ISTITUTO NAZIONALE DI FISICA NUCLEARE

Centro Nazionale Analisi Fotogrammi

INFN/TC-92/02
18 Febbraio 1992

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

ULTRIX: GUIDA ALLA GESTIONE DEL SISTEMA

Istituto Nazionale di Fisica Nucleare
Centro Nazionale Analisi Fotogrammi

INFN/TC-92/02
18 Febbraio 1992

ULTRIX: GUIDA ALLA GESTIONE DEL SISTEMA

A.Bassi, M.L.Luvisetto, E.Ugolini
I.N.F.N., Bologna

Il presente articolo è concepito come introduzione 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, eventuali problemi incontrati, la manutenzione dei file system.

1. INTRODUZIONE

Lo sviluppo di macchine con architteture 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 e il suo settaggio sulla macchina.
I marchi registrati citati sono referenziati in appendice.

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 ***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.

In Unix l'hardware è nascosto dall'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

3.1. 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:KN0Z-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 con 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 5, quello della rete è 6.

Per vedere il boot device:

```
>> cnfg 5
```

Per 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
1 = unita' disco
0 = partizione
```

in particolare:

```
>> setenv boot "0/rz1/vmunix -a      (per ambiente multiuser)
>> setenv boot "0/rz1/vmunix      (per ambiente single user)
>> setenv boot "6/mop -a      (per boot dalla rete)
```

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

Per boot automatico: `>> setenev haltaction b`

Per boot manuale: `>> setenev haltaction h`

Per restart forzato con dump della memoria: `>> setenev haltaction r`

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

Boot da disco di sistema: `>> boot`

Boot da disco alternativo: `>> boot 0/rz5/vmunix.new -a`

dove `vmunix.new` è il nuovo file di sistema (default vmunix).

Boot da tk50: `>> boot 5/tz5`

Boot da CDROM: `>> boot 1/rz4/vmunix -a` dove z è lo slot del CDROM unità RRD40

Per boot single user: `>> boot -s`

Per boot multi user: `>> auto`

Per boot alternativo:

```
>> boot -s -f rz(0,#,0)vmunix.new      (single user)
>> boot    -f rz(0,#,0)vmunix.new      (multi user)
```

Per maggiori informazioni sull'uso di console:

```
>> help
```

oppure:

```
>> ?
```

Per i test dello SCSI:

```
>> test 5
```

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

3.2. 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 indefferentemente " 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/shutdwon -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

Il sistemista deve essere in grado di installare e configurare il software, gestire gli utenti ed eventualmente le quote, garantire l'integrità del sistema eseguendo periodici backup, aggiornare il sistema, installare e gestire prodotti software, analizzare il comportamento del sistema 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, tenedo presente che la radice è / e che i file di gestione si trovano sotto `/etc`. Nel seguito i file verrano 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 `/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 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 programmi in base al contenuto di `/etc/crontab`
   - `/etc/update` per la scrittura forzata su disco ogni 30 secs
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`

4.2. LOGIN

Al login hanno luogo i seguenti steps:
`/etc/init`

se c'è un'entry in `/etc/ttys`, crea un processo tramite `/etc/getty`

`/etc/getty`

esegue login al terminale `ttyn`

dà il messaggio da `/etc/gettytab` con la versione del sistema e 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

È analogo al 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 la 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

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

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 un ***raw device*** cioè un device a caratteri. Per visualizzare le informazioni di un disco, occorrono i privilegi di superuser e i comandi:

```
# 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
```

```
# 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
#
```

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
# 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.

Il file system è agibile dopo essere stato montato con:

```
# 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:

```
# 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:

```
# 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. 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 irragiungibili 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 piú 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/cnl201/bin/ypatchy ypatchy
```

In questo modo viene creato un file simbolico ypatchy nella directory /usr/local (che è nel path di ogni utente) al file fisico /cern/cnl201/bin/ypatchy e si consente ad ogni utente di poter eseguire 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 file fisico viene rimosso dal sistema solo quando sono stati cancellati tutti i link al medesimo. La creazione di un link 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

Per l'installazione si deve pianificare lo spazio disco in modo che ci sia spazio di swap pari a 2 o 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. Il software installato (supported e unsupported) richiede circa 150 MB tra /usr e /var, quindi la

partizione ***g*** non dovrebbe essere inferiore a 30 MB per consentire anche l'installazione di altri prodotti come Fortran, DECnet, etc. Configurare il disco in modo da avere le partizioni adeguate.
L'installazione di un sistema nuovo parte da cassetta. Le cassette da usare sono quelle dei ***supported*** tools. Per fare il boot da cassetta, battere:

```
>> boot 5/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 invoca 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 TIME ZONE si risponde ***g***, per ***Daylight Savings*** si risponde ***y***, per l'area geografica, rispondere ***c***.
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 i files 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

(vedi appendice con esempi di files)
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	lmf
Gestiore licenze	lmfsetup
Inserire utenti	adduser
Aggiungere device	MAKEDEV
Settare stampanti	lprsetup
Settare rete locale	netsetup
Settare nfs	nfssetup
Settare installazione remota	ris
Settare diskless	dms
Settare terminali	ttys – gettytab
Settare messaggi	/etc/motd

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:

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

Dopo l'installazione, per corregere l'ora:

```
# date 1730
```

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`.

6.2. INSTALLAZIONE E GESTIONE LICENZE

Il sistema Ultrix è stato modificato per funzionare solo su prodotti per cui è disponibile la licenza con il sistema 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
```

e registrare la licenza secondo i dati del PAK inserendo 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. Quando vengono apportate delle modifiche, queste agiscono sui file e devono essere riportate nel kernel perchè diventino operative. 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:

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

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

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

Le licenze appena registrate vengono abilitate automaticamente.

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, quindi il file su cui viene scritto il PAK va conservato per ripristinare la licenza:

```
# cd /usr/lic.ark                   (spostarsi su una subdir di archivio)
# /etc/lmf ult.lic ultrix           (ricreazione PAK Ultrix su ult.lic)
# /etc/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, 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**.

6.3. INSTALLAZIONE PRODOTTI DI RETE

Dopo l'installazione del sistema si devono attivare i prodotti di rete, quello nativo (***tcp/ip***) ed eventualmente DECnet. 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/nerworks**
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 131.154.1.2 1
```

dove:

```
131.154.1.2 = nodo di routing
1           = no.  hops per raggiungerlo
```

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

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:

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

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***.

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
```

6.4. 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` a livello di `halt`, supponendo che il device sia `rz2`, fare:

```
# MAKEDEV rz2
```

Creare le partizioni disco consultando `/etc/disktab`, ad esempio:

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

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 in precedenza.

Creare la directory a cui associare il disco, ad esempio:

```
# mkdir /libdisk
```

Definire il disco in `/etc/fstab` e montarlo per verificare che il file system sia 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:

`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 bigUltrix e `nfs` per SUN o network
`options`	opzioni di mount (vedi man mount)

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

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

6.5. 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 verrano 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
......
```

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 è:

```
nome comando tipo flags
```

dove:

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

I possibili valori dei flags sono:

`on`	abilita login per il terminale
`off`	disabilita login (default)
`secure`	consente il login di root sul terminale
`su`	consente di collegarsi con su
`nomodem`	ignora segnali di modem (default)
`modem`	riconosce segnali di modem
`window`	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.

