggio **high** indica errori recuperabili hardware. Un messaggio **low** indica errori soft, restart e CRD (corrected read data).

Lo stato di default è 1. Lo stato 2 viene usato se il sistema è un server e si vuole il log di un cliente. Lo stato 3 implica il log sia del server che del cliente. Lo stato 4 indica che il log del cliente va fatto sul server. Lo stato 5 indica che il log di tutti i messaggi viene fatto sul server, mentre il log dei messaggi **severe** e **high** viene fatto anche sul cliente.

Se si specifica la dimensione del file di log si evita la situazione di **File system full** che viene segnalata nel file `/usr/adm/elcsdlog`. Si deve comunque tenere sotto controllo la dimensione del log perché il sistema sospende la memorizzazione dei messaggi di errore se lo spazio è insufficiente. Se non si specifica una directory di log, il path usato è

12 MANUTENZIONE FILE SYSTEM

`/usr/adm/syserr`. Quando il file system si riempie, si può indicare una directory di backup su un file system alternativo. Non esistono default per questo parametro.

I file di errore di `elcsd` si trovano nella directory `/usr/adm/syserr/` e vengono creati con un nome del tipo `syserr.nodename`. I file di errore di `syslog` si trovano nella directory `/usr/spool/mqueue/`, si chiamano `syslog`, `syslog.0`, `syslog.1`, etc. fino a `syslog.7` e vengono periodicamente rinominati su base giornaliera, sono segnalati gli errori di priorità 8. In generale i file vengono creati alle ore 4:05 di ogni nuovo giorno e memorizzati quindi per circa una settimana. Oltre alla scrittura su file, in caso di errore grave il messaggio viene segnalato anche agli utenti indicati in `syslog.conf` (default root). Il motivo di shutdown viene registrato nel file `/usr/adm/shutdownlog`.

12.3 RESIZE SYSERR. USO DI ELI E UERF

I log di errore vengono trascritti nella directory `/usr/adm/syserr`.
Per disabilitare il log:

```
# /etc/eli -d
```

Per riabilitare il log in single-user, girare `fsck` sul root file system, poi:

```
# /etc/eli -s
```

i file di log vengono scritti su `/`.

Per riabilitare il log in multiuser:

```
# /etc/eli -f -e
```

Per vuotare il buffer di errore:

```
# /etc/eli -i
```

Per interrompere la scrittura dei messaggi di `eli` sulla console:

```
# /etc/eli -q
```

Per riabilitarli:

```
# /etc/eli -w
```

Per trasferire un messaggio di log sul file di errore:

```
# /etc/eli -l
```

seguito dal messaggio al prompt, se il messaggio è su file:

```
# /etc/eli -l < mylog > /dev/null
```

con eliminazione del prompt di `eli` (`>/dev/null`).

Per far ripartire il daemon `elcsd` dopo aver modificato `elcsd.conf`:

```
# /etc/eli -r
```

12 MANUTENZIONE FILE SYSTEM

Per un controllo della dimensione dei file di log:

```
# cd /usr/adm/syserr
# ls -al syserr.*
```

Esiste un file per ogni nodo con nome `syserr.nodename`.

Per analizzare il contenuto del file usare `/etc/uerf` e precisamente:

```
# uerf -f file_name
```

Per il resize:

```
# rm file_name
```

seguito dalla riabilitazione dell'error log con:

```
# eli -f -e
```

Per maggiori informazioni, si rimanda al manuale *Guide to Error Logger System*.

Per formattare i file di errore si usa `/etc/uerf` che non richiede privilegi. Per help:

```
# /etc/uerf -h
```

Si possono selezionare gli errori, le date, gli hosts, etc. con formato di stampa breve (`-o brief`), completo (`-o full`) o condensato (`-o terse`). Il programma `/etc/uerf` fa riferimento ai file ausiliari `uerf.bin` `uerf.err` `uerf.hlp` che possono trovarsi in `root (/)` o in `/etc`. Di seguito sono indicati gli esempi di utilizzo più frequente.

Log errori di memoria:

```
# /etc/uerf -M mem
```

Log condensato:

```
# /etc/uerf -o terse -M mem
```

Log da file specifico:

```
# /etc/uerf -f /usr/adm/syserr/syserr.ds5ze7.old
```

Log in tempo reale:

```
# /etc/uerf -n
```

Log errori specifici (vedi `man` per il significato) in ordine cronologico inverso:

```
# /etc/uerf -o terse -R r 300 | more
```

con questo comando si ottengono informazioni sulle operazioni di startup. Se si usa il codice 301 si ottengono informazioni di shutdown. Indirettamente il comando **fornisce informazioni** sull'hardware quale memoria disponibile, floating point, processore, dischi, ethernet, etc.

Log degli errori della giornata:

12 MANUTENZIONE FILE SYSTEM

```
# /etc/uerf -t s:00 -S | more
```

Log degli errori hardware di memoria e cpu:

```
# /etc/uerf -M mem,cpu | more
```

Log degli errori hardware e software:

```
# /etc/uerf -c err -R | more
```

Per produrre un Log analogo al file `errlog.sys` del VMS:

```
# /etc/uerf -c oper _R | more
```

Log degli errori di sistema operativo:

```
# /etc/uerf -O -R | more
```

Log di errori specifici:

```
# /etc/uerf -r 101,102,104,111 -R | more
```

dove:

100	machine check
102	disk error
104	device controller
111	console timeout

Per altri errori hardware vedi `man uerf`.

Per la selezione temporale del log:

```
# /etc/uerf -t s:23-oct-1991,00:00:00 e:25-oct-1991,23:59:59
```

in cui `-t` specifica selezione temporale ed `s:` `e:` indicano data e ora rispettivamente iniziali e finali.

Log degli errori eccetto (`-x`) quelli di sistema operativo (`-O`) e quelli di disco (`-D`):

```
# /etc/uerf -O -x -o full -D
```

12.4 RESIZE DEI FILE DI ACCOUNTING

Il sistema produce due tipi di file di accounting:

<code>/usr/adm/wtmp</code>	account user
<code>/usr/adm/acct</code>	account processi

Per analizzare l'account utenti usare:

```
$ /etc/ac -p      totale utenti  
$ /etc/ac -p -d  totale utenti su base giornaliera
```

Per il resize dello user account, ricreare il file vuoto con:

12 MANUTENZIONE FILE SYSTEM

```
# cat /dev/null > /usr/adm/wtmp
```

Per analizzare l'account dei processi:

```
# /etc/sa
```

Per il resize del process account:

```
# /etc/sa -s
```

Il file `acct` viene ridotto di dimensione e viene creato un file di archivio compresso di nome `savacct`.

Per disabilitare l'account di processi:

```
# /etc/accton
```

Per la disabilitazione permanente, oltre a disabilitare l'account attivo con:

```
# /etc/accton
```

con l'editor in `/etc/rc` inserire `#` per commentare la riga:

```
# /etc/accton /usr/adm/acct
```

in modo da disabilitare l'account al boot. Infine per liberare lo spazio disco:

```
# rm /usr/adm/acct
```

```
# rm /usr/adm/savacct
```

Per avere statistiche sulle stampanti del tipo: no. di pagine per utente, no. di accessi per utente, costo della carta basato sul default di \$ 0.02 per pagina, etc. si deve creare il file `lpacct`:

```
# cat /dev/null > /usr/adm/lpacct
```

Per abilitare l'accounting della stampante, verificare che nel file `/etc/printcap` l'entry relativo contenga il parametro `af=/usr/adm/lpacct:.` La statistica viene prodotta da:

```
# /etc/pac
```

Per cambiare il costo della carta:

```
# /etc/pac -p2.2
```

Il file di account della stampante può essere compresso con:

```
/etc/pac -s
```

L'account viene disabilitato eliminando il campo `af` dal `/etc/printcap` e cancellando i file di account per risparmiare spazio.

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

Il sistemista deve essere in grado di aggiornare il sistema operativo ogni volta che interviene qualche variazione dovuta a una versione nuova del software oppure a una modifica dell'hardware. L'aggiornamento può interessare il kernel o un applicativo, ad esempio un compilatore o una libreria grafica. Se si aggiorna il kernel, si deve avere un backup del sistema e si deve esercitare la massima cautela perché un errore operativo può portare ad un sistema che non fa il boot e che costringe ad operazioni di emergenza con possibile perdita di dati. L'installazione di prodotti applicativi è usualmente meno critica e pone dei vincoli solo sullo spazio disco disponibile. Si consiglia quindi di creare *sempre* delle partizioni dimensionate adeguatamente per consentire aggiornamenti e ampliamenti del software.

La configurazione della macchina viene testata al boot con autoconf. Informazioni relative ad autoconf si ottengono con:

```
$ man autoconf rz tz
```

13.1 FILE DI CONFIGURAZIONE E COSTRUZIONE DEL KERNEL

Tutte le capacità del sistema sono descritte nel file di configurazione che pilota la costruzione del kernel. Il file di configurazione si trova per ciascun nodo su:

```
/usr/sys/conf/mips/HOST
```

In caso di sistemi client-server, i file di configurazione dei clienti si trovano sulla directory del server:

```
/dlclient0/host.root/sys/conf/mips/
```

il nome del file di configurazione è HOST, dove HOST è il nome del nodo in *maiuscole*. Analogamente per i clienti il file si trova in una subdirectory della radice dlclientN dove N vale usualmente 0, ma potrebbe prendere anche altri valori ad esempio (1,2,, etc. per ogni nuova famiglia di cliente (es. MIPS, VAX, etc.) e la subdirectory ha nome host.root dove host è il nome del nodo in minuscole. Il nome del nodo si ottiene con:

```
$ hostname
```

Ogni volta che viene fatta una modifica hardware o software occorre verificare che il kernel sia adeguato alla nuova situazione analizzando il file di configurazione. Ad esempio, nel config file compaiono i campi:

```
cpu      "DS5000"  
physmem  24
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

che indicano una macchina di tipo DECstation 5000 con 24 Mbyte di memoria. Se si aggiunge memoria, ad esempio $24+16=40$ Mbyte, si deve modificare il campo `physmem` e rigenerare il kernel.

Il config file viene generato da un template che si trova alla stessa directory e si chiama `GENERIC`. Il file di configurazione contiene le informazioni relative all'hardware e al software del sistema sotto forma di definizioni appartenenti alle seguenti classi:

definizioni globali
definizioni opzionali
definizioni makeoptions
definizioni per le immagini di sistema
definizioni dei device
definizioni degli pseudodevice

Le definizioni *globali* si applicano a tutti i kernel generati dal file di configurazione. Ogni definizione compare su una linea separata. I parametri globali servono per il tuning della macchina. Parametri globali sono: `machine`, `cpu`, `maxuser`, `physmem`, etc.

Le definizioni *opzionali* indicano codice opzionale che deve essere compilato nel kernel. Non dovrebbero essere cambiate dal sistemista. Opzioni sono: `DECNET`, `LAT`, `QUOTA`, etc.

Per i processori RISC si ha una sola definizione *makeoptions* che riguarda l'ordine dei byte all'interno delle parole usate dal processore e che deve valere:

```
makeoptions  ENDIAN="-EL"
```

Una descrizione completa delle opzioni di sistema si trova nel manuale `Guide to System Configuration File Maintenance`.

C'è una definizione di sistema nel file di configurazione per ogni kernel che si vuole generare. Tale definizione inizia con la parola chiave `config` e contiene il nome del kernel e i device per `root`, `page-swap` e `dump`, ad esempio:

```
config  vmunix  root on rz0a  swap on rz0b  dumps on rz0b
```

Le definizioni dei device descrivono i device fisici connessi o previsti, ad esempio: `adapter` (per la connessione fisica), `disk`, `tape`, etc.

Le definizioni di pseudodevice si riferiscono a componenti per cui non esiste un hardware associato, come uno pseudoterminale o un protocollo. Ogni definizione specifica il driver che gestisce lo pseudodevice, ad esempio: `ether`, `lat`, `nfs`, `ufs`, `rpc`, etc.

13.2 ESEMPI DI FILE DI CONFIGURAZIONE

Di seguito è listato il file di configurazione tipico di una DECstation *5000/200*.

```
ident          "GETTIM"
machine        mips
cpu            "DS5000"
maxusers      32
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

```
processors      1
maxuprc         50
physmem         32
timezone        -1 dst 4
smmax           1024
smseg           12

options         QUOTA
options         INET
options         NFS
options         RPC
options         DLI
options         UFS
options         NETMAN
options         LAT
options         CDFS
options         PACKETFILTER
options         AUDIT
options         SYS_TPATH
options         DECNET
options         UWS

makeoptions     ENDIAN="-EL"

config          vmunix  root on rz2a  swap on rz2b  dumps on rz2b

adapter         ibus0      at nexus?
adapter         ibus1      at nexus?
adapter         ibus2      at nexus?
adapter         ibus5      at nexus?
adapter         ibus6      at nexus?
adapter         ibus7      at nexus?
controller      asc0        at ibus?      vector ascintr
disk            rz0        at asc0        drive 0
disk            rz1        at asc0        drive 1
disk            rz2        at asc0        drive 2
disk            rz3        at asc0        drive 3
disk            rz4        at asc0        drive 4
disk            rz5        at asc0        drive 5
disk            rz6        at asc0        drive 6
disk            rz7        at asc0        drive 7
tape            tz0        at asc0        drive 0
tape            tz1        at asc0        drive 1
tape            tz2        at asc0        drive 2
tape            tz3        at asc0        drive 3
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

tape	tz4	at asc0	drive 4
tape	tz5	at asc0	drive 5
tape	tz6	at asc0	drive 6
tape	tz7	at asc0	drive 7
device	ln0	at ibus?	vector lnintr
device	dc0	at ibus?	vector dcintr
device	fb0	at ibus?	vector fbint

scs_sysid 1

pseudo-device nfs
pseudo-device rpc
pseudo-device dli
pseudo-device pty
pseudo-device loop
pseudo-device ether
pseudo-device ufs
pseudo-device netman
pseudo-device inet

pseudo-device lat
pseudo-device lta

pseudo-device scsnet

pseudo-device msdup

pseudo-device cdfs

pseudo-device packetfilter

pseudo-device sys_tpath
pseudo-device audit

pseudo-device decnet
pseudo-device xcons
pseudo-device ws
pseudo-device tc

Il file che segue è relativo alla stessa macchina dopo l'upgrade a **5000/240**. Le modifiche e le definizioni aggiunte rispetto al file di configurazione precedente sono indicate dal simbolo <---- seguito rispettivamente da modi oppure da add. Inoltre sono indicati con **verify** i parametri che potrebbero essere inesatti senza però influenzare il sistema. Ad esempio se il valore di **timezone** è errato si hanno risultati errati nel sincronismo dell'ora mentre tutte le altre funzioni del sistema sono corrette.

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

```
ident          "GETTIM"
machine        mips
cpu            "DS5000_300"          <---- modi
maxusers       32
processors     1
maxuprc        50
physmem        32
timezone       -1 dst 4             <---- verify
smmax          1024
smseg          12

options        QUOTA
options        INET
options        NFS
options        RPC
options        DLI
options        UFS
options        NETMAN
options        LAT
options        CDFS
options        PACKETFILTER
options        AUDIT
options        SYS_TPATH
options        DECNET
options        UWS

makeoptions    ENDIAN="-EL"

config         vmunix root on rz2a swap on rz2b dumps on rz2b

adapter        ibus0      at nexus?
adapter        ibus1      at nexus?
adapter        ibus2      at nexus?
adapter        ibus3      at nexus?          <---- add

adapter        ibus5      at nexus?
adapter        ibus6      at nexus?
adapter        ibus7      at nexus?
controller     asc0       at ibus?      vector ascintr
disk           rz0        at asc0       drive 0
disk           rz1        at asc0       drive 1
disk           rz2        at asc0       drive 2
disk           rz3        at asc0       drive 3
disk           rz4        at asc0       drive 4
disk           rz5        at asc0       drive 5
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

disk	rz6	at asc0	drive 6	
disk	rz7	at asc0	drive 7	
tape	tz0	at asc0	drive 0	
tape	tz1	at asc0	drive 1	
tape	tz2	at asc0	drive 2	
tape	tz3	at asc0	drive 3	
tape	tz4	at asc0	drive 4	
tape	tz5	at asc0	drive 5	
tape	tz6	at asc0	drive 6	
tape	tz7	at asc0	drive 7	
device	ln0	at ibus?	vector lnintr	
device	scc0	at ibus?	vector sccintr	<---- modi
device	fb0	at ibus?	vector fbint	
scs_sysid	1			
pseudo-device	nfs			
pseudo-device	rpc			
pseudo-device	dli			
pseudo-device	pty			
pseudo-device	loop			
pseudo-device	ether			
pseudo-device	ufs			
pseudo-device	netman			
pseudo-device	inet			
pseudo-device	lat			
pseudo-device	lta			
pseudo-device	scsnet			
pseudo-device	msdup			
pseudo-device	cdfs			
pseudo-device	packetfilter			
pseudo-device	sys_tpath			
pseudo-device	audit			
pseudo-device	decnet			
pseudo-device	xcons			
pseudo-device	ws			
pseudo-device	tc			

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

13.3 COSTRUZIONE AUTOMATICA DEL KERNEL

Per costruire un nuovo kernel si può usare il comando `/etc/doconfig` oppure procedere manualmente. Un nuovo kernel deve essere costruito ogni volta che vengono modificati i parametri del file di configurazione, ad esempio per l'aggiunta di device o pseudodevice, la modifica di un parametro globale, un upgrade hardware e/o software del sistema. Per aggiornare il kernel partendo da un file di configurazione esistente con `/etc/doconfig`, il sistema deve usare il kernel generico `genvmunix`. Come *su* eseguire i seguenti steps:

```
1. # /etc/shutdown +5 "Aggiornamento kernel"
   # mv /vmunix /sys/vmunix.old
   # cp /genvmunix /vmunix
   # /etc/halt
```

2. Fare reboot del processore in single-user, verificare il file system e montare solo il file system locale (ufs):

```
# /etc/fsck -p
# /etc/mount -a -t ufs
```

3. Attivare error log e girare update:

```
# eli -s
# /etc/update
```

4. Salvare il file di configurazione esistente:

```
# cd /sys/conf/mips
# cp HOST HOST.old
```

5. Definire l'editor che si vuole usare durante l'operazione:

```
# EDITOR=vi
# export EDITOR
```

Usare `setenv` per `csh`.