```
# @(#)ttys3.1 (ULTRIX) 4/20/90
#
#
#
# namegettytypestatuscomments
#
console"/etc/getty std.9600" vt100 offsecure# console terminal
tty00"/etc/getty std.9600" vt100on modem# lat
tty01"/etc/getty std.9600" vt100on modem# lat
tty02"/etc/getty std.9600" vt100off nomodem# laser
```

```
tty03"/etc/getty std.9600" vt100off nomodem# laser_ps
tty04"/etc/getty std.9600" vt100on modem# lat
tty05"/etc/getty std.9600" vt100on modem# lat
tty06"/etc/getty std.9600" vt100on modem# lat
tty07"/etc/getty std.9600" vt100on modem# lat
ttyd0   "/etc/getty std.9600" vt100     off shared secure # modem line
ttyp0nonenetwork
ttyp1nonenetwork
ttyp2nonenetwork
ttyp3nonenetwork
.....
.....
ttyqfnonenetwork
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession -e" none on
secure window="/usr/bin/Xcfb"
```

Il comando `/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:

```
$ 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
```

```
$ 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
-iexten -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
```

6.6. 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
```

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, cioè 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.

6.7. FORMATO DI TERMCAP

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. `d0 = 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, `cl=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 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[?3l\E[?4l\E[?5l\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:
#----------------------------------------------
```

6.8. 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 termibali le stampanti sono definite dal file `/etc/printcap` che descrive le capacità delle stampanti. 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:

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

Dato che `/usr/lib` è nel `PATH`, non occorre specificarla 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 `/usr/spool/lpd` dove si trovano anche i file:

`lock`	ID del daemon di stampa
`status`	stato della stampante

sono file ASCII stampabili da qualsiasi utente. Per ogni stampa vengono creati il file `cf*` che contiene le instruzioni 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/lpdfilters/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/lpdfilters/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/lpdfilters/ln03rof:\
:lf=/usr/adm/lperr:\
:lp=/dev/tty02:\
:mc#20:\
:mx#0:\
:of=/usr/lib/lpdfilters/ln03rof:\
:op=PSJET_GRI:\
:os=:\
:pl#66:\
:pw#80:\
:rw:\
:sd=/usr/spool/lpd:\
:ts=GRI2:\
:uv=4.0:\
:xc#0177777:\
```

7. GESTIONE UTENTI IN AMBIENTE STANDALONE

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 (`root`, `operator`, 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 `/etc/group`
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. `/bin/sh`, `/bin/csh`, etc.)

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

1	`nome gruppo`	i.e. `system`, `users`, etc.
2	`password`	crittografata, di solito ignorata e sostituita da * per eliminare il test
3	`Group ID`	
4	`nomi utenti`	eventuale elenco dei componenti del gruppo seperati 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:

```
# 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***.

8. 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`.
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:

/	root
/etc	manutenzione, accounting, management
/bin	utilities di shell (es. sh, csh, etc.)
/tmp	temporanea
/dev	device
/usr	file system usr
/usr/adm	funzioni amministrative
/usr/lib	librerie
/usr/ucb	programmi di utilità (es. grep, edit, ftp,etc.)
/usr/hosts	informazioni di LAN
/usr/man	documentazione on–line
/usr/users	utenti (usualmente file system su altro disco, obbligatorio nei sistemi client–server)
/var	log file (su file system separato per client–server)
/var/adm	account, error log, crash dump
/var/spool	per printer, mail, etc.
/var/adm/ris	installazione remota

8.1. 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 da wd:

```
$ cd /any
$ du
```

Per singolo file a partire da wd:

```
$ du -a
```

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

8.2. 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/quotaon` in `/etc/rc.local` e fare reboot. Per disattivare la quota usare `/etc/quotaoff`. Verifiche sulla quota vengono eseguite da:

`quotacheck`	verifica
`quota`	valore attuale
`repquota`	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.

8.3. CREAZIONE DI NFS

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.
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 `/etc/fstab`	Lettura di `/etc/exports`
Verifica server conosciuto	Verifica client conosciuto

Il mount remoto inzia 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`. 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 del server per indicare ai clienti i file system che possono montare
RPC	metodo di Remote Procedure Call per la gestione delle comunicazioni tra server e cliente con semantica di subroutine
XDR	metodi di External Data Representation per la descrizione dei dati remoti

I processi connessi all'uso e la gestione di NFS sono:

Programmi	Operazioni
`nfs`	Network File System
`biod`	Start dei daemon di I/O asincrono
`exports`	definizione sistemi NFS da esportare
`fsirand`	installazione generatore random di i–node
`mount, umount`	monta, smonta NFS
`mountd`	server per le richieste di mount NFS
`nfsasyncdaemon`	invoca daemon NFS
`nfsmount`	monta Network File System
`nfssvc`	invoca daemon NFS
`nfsumount`	smonta NFS
`nfsd`	daemon server di NFS
`nfssetup`	setup di NFS

`nfsstat`	statistica di NFS
`portmap`	mapper per DARPA INTERNET
`mtab`	tabelle dei sistemi locali montati da clienti NFS remoti
`rpcinfo`	informazione su RPC
`showmount`	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.

8.4. 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`.

8.5. 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
```

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

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

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

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.

8.6. 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.
Esempi:
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 acesso di ***su*** :

```
/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 subdiretories. 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=o`, 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 infomazioni statiche da `/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.
Configurazione ethernet:

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

Attivazione del daemon `portmap`:

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

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
```

Attivazione il daemon nfsd:

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

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

```
# /etc/shutdwon -r now
```

In ambiente multiuser dare i comandi:

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

8.7. 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.

Configurare ethernet:

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

Attivare il daemon portmap:

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

Attivare il daemon biod:

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

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/shutdwon -r now
```

in ambiente multiuser dare i comandi:

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

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

```
# ps -aux | egrep "biod|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/rcpinfo`, si può verificare lo stato di ogni singolo processo usando il numero del processo listato nella colonna `program`:

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

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 che il file system è di tipo network
`-o soft`	error code (soft) se il nodo remoto è 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`	il file system remoto è 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
`wsize=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/mountsenza` parametri fornisce l'informazione completa sui file system in uso:

```
$ /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 `sccs` per centrallizare gli update
`/usr/sys`	file di costruzione del sistema per centralizzare il kernel
`/usr/local`	utilità locali (es. librerie CERN)

File system da ***NON*** esportare:

`/etc`	versione locale di `ttys, fstab, passwd`
`/bin`	routine critiche per crash di sistema
`/dev`	l'accesso remoto ai device ***NON*** è 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` può 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 `rwalld`.

8.8. 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.

9. MANUTENZIONE FILE SYSTEM

9.1. 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 `rwx` 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).

9.2. 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/consol`

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

I file vengono creati dai daemon:

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

I daemon `accton` e `elcsd` vengono attivati da `/etc/rc`, `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:

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

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 ad alta priorità

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 `/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 sulla directory `/usr/adm/syserr/` e vengono creati con un nome del tipo `syserr.nodename`. I file di errore di syslog si trovano sulla 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. 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 del file `/usr/adm/shutdownlog`.

9.3. 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
```

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. Indirettamente il comando ***fornisce informazioni*** sull'hardware quale memoria disponibile, floating point, processore, dischi, ethernet, etc.

Log degli errori della giornata:

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

Log degli errori hardware di memoria e cpu:

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

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
```

9.4. RESIZE DEI FILE DI ACCOUNTING

Il sistema produce due tipi di file di accounting:

/usr/adm/wtmp	account user
/usr/adm/acct	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:

```
# 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.

10. AGGIORNAMENTO E MANUTENZIONE DEL SISTEMA

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
```

10.1. 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 su `/usr/sys/conf/mips/HOST` dove HOST è il nome del nodo in ***maiuscole***. 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
```

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:

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 inzia 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.

10.2. 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 un file di configurazione esistente con `/etc/doconfig`, il sistema deve usare il kernel generico `genvmunix`. Come ***su*** eseguire i seguenti steps:

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

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
```

Attivare error log e girare `update`:

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

Salvare il file di configurazione esistente:

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

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

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

Invocare `doconfig`:

```
# cd /
# /etc/doconfig
```

Il programma opera in modo interattivo con domande sul nodo, la data, l'editor del file di configurazione, etc. 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
```

10.3. 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
  ```

- boot del nuovo kernel

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

10.4. 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 puoi 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 qualsisi 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
```

10.5. 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.