6. Invocare `doconfig`:

```
# cd /
# /etc/doconfig -c DS5ZE1 -e vi
```

Se non si specifica il nome del file, `doconfig` crea un template non necessariamente adeguato al proprio hardware e software. Il programma opera in modo interattivo con domande sul nodo, la data, l'editor del file di configurazione, etc. Il nome del nodo deve essere minuscolo o come compare in `/etc/rc.local` nella definizione:

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

```
/bin/hostname ds5ze1
```

Da non confondere il nome del file che è *sempre maiuscolo* con il nome del nodo che generalmente è minuscolo. Prima di procedere si consiglia di verificare la validità del nuovo file di configurazione richiamando `diff` dall'editor:

```
!diff /sys/conf/mips/HOST /sys/conf/mips/HOST.old
```

Per rendere attivo il nuovo kernel:

```
# mv /sys/MIPS/HOST/vmunix /vmunix
# chmod 755 /vmunix
# /etc/reboot
```

13.4 COSTRUZIONE MANUALE DEL KERNEL

Si può fare sia in `single-user` che in `multiuser` sebbene, per garantire l'integrità del sistema, sia più sicuro operare in `single-user`. Le operazioni richieste sono:

- con `shutdown` portare il sistema a `single-user`

```
# /etc/shutdown +5 "Nuovo kernel"
```
- correggere il file di configurazione

```
# cd /sys/conf/mips
# cp HOST HOST.old
# chmod +w HOST
# vi HOST
```
- usare l'utility `config`

```
# config HOST
```

l'utility crea (se non esiste già) la directory `/sys/MIPS/HOST`
- definire le dipendenze

```
# cd /sys/MIPS/HOST
# make clean
# make depend
```

con le due chiamate di `make` si ripulisce la directory e si creano i nuovi binari richiesti dalla variazione di configurazione
- compilare il kernel

al seguito delle operazioni precedenti viene creato un `makefile` che consente di compilare il nuovo kernel:

```
# make
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

- boot del nuovo kernel

```
# mv /vmunix /sys/vmunix.old
# mv vmunix /vmunix
# chmod 755 /vmunix
# /etc/reboot
```

13.5 ESEMPIO DI UPDATE

Occasionalmente l'installazione di un subset (ad esempio un compilatore) richiede che venga applicato un aggiornamento al kernel. In questo caso vengono forniti i file da modificare (ad esempio `fp_intr.o` e `softfp.o`) che vanno copiati nella directory `/sys/MIPS/BINARY`. Si deve poi rigenerare il kernel con i comandi:

```
# EDITOR=vi; export EDITOR
# /etc/doconfig -c NODE_NAME      (es: /etc/doconfig -c DS5ZE2)
# /etc/shutdown +5 "Update"
```

Si deve fare il rename del vecchio kernel (`vmunix` che si trova nella root di sistema `/`), chiamandolo `vmunix.old` o qualsiasi nome adeguato alla propria configurazione. Si deve copiare il nuovo kernel da `/sys/MIPS/NODE_NAME` sulla root `/`:

```
# mv /sys/MIPS/DS5ZE2/vmunix /vmunix
```

Infine rifare il boot del sistema per rendere operativo il kernel appena costruito:

```
# /etc/reboot
```

13.6 SETUP DI UN NUOVO SISTEMA OPERATIVO

Con l'evolversi delle esigenze e l'aggiornamento dell'hardware può essere necessario modificare l'assetto delle macchine con l'obbligo di spostare il sistema operativo su un nuovo disco o di uno scambio di funzioni tra client e server.

Per operare uno spostamento del sistema operativo dal disco attuale ad un nuovo disco, ad esempio per disporre di maggiore memoria di massa, si deve fare il dump di tutti i file system esistenti sul sistema in uso. Si creano le partizioni necessarie con dimensioni adeguate sul nuovo disco e si fa il restore del software con comandi del tipo:

```
# mount /dev/rz1a /mnt
# cd /mnt
# restore -rvsf 4 /dev/nrmt1h      (restore 4-th tape file)
# cd /
# umount /mnt
# mount /dev/rz1g /mnt
# cd /mnt
# restore -rvf /dev/nrmt1h
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

Da notare che il mount senza `fsck` è possibile solo se la partizione è vuota. In ogni caso il restore va fatto su partizioni vuote per evitare di riempire il file system e specialmente per non correre il rischio di avere vecchi file non usati dalla nuova versione del sistema che generano interferenze.

Nell'esempio su indicato vengono ripristinate le partizioni **a** e **g** da un dump multi-file. I parametri indicati nelle operazioni di restore hanno il seguente significato:

```
r    restore di un full dump sulla directory corrente
v    verbose
s    posiziona il nastro sul file indicato nell'argomento (ad esempio 4)
f    richiesto per unita' nastro diversa dal /dev/rmt0h
```

Inoltre usare il device `/dev/nrmt0h` invece di `/dev/rmt0h` per inibire il rewind.

Ripristinate le partizioni si deve creare un nuovo kernel, correggendo nel file di configurazione `/usr/sys/conf/mips/HOST-NAME` la riga `config` che contiene le informazioni di boot, cioè file di kernel, disco e partizioni di root, swap e dump. A questo punto per rifare il boot dal nuovo disco usare i comandi:

```
# shutdown -h now
# boot 5/rz2/vmunix -a
```

Se l'operazione avviene correttamente, ripristinare l'ambiente di boot usando `setenv` in modo da attivare il nuovo disco.

Se, oltre a modificare il disco di sistema, si vuole anche scambiare il ruolo di server tra due diverse cpu, occorre modificare l'assetto hardware di modo che la cpu server disponga di memoria di massa adeguata alle sue funzioni. Si devono analizzare i file di configurazione delle macchine che si vogliono scambiare per correggere tutti i parametri che intervengono nel ruolo server-client.

In particolare, oltre a correggere i dati di boot (da disco per il server, da eth per il cliente) e di swap-dump, verificare che il parametro `UWS` sia presente solo sulle macchine che sono workstation e che la configurazione del server contenga i parametri `AUDIT` (controllo dei login) e `SYS_TPATH` (yellowpages). Verificare inoltre che il parametro `EMULFLT` (emulazione floating point) compaia nelle macchine che già lo contenevano.

13.7 INSTALLAZIONE SOFTWARE SUBSET

Come già detto in precedenza, il sistema operativo dovrebbe venire installato con tutti i subset anche se alcuni non vengono utilizzati in modo esplicito, dato che applicativi che verranno installati in un secondo tempo potrebbero fare riferimento a subset mancanti.

Per ogni subset installato nella directory `/usr/etc/subsets` vengono memorizzati i file: `*.ctrl *.inv *.lk *.scp` dove `*` è il nome del subset, ad esempio `UDTBASE420`. I file contengono le seguenti informazioni:

```
ctrl  informazioni di controllo (nome prodotto, parametri, etc)
inv   elenco file installati
lk    elenco eventuali subset connessi (generalmente vuoto)
scp   shell di installazione con config e IVP
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

A titolo di esempio, seguono i file CXXBASE125.* per l'installazione del prodotto c++.

```
$ page /usr/etc/subsets/CXXBASE125.ctrl
```

```
NAME='DEC C++ for ULTRIX on RISC CXXBASE125'  
DESC='DEC C++ (cxx) For RISC'  
NVOLS=1:7  
MTLOC=1:1  
DEPS="."  
FLAGS=2
```

Il file .inv contiene la lista dei file creati dall'installazione come indicato nel frammento seguente, in cui la lettera indica se si tratta di file f o di link s:

```
$ page /usr/etc/subsets/CXXBASE125.inv  
0      6      00000  0      0      120755  7/28/93 125      s  
./usr/bin/cxx  cxx2.1  CXXBASE125  
0      27      00000  0      0      120755  7/28/93 125      s  
./usr/bin/cxx2.1  ../lib/cmplrs/cxx2.1/driver  CXXBASE125  
0      11      00000  0      0      120755  7/28/93 125      s  
./usr/bin/demangle  demangle2.1  CXXBASE125  
0      31      00000  0      0      120755  7/28/93 125      s  
./usr/bin/demangle2.1  ../lib/cmplrs/cc2.1/demangle2.1  CXXBASE125  
0      21832  06486  0      0      100644  8/24/93 125      f  
./usr/lib/cmplrs/DECCXX120A.release-notes  none  CXXBASE125  
.....
```

Il file .lk è quasi sempre vuoto e viene usato con l'opzione setld -d per disinstallare il subset. Il file .scp è uno shell script che contiene la procedura di installazione ed eventualmente la procedura di IVP.

L'informazione di /usr/etc/subsets viene usata da /etc/setld -i per produrre la tabella di tutti i subset esistenti con il flag installed se il subset è stato installato, senza detto flag se il subset è stato disinstallato. Se il subset non compare non è mai stato installato. Ogni subset può essere disinstallato in qualsiasi momento. L'operazione rimuove anche tutti i link connessi al subset. I subset possono essere configurati e verificati. Per alcuni prodotti viene fornita una procedura di test come per il VMS nota come Installation Verification Procedure (IVP).

Le operazioni sui subset al di fuori dell'installazione del sistema, vengono eseguite da /etc/setld che consente la gestione dei subset con operazioni di indice, verifica, installazione, rimozione, caricamento da nastro a disco del distribution etc. Le opzioni previste sono indicate nella tabella seguente.

- l installa il subset
- d cancella il subset
- i produce l'indice dei subset esistenti e installati
- v verifica che il subset sia integro ed esegue, se disponibile, IVP
- c configura il subset
- x copia il subset da nastro a disco

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

Esempi:

Installazione dei subset dall'unità nastro 2:

```
# setld -l /dev/rmt2h
```

Normalmente si ha un solo device di tipo nastro (TK50, DAT, EXABYTE, etc.) che viene visto come /dev/rmt0h.

Installazione del subset UDTUUCP400 dall'unità nastro 2:

```
# setld -l /dev/rmt2h UDTUUCP400
```

Installazione del subset da CDROM nel caso in cui il CD contenga dei programmi applicativi e non contenga il sistema operativo:

```
# mount -r /dev/rz4c /mnt  
# setld -l /mnt/TPU300/bin
```

Dato che il CDROM è un disco a tutti gli effetti, dopo averlo montato è consigliabile esaminarne il contenuto prima di procedere con le installazioni.

Installazione del subset UDTUUCP400 da nastro per un sistema offline con radice /mnt:

```
# setld -D /mnt -l /dev/rmt2h UDTUUCP400
```

Installazione del subset UDTUUCP400 da nastro per un sistema offline con radice /mnt dall'installation server mmbly:

```
# setld -D /mnt -l mmbly: UDTUUCP400
```

Installazione del subset UDTUUCP400 per un sistema offline con radice /mnt dall'area di distribuzione /mnt2/RISC/BASE:

```
# setld -D /mnt -l /mnt2/RISC/BASE UDTUUCP400
```

Rimozione dei subset UDTUUCP400 e UDTCOMM400:

```
# setld -d UDTUUCP400 UDTCOMM400
```

Rimozione dei subset UDTUUCP400 e UDTCOMM400 da un sistema offline con radice /mnt:

```
# setld -D /mnt -d UDTUUCP400 UDTCOMM400
```

Indicazione dello stato di tutti i subset noti al sistema:

```
$ setld -i
```

Indicazione dello stato di tutti i subset noti al sistema offline con radice /mnt:

```
$ setld -D /mnt -i
```

Lista del contenuto del subset UDTUUCP400:

```
$ setld -i UDTUUCP400
```

13 SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE

Verifica del subset ULTVAXC400 sul sistema corrente:

```
# setld -v ULTVAXC400
```

Invio del messaggio di configurazione Config Subset al subset UWSX11400:

```
# setld -c UWSX11400 "Config Subset"
```

Estrazione dei subset da nastro sulla directory corrente:

```
# setld -x /dev/nrmt0h
```

Estrazione da disco dei subset contenuti nella directory /mnt/RISC/UNSUPPORTED sulla directory /usr/bigdisk:

```
# setld -D /usr/bigdisk -x /mnt/RISC/UNSUPPORTED
```

Con questo comando alla directory /usr/bigdisk vengono ricreate le strutture di sistema:

```
/usr/var/subsets  
/usr/etc/subsets
```

Per evitare la creazione della struttura, si deve creare un tar file contenente i subsets.

Infine tenere presente che alcuni subset richiedono di essere configurati. L'informazione compare nel manuale di guida all'installazione o nelle *Release Notes*. In particolare la **configurazione** è richiesta dal Fortran e va fatta con il comando:

```
# setld -c DFRBASE3xx SELECT
```

ed è descritta in /usr/lib/complrs/fort_310/relnotes.

Normalmente la configurazione crea dei link a files. Per esempio, nel caso del Fortran il link consente di accedere alla descrizione esplicita dei messaggi d'errore e di evitare ABORT del programma in presenza di errori del tipo ZERO DIVIDE. Se non si esegue lo step di configurazione al momento dell'installazione o se l'installazione fallisce perchè trova link vecchi (che devono essere cancellati prima di procedere), il prodotto in fase di esecuzione non funziona regolarmente.

14 SISTEMA OPERATIVO SU SERVER

14 SISTEMA OPERATIVO SU SERVER

14.1 SERVER SETUP

Unix consente di costruire un ambiente con funzioni analoghe a quelle svolte dai cluster VMS in cui una o più macchine possiedono il sistema operativo da cui viene fatto il boot di tutte le altre. Il nodo che gestisce il sistema è un server da cui i clienti caricano il sistema operativo via ethernet. Possono essere supportati sia clienti VAX-Ultrix che RISC. Il server possiede un proprio sistema operativo **NON** condiviso e gestisce i **daemons**:

dms installazione di software per clienti diskless
ris installazione di software remoto (opzionale)

Il server deve contenere i file system per i clienti e precisamente un file system **dlenv** per ciascun ambiente (VAX o RISC). Il file system **/dlenv** contiene anche l'area **shared /usr** dove risiedono i file read-only comuni a tutti i clienti. Almeno un sistema **/dlclient** in cui risiede una copia della porzione comune di **root** e i file specifici di ogni cliente. Detti file system devono essere configurati con dimensioni sufficienti per il sistema operativo e i prodotti **layered** e per contenere le aree specifiche per tutti i clienti supportati.

Il file system **/dlclient** viene occupato per circa **10 MB** per ogni cliente installato e richiede **50 MB** aggiuntivi usati temporaneamente ad ogni aggiunta di un nuovo cliente, quindi ad esempio per **10 clienti** occorrono almeno **150 MB**.

14.2 DMS

Un server fornisce servizi di management a processori clienti da un'area opportuna. Il processo viene gestito dall'utility **dms** (Diskless Management Services). L'area server può contenere software per uno o più processori indifferentemente VAX o RISC. Perché un processore possa agire da server deve essere installato il **MOP** (Maintenance Operations Protocol). Il processo di server viene gestito tramite un apposito file system in cui vengono abilitati i vari clienti. I file system richiesti sono:

/dlclient0 contiene le **root** dei vari clienti
 (i.e. **ClientA.root**, **ClientB.root**, etc.)
/dlenv0 contiene le **root** comuni per i vari sistemi
 (i.e. **root0.mips**, **root0.vax**, etc.)

Il server ha accesso all'intera struttura di directory, mentre il cliente ha accesso solo all'area **/usr** condivisa che esiste sotto **root0.mips** e al proprio **ClientA.root**. Un sistema può diventare server se è dotato di sistema operativo, local area network e network file system. Devono essere attivi i **daemons**:

biod
mountd
nfsd
portmap

14 SISTEMA OPERATIVO SU SERVER

Per verificarne la presenza:

```
$ ps -aux | egrep "biode|mountd|nfsd|portmap"
```

I file system `dlenv` e `dlclient` possono stare su dischi diversi. Il file system `dlenv` deve avere spazio sufficiente per il sistema operativo. Per un server con soli clienti RISC completo di supported, Fortran e DECnet + 20% di management, in genere sono richiesti circa 100 MB. Si consiglia di verificare il valore esatto consultando i manuali di installazione e di calcolare un margine di lavoro adeguato per futuri aggiornamenti. Per ciascun cliente sono richiesti circa 10 MB più spazio per page e swap che può essere allocato sui dischi locali dei clienti. Sono necessari circa 50 MB temporanei per l'operazione di aggiunta di ogni nuovo cliente.

14.3 CREAZIONE CLIENTI

Per registrare un cliente occorrono le seguenti informazioni:

- nome nodo e numero relativo
- password di root
- indirizzo ethernet
- disco di swap

Verificare le partizioni prima di iniziare ed eventualmente modificare le partizioni del disco tramite `chpt`. Si invoca `/etc/dms` che è una procedura a menu per il management dei clienti. Alla prima installazione si usa l'opzione `C` per la creazione del file system `diskless` sulla partizione specificata. La procedura verifica che ci sia abbastanza spazio disco e chiede conferma. Dopo la creazione di `dlclient` e `dlenv`, invocare l'opzione `i` di `dms` per installare il sistema operativo per i clienti. L'installazione è uguale a quella di un sistema stand-alone ma opera sul `/dlenv` designato.

Dopo aver installato il sistema si procede all'aggiunta dei clienti. Si deve specificare l'ambiente (RISC o VAX), il nome del cliente, l'indirizzo ethernet, il file system `/dlclient` corrispondente, lo swap space, l'opzione per il crash dump, la password di root.

Il boot del cliente deve avvenire da ethernet quindi occorre inserire i seguenti comandi di hardware per consentire il boot da `eth` all'accensione:

```
>> setenv bootpath mop(0)
>> auto
```

Da tener presente che il sistema operativo condiviso si trova su `/dlenv` mentre la parte personalizzata per ogni cliente si trova su `dlclient` a partire dalla directory `host.root`. Se il cliente non possiede un'area di swap locale, viene creato il file: `/dlclient0/host.root/dev/swap`.

Con `dms` si possono inoltre modificare clienti (swap - dump), eliminare clienti, listare i clienti registrati, vedere i prodotti installati nell'area server, ricostruire o copiare il kernel. Inoltre `dms` crea un database `/usr/diskless/dmsdb` che può essere utilizzato direttamente per la gestione di più clienti. Le informazioni contenute sono: `nome`, `eth`, `dlenv root`, `dlclient root`, `swap`, `dump` con `:` per la separazione dei campi, ad esempio:

14 SISTEMA OPERATIVO SU SERVER

```
$ cat /usr/diskless/dmsdb
ds5ze3:08-00-2b-1d-7b-05:/dlenv0/root0.mips:/dlclient0:rz1b:n
ds5ze2:08-00-2b-1a-e0-a0:/dlenv0/root0.mips:/dlclient0:64:n
$
```

I nuovi clienti vengono aggiunti via editor e registrarli con:

```
# /etc/dms -a new1 new2 new3
```

Le operazioni di `/etc/dms` sono pilotate dalle opzioni:

```
-a    add
-r    remove
-k    build new kernel
-l    list
-s    list software subsets
```

Per la gestione dei prodotti installati:

```
# setld /dlenv0/root0.mips -i          lista subset
# setld /dlenv0/root0.mips -d PRODUCT eliminazione subset
```

N.B. Il nome del cliente deve comparire localmente nel file `/etc/hosts` per consentire il boot in ogni circostanza. In questo modo se il nodo che svolge le funzioni di name server non è raggiungibile, il sistema può comunque fare il boot basandosi sul nome locale, operazione che non avrebbe luogo se i nomi dei nodi venissero gestiti esclusivamente tramite name server (`bindsetup`).

14.4 RIS

Per operare installazioni remote (cioè su clienti o nodi che non dispongano di device esterni adeguati) da un server dotato di tk50 o di cdrom si fa uso dell'utility `ris` (Remote Installation Services). Il software si trova sotto `/var/adm/ris` in cui si trova l'ambiente `ris0.mips`. `/etc/ris` consente di installare il software in forma di distribution, registrare e gestire i clienti che possono accedervi, listare i prodotti contenuti nel server. La gestione avviene tramite i seguenti steps:

- il system manager del server crea l'area di distribuzione tramite `/etc/ris`
- carica il software in detta area
- autorizza i clienti che possono accedere al software
- gestisce il database `/var/adm/ris/clients/risdb`

A questo punto, il system manager del cliente è in grado di installare il software sul proprio nodo con:

```
# setld -l server_name SUBSET
```

15 PAGINE GIALLE

15 PAGINE GIALLE

Un sistema stand-alone non ha bisogno di installare il sistema di database di unix YP (denominato Yellow Pages), che invece è richiesto per sistemi client-server in cui alcune risorse debbano essere condivise. YP si occupa di gestire i database attraverso la rete, in particolare gestisce i nomi dei nodi di tcp/ip, le passwords, etc.

I database di YP si chiamano *mappe*, un set di mappe cui venga assegnato un nome si chiama *dominio*. Le funzioni di YP sono gestite con metodo client-server che agisce su processi (NFS agisce su nodi). YP è formato da un master server, uno o più slave server e un numero indefinito di clienti. L'unico database modificabile è quello del master. Gli slave vengono aggiornati periodicamente. Ogni dominio può avere un solo master. Tutte le mappe di un dominio YP vengono mantenute in opportune subdirectory della directory /etc/yp. Ogni subdirectory ha il nome del dominio corrispondente. Il nome del dominio viene listato da /bin/domainname. /etc/yp è linkata simbolicamente a /var/yp. Il nome del dominio è determinato in /etc/rc.local con un entry del tipo:

```
/bin/domainname NOME
```

La maggior parte delle mappe del dominio è stata generata usando file ASCII come /etc/passwd, /etc/group, /etc/hosts e /etc/networks. L'ordine in cui i servizi vengono gestiti viene stabilito nel file /etc/svc.conf:

```
# @(#)svc.conf 4.1      (ULTRIX)      7/2/90
#
# WARNING: This file is MANDATORY !
#
# Setup recommendation: As you add distributed services to database
#      entries, it is recommended that "local" is the first service.
#      For example:
#
#                          passwd=local,bind
#
# Note: White space allowed only after commas or newlines.
#
# File Format
# -----
# database=service,service
#
# The database can be:
#      aliases
#      auth
#      group
#      hosts
#      netgroup
#      networks
#      passwd
#      protocols
```

15 PAGINE GIALLE

```
#      rpc
#      services
# The service can be:
#      local
#      yp
#      bind
#
aliases=local
auth=local
group=local,yp
hosts=local,bind
netgroup=local
networks=local,yp
passwd=local,yp
protocols=local,yp
rpc=local,yp
services=local,yp

PASSLENMIN=6
PASSLENMAX=16
SOFTEXP=604800          # 7 days in seconds
SECLEVEL=BSD           # (BSD | UPGRADE | ENHANCED)
#
#---- end /etc/svc.conf ----
```

I file di mappa sono di tipo *dbm*. Se si crea un dominio *netuser* che contiene le informazioni di */etc/hosts*, nel dominio si troveranno i files:

```
/var/yp/netuser/hosts.byaddr.dir
/var/yp/netuser/hosts.byaddr.pag
/var/yp/netuser/hosts.byname.dir
/var/yp/netuser/hosts.byname.pag
```

I nomi dei files indicano il tipo di indirizzamento, ad esempio i files *.byaddr* indirizzano i dati per indirizzo (es. Internet), i files *.byname* li indirizzano per nome. YP è attivo se gira */etc/ypbind*. I database possono essere stampati con *ypcat*. La lista dei file esistenti è generata da:

```
$ ypcat -x
```

che produce un output del tipo:

```
$ ypcat -x
Use "passwd" for map "passwd.byname"
Use "group" for map "group.byname"
Use "networks" for map "networks.byaddr"
Use "hosts" for map "hosts.byaddr"
```

15 PAGINE GIALLE

Use "protocols" for map "protocols.bynumber"

Use "services" for map "services.byname"

Use "aliases" for map "mail.aliases"

Use "ethers" for map "ethers.byname"

Use "rpc" for map "rpc.bynumber"

Il comando `makedbm` trasforma un file ASCII in un set di files dbm. Per l'uso corretto, adoperare il `Makefile` di `/var/yp`. Consultare anche `ypmake`. I database di default gestiti da YP sono:

`/etc/hosts` - `/etc/passwd` - `/etc/group` - `/etc/networks` - `/etc/rpc` -
`/etc/services` - `/etc/protocols` - `/etc/netgroups`

I comandi e i file di YP sono:

Comandi	Definizioni
<code>domainname</code>	lista o setta il nome del dominio corrente
<code>endnetgrent</code>	preleva l'entry del gruppo di rete
<code>getdoaminname</code>	preleva il nome del dominio corrente
<code>getnetgrent</code>	preleva l'entry del gruppo di rete
<code>group</code>	raggruppa i file in un ambiente YP
<code>hosts.equiv</code>	lista dei nodi di fiducia
<code>innetgr</code>	preleva l'entry del gruppo di rete
<code>makedbm</code>	crea un file dbm
<code>netgroup</code>	lista i gruppi di rete
<code>passwd</code>	file delle password per il servizio YP
<code>setdomainname</code>	setta il nome del dominio corrente
<code>setnetgrent</code>	preleva l'entry del gruppo di rete
<code>ypcat</code>	stampa i valori di un database YP
<code>ypclnt</code>	pacchetto di interfaccia cliente per YP
<code>ypfiles</code>	struttura database e directory di YP
<code>ypmake</code>	ricostruisce database YP tramite <code>make</code>
<code>ypmatch</code>	stampa il valore di una o più chiavi di una mappa YP
<code>yppasswd</code>	cambia e/o aggiorna la password nella mappa YP
<code>ypaswdd</code>	daemon per la gestione delle password YP
<code>yppoll</code>	determina la versione di una mappa sul master
<code>yppush</code>	forza la propagazione di una mappa variata
<code>ypserv</code>	process server e binder di YP
<code>ypset</code>	indirizza il processo <code>ypbind</code> ad un server particolare
<code>ypsetup</code>	set dell'ambiente YP
<code>ypwhich</code>	determina il nodo che è il server o il master corrente di YP
<code>ypxfr</code>	trasferisce una mappa YP da un server al nodo locale

YP usa il protocollo *tcp/ip* e richiede che `/usr` sia montato. L'installazione può essere semplificata da `ypsetup`, una procedura a menu con ampia descrizione di ogni step e con funzionamento analogo a `nfsetup` e alle altre procedure di setup descritte

15 PAGINE GIALLE

precedentemente. Come già indicato, i database possono essere visualizzati con `ypcat` e `ypmatch`.

Esempi:

Lista nomi hosts:

```
# ypcat hosts
```

Lista utenti:

```
# ypcat passwd
```

Ricerca elemento database:

```
# ypmatch librarian passwd
```

```
# ypmatch ds5ze7 hosts
```

Per *aggiungere* un utente in *ambiente YP*, seguire i seguenti steps:

```
# cd /var/yp/src
# vi passwd
# cd /var/yp
# make DIR=/var/yp/src passwd
# mkdir directory
# chown
# chgrp
# cp /usr/skel/.??* /usr/users/newone
```

Per *cambiare la password* in ambiente YP si deve usare `yppasswd`.

Per *aggiornare* la tabella degli hosts:

```
# cd /usr/var/yp/src
# vi hosts
# cd /var/yp
# make hosts
```

15.1 SETUP MANUALE DI YP

Oltre a settare le mappe di YP, si devono girare i daemons di gestione YP `/etc/portmap` e `/usr/etc/ypserv`. Il master server deve anche girare `/usr/etc/rpc.yppasswd`, qualsiasi altro sistema che funzioni da cliente deve girare `/etc/ypbind` che alla partenza lancia una richiesta su `eth` (ethernet) per la ricerca di un server. Il comando `ypwhich` fornisce il nome del server attuale.

`ypcat` lista le informazioni del database:

```
$ ypcat passwd
```

`ypmatch` cerca un entry specifico nel database:

15 PAGINE GIALLE

```
$ ypmatch jones passwd
```

Notare la differenza tra i 2 comandi:

```
$ ypcat passwd | grep jon
```

```
$ ypmatch jones passwd
```

Per poter settare YP il sistema deve essere in multiuser con il file system /usr montato. Inoltre deve essere stata creata la topologia dei server: **master, slave, clienti** tenendo presente che può esistere un solo master nel dominio. I file /etc/ indicati sopra devono esistere ed essere aggiornati.

Per creare **l'ambiente** di YP procedere come descritto di seguito.

- Creare il dominio:

```
# /bin/domainname lep          (creazione del dominio di nome lep)
# mkdir /var/yp/lep
```

- Costruire le mappe di default:

```
# cd /var/yp
# make NOPUSH="Y"
# /etc/portmap
# /usr/etc/ypserv
```

- Se il sistema agisce anche come **cliente** creare la directory:

```
# mkdir /var/yp/src
```

e copiare i master file di /etc/ in tale directory. Se i master vengono mantenuti in questa directory allora make verrà invocato con il parametro DIR=/var/yp/src.

- Assicurarsi che in /etc/rc.local compaia:

```
/usr/etc/rpc.yppasswd /var/yp/src/passwd -m passwd DIR=/var/yp/src
```

per attivare il daemon che gestisce le password in ambito YP.

- Infine creare le mappe di YP per i server:

```
# cd /var/yp
# makedbm - lep/ypservers
```

(legge da standard input e scrive su ypservers)

```
lepslav1
lepslav2
```

15 PAGINE GIALLE

dove `lep*` sono i nomi dei servers slave. Oppure con `vi` creare il file che contiene l'informazione e usare `makedbm` su detto file:

```
# cd /var/yp/src
# vi lepserv.asc
# /var/yp/makedbm lepserv.asc ../lep/ypservers
```

Il file `/etc/svc.conf` controlla l'ordine in cui i database vengono usati. Le variazioni vengono apportate tramite `svcsetup` (invocato da `su`) che consente di verificare e modificare il file di configurazione. L'utility è pilotata da menu. Nel setup dei servizi, specificare sempre prima `local` di `yp`.

Gli *slave server* si possono settare solo dopo aver stabilito un *master* che giri `ypserv` e possieda le informazioni da cui gli slave possono copiare i database. I passi per creare lo *slave* sono:

- Stabilire il dominio:

```
# /bin/domainname lep
# mkdir /var/yp/lep
```

- Copiare le mappe dal master (es. nodo `ds5ze7`):

```
# ypxfr -h ds5ze7 -c -d lep passwd.byname
# ypxfr -h ds5ze7 -c -d lep passwd.byuid
```

- Far partire i daemon sul server nell'ordine:

```
# /etc/portmap
# /usr/etc/ypserv
```

- Se lo slave sarà anche *client*, editare i database come descritto di seguito e attivare:

```
# /etc/ypbind -S
```

(usare `-S` per security - vedi `man`)

- Editare `/etc/rc.local` per aggiungere il nome del dominio e l'attivazione del daemon `portmap`, `ypserv` e `ypbind`.

- Modificare `svc.conf`:

```
# svcsetup
```

- Editare `/usr/lib/crontab` per l'aggiornamento periodico delle mappe inserendo:

15 PAGINE GIALLE

```
/etc/yp/ypxfr_1perday
/etc/yp/ypxfr_2perday
/etc/yp/ypxfr_1perhour
```

a seconda della frequenza di aggiornamento e verificare l'esistenza del file `/etc/yp/ypxfr.log` per il controllo dei trasferimenti.

- Aggiungere il nuovo slave al dominio come descritto oltre.

Per settare un *cliente* deve esistere un *server* con le mappe aggiornate che gira `ypserv`, quindi procedere come segue:

- Stabilire il dominio:

```
# /bin/domainname lep
```

- Fare partire i daemon:

```
# /etc/portmap
```

- Modificare i database come descritto oltre e far partire `/etc/ypbind`:

```
# /etc/ypbind -S (per security)
```

- Editare `/etc/rc.local` per aggiungere il nome del dominio e l'attivazione dei daemon `portmap` e `ypbind`.

Modificare `svc.conf`:

```
# svcsetup
```

Per evitare conflitti di contenuto nei file di database eliminare dal cliente i files seguenti:

```
# mv /etc/networks /etc/networks.old
# mv /etc/protocols /etc/protocols.old
# mv /etc/rpc /etc/rpc.old
# mv /etc/services /etc/services.old
# mv /etc/netgroup /etc/netgroup.old
```

Per indicare che `/etc/hosts.equiv` deve fare ricorso ad YP far precedere ciascun entry da `@` che indica elemento di rete. Per motivi di sicurezza usare informazioni esplicite nel file `/.rhosts` che non viene servito da YP. Normalmente tali files sono vuoti all'installazione. Nel file `/etc/hosts` deve comparire il nome del nodo locale e l'entry loopback. Il file `/etc/passwd` può contenere le informazioni di root e degli utenti locali e deve terminare con `+`: perché la ricerca prosegua su YP per gli altri utenti. Analogamente per il file `/etc/group`.

Le mappe devono essere *modificate* nel master e *propagate* agli slave. Le modifiche si possono fare editando il file e usando `make`. Ad esempio per aggiungere o modificare un utente alla mappa:

15 PAGINE GIALLE

```
# cd /var/yp/src
# vipw passwd
# cd /var/yp
# make DIR=/var/yp/src passwd
```

Per creare o aggiornare una mappa invocando direttamente `/var/yp/makedbm`:

```
# cd /var/yp/src
# vi lepmap.asc
# /var/yp/makedbm lepmap.asc ../lep/lepmap
```

Ricordarsi di correggere *sempre* il file ASCII *prima* di usare `makedbm`. Il file ASCII si può anche creare dalla mappa con:

```
# cd /var/yp/lep
# /var/yp/makdbm -u lepmap > lepmap.tmp
# vi lepmap.tmp
# /var/yp/makedbm lepmap.tmp lepmap
# rm lepmap.tmp
```

Per *propagare* le mappe YP da master a server esistono tre diverse possibilità.

1. Usare `make`:

```
# cd /var/yp
# make hosts
```

in questo modo viene girato automaticamente `yppush` che copia dal master agli slave usando l'elenco dei server YP del dominio contenuto in `ypservers`.

2. Usare direttamente `yppush` per il database:

```
# cd /var/yp/src
# /var/yp/makedbm lepcal.asc ../lep/lepcal
# yppush lepcal
```

3. Usare `ypxfr` per trasferire le mappe sia manualmente che da cron.

(a) *Manualmente* dallo slave:

```
# /usr/etc/ypxfr group.byname
# /usr/etc/ypxfr group.bygid
```

con una chiamata per ogni database da aggiornare.

(b) *Automaticamente* con `/etc/cron`

Se si usa `ypxfr` con cron, si devono usare o personalizzare i comandi di shell degli script `ypxfr_1perhour`, `ypxfr_1perday`, `ypxfr_2perday` in cui si indicano i database da aggiornare in base alla presunta frequenza di modifica. Indicare in `/usr/lib/crontab` quali shell girare e quando, in tempo assoluto, con differenze tra i tempi di trasferimento dei vari slave per evitare un carico eccessivo del master.

15 PAGINE GIALLE

15.2 MODIFICA DELL'AMBIENTE YP

Le modifiche all'ambiente riguardano variazioni ai server di YP quali aggiunta o eliminazione di server, aggiornamento mappe, gestione di utenti, etc. che vengono gestite con le operazioni indicate di seguito.

- **Aggiunta del server lepslav3:**

```
# cd /var/yp
# (/var/yp/makedbm -u lep/ypservers ; \
    echo lepslav3)|/var/yp/makedbm - tmpmap
# mv tmpmap.dir lep/ypservers.dir
# mv tmpmap.pag lep/ypservers.pag
# yppush ypservers
# vi /etc/hosts
... inserire i dati di lepslav3 ...
# make hosts
```

Collegarsi con il nuovo server, creare i database e propagarne il contenuto dal master.

- **Eliminazione del server lepslav3:**

```
# cd /var/yp
# makedbm -u lep/ypservers | grep -v lepslav3 | makedbm - tmpmap
# mv tmpmap.dir lep/ypservers.dir
# mv tmpmap.pag lep/ypservers.pag
# yppush ypservers
```

- **Trasferimento mappa ad un nuovo master:**

Collegarsi sul nuovo master e creare la mappa voluta:

```
# cd /var/yp
# make lepmap.asc
```

oppure, in mancanza del file ASCII:

```
# cd /var/yp
# ypcat -k lepmap | /var/yp/makedbm - lep/lepman
```

aggiornare i dati di master e la mappa:

```
# /var/yp/ypxfr -h newmaster lepmap
```

- **Aggiunta utenti agendo su /etc/passwd:**

15 PAGINE GIALLE

```
# /etc/vipw
# cd /var.yp
# make passwd
```

• *Aggiunta utenti agendo su /var/yp/src/passwd:*

```
# vi /var/yp/src/passwd
# cd /var/yp
# make DIR=/var/yp/src passwd
```

Da notare che l'argomento `DIR=/var/yp/src` non è necessario dato che tale directory è definita come default. In entrambi i casi l'operazione viene completata con:

```
# cd /usr/users
# mkdir newone
# chown newone newone
# chgrp 15 newone
# cp /usr/skel/.??* /usr/users/newone
# cd newone
# chown newone .??*
```

Se la mappa non è stata aggiornata, `chown` dà un errore. In questo caso invece del nome usare `userid`:

```
# chown 273 newone
```

Infine propagare le nuove mappe.

16 BACKUP

16 BACKUP

16.1 DUMP E RESTORE

Per l'archiviazione e il ripristino dei file esistono comandi *locali* e *remoti*. Con i comandi remoti si può agire solo su interi file system. I comandi sono:

<code>dump (locale), rdump (remoto)</code>	usabili da <i>su</i> - single user
<code>tar, mdtar</code>	senza privilegi - multiuser - file, directory no file system - no remote <code>tar su /dev/rmt0h</code> <code>mdtar su /dev/rra1a (floppy)</code>
<code>opser</code>	utility interattiva - locale: solo su tape remota: su file e su tape da operator - single-user
<code>restore (locale), rrestore (remoto)</code>	senza privilegi - multiuser

I comandi remoti vengono usati in sistemi client-server. I comandi di dump agiscono utilizzando le informazioni di inode (tabelle e statistiche), perciò se un file viene alterato dopo la registrazione di inode per il backup, il backup stesso potrebbe essere corrotto. Di conseguenza per il dump è opportuno operare in single-user con i file system smontati. I comandi di tar e mdtar non sono altrettanto critici perché non si basano sul contenuto di inode, inoltre funzionano sia come save che come restore e sono accessibili a tutti gli utenti. Per il restore di un dump file esistono i comandi `restore` ed `rrestore` che consentono un uso interattivo. Per il salvataggio con `dump` è conveniente usare la procedura `opser` pilotata in modo interattivo.

Per il backup di un intero file system esiste il comando `/etc/dump` che normalmente richiede l'intervento dell'operatore. I file vengono copiati su di un file, una pipe, un disco o un nastro. Il dump fallisce se si verificano più di 32 errori sul device di output. Il formato del comando è:

```
/etc/dump opzione [argomento] filesystem
```

L'argomento dipende dal contesto: è un nome di file con l'opzione `f`, la densità del nastro con l'opzione `d`. Le opzioni sono:

16 BACKUP

Chiavi	Descrizione
d	densità nastro (default 1600)
f	scrive il dump sul file indicato dall'argomento
n	notifica agli utenti del gruppo operator la necessità di intervento
s	dimensione del nastro in piedi (default 1200)
u	se il dump è ok, scrive la data in /etc/dumpdates
w	lista i file system di cui è richiesto il dump
W	lista i file system di cui è richiesto il dump, la data più recente di dump e il livello
0-9	specifica il livello di dump da 0 a 9
9u	valore di default delle chiavi
argomento	nome del file con f, densità nastro con d
filesystem	nome del file system montato. Nome del device per file system smontato non indicato in fstab

Il file system su cui si esegue il dump non deve essere attivo e non deve essere montato. L'operazione avviene da superuser. Usare `fsck` per verificare l'integrità del file system. Il backup può essere completo o incrementale su base mensile, settimanale o giornaliera.

Di seguito è indicato un *esempio di dump* completo dei comandi e delle risposte del sistema:

```
# shutdown now "Backup del Sistema"
shutdown at 11:30 (in 0 minutes) [pid 1410]
#
System shtdown time has arrived
erase ^?, kill ^U, intr ^C
# umount -a
# mount
```

Il comando `/etc/mount` senza argomenti serve per controllare che sia montato solo *root*. Il dump deve essere fatto con *livello 0* per fare il dump totale. Il parametro `u` indica che le date devono essere registrate su `/etc/dumpdates`.

```
# dump 0u /usr/users
DUMP: Date of this level 9 dump: Mon Dec 16 11:31:27 1991
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rra1h (usr/users) to /dev/rmt0h
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 28750 tape blocks on 0.74 tape(s)
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: 32.17% done, finished in 0:10
DUMP: 66.68% done, finished in 0:05
DUMP: DUMP: 28750 tape blocks on 1 tape(s)
DUMP: DUMP: IS DONE
DUMP: level 0 dump on Mon Dec 16 11:31:27 1991
```

16 BACKUP

```
DUMP: tape rewinding
#
# cat /etc/dumpdates
/dev/rraih      9 Mon Dec 16 11:31:27 1991
#
# /etc/dump W
Last dump(s) done (Dump '>' file systems):
  /dev/raih (/usr/users) Last dump: Level 0, Date Mon Dec 16 11:31
```

Il backup può essere schedulato automaticamente tramite `/usr/lib/crontab`. Il backup può essere eseguito a vari livelli, con il livello 0 si ha il backup totale, mentre i livelli 1-9 consentono backups incrementali per data e livello secondo le direttive del file `/etc/dumpdates`. Il file `/etc/dumpdates` deve esistere anche se vuoto, altrimenti il dump fallisce. Fare attenzione al `/dev/` che si indica nel comando, tenendo presente che:

- `/dev/rmt0h` esegue `rewind` dopo il dump
- `/dev/nrmt0h` *non* esegue `rewind` dopo il dump e dovrebbe essere il default.

Esempi:

Dump del file system su nastro:

```
# dump 0undf 6250 /dev/rmt0h /usr/users
```

Dump del file system su exabyte:

```
# dump 0usdf 35000 10240 /dev/nrmt1h /usr/users
```

Dump del file system su DAT:

```
# dump 0usdf 17000 10240 /dev/nrmt1h /usr/users
```

I parametri usati significano:

```
0  tutto il file system
u  scrive data su /etc/dumpdates
s  lunghezza nastro
d  densita'
f  device se diverso da /dev/rmt0h
```

Bytes richiesti per un dump di livello 0 del root file system:

```
# dump 0Sf test /
```

I file salvati con `dump` vengono ripristinati con `restore` la cui sintassi è:

```
# /etc/restore opzione [nome]
```

16 BACKUP

`restore` opera di default da `/dev/rmt0h` e va eseguito in `single-user` come si opera per `dump`. L'opzione `i` consente un'operazione di restore interattiva. Le opzioni sono:

Chiavi	Descrizione
<code>f</code>	usa il prossimo argomento come input invece di <code>/dev/rmt?</code>
<code>h</code>	estrae le directory invece dei file
<code>i</code>	opera in modo interattivo
<code>m</code>	estrae per inode invece che per nome
<code>r</code>	carica l'intero nastro nella directory corrente, usato per ripristinare un intero disco o un dump incrementale dopo un restore di livello 0
<code>R</code>	inizia restore da un volume specifico di un dump multivolume
<code>t</code>	lista le occorrenze dei file indicati
<code>v</code>	invoca il modo <code>verbose</code> durante il restore
<code>x</code>	estrae i file indicati
<code>y</code>	ignora gli errori

Il comando `restore` non richiede privilegi. Poichè `dump` scrive i file con un path relativo, `restore` va attivato dalla **stessa directory** da cui è stato fatto il dump.

Esempi:

`dump` e `restore` possono essere usati in una pipe:

```
# dump Of - /usr | (cd /mnt; restore xf -)
```

dove: `/usr` è la directory sorgente, che **deve** essere montata da `/fstab`

`/mnt` è la directory destinazione

`Of` indica tutto il disco

Per avere una lista del dump file:

```
# /etc/restore -t
```

Per un uso interattivo:

```
# /etc/restore -i
```

se il dump file non è sul nastro di default:

```
# /etc/restore -if /usr/backups
```

Quando `restore` è in modo interattivo compare il prompt:

```
restore>
```

I numeri inode vengono registrati in ordine crescente e possono essere usati per localizzare i file su nastro (per avere le informazioni di inode usare `ls -i`). Da `restore` si possono dare i comandi: `pwd`, `ls`, `cd` relativi al dump file. Per vedere inode dare il comando `verbose` prima di `ls`.

Per il restore parziale di `/usr/lib`, porsi su `/usr`, creare una `extraction list` con `add`, ad esempio per estrarre i file `bck.doc`, `bck.lis` e la directory `bckdir` battere:

16 BACKUP

```
restore> add bck.doc bck.lis bckdir
```

Il comando `ls` lista i file della extraction list prefissati con `*`. Se si vuole eliminare un file usare `delete`. Una volta verificata la lista per procedere al restore, dare il comando `extract`. Tenere sempre presente che la directory è relativa, cioè se si fa il dump di `/usr/lib/lpf` dalla directory `/usr` nel dump media il path è `./lib/lpf`. L'operazione di restore è pilotata dal programma e richiede l'intervento dell'operatore. Domande tipiche riguardano il no. di nastri e le protezioni delle directory. Per terminare usare il comando `quit`.

Esempi:

Restore totale da nastro default:

```
# /etc/restore -x
```

Restore totale da nastro alternativo:

```
# /etc/restore -xf /dev/rmt1h
```

Restore di file:

```
# /etc/restore -v /usr/lib/file1 /usr/bin/file2
```

Per restore interattivo:

```
# /etc/restore i
```

Per creare un nuovo file system sul device `rz1g`, dopo avere portato il sistema a single-user, montare il nastro *read-only*, creare il file system con `/etc/newfs` e verificarlo con `/etc/fsck`, montare il file system da ripristinare, porsi nella root directory, infine invocare `restore`, secondo gli step indicati:

```
# /etc/newfs /dev/rrz1g rz57
# /etc/fsck /dev/rz1g
# /etc/mount /dev/rz1g /usr/users
# cd /usr/users
# /etc/restore r          per dump mono tape
# /etc/restore R          per dump multi tape
# rm restorestymtable    per cancellare la restore table creata da restore
# cd /
# /etc/umount /dev/rz1g
# /etc/fsck /dev/rz1g
```

16.2 ESEMPIO PRATICO DI RESTORE

Per fare il restore del file system `/usr` che sia il secondo file di una cassetta:

```
# mnt fsf 1                # ! skip 1rst file
# cd destination_directory # ! N.B. verify
```

16 BACKUP

Verificare di essere alla directory voluta. Ad esempio:

```
# mount /dev/rz1g /mnt  
# cd /mnt
```

Per il restore:

```
# restore -i
```

Viene presentato un menu, battere ? per help. Conviene scegliere l'opzione v per verbose in modo da seguire l'operazione. Scegliere add senza argomenti per selezionare tutte le directory del nastro. Una volta terminato il restore, rispondere Y alla domanda di ripristino date e owner.

Per il restore di un cliente *dms*, non si consiglia di usare la procedura di restore da dms. È più conveniente cancellare o rinominare la directory per cui si richiede il ripristino e ricaricarla direttamente da restore senza passare per il menu di dms.

16.3 TAR

Per il backup non privilegiato di singoli file, directory o alberi usare tar, la cui sintassi è:

```
$ tar opzioni nome
```

Il device di default di tar è /dev/rmt0h. Le opzioni e i relativi valori sono:

Chiavi	Descrizione
Opzioni	
c	crea un nuovo nastro
r	aggiunge file alla fine del nastro
t	lista i nomi dei file dal nastro
u	aggiunge file nuovi o modificati sul nastro
x	estrae file dal nastro
Valori	
b	setta il block size (default 20)
f	usa il prossimo argomento come nome dell'archivio
o	sopprime l'informazione di directory
p	ripristina i file indicati nei modi originali
v	scrive su terminale il nome di ogni file trattato
w	chiede conferma prima di ogni opzione
0-9	sceglie il drive (default 0)

Esempi:

Per help delle opzioni:

```
$ tar H
```

Backup della directory corrente su file temporaneo e verifica:

16 BACKUP

```
$ cd $HOME/libs
$ tar -cvf $TMPDIR/libs.tar .
$ tar -tvf $TMPDIR/libs.tar
```

Backup di una subdir sul nastro di default:

```
$ tar c ./ajw
$ mdtar cf /dev/rmt0h ./ajw
```

Backup di una subdir su nastro alternativo:

```
$ tar cf /dev/rmt1h ./ajw
```

Backup di file in coda al nastro:

```
$ tar rv main.*
```

Backup di una nuova versione di file in coda al nastro:

```
$ tar -uv main.*
```

Recupero di un file da nastro:

```
$ tar -xv main.f
```

Lista archivio nastro default:

```
$ tar t
```

Lista archivio nastro alternativo:

```
$ tar tf /dev/rmt1h
```

Estrazione file:

```
$ tar xp file2 file4 file7
```

Se il file è stato compresso, ad esempio file.tar.Z con l'utility `compress`, per evitare lo spreco di spazio disco ricreando il file file.tar con l'utility `uncompress` si può usare l'utility `zcat`.

Backup da file compresso:

```
$ zcat file.tar.Z | tar xvf -
```

dove - indica a tar di prendere i dati dalla pipe.

16 BACKUP

16.4 OPSER

Per il dump pilotato si può usare `/opr/opser` che è uno shell speciale ed è il default dell'utente `operator`. Il prompt è `opr>`. Lo shell consente le seguenti opzioni:

<code>h</code>	lista opzioni
<code>!sh</code>	escape allo shell (richiede la password di root)
<code>u</code>	show users
<code>s</code>	shutdown multiuser (logout utenti, stop processi, sync del file system)
<code>d</code>	dismount file system
<code>f</code>	check file system
<code>r</code>	restart multiuser
<code>b</code>	backup
<code>halt</code>	halt cpu
<code>n</code>	inizializza opser remoto
<code>q</code>	quit opser (logout)

L'opzione `b` si presenta con i prompt necessari per il backup. Per poter uscire da `opser` con `quit`, occorre prima ripristinare il modo `multiuser` con `r`.

16.5 NASTRI ANSI

Se si vogliono creare dei nastri di archivio con label ANSI usare il comando `ltf` con il quale si possono creare nastri compatibili VMS.

Esempi:

Copia e verifica file:

```
$ ltf -c /dev/rmt0h main.*
$ ltf -t
```

Recupero file su direcorey temporanea:

```
$ mkdir save
$ cd save
$ ltf -x
$ ls -l
```

16.6 BACKUP REMOTO

In un sistema complesso è abituale che solo una macchina sia dotata di un device di backup (es. TK50, dat, etc.) di conseguenza gli altri nodi eseguono il backup remoto in un ambiente master-slave. Il nome dello slave deve comparire in `/etc/hosts` del master e viceversa. Analogamente il file `/.rhosts` deve avere accesso di root per master e reciprocamente per slave. Ci si collega quindi sul nodo master come `opser` e si sceglie l'opzione `n`:

16 BACKUP

```
opr> n  
Enter Slave System name: ds5ze9  
ds5ze9_opr>
```

Per accedere allo shell dello slave:

```
ds5ze9_opr>!sh
```

Per accedere allo shell del master:

```
ds5ze9_opr>lsh
```

Esistono due metodi di backup: diretto o con **stage**. Quest'ultimo è necessario per device streaming come tk50, in questo caso creare un area di stage , es. /staging. I comandi sono essenzialmente quelli del backup locale. Per maggiori informazioni vedi ***Guide to Backup and Restore***. Ricordare che i livelli di backup sono, ad esempio, 0 (backup totale o mensile), 9 (giornaliero), 5 settimanale, etc.

17 PROBLEMI E RIMEDI

17 PROBLEMI E RIMEDI

Nei paragrafi seguenti sono indicati alcuni dei problemi che si presentano con maggiore frequenza nella gestione del sistema e i relativi rimedi raggruppati per categoria.

17.1 PROBLEMI E RIMEDI: ERRORE TIME

In un sistema di rete in cui sia presente un nodo con funzione di *name server*, i nodi si possono sincronizzare al boot se in `/etc/rc` è presente il comando `/etc/rdate` che sincronizza la data e l'ora su tutta la rete. Se il parametro `timezone` del file di configurazione è sbagliato tale operazione produrrà sul nodo un'ora sbagliata. Se non si può ricreare il kernel per correggere l'errore, commentare `/etc/rdate` in `/etc/rc`. Fare attenzione che la sincronizzazione di `/etc/rdate` non sia in contrasto con la sincronizzazione di DECnet-OSI.

17.2 PROBLEMI E RIMEDI: DISTRUZIONE SISTEMA

Se il sistema operativo è andato distrutto a causa di un malfunzionamento hardware o di un errore del sistemista, per ricrearlo da TK50, fare boot dal sistema di distribuzione ed entrare nel menu di system management. Seguire la procedura:

```
# cd /dev
# MAKEDEV tz           creare il device nastro
# MAKEDEV rz           creare il device disco
# cd /
# /etc/mkfs /dev/rrz0a ...   ricreare il file system
# /etc/fsck /dev/rz0a
# /etc/mount /dev/rz0a /mnt
# cd /mnt
# restore -r
```

Per il reboot del sistema:

```
# sync
# sync
# halt
>>>b
```

17.3 PROBLEMI E RIMEDI CON UN NUOVO KERNEL

Se il nuovo kernel non parte o funziona male, per ripristinare il vecchio kernel, dopo reboot in single-user da `/genvmunix`:

```
# /etc/fsck -p
# /etc/mount -a -t ufs
# cp /sys/vmunix.old /vmunix
# /etc/reboot
```

17 PROBLEMI E RIMEDI

Con alcune versioni di Ultrix (es. 4.2a) una copia di `genvmunix` è presente anche sotto `/usr` se c'è spazio sufficiente. Si può così fare il boot in single-user anche dalla partizione `g`. È perciò buona norma definire partizioni abbastanza ampie da consentire questa ulteriore possibilità di boot *in extremis*.

Durante la creazione del kernel dopo un upgrade di sistema tramite `dms` vengono configurati i prodotti installati *solo* sul primo cliente che fa il boot, dato che il sistema è condiviso per tutti i clienti. Di conseguenza durante la creazione del kernel degli altri clienti compare il messaggio:

```
LOGNAME set to null
setld -c can be done only by su
```

Questo messaggio non è un messaggio d'errore ma solo un warning che va ignorato. Da notare che il messaggio viene ripetuto per *ogni* prodotto installato, quindi non interrompere la procedura.

17.4 PROBLEMI E RIMEDI CON IL FILE SYSTEM E I PROCESSI

I problemi più frequenti riguardano l'integrità del file system e le prestazioni del sistema. Per il controllo del file system usare `/etc/fsck`. In caso di emergenza:

```
# su
Password:
# /etc/shutdown now
# umount /dev/rz1d
# /etc/fsck /dev/rz1d
^D
```

Il sistema controlla il file system in questione e torna a multiuser.

L'occupazione del disco può essere tenuta sotto controllo con i comandi `df` e `du`. `df` dà informazioni globali, mentre `du` consente di identificare l'occupazione delle singole directory.

Alcuni processi usano la directory temporanea `/tmp` che viene vuotata ad ogni reboot. In caso di carico eccessivo (File System full) verificare lo stato di `/tmp` ed eventualmente cancellare i file.

N.B. Non cancellare la directory `/tmp` altrimenti il sistema rimane in single user.

Per identificare file molto grossi usare il comando `find` nella forma:

```
# find /usr/users -size +200 -atime +60 -print
```

che ricerca nella directory `/usr/users` i file più grandi di 200 blocchi, il cui ultimo accesso risale a due mesi prima e che potrebbero essere candidati ad un archivio in caso di problemi di spazio.

Per comunicare con gli utenti esiste il comando `wall` (Write to All Users).

Per gestire i processi esistono i comandi `ps` e `kill`. Il comando `ps` fornisce l'elenco dei processi attivi con una selezione che dipende dall'opzione:

17 PROBLEMI E RIMEDI

-a tutti i processi connessi a terminali
-x tutti i processi compresi quelli in background
-l lista completa dei processi con PPID (Parent process ID) e PRI (priority)

Per fermare un processo:

```
$ kill -9 pid
```

dove pid è il numero del processo.

Per variare la priorità di un processo usare `nice` o `renice`, ad esempio:

```
$ nice +10 cc -c driver.c
```

sottopone la compilazione del file `driver.c` con priorità minore.

```
$/etc/renice -10 248
```

aumenta la priorità del processo 248

17.5 PROBLEMI E RIMEDI PER ERRORE DISCO

In caso di errori sui dischi, verificare il log con:

```
# /etc/uerf -D rz
```

oppure:

```
/etc/uerf -r 102        errori disco  
/etc/uerf -r 104        errori controller
```

Le verifiche e i tentativi di restore vanno fatti da *su* in `single-user`, se l'errore è sul file system di root oppure dopo `umount` sugli altri file system. Una volta che si hanno le informazioni di errore dall'`error log` (che dovrebbe indicare il blocco anomalo sul disco), le possibili operazioni di verifica sono:

```
# icheck -b 2300 /dev/rrz1g        per il no. inode  
# ncheck -i 354 /dev/rrz1g        per il nome del file
```

Per sostituire manualmente un blocco errato usare `rzdisk`.

Se un disco risulta danneggiato ed è montato via `fstab`, per evitare problemi all'intero sistema, occorre copiare `fstab` come `fstab.old` e correggere `fstab` eliminando il disco in questione che va smontato da tutte le macchine. In casi estremi, si dovrà fare il `reboot` per ripristinare la funzionalità del sistema.

17.6 PROBLEMI E RIMEDI CON FSTAB

Se si modifica `/etc/fstab` e si cambia un file system, ad esempio `/usr/users` da una partizione ad un'altra per aumentarne l'estensione (es. da `rz1h` a `rz4c`) e ognuna delle due partizioni era esportabile, si deve correggere sia `/etc/fstab` che `/etc/exports`. Notare che il sistema ha in memoria la precedente definizione di `exports`. Per aggiornare le definizioni, usare il comando:

```
# showmount -e
```

oppure fare il `reboot` di tutti i nodi dopo la modifica.

17 PROBLEMI E RIMEDI

17.7 PROBLEMI E RIMEDI CON NFS

Se si cerca di far condividere il file `/etc/passwd` (authorize) da più nodi che non supportano i database condivisi YP di Unix, un metodo può essere la condivisione del file system `/etc`, metodo pericoloso e sconsigliabile di cui segue una descrizione informativa. Per ottenere questa condivisione, dopo avere attivato NFS, come indicato sopra, inserire nel nodo master nel file `/etc/exports` il seguente comando:

```
/etc -r=0
```

inserire nel nodo slave in `/etc/fstab`

```
/usr/etc@master:/etc:rw:1;2:nfs:soft,bg:
```

Per prova fare il mount manuale sul nodo slave con:

```
# mount master:/etc /etc
```

oppure:

```
# mount -a
```

per fare il mount da `/etc/fstab`.

In questo modo quando si usa `/etc` si vedono solo i file del master e non si vedono più i file dello slave. Dato che il master ha un `/etc/fstab` diverso non è più possibile accedere al file `/etc/fstab` dello slave per correzioni e modifiche a `/etc/fstab`. È quindi necessario mettere in halt il master con:

```
# shutdown -h now
```

poi far ripartire lo slave. Quando vengono incontrate su `/etc/fstab` le istruzioni di mount per un file system remoto il cui nodo è in halt il sistema arresta il boot fino al time-out, poi procede normalmente. È quindi possibile agire su `/etc/fstab` e ripristinare le condizioni di lavoro corrette.

17.8 PROBLEMI DI BOOT CON UN CLIENTE

In un sistema client-server, lo stato del sistema è monitorato sul file:

```
/usr/spool/mqueue/syslog.
```

Se un cliente non fa il boot verificare il server con il comando:

```
# ps -aux | egrep "biold|mountd|nfsd|portmap"
```

Verificare il file `/etc/exports` che deve contenere informazioni del tipo:

```
/dlenv0/root0.mips/usr -o -r=0 node
```

```
/dlclient0/node.root -r=0 node
```

Verificare il path del kernel del nodo in questione con `/etc/getnode`:

```
# /etc/getnode node
```

Verificare il file `/etc/hosts` che deve contenere il nome del cliente. Tale nome potrebbe mancare accidentalmente se si fa uso di un name server. Nel caso in cui il name server risulti irraggiungibile, il cliente non fa il boot.

17 PROBLEMI E RIMEDI

17.9 PROBLEMI E RIMEDI CON DMS

Non mettere `dlenv` e `dlclient` in `/etc/fstab` durante le modifiche a `dms`. Infatti se esiste `dlenv0` e lo si vuole ricreare, `dms` cerca di creare `dlenv1` qualora `dlenv0` sia montato da `/etc/fstab`. Verificare anche che sotto `/` non esistano le directory che vengono create da `dms` con l'opzione `c`.

Ricordare inoltre che la voce di menu:

`Install new operating system`

serve *solo* per installare il sistema operativo. Qualsiasi altro prodotto viene installato con l'opzione:

`Add software to existing system`

Da notare infine che le modifiche apportate con `dms` ad un cliente, ad esempio cambio della swap area da file sul server a partizione sul cliente diventano attive *solo* al prossimo reboot del cliente.

17.10 PROBLEMI E RIMEDI CON MOUNT e UMount

Quando si cerca di montare un file system, specialmente durante le operazioni di installazione può capitare che l'operazione di `mount` fallisca perchè il file system non è stato smontato regolarmente. Per ovviare a questo problema occorre verificare lo stato degli inode con `fsck` e precisamente:

```
# fsck          verifica i file system di /etc/fstab
# fsck /dev/rz2e  verifica il file system /dev/rz2e
```

Se si deve portare un nodo in single user per il backup e in `fstab` del nodo sono presenti dischi di altre macchine, l'operazione:

```
# umount -a
```

fallisce. In questo caso occorre specificare l'opzione `-f` per ignorare le informazioni di NFS, si avrà quindi:

```
# umount -af
```

17.11 PROBLEMI E RIMEDI CON CHPT e MOUNT

Se si devono modificare le partizioni di un disco e `chpt` dà errore specificando la partizione `a`, usare la partizione `c`. Da notare che non è sufficiente usare `chpt` per poter accedere alle partizioni del disco. Infatti se si cerca di fare il `mount` della partizione si avrà il messaggio di errore:

```
"bad argument for mount"
```

In questo caso il problema rimane anche tentando di usare `fsck`, perchè il file system non è stato creato, occorre quindi invocare `newfs` prima di procedere.

17 PROBLEMI E RIMEDI

17.12 PROBLEMI E RIMEDI CON YPPASSWD

Quando si fa uso di YP per la gestione degli utenti, è buona norma definire la password di root localmente, cioè al di fuori di YP per poter fare login come *su* anche senza YP almeno per il server. Se ci si attiene a questa norma, il cambio della password di root è anomalo rispetto a quello che avviene per tutti gli altri user che saranno definiti solamente in ambiente YP. In pratica si possono presentare i seguenti casi distinti:

1. tutti i nodi hanno lo user root definito localmente, cioè al di fuori di YP
2. nel server root è definito localmente, mentre nei clienti è definito tramite YP
3. nel server e in parte dei clienti lo user root è definito localmente, mentre nei rimanenti clienti è definito tramite YP

Per cambiare la password si dovrà agire in modo diverso per i nodi definiti localmente rispetto ai nodi definiti in YP e precisamente secondo le seguenti modalità.

- sul server e su tutti i nodi locali la password di root si cambia con `passwd`
- sui clienti definiti con YP, la password di root si cambia con `yppasswd` invocato *dal server*

Se non si rispettano le regole enunciate sopra si ottengono messaggi di errore. Da notare che il cambio della password va fatto contemporaneamente su tutti i nodi per evitare il rischio di avere password diverse e non riuscire ad accedere come *su* ai nodi.

Quale struttura di password definire è una scelta a discrezione del sistemista in dipendenza delle esigenze locali e del numero di nodi da gestire. Ad esempio, se tutti i nodi hanno root definito localmente, occorre modificare la password su ogni nodo. La definizione più razionale potrebbe essere quindi la definizione locale di root per il solo server e tramite YP per tutti i clienti. In questo caso, da *su* sul server, si cambia la password due volte soltanto:

1. per il server con `passwd`
2. per i clienti con `yppasswd`

17.13 PROBLEMI E RIMEDI CON DECNET-OSI

Quando si installa DECnet-OSI via `dms` l'installazione richiede la creazione del kernel su ogni nodo. Apparentemente l'operazione avviene in modo corretto solo sul primo cliente del sistema. Sugli altri nodi fallisce perchè non vengono creati gli appositi link sotto:

```
/dlclient0/hist.root/sys/conf/mips.
```

In questo modo durante la creazione del kernel vengono compilati dei moduli, ad esempio `../../../../data/tc_options_data.c` e l'installazione fallisce con il messaggio:

```
loading vmunix
/bin/ld:
Undefined:
```

17 PROBLEMI E RIMEDI

```
dli_swstack_
netmandomain
*** Error code 1
```

Stop.

Da notare che normalmente la creazione del kernel non dovrebbe implicare la compilazione e il link di moduli.

Per ovviare al problema occorre creare a mano i seguenti link:

```
# ln -s /usr/sys/conf/mips/DNU.script      DNU.script
# ln -s /usr/sys/conf/mips/NETMAN.script   NETMAN.script
# ln -s /usr/sys/conf/mips/files.dli       files.dli
# ln -s /usr/sys/conf/mips/files.netman    files.netman
# ln -s /usr/sys/conf/mips/files.osi       files.osi
```

18 MISCELLANEA

18 MISCELLANEA

In questa sezione vengono fornite informazioni supplementari su compiti occasionali che non fanno parte delle usuali operazioni di routine di manutenzione del sistema.

18.1 INSTALLAZIONE CDROM

La distribuzione del sistema operativo viene fatta oltre che su nastro (es. TK50) anche su disco ottico. Analogamente la documentazione è disponibile oltre che in forma cartacea anche su CD consentendo l'accesso tramite *bookreader* da parte di più stazioni DECW contemporaneamente.

Per accedere al CD disponendo di un'unità di lettura con interfaccia SCSI collegata ad una delle macchine del sistema, occorre procedere secondo gli steps seguenti per rendere disponibile l'informazione suddetta.

Installata l'unità CD sulla macchina, controllare l'indirizzo SCSI con il comando:

```
>> cnfg 3
```

Di solito l'unità, che può essere un modello RRD40 oppure RRD42, viene vista come rz4. Ripartire e dopo il boot verificare se esiste il device:

```
# cd /dev
# ls *rz4*
```

Se il device non esiste, crearlo con:

```
# MAKEDEV rz4
```

Per esaminare il contenuto del CDROM che va inserito nell'unità con opportuno caricatore (RRD4X-AA per RRD40 e RRD4X-CA per RRD42), eseguire il mount read-only (parametro *-r*):

```
# /etc/mount -r /dev/rz4c /mnt
```

Se si tratta del CD con la documentazione, per poter accedere all'informazione, si deve registrare la licenza:

```
# cd /mnt
# ls -l bookbrowser
# bookbrowser
```

Verificare che la licenza sia caricata (stato *active*) con:

```
# lmf list
```

Dopo la registrazione smontare il disco con:

```
# umount /mnt
```

da notare che se ci si trova ancora su *mnt*, bisogna spostarsi su */* per smontare */mnt*, altrimenti il file system risulta attivo.

Verificare che esista la directory che collega *bookreader* ai documenti e precisamente:

18 MISCELLANEA

```
# ls -ld /usr/lib/dxbook
```

Se non esiste crearla e montare il disco con:

```
# mount -r /dev/rz4c /usr/lib/dxbook
```

Per leggere la documentazione fare partire bookreader usando il menu application del session manager oppure usando il comando:

```
$ /usr/bin/dxbook &
```

che attiva il processo in background.

18.2 ESEMPIO DI UPGRADE

Per mantenere il software ad un livello compatibile con l'hardware e con le versioni aggiornate dei prodotti applicativi, si presenta frequentemente la necessità di fare un aggiornamento del sistema operativo. Il problema è più o meno complesso a seconda che si operi in ambiente stand-alone o in ambiente client-server.

In generale quando si deve aggiornare il sistema operativo, si deve operare prima sul server. Una volta che il sistema del server è aggiornato e funzionante, si ripete la procedura con dms per il sistema dei clienti. Usualmente le istruzioni di aggiornamento allegate al *distribution* funzionano correttamente, talvolta però durante un upgrade si possono incontrare dei problemi in particolare con i clienti per i quali il dms opera solo l'installazione, mentre la costruzione del kernel avviene al primo boot del nodo.

Quanto segue è una guida per l'aggiornamento da Ultrix 4.2 a Ultrix 4.2a. Con versioni più recenti di Ultrix il processo di upgrade è stato migliorato e automatizzato, quindi queste note sono date a titolo indicativo e per fornire notizie su alcune *feature nascoste* di Ultrix che normalmente sfuggono al sistemista. In ogni caso si consiglia cautela nelle operazioni di aggiornamento, leggere attentamente le istruzioni, fare il dump del sistema e copiare in un'area opportuna di facile accesso i file personalizzati, quali *fstab*, *rc.local*, *passwd*, etc.

Indicativamente i passi di installazione sono i seguenti. Fare login come root in single-user, smontare tutti i file system di *fstab* (-a) senza aspettare notifiche dal server (-f) per i file system remoti:

```
# umount -a -f
```

Rimontare solo i file system locali:

```
# mount -t ufs
```

Verificare la consistenza dei file system:

```
# /etc/update
```

in cui il comando */etc/update* fa il sync di aggiornamento dei file system. Da notare che tale comando non dovrebbe essere usato direttamente ma da */etc/rc*. Se si installa l'upgrade da TK50 dare il comando:

18 MISCELLANEA

```
# /etc/setld -l /dev/nrmt0h
```

Se si installa l'upgrade da CDROM dare i comandi:

```
# /etc/mount -r /dev/rz4c /mnt
# /etc/setld -l /mnt/RISC/BASE_UPGRADE
```

Dopo l'installazione il sistema riparte in single-user, per ricostruire il kernel:

- Copiare genvmunix su vmunix:

```
# cp /usr/genvmuinx /vmunix
```

- Ripartire e passare in single-user con:

```
# /etc/shutdown -r now
# shutdown now
```

- Conservare il vecchio file di configurazione:

```
# cp /sys/conf/mips/HOSTNAME /sys/conf/mips/HOSTNAME.V4.2
```

- Riconfigurare il sistema interattivamente con:

```
# /etc/doconfig
```

- Dopo doconfig ripristinare la situazione:

```
# mv /vmunix /genvmunix
# mv /sys/MIPS/HOSTNAME/vmunix /vmunix
```

Se si fa uso di un name server, correggere il nome del nodo nella nuova versione di rc.local. Infatti dopo l'upgrade in rc.local compare un'istruzione del tipo:

```
/bin/hostname ds5ze1
```

che va corretta con il nome del dominio, ad esempio:

```
/bin/hostname ds5ze1.bo.infn.it
```

Inoltre se il sistema è una workstation, per abilitare il sistema di finestre girare lo script:

```
# /etc/server_scps
```

che completa l'installazione e aggiorna la tabella /etc/ttys con le informazioni sul video grafico. Alla fine di /etc/ttys deve comparire:

```
:0 "/ust/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession -e"
none on secure window="/usr/bin/Xws bc"
```

18 MISCELLANEA

N.B. Tale riga varia con le versioni di Ultrix come si può vedere confrontando la definizione precedente valida con Ultrix 4.2a con quella che compare nel paragrafo dedicato ai terminali valida per Ultrix 4.1.

Per aggiornare i clienti, teoricamente si dovrebbe usare `/etc/dms` con l'opzione:
`install`

a cui si fornisce il nome del device come indicato per l'installazione sul server. Se il sistema così generato non facesse il boot, si può provare a creare il nuovo sistema copiando il file `genvmunix` del server nell'area del cliente e creare la nuova configurazione usando `doconfig`. Fare attenzione che `doconfig` deve agire sul file di configurazione del cliente ad esempio:

```
/dlclient0/ds5ze2.root/sys/conf/mips/DS5ZE2
```

Inoltre controllare i file e conservare i `vmunix` corretti.

18.3 SCAMBIO RUOLI CLIENT-SERVER

Durante la riconfigurazione hardware di un sistema client-server può rendersi necessario lo scambio di ruoli tra le macchine per cui una macchina che veniva vista da Ultrix come cliente deve assumere il ruolo di server, mentre il precedente server diventa cliente. Il modo più corretto di operare in questa situazione dovrebbe comportare l'installazione da zero del sistema con la successiva riconfigurazione di tutto il complesso. D'altra parte una installazione a partire dalla distribuzione è un'operazione molto onerosa in termini di tempo e di criticità in quanto in questo modo si deve conservare con procedure manuali copia di tutti i files di personalizzazione quali `passwd`, `rc.local`, pagine gialle, etc. per ripristinare l'ambiente di lavoro esistente.

Nel caso in cui si conservino i nomi dei nodi e i relativi ruoli logici, scambiando solo le CPU e non le unità periferiche, un sistema relativamente più rapido consiste nella riconfigurazione hardware del sistema seguita dall'opportuna ricostruzione del kernel del server e del cliente interessati. Unica cautela è la preventiva verifica di `/etc/tty` nel caso in cui i nodi scambiati non siano entrambi dello stesso tipo, cioè tipicamente la macchina destinata a svolgere il ruolo di server non dovrebbe essere una stazione grafica.

Una volta operata la sostituzione hardware delle macchine, si fa il boot da `genvmunix` e si riconfigura il kernel del server usando `doconfig` come già più volte illustrato. Per il cliente si dovrebbe poter procedere da `/etc/dms` chiedendo la rigenerazione del kernel. Purtroppo, almeno con Ultrix 4.2a, dopo tale operazione il cliente non riesce a fare il boot. In questo caso occorre creare un archivio di tutti i file di personalizzazione del cliente o tramite `tar` del cliente in questione (es. `tar cv /dlclient0/ds5ze7.root`) oppure copiando su una directory di archivio i file essenziali, descritti nel paragrafo seguente.

Si usa poi `/etc/dms` per cancellare il cliente e ricrearlo. A questo punto si fa il boot del cliente. Nel nostro caso, il boot non terminava correttamente per la mancanza della directory `/tmp`, il sistema segnalava la mancanza della directory e si fermava in `single-user`. Creando tale directory a mano e ripartendo:

```
# cd /  
# mkdir /tmp  
# reboot
```

18 MISCELLANEA

il processo di boot avviene correttamente, viene attivato `doconfig` e si procede ad una creazione corretta del cliente.

18.4 ARCHIVIO FILE PERSONALIZZATI

I file di sistema che vengono personalizzati in ogni installazione e che quindi dovrebbero essere copiati periodicamente in aree di archivio per garantire l'integrità del sistema in ogni situazione, sono elencati di seguito. Sono divisi in gruppi secondo le loro funzioni: accesso alla macchina, accesso alla rete, licenze, pagine gialle.

accesso alla macchina:

<code>/etc/rc.local</code>	startup di sistema
<code>/etc/passwd</code>	accesso utenti
<code>/etc/groups</code>	accesso gruppi di utenti (se abilitato)
<code>/etc/fstab</code>	montaggio file system
<code>/etc/exports</code>	tabella file system esportati (se abilitato)
<code>/etc/ttys</code>	personalizzazione accesso terminali e stampanti
<code>/etc/motd</code>	messaggio all'atto del login
<code>/etc/svc.conf</code>	configurazione pagine gialle (se abilitate)

accesso alla rete:

<code>/etc/hosts</code>	nodi per tcp/ip
<code>/etc/networks</code>	reti per tcp/ip
<code>/var/dnet/nodes_p</code>	nodi DECnet

Per DECnet fase IV oltre alla personalizzazione dei nodi, potrebbe essere personalizzato lo stato dell'executor ad esempio inibendo alcuni messaggi d'errore. È quindi consigliabile fare il tar di tutta la directory `/var/dnet`.

Per le licenze copiare tutti i file che si trovano su `/usr/var/adm/lmf`. Il data base attivo è il file `ldb`.

Per le pagine gialle oltre a copiare il file di configurazione `/etc/svc.conf` indicato sopra, copiare tutti i domini facendo il tar dell'albero `/etc/yp`.

Oltre ai file indicati sopra, copiare i file con gli script di `dump` e di `tar` che si trovino sotto `/` e, per ogni nodo, conservare su file l'uscita del comando:

```
# cd /  
# ls -ld *
```

per essere in grado di ripristinare le protezioni delle directory e i link in caso di problemi. Ad esempio:

18 MISCELLANEA

NODE-1 ROOT DIRS AND PROTECTION

```
=====
# pwd
/
# ls -ld *
drwxr-xr-x  2 root          2048 Feb 23  1993 bin
drwxr-xr-x  2 root           512 Nov 13  1992 cdrom
lrwxr-xr-x  1 root              8 Feb 26  1993 cern -> /d0/cern
drwxr-xr-x  4 root          512 Sep 29 14:42 d0
drwxr-xr-x  7 root         1024 Feb  5 23:55 d1
drwxr-xr-x  5 root          512 Feb  3 14:28 d2
drwxr-xr-x 11 root         1024 Feb  5 23:04 d3
drwxr-xr-x  3 root         4096 Feb  7 08:46 dev
drwx----- 5 root          512 Nov 11 09:12 dlclient0
drwxr-xr-x  5 root          512 Oct  6 15:19 dlenv0
drwxr-xr-x  6 root         4608 Feb  7 08:46 etc
lrwxr-xr-x  1 root              7 Jun 29  1992 lib -> usr/lib
drwxr-xr-x  2 root         8192 Jun 30  1991 lost+found
drwxr-xr-x  2 root          512 Nov 15  1991 mnt
lrwxr-xr-x  1 root           10 Jul 10  1992 newsdir -> /cern/news
drwxr-xr-x  2 root          512 Jun 29  1992 opr
lrwxr-xr-x  1 root              7 Jun 29  1992 sys -> usr/sys
drwxrwxrwt  2 root          512 Feb  8 08:34 tmp
drwxr-xr-x 32 root         1024 Aug 13 08:53 usr
lrwxr-xr-x  1 root              8 Jun 29  1992 var -> /usr/var
#
=====
```

Se il nome dei file system di lavoro sono del tipo d1, d2, etc., per cambiare la protezione alle directory d1, d2, etc.