Quando viene installato il sistema operativo, viene creata una tabella di tutti i subset esistenti con il flag `installed` se il subset è stato installato. Ogni subset può essere disinstallato in qualsiasi momento. L'operazione rimuove anche tutti i file 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).

I subset vengono gestiti da `/etc/setld` che consente la gestione dei subset con operazioni di indice, verifica, installazione, rimozione, caricamento da nastro a disco del `distribution` etc.

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

Esempi:

Installazione dei subset dall'unità nastro 2:

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

Installazione del subset `UDTUUCP400` dall'unità nastro 2:

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

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 mumbly:  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 UDTUUCT400
```

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
```

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
```

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
scp	shell di installazione con config e IVP

11. SISTEMA OPERATIVO SU SERVER

11.1. 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 files 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***.

11.2. 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
```

Per verificarne la presenza:

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

I file system `dlenv` e `dlclient` possono stare su dischi diversi. Il sistema `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.

11.3. 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
```

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 separzione dei campi. I nuovi clienti vengono aggiunti via editor e registrarti 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

11.4. 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
```

12. PAGINE GIALLE

Un sistema stand–alone non ha bisogno di installare YP (Yellow Pages), che invece è richiesto per sistemi client–server in cui alcune risorse debbano essere condivise. YP si occupa di gestire database attraverso la rete, in particolare i 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
#       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
```

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

Per aggiornare la tabella degli hosts:
```
# cd /var/yp
# make hosts
```

12.1. SETUP MANUALE DI YP

Oltre a settare le mappe di YP, si devono girare i daemons `/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 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:
```
$ 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
# 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 files 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
```
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 di portmap, ypserv e ypbind.

Modificare svc.conf:

svcsetup

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

/etc/yp/ypxfr_1perday

/etc/yp/ypxfr_2perday

/etc/yp/ypxfr_1perhour

a seconda della frequenza di aggiornamento e verificare l'esistenza di /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 di 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 un utente alla mappa:

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.

- 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`.

- Usare direttamente `yppush` per il database:

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

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

  Manualmente dallo slave:

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

  con una chiamata per ogni database da aggiornare.

  Automaticamente con `/etc/cron`

Se si usa `ypxfr` con `cron`, si devono usare o personalizzare i comandi di shell `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 in carico eccessivo del master.

12.2. 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`:

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

Aggiunta utenti agendo su `/var/yp/src/passwd`:

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

In entrambi i casi l'operazione viene completata con:

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

13. BACKUP

13.1. 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:

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

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 2400)
u	se il dump è ok, scrive la data in /etc/dumptapes
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

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
# dump 9u /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 9 dump on Mon Dec 16 11:31:27 1991
DUMP: tape rewinding
#
# cat /etc/dumpdates
/dev/rra1h     9 Mon Dec 16 11:31:27 1991
#
# /etc/dump W
Last dump(s) done (Dump '>' file systems):
/dev/ra1h  (/usr/users) Last dump:  Level 9, 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.

Esempi:
Dump del file system su nastro:

```
# dump 0undf 6250 /dev/rmt0h /usr/users
```
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]
```
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
f	usa il prossimo argomento come input invece di /dev/rmt?
h	estrae le directory invece dei file
i	opera in modo interattivo
m	estrae per inode invece che per nome
r	carica l'intero nastro nella directory corrente, usato per ripristinare un intero disco o un dump incrementale dopo un restore di livello 0
R	inizia restore da un volume specifico di un dump multivolume
t	lista le occorenze dei file indicati
v	invoca il modo **verbose** durante il restore
x	estrae i file indicati
y	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 0f - /usr | (cd /mnt; restore xf -)
```
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:
```
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 `ra1g`, 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 siystem da ripristinare, porsi nella root directory, infine invocare `restore`, secondo gli step indicati:

```
# /etc/newfs /dev/rra1g rz57
# /etc/fsck /dev/ra1g
# /etc/mount /dev/r1ag /usr/users
# cd /usr/users
# /etc/restore r     (per dump mono tape)
# /etc/restore R     (per dump multi tape)
# rm restoresymtable (per cancellare la restore table creata da restore)
# cd /
# /etc/umount /dev/ra1g
# /etc/fsck /dev/ra1g
```

13.2. 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:
```
$ 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
```

13.3. 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:

h	lista opzioni
!sh	escape allo shell (richiede la password di root)

u	show users
s	shutdown multiuser (logout utenti, stop processi, sync del file system)
d	dismount file system
f	check file system
r	restart multiuser
b	backup
halt	halt cpu
n	inizializza opser remoto
q	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.

13.4. 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 direcory temporanea:

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

13.5. 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:

```
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.

14. PROBLEMI E RIMEDI

Nel paragrafo seguente sono indicati alcuni dei problemi che si presentano con maggiore frequenza nella gestione del sistema e i relativi rimedi raggruppati per categoria.

14.1. 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 ...      (per 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
```

14.2. 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
```

14.3. 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
# unmount /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.
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:

`-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 (priorità)

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

14.4. 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`.

14.5. 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 in `/etc/exports`

```
/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 normalemente. È quindi possibile agire su `/etc/fstab` e ripristinare le condizioni di lavoro corrette.

14.6. 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 "biod|mountd|ndsf|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`.

15. APPENDICE

15.1. 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`	dir dei programmi di utilità
`/etc`	dir 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
`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
`seltd`	carica software
`shutdown`	regolare shutdown
`syslog`	gestione errori
`termcap`	caratteristiche terminali
`/tmp`	dir per i file temporanei
`uerf`	error log report generator
`/usr/adm`	dir per l'informazione amministrativa
`/usr/bin`	dir di utility
`/usr/lib`	dir delle librerie e dei programmi al di fuori dello shell
`/usr/ucb`	dir di utility
`utmp`	storia corrente del login
`vipw`	editor e consistency check di `/etc/passwd`
`wtmp`	storia del login

15.2. STATISTICHE

Il sistema è in grado di produrre statistiche operative con i seguenti comandi:

`iostat`	statistica I/O
`ps`	stato processi
`uptime`	attività dall'ultimo boot
`w`	attività in corso
`pstat`	tabelle di sistema
`netstat`	report sui vari protocolli di rete
`vmstat`	memoria virtuale

I comandi si trovano su `/etc/`. Per maggiori informazioni consultare man.

15.3. COMANDI UTILI USER

editors:	`ed, vi`
compilatori:	`cc, f77, awk`
search:	`grep, egrep, fgrep`
compare:	`comp, diff`
conversioni:	`tr`
file:	`file, find`

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
```

cambio di protezione di tutti i file nella dir 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`.

Essempio:

```
man f77 > man.man
tr -d '\010\137' <man.man > f77.man
man ls | tr -d '\010'_ > ls.man
```

15.4. LOG FILE DI INSTALLAZIONE

Tutte le operazioni di installazione vengono registrate nella directory `/usr/adm` nei file:

`install.log`	installazione del sistema
`install.FS.log`	installazione del file system
`install.DEV.log`	installazione dei device

Il file `install.log` viene aggiornato ad ogni installazione e cresce nel tempo. In detto file sono riportate tutte le informazione 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*:

```
$ 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
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, 398144, 406832, 414752, 423440,
432128, 440816, 449504, 458192, 466880, 475568,
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 rz1g rrz1g rz
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 tty13 tty14 t
MAKEDEV: special file(s) for audit:
audit
```

16. 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.

17. MARCHI REGISTRATI

Nella documentazione sono stati citati i seguenti marchi registrati.

YP	Sun Microsystems, Inc.
NFS	Sun Microsystems, Inc.
Unix	AT&T
Ultrix	DEC

Bibliografia

[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.
[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.

INDICE