```
# chmod 755 d?
```

18.5 SESSION MANAGER: FINESTRE E MOTIF

Il sistema di finestre delle stazioni Ultrix è gestito dal *session manager* che di default è DECwindow. Se è installato *motif* l'utente può modificare il proprio ambiente per richiedere la versione *motif* del *session manager*. Per la personalizzazione operare come segue:

- fare login sulla stazione
- invocare il menu **Customize** della finestra **sm** e selezionare **Window**
- attivare la voce **Other** nella lista **Window Manager** che compare sotto **Window**
- nel campo testo di **Other** battere:

18 MISCELLANEA

```
/usr/lib/DXM/clients/mwm/mwm
```

- eseguire le operazioni di save per memorizzare la modifica che sarà attiva solo al prossimo login

Le operazioni suddette agiscono sul file `.Xdefaults` che si trova nella *home directory* di ciascun utente. Quando si modifica l'ambiente di window il setup precedente viene conservato nel file `.Xdefaults.old`. Segue un esempio di file `.Xdefaults` per DECWindow.

```
#ifdef COLOR
sm.pointer_foreground: #ffffb3b30000
#endif
sm.AutoStart: Clock,DECterm,DECterm
sm.mouse_accel_denominator: 1
sm.mouse_accel_numerator: 7
sm.mouse_accel_threshold: 3
sm.num_AutoStart: 3
#ifdef COLOR
*Background: #2b2b4a4a1717
*Foreground: #d1d1e6e6d7d7
sm.display_background: #616161617f7f
sm.display_pattern: 1
wm*WmForm.BorderColor: #2b2b4a4a1717
wm*WmForm.Foreground: #d1d1e6e6d7d7
#endif
#ifdef COLOR
sm.display_foreground: #545454545454
#endif
*doubleClickDelay: 500
sm.host_list: DS5ZE3::,localhost,VS3BON::
sm.num_hosts: 3
sm.windowManagerName: /usr/bin/dxwm
sm.x: 416
sm.y: 329
```

<---- DECwindow sm

Nel caso di motif il file `.Xdefaults` viene modificato come segue:

```
#ifdef COLOR
sm.pointer_foreground: #ffffb3b30000
#endif
sm.AutoStart: Clock,DECterm,DECterm
sm.mouse_accel_denominator: 1
sm.mouse_accel_numerator: 7
sm.mouse_accel_threshold: 3
sm.num_AutoStart: 3
#ifdef COLOR
*Background: #2b2b4a4a1717
```

18 MISCELLANEA

```
*Foreground: #d1d1e6e6d7d7
sm.display_background: #616161617f7f
sm.display_pattern: 1
wm*WmForm.BorderColor: #2b2b4a4a1717
wm*WmForm.Foreground: #d1d1e6e6d7d7
#endif
#ifdef COLOR
sm.display_foreground: #545454545454
#endif
*doubleClickDelay: 500
sm.host_list: DS5ZE3::,localhost,VS3BON::
sm.num_hosts: 3
sm.windowManagerName: /usr/lib/DXM/clients/mwm/mwm      <---- motif
sm.x: 416
sm.y: 329
```

I due file differiscono per il parametro `sm.windowManagerName` che vale:
per motif: `/usr/lib/DXM/clients/mwm/mwm`
per DECwindow: `/usr/bin/dxwm`

Problemi:

- **Finestre senza cornice:**
se la modifica del *session manager* non avviene in modo corretto, non vengono create le cornici delle finestre, perciò non si possono attivare le medesime, invocare menu, etc. con la conseguenza di non potere neanche chiudere la seduta e di dover fare *reboot* per poter ripristinare il video grafico. Si potrà fare login solo dopo aver corretto da un terminale non grafico il file `.Xdefaults` eventualmente copiando la versione *skeleton* inserita al momento della creazione dello user.
- **Finestre senza scroll bar:**
con Ultrix le finestre vengono create senza *scroll bar* quindi non è possibile rivedere il testo comparso precedentemente su video. Per attivare tale *facility*:
 - attivare il menu `Options` della finestra terminale
 - selezionare la voce `Display`
 - nella finestra `Display` attivare:
`Record Lines Off Top`
`Vertical Scroll Bar`

18.6 GRAFICA: STAZIONE REMOTA, PHIGS, GKS

Il software grafico DECwindow consente la comunicazione tra CPU locali e remote. Se l'utente è loggato su di una stazione, le immagini grafiche possono essere create su di una CPU remota e visualizzate sullo schermo della stazione locale. Per accedere alla stazione locale dalla CPU remota eseguire le seguenti operazioni:

18 MISCELLANEA

1. autorizzare la CPU remota tramite il menu **Security** del *session manager*
2. fare login sul nodo remoto e definire il **DISPLAY**
3. eseguire l'applicazione

In particolare:

1. **Security**

Per abilitare una CPU remota si deve specificare il nome del nodo e il protocollo di trasporto. Se il sistema richiede anche il nome dello user e c'è incompatibilità di sintassi, specificare l'utente come *. Per il trasporto specificare *decnet* o *tcpip*. In pratica:

- attivare il *session manager*
- scegliere **Customize**
- scegliere **Security**
- battere il nome del nodo nelle forma:

nodo: per *tcp/ip*
nodo:: per *DECnet*

- inserire i dati con **Add** e verificare che vengano inseriti correttamente nella tabella **Authorized Users**
- attivare **Apply** e **OK**

2. **DISPLAY**

Per consentire all'applicazione remota l'accesso allo schermo locale, bisogna definire la variabile **DISPLAY** con il nome del nodo locale. Se l'utente è collegato localmente al nodo **LOCAL1** e vuole girare un'applicazione disponibile sul nodo remoto **REMO TX** già autorizzato, si deve loggare sul nodo remoto **REMO TX** e deve specificare che lo schermo del nodo **LOCAL1** deve essere usato per l'uscita grafica. Nel caso di *Ulrix* la definizione della variabile **DISPLAY** dipende dall'applicazione, ad esempio:

```
paw:                    setenv DISPLAY local1::0  
phigs:                 setenv PHIGSconid local1::0  
gks3d:                 setenv GKS3Dconid local1::0
```

Se il trasporto è *tcp/ip*, il nodo è indicato con **local1:0**.

Sia per **phigs** che per **gks** il valore di **wstype** deve essere 211. In caso di problemi verificare le variabili **PHIGSwstype** e **GKS3Dwstype**.

Per visualizzare file **PostScript** :

```
/usr/bin/dxpsview file.ps
```

Per produrre file **PostScript** da **phigs** definire:

18 MISCELLANEA

```
% setenv PHIGSconid file.ps
% setenv PHIGSwstype 61
```

La definizione indicata definisce una stampante BW con orientamento Landscape. La definizione completa di `wstype` è un valore esadecimale del tipo:
`%xm0nn003d` dove:

```
003d      stampante
m         orientamento (0= landscape, 1= portrait)
nn       dimensione carta (per A4, nn=50)
```

Quindi per produrre un'uscita con orientamento Portrait in formato A4 definire:
`% setenv PHIGSwstype %x1050003d`

Errori:

In caso di problemi i messaggi d'errore più frequenti e relativi rimedi sono:

Can't open DISPLAY	verificare il valore di DISPLAY
Client is not authorized to access server	verificare security
Specified workstation cannot be opened	verificare transport

18.7 ANONYMOUS FTP

Una delle prestazioni più utili per concedere accesso pubblico ad una macchina senza compromettere la sicurezza è l'`anonymous login` via `ftp`. Questo login speciale consente un accesso ristretto alla macchina per la copia di file di tipo *pubblico*. Per creare un `anonymous login` procedere come segue:

1. Creare un utente `ftp` e disabilitarlo ponendo come password `*`. Assegnarli una home directory speciale, ad esempio `/usr/ftp`. Si avrà quindi:

```
ftp:*:279:15:anonymous test:/usr/ftp:/bin/csh
```

2. Modificare la directory in modo che `ftp` sia `owner` e nessuno possa scrivere su detta directory:

```
# chown ftp ~ftp
# chmod 555 ~ftp
```

3. Creare la directory `~ftp/bin` di proprietà di `su` su cui nessuno abbia accesso in `write`, porre una copia di `ls` in detta directory. Nella copia usare l'opzione `-p` per conservare data, ID, etc. Proteggere `ls` in modo che sia solo eseguibile. Procedere come indicato di seguito.

```
# mkdir ~ftp/bin
# chown root ~ftp/bin
# chmod 555 ~ftp/bin
# cp -p /bin/ls ~ftp/bin
# chmod 111 ~ftp/bin/ls
```

18 MISCELLANEA

4. Creare la directory `ftp/etc` di proprietà di `su` su cui nessuno abbia accesso in `write`, porre eventualmente una copia *protetta* di `passwd` in cui tutti i campi password sia stati modificati in `*`. Se si lavora in ambiente YP questo passo non è necessario. In ogni caso l'unico account che è richiesto è `ftp`. Procedere come segue:

```
# mkdir ~ftp/etc
# chown root ~ftp/etc
# chmod 555 ~ftp/etc
#
# cp -p /etc/passwd /etc/group ~ftp/etc          <---- non necessario
# chmod 444 ~ftp/etc/passwd ~ftp/etc/group      <---- con YP
```

5. Creare la directory `ftp/pub` di proprietà di `ftp` e leggibile da tutti. Questa directory potrebbe anche essere scrivibile da tutti per lo scambio dei dati, ma per motivi di sicurezza e di gestione spazio si sconsiglia vivamente di attivare la scrittura. Procedere come segue:

```
# mkdir ~ftp/pub
# chown ftp ~ftp/pub
# chmod 555 ~ftp/pub          <--- READ ONLY
#
# chmod 777 ~ftp/pub          <--- read/write !!!!
```

Dopo la creazione delle directory, la situazione di `ftp` darà la seguente:

```
dr-xr-xr-x  5 ftp          512 Sep  9 1992 .
drwxr-xr-x 32 root        1024 Jun 28 16:01 ..
-rw-r--r--  1 root         405 Mar  1 10:43 README
dr-xr-xr-x  2 root         512 Jul 23 1992 bin
dr-xr-xr-x  2 root         512 Jul 23 1992 etc
drwxr-xr-x  3 ftp          512 Mar  1 10:28 pub
```

6. Creare eventualmente un file `README` per illustrare il contenuto di `pub`. Da notare che la protezione `read-only` di `ftp/pub` obbliga ad eseguire il caricamento dei file da parte di `su`, quindi tale situazione è accettabile solo in un ambiente limitato.

A DIRECTORY E COMANDI UTILI SISTEMA

A DIRECTORY E COMANDI UTILI SISTEMA

Le directory principali, i file e i comandi di system management di uso più frequente sono listati nella tabella seguente.

Dir, File e Comandi	Descrizione
/bin	directory dei programmi di utility
/etc	directory di manutenzione e accounting
ac	stampa account
acct	file di history dei processi
accton	programma di account
atrun	scheduler di at (esecuzione dilazionata)
adduser	aggiunge user
bindsetup	link a name server
chown	change owner dei file
chpt	change disk partition
clear	erase del video
config	build configuration files
crash	analisi crash dump
cron	daemon di clock
crontab	tabella per cron (gestione temporizzata processi)
doconfig	configurazione sistema
elcsd	daemon di error log
eli	gestione error log
fsck	consistency check del file system
file	indagine sulla natura di un file
getty	set del terminale
halt	stop processore
hostname	fornisce il nome del nodo
ifconfig	configurazione rete
init	boot
lcp	gestione LAT
lmf	gestione licenze
/etc/login	fa login ed esegue lo shell utente
motd	messaggio del giorno (compare al login)
newfs	crea file system
pac	programma di account per printer
printcap	caratteristiche stampanti
rc	procedura di boot
rc.local	procedura personalizzata di boot (affianca rc)
sa	programma di account del sistema
savecore	salva l'immagine di memoria

A DIRECTORY E COMANDI UTILI SISTEMA

<code>set</code>	set ambiente di lavoro (*)
<code>set -e</code>	termina lo script in caso di errore
<code>set +e</code>	lo script ignora gli errori e prosegue
<code>setld</code>	carica software
<code>shutdown</code>	regolare shutdown
<code>syslog</code>	gestione errori
<code>termcap</code>	caratteristiche terminali
<code>/tmp</code>	directory per i file temporanei
<code>uerf</code>	error log report generator
<code>/usr/adm</code>	directory per l'informazione amministrativa
<code>/usr/bin</code>	directory di utility
<code>/usr/lib</code>	directory delle librerie e dei programmi al di fuori dello shell
<code>/usr/ucb</code>	directory di utility
<code>utmp</code>	storia corrente del login
<code>vipw</code>	editor e consistency check di <code>/etc/passwd</code>
<code>wtmp</code>	storia del login

(*) - Da notare che `set` è un comando intrinseco di shell, quindi non esiste la voce `man set`, per la documentazione occorre fare riferimento alla shell specifica, ad esempio `man sh` in cui compare anche la descrizione di `set` con le relative opzioni.

A.1 STATISTICHE

Il sistema è in grado di produrre statistiche operative con i seguenti comandi:

<code>iostat</code>	statistica I/O
<code>ps</code>	stato processi
<code>uptime</code>	system activity dall'ultimo boot
<code>w</code>	system activity in corso
<code>pstat</code>	tabelle di sistema
<code>netstat</code>	report sui vari protocolli di rete
<code>vmstat</code>	memoria virtuale

I comandi si trovano su `/etc/`. Per maggiori informazioni consultare `man`.

A.2 COMANDI UTILI USER

editors:	<code>ed, vi</code>
compilatori:	<code>cc, f77, awk</code>
search:	<code>grep, egrep, fgrep</code>
compare:	<code>comp, diff</code>
conversioni:	<code>tr</code>
file:	<code>file, find</code>

A DIRECTORY E COMANDI UTILI SISTEMA

Se si deve operare globalmente su file e/o su directory, si utilizza il comando `find` che opera una ricerca su file e directory.

Esempi:

Ricerca di tutti i file `*.h` di un dato albero:

```
find /dir -print | grep [.]h > allh.out
```

ricerca di tutti i file cambiati negli ultimi 7 giorni:

```
find / -ctime 7 -print
```

ricerca nella directory `/usr/users` i file più grandi di 200 blocchi, il cui ultimo accesso risale a due mesi prima

```
# find /usr/users -size +200 -atime +60 -print
```

cambio di protezione di tutti i file nella directory corrente con esecuzione immediata o con la creazione di un file di shell:

```
find . -print | sed 's/./chmod 755 ./' | sh; dir
```

```
find . -print | sed 's/./chmod 744 ./' | sh; dir
```

```
find . -print | sed 's/./chmod 755 ./' >f2
```

Per convertire o eliminare caratteri in un file si usa `tr`. In particolare è necessario usare `tr` per rendere editabili i file creati dirottando l'uscita di `man`.

Esempio:

```
man f77 > man.man
```

```
tr -d '\010\137' <man.man > f77.man
```

```
man ls | tr -d '\010' > ls.man
```

A.3 LOG FILE DI INSTALLAZIONE

Tutte le operazioni di installazione vengono registrate nella directory `/usr/adm` nei file:

<code>install.log</code>	installazione del sistema
<code>install.FS.log</code>	installazione del file system
<code>install.DEV.log</code>	installazione dei device

Il file `install.log` viene aggiornato ad ogni installazione e cresce nel tempo. In detto file sono riportate tutte le informazioni che compaiono su video durante l'installazione. A titolo di esempio, segue un estratto in cui compaiono le informazioni relative alla scelta dei subset e alla creazione del kernel.

File `/usr/adm/install.log`:

A DIRECTORY E COMANDI UTILI SISTEMA

```
$ cat /usr/adm/install.log
```

Enter the selection number for each kernel option you want.

For example, 1 3 : 1 4 7 8

You specified the following kernel options:

```
Local Area Transport (LAT)
Diagnostic/Utilities Protocol (DUP)
Enhanced Security Features
DECnet
```

Is this correct? (y/n) [n]: y

*** SYSTEM CONFIGURATION PROCEDURE ***

Configuration file complete.

Do you want to edit the configuration file? (y/n) [n]: n

*** PERFORMING SYSTEM CONFIGURATION ***

```
working ..... Wed Oct 9 14:20:58 EDT 1991
```

```
working ..... Wed Oct 9 14:22:59 EDT 1991
```

```
working ..... Wed Oct 9 14:25:00 EDT 1991
```

*** DEVICE SPECIAL FILE CREATION ***

```
working ..... Wed Oct 9 14:26:00 EDT 1991
```

*** SOFTWARE INSTALLATION PROCEDURE COMPLETE ***

The following files were created during the installation procedure:

```
/vmunix           - customized kernel
/genvmunix        - generic kernel
/usr/adm/install.log - installation log file
/usr/adm/install.FS.log - file systems log file
/usr/adm/install.DEV.log - special device log file
```

Gli altri file di log vengono creati e/o aggiornati quando viene creato il file system e quando vengono aggiunti nuovi device con MAKEDEV. Segue un esempio per ciascun file.
File */usr/adm/install.FS.log*:

```
$ cat /usr/adm/install.FS.log
```

A DIRECTORY E COMANDI UTILI SISTEMA

Making the root file system on rz0 RZ55.

Warning: 80 sector(s) in last cylinder unallocated

/dev/rrz0a: 40960 sectors in 76 cylinders of 15 tracks, 36 sectors
21.0Mb in 5 cyl groups (16 c/g, 4.42Mb/g, 1920 i/g)

super-block backups (for fsck -b#) at:

32, 8720, 17408, 26096, 34784,

Making the new file system for /usr on /dev/rrz0g RZ55

Warning: 352 sector(s) in last cylinder unallocated

/dev/rrz0g: 477008 sectors in 884 cylinders of 15 tracks, 36 sectors
244.2Mb in 56 cyl groups (16 c/g, 4.42Mb/g, 2048 i/g)

super-block backups (for fsck -b#) at:

32, 8720, 17408, 26096, 34784, 43472, 52160, 60848, 69536, 78224,
86912, 95600, 104288, 112976, 121664, 130352, 138272, 146960, 155648,
164336, 173024, 181712, 190400, 199088, 207776, 216464, 225152, 233840,
242528, 251216, 259904, 268592, 276512, 285200, 293888, 302576, 311264,
319952, 328640, 337328, 346016, 354704, 363392, 372080, 380768, 389456,
.....

Warning: partition table overriding /etc/disktab

File */usr/adm/install.DEV.log*:

```
$ cat /usr/adm/install.DEV.log
```

```
MAKEDEV rz0 rz1 tz5 dc0 xcons lta0 audit
```

```
MAKEDEV: special file(s) for rz0:
```

```
MAKEDEV: special file(s) for rz1:
```

```
rz1a rrz1a rz1b rrz1b rz1c rrz1c rz1d rrz1d rz1e rrz1e rz1f rrz1f ....
```

```
MAKEDEV: special file(s) for tz5:
```

```
rmt0l nrmt0l rmt0h nrmt0h rmt0m nrmt0m rmt0a nrmt0a
```

```
MAKEDEV: special file(s) for dc0:
```

```
mouse tty00 tty01
```

```
MAKEDEV: special file(s) for xcons:
```

```
xcons
```

```
MAKEDEV: special file(s) for lta0:
```

```
tty02 tty03 tty04 tty05 tty06 tty07 tty08 tty09 tty10 tty11 tty12 ....
```

```
MAKEDEV: special file(s) for audit:
```

```
audit
```

B RINGRAZIAMENTI

B RINGRAZIAMENTI

Ringraziamo il gruppo ZEUS della sezione INFN di Bologna che ci ha messo a disposizione i mezzi per la sperimentazione dei sistemi client-server descritti nel presente articolo.

C MARCHI REGISTRATI

Nella documentazione sono stati citati i seguenti marchi registrati.

YP Sun Microsystems, Inc. NFS
Sun Microsystems, Inc. Unix AT&T Ultrix DEC

D

References

- [1] KERNIGHAN B.W., PIKE R. The Unix Programming Environment. Prentice-Hall.
- [2] KERNIGHAN B.W., RITCHIE D.M. The C Programming Language. Prentice-Hall.
- [3] AHO A.V., KERNIGHAN B.W., WEINBERGER P.J. The AWK Programming Language. Addison-Wesley Publishing Company
- [4] **digital educational services.** ULTRIX-32 System Management Student Workbook.
- [5] **digital equipment corporation.** Guide to Remote Installation Services.
- [6] **digital equipment corporation.** Guide to Disk Maintenance.
- [7] **digital equipment corporation.** Guide to Backup and Restore.
- [8] **digital equipment corporation.** Guide to the Yellow Pages Service.
- [9] **digital equipment corporation.** Guide to the Network File System.
- [10] **digital equipment corporation.** Guide to Configuration File Maintenance.
- [11] **digital equipment corporation.** Guide to the Error Logger.
- [12] **digital equipment corporation.** Guide to the BIND/Hesiod Service.
- [13] **digital equipment corporation.** Guide to System Shutdown and Startup.
- [14] **digital equipment corporation.** Advanced Installation Guide.
- [15] **digital equipment corporation.** Guide to Diskless Management services.

REFERENCES

- [16] digital equipment corporation. Introduction to Networking and Distributed System Services.
- [17] digital equipment corporation. Guide to server Setup.
- [18] LAMB L. Learning the vi Editor. O'Reilly & Associates, Inc.
- [19] TALBOTT S. Managing Projects with Make. O'Reilly & Associates, Inc.
- [20] STRANG J., O'REILLY T., MUI L. Termcap and Terminfo. O'Reilly & Associates, Inc.

CONTENTS

Contents

1	Introduzione	1
2	GENERALITÀ	2
3	Hardware DECstation 5000	3
3.1	Firmware	3
3.2	FIRMWARE PER DS 5000/240	4
3.3	ERRORI HARDWARE	5
3.4	INDIRIZZI SCSI	5
3.5	SHUTDOWN E REBOOT	6
4	SISTEMA OPERATIVO - INTERNALS	7
4.1	BOOT	7
4.2	LOGIN	8
4.3	PAGING	8
5	FILE SYSTEM	9
5.1	ERRORI DI FSCK	12
5.2	ESEMPIO DI PARTIZIONI PER UN DISCO RZ58	13
5.3	DIRECTORY, FILE, LINK	14
6	INSTALLAZIONE DEL SISTEMA	15
6.1	CONFIGURAZIONE	16
6.2	INSTALLAZIONE E GESTIONE LICENZE	19
7	INSTALLAZIONE PRODOTTI DI RETE	21
7.1	tcp/ip	21
7.2	DECnet phase IV	22
7.3	LAT	24
8	INSTALLAZIONE DEVICE E DISCHI	25
9	INSTALLAZIONE TERMINALI E STAMPANTI	27
9.1	GESTIONE TERMINALI UTENTI	31
9.2	FORMATO DI TERMCAP	32
9.3	GESTIONE STAMPANTI	33
10	GESTIONE UTENTI	36
10.1	GESTIONE UTENTI IN AMBIENTE STAND-ALONE	36
10.2	GESTIONE UTENTI IN AMBIENTE YELLOW PAGES	37
11	GESTIONE DISCHI: UFS E NFS	38
11.1	GESTIONE SPAZIO	39
11.2	QUOTE	40
11.3	NETWORK FILE SYSTEM	41

CONTENTS

11.4	CREAZIONE DI NETWORK FILE SYSTEM	42
11.5	FORMATO DEL FILE /ETC/FSTAB	43
11.6	SETUP MANUALE DI UN SERVER NFS	43
11.7	SETUP MANUALE DI UN CLIENTE NFS	45
11.8	NFS SECURITY	48
12	MANUTENZIONE FILE SYSTEM	49
12.1	CONTROLLO FILE	49
12.2	LOG FILE	49
12.3	RESIZE SYSERR. USO DI ELI E UERF	51
12.4	RESIZE DEI FILE DI ACCOUNTING	53
13	SISTEMA OPERATIVO: AGGIORNAMENTO E MANUTENZIONE	55
13.1	FILE DI CONFIGURAZIONE E COSTRUZIONE DEL KERNEL	55
13.2	ESEMPI DI FILE DI CONFIGURAZIONE	56
13.3	COSTRUZIONE AUTOMATICA DEL KERNEL	61
13.4	COSTRUZIONE MANUALE DEL KERNEL	62
13.5	ESEMPIO DI UPDATE	63
13.6	SETUP DI UN NUOVO SISTEMA OPERATIVO	63
13.7	INSTALLAZIONE SOFTWARE SUBSET	64
14	SISTEMA OPERATIVO SU SERVER	68
14.1	SERVER SETUP	68
14.2	DMS	68
14.3	CREAZIONE CLIENTI	69
14.4	RIS	70
15	PAGINE GIALLE	71
15.1	SETUP MANUALE DI YP	74
15.2	MODIFICA DELL'AMBIENTE YP	79
16	BACKUP	81
16.1	DUMP E RESTORE	81
16.2	ESEMPIO PRATICO DI RESTORE	85
16.3	TAR	86
16.4	OPSER	88
16.5	NASTRI ANSI	88
16.6	BACKUP REMOTO	88
17	PROBLEMI E RIMEDI	90
17.1	PROBLEMI E RIMEDI: ERRORE TIME	90
17.2	PROBLEMI E RIMEDI: DISTRUZIONE SISTEMA	90
17.3	PROBLEMI E RIMEDI CON UN NUOVO KERNEL	90
17.4	PROBLEMI E RIMEDI CON IL FILE SYSTEM E I PROCESSI	91
17.5	PROBLEMI E RIMEDI PER ERRORE DISCO	92
17.6	PROBLEMI E RIMEDI CON FSTAB	92

CONTENTS

17.7	PROBLEMI E RIMEDI CON NFS	93
17.8	PROBLEMI DI BOOT CON UN CLIENTE	93
17.9	PROBLEMI E RIMEDI CON DMS	94
17.10	PROBLEMI E RIMEDI CON MOUNT e UMount	94
17.11	PROBLEMI E RIMEDI CON CHPT e MOUNT	94
17.12	PROBLEMI E RIMEDI CON YPPASSWD	95
17.13	PROBLEMI E RIMEDI CON DECNET-OSI	95
18	MISCELLANEA	97
18.1	INSTALLAZIONE CDROM	97
18.2	ESEMPIO DI UPGRADE	98
18.3	SCAMBIO RUOLI CLIENT-SERVER	100
18.4	ARCHIVIO FILE PERSONALIZZATI	101
18.5	SESSION MANAGER: FINESTRE E MOTIF	102
18.6	GRAFICA: STAZIONE REMOTA, PHIGS, GKS	104
18.7	ANONYMOUS FTP	106
A	DIRECTORY E COMANDI UTILI SISTEMA	108
A.1	STATISTICHE	109
A.2	COMANDI UTILI USER	109
A.3	LOG FILE DI INSTALLAZIONE	110
B	RINGRAZIAMENTI	113
C	MARCHI REGISTRATI	113
D		113

Index

- account in ambiente yp, 74
- account stand-alone, 36
- account yellow pages, 37
- accounting file, 53
- accounting stampanti, 54
- acct, 53
- administrator, 2
- anonymous ftp, 106
- archivio file personalizzati, 101
- autoconf, 55
- automount, 41
- backup remoto, 88
- bindsetup, 22
- biod, 68
- bookbrowser, 97
- bookreader: dxbook, 98
- boot path, 3
- boot: processi, 7
- cdrom: boot, 4
- chpt, 10
- chpt -q, 10
- chpt: creazione partizioni, 11
- chpt: modifica partizioni, 11
- client, 2
- client-server: scambio ruoli, 100
- clienti: creazione, 69
- clienti: gestione, 69
- cnfg, 3
- comandi utili, 108
- configurazione, 16
- configurazione kernel: esempi, 55
- configurazione yp, 71
- cp: wild card, 36
- date: decnet-osi, 17
- decnet: database, 23
- decnet: fase iv, 22
- decnet: update nodi, 23
- df, 10, 39
- directory di sistema, 38
- directory etc, 7
- dischi: spazio, 39
- dischi: ufs e nfs, 38
- disinstallazione subset, 66
- disktab, 9
- disktab: missing disk, 13
- dlclient, 69
- dlenv, 69
- dms, 68
- dms: dmsdb, 69
- doconfig, 16, 61
- du, 39
- dump, 81
- dump forzato per crash, 49
- dump totale file system, 82
- dump: esempi, 83
- dump: pipe restore, 84
- dumpdates, 82
- eli, 51
- error log, 50, 51
- errore boot cliente, 93
- errore boot, 90
- errore chpt, 94
- errore decnet-osi, 95
- errore disco, 92
- errore display, 106
- errore dms, 94
- errore file system, 91
- errore fstab, 92
- errore kernel, 90
- errore logname, 91
- errore mount, 94
- errore nfs, 93
- errore passwd, 95
- errore time, 90
- errore umount, 94
- errori di fsck, 12
- errori finestre, 104
- errori hardware, 5
- exports, 43
- exports: modifica, 44
- file di configurazione, 15
- file di configurazione, 55
- file di configurazione: generic, 56
- file system, 2

INDEX

- find, 40, 91, 110
- find: esempi, 110
- find: ricerca file grandi, 91
- firmware, 3
- firmware: 500/240, 4
- fsck, 11
- fsck: errori, 12
- fstab, 43
- fstab: esempio, 25
- gks, 104, 105
- halt action, 4
- hesiod.conf, 22
- hosts in ambiente yp, 74
- indirizzi scsi, 5
- inode, 14
- installazione cdrom, 97
- installazione dischi, 25
- installazione nastri, 25
- installazione sistema, 15
- installazione software subset, 64
- installazione: file creati, 65
- ivp, 17, 65
- kernel, 2
- kernel: config, 62
- kernel: doconfig, 61
- kernel: modifiche, 63
- kernel: update, 63
- kill, 92
- lat, 24
- lat: comandi lcp, 35
- lat: stampanti, 35
- licenze, 19
- lmf, 19
- log daemon, 49
- log file, 49
- log file di shutdown, 6
- log file: esempi, 110
- log file: installazione, 17
- logname set to null, 91
- lpacct, 54
- ls: wild card, 102
- makedev, 5
- makedev: dischi, 5
- makedev: nastri, 5
- man, 47
- motif, 102
- mount, 11, 26
- mount manuale nfs, 47
- mount: filesystem locale, 98
- mount: opzioni, 47
- mount: update, 98
- mountd, 45
- name server ip, 22
- nastri ansi, 88
- newfs, 9
- nfs, 41
- nfs: rc.local, 44
- nfs: setup client, 45
- nfs: setup server, 43
- nfs: start manuale, 45
- nfsd, 45
- nice, renice, 92
- opser, 88
- pac, 54
- pagine gialle (vedi yp), 71
- partizioni, 9, 25
- partizioni iniziali del sistema, 9
- partizioni: esempio, 13
- partizioni: paging, 9
- passwd, 36
- password in ambiente yp, 74
- phigs, 104, 105
- portmap, 45, 68
- postscript, 105
- printcap, 34
- printenv, 4
- processi attivi, 7
- processi attivi dms, 69
- processi attivi nfs, 46
- quot, 39
- quota, 40
- rc e rc.local, 7
- rc.local per nfs client, 45
- rc.local per nfs server, 44
- reboot, 6
- resolv.conf, 22
- restart ttys, 28
- restore, 81, 83
- restore: dump pipe, 84
- restore: esempi, 85

INDEX

restore: partizioni a e g, 63
rete: decnet, 22
rete: tcp, 21
ris, 68, 70
rmtab, 44
rpcinfo, 46
scsi: indirizzi, 5
setld, 17
server not responding, 41
server, 2
setenv, 3
setld: configurazione, 67
setld: uso, 64
showmount, 44
shutdown, 6
software subset, 64
spazio disco, 39
stampanti, 27, 33
stop processi, 91
su, 2
subset: control file, 64
superuser, 2
svc.conf, 71
svcsetup, 76
tar, 86
tar su file compressi, 87
tar: esempi, 86
tcp/ip, 21
tcp/ip: modifica nodo, 21
termcap: formato, 32
terminal server, 24
terminal server: stampanti, 35
terminali, 27
timezone, 15, 58, 90
tmp, 7
tr: esempio di formattazione, 110
ttys, 28
ttys: restart, 28
uerf, 52
ufs, 38
umount, 26, 41
umount file system remoto, 47
uncompress e zcat, 87
unsetenv, 4
upgrade, 98
utenti: stand-alone, 36
utenti: yellow pages, 37
wtmp, 53
yellow pages, 37
yp: /bin/domainname, 71
yp: aggiunta server, 79
yp: aggiunta utenti, 79
yp: creazione slave, 76
yp: dominio, 71
yp: eliminazione server, 79
yp: gestione, 73
yp: lista files con ypcat -x, 72
yp: mappe, 71
yp: modifica ambiente, 79
yp: modifica e propagazione mappe, 77
yp: push mappa, 79
yp: setup manuale, 74
yp: struttura dominio, 72
yppasswd, 37
ypwhich, 74
ypxfr: trasferimento mappe yp, 78
zcat: lista file compressi, 87