Sezione 8: comandi per l'amministrazione del sistema

92.1 os32: arp(8)

NOME

'**arp**' - mostra la tabella ARP

SINTASSI

```
arp
```

DESCRIZIONE

Il programma '**arp**' consente di visualizzare la tabella ARP (*Address resolution protocol*, ma senza la possibilità di poter-vi intervenire per modificarla. Pertanto, l'uso del programma è ammissibile per qualunque utente.

os32 fornisce l'accesso alla tabella ARP attraverso il file di dispositivo '`/dev/kmem_arp`', da cui attinge il programma '**arp**'.

FILE

'`/dev/kmem_arp`'
: È il file di dispositivo attraverso il quale è possibile leggere la tabella ARP del kernel di os32.

FILE SORGENTI

'`applic/crt0.mer.s`' [96.1.12]

'`applic/crt0.sep.s`' [96.1.13]

'`applic/arp.c`' [96.1.4]

VEDERE ANCHE

kmem_arp(4) [89.3], *ipconfig(8)* [92.5], *route(8)* [92.9].

92.2 os32: getty(8)

«

NOME

‘**getty**’ - predisposizione di un terminale e avvio di ‘**login**’

SINTASSI

```
getty terminale
```

DESCRIZIONE

Il programma ‘**getty**’ viene avviato da ‘**init**’ per predisporre il terminale, ripristinando anche i permessi predefiniti, e per avviare successivamente il programma ‘**login**’. Prima di avviare ‘**login**’, ‘**getty**’ visualizza il contenuto del file ‘`/etc/issue`’, se disponibile, inoltre mostra almeno l’indicazione del terminale attuale. Va osservato che questa realizzazione di ‘**getty**’ lascia a ‘**login**’ il compito di chiedere l’inserimento del nominativo utente.

FILE

‘`/etc/issue`’
: ‘**getty**’ visualizza il contenuto di questo file prima di avviare ‘**login**’.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

'applic/getty.c' [96.1.16]

VEDERE ANCHE

login(1) [86.14], *issue(5)* [90.3].

92.3 os32: http(8)

NOME

'**http**' - servente HTTP elementare

SINTASSI

```
http porta directory_radice
```

DESCRIZIONE

Il programma '**http**' si comporta come servente HTTP, in ascolto nella porta indicata come primo argomento, per pubblicare i file che si trovano a partire dalla directory specificata come secondo argomento della riga di comando.

Dal momento che os32 non dispone di una shell in grado di interpretare script, il modo migliore per avviare il servizio è quello di inserire la richiesta di avvio del programma '**http**' nel file '/etc/inittab', con una direttiva simile a quella seguente:

```
s0:1:respawn:/bin/http 80 /var/www
```

FILE SORGENTI

'applic/crt0.mer.s' [96.1.12]

'applic/crt0.sep.s' [96.1.13]

'applic/http.c' [96.1.17]

VEDERE ANCHE

inittab(5) [90.2], *arp(8)* [92.1], *ipconfig(8)* [92.9], *route(8)* [92.9], *ping(8)* [92.8], *nc(8)* [86.20].

92.4 os32: init(8)

«

NOME

‘**init**’ - progenitore di tutti gli altri processi

SINTASSI

```
init
```

DESCRIZIONE

Il programma ‘**init**’ viene avviato dal kernel (deve trattarsi precisamente del file ‘`/bin/init`’) come primo e unico processo figlio del kernel stesso. Pertanto, ‘**init**’ deve assumere il numero PID uno.

Questa realizzazione di ‘**init**’ si limita a leggere il file ‘`/etc/inittab`’ per determinare quali programmi figli avviare, senza poter distinguere da diversi livelli di esecuzione. In pratica, all’interno di questo file si indica l’uso di ‘**getty**’, per la gestione dei terminali disponibili.

FILE

‘`/etc/inittab`’

Contiene l’indicazione dei processi che ‘**init**’ deve avviare.

DIFETTI

Con os32 non è possibile associare ai segnali un’azione diversa da quella predefinita; quindi ‘**init**’ non può essere informato

dell'intenzione di arrestare il sistema. Pertanto, tale funzionalità non è stata realizzata nella versione di '**init**' di os32.

FILE SORGENTI

'applic/crt0.mer.s' [96.1.12]

'applic/crt0.sep.s' [96.1.13]

'applic/init.c' [96.1.18]

VEDERE ANCHE

inittab(5) [90.2].

92.5 os32: ipconfig(8)

NOME

'**ipconfig**' - mostra la configurazione delle interfacce di rete

SINTASSI

```
ipconfig
```

DESCRIZIONE

Il programma '**ipconfig**' consente di visualizzare la tabella delle interfacce di rete, con la loro configurazione, ma senza la possibilità di potervi intervenire per modificarla (la configurazione delle interfacce di rete avviene esclusivamente attraverso le opzioni di avvio del kernel). Pertanto, l'uso del programma è ammissibile per qualunque utente.

Va osservato che l'interfaccia '**net0**' è costituita sempre dal dispositivo interno associato all'indirizzo 127.0.0.1 (*loopback*).

os32 fornisce l'accesso alla tabella delle interfacce di rete attraverso il file di dispositivo '`/dev/kmem_net`', da cui attinge il programma '**`ipconfig`**'.

FILE

'`/dev/kmem_net`'
È il file di dispositivo attraverso il quale è possibile leggere la tabella delle interfacce del kernel di os32.

FILE SORGENTI

'`applic/crt0.mer.s`' [96.1.12]

'`applic/crt0.sep.s`' [96.1.13]

'`applic/ipconfig.c`' [96.1.19]

VEDERE ANCHE

kmem_net(4) [89.7], *arp(8)* [92.1], *route(8)* [92.9].

92.6 os32: MAKEDEV(8)

«

NOME

'**`MAKEDEV`**' - creazione dei file di dispositivo

SINTASSI

```
MAKEDEV
```

DESCRIZIONE

'**`MAKEDEV`**' è un programma che crea, nella directory corrente, tutti i file di dispositivo previsti per os32. Tali file devono trovarsi normalmente nella directory '`/dev/`', pertanto, prima di usa-

re '**MAKEDEV**' è necessario che la directory corrente corrisponda precisamente a tale posizione.

OPZIONI

Non sono previste opzioni per l'uso di '**MAKEDEV**', dal momento che vengono creati tutti i file di dispositivo, considerato il loro numero limitato.

NOTE

Tradizionalmente '**MAKEDEV**' viene realizzato in forma di script, ma os32 non dispone di una shell adeguata e non è possibile utilizzare script.

FILE SORGENTI

'`applic/crt0.mer.s`' [96.1.12]

'`applic/crt0.sep.s`' [96.1.13]

'`lib/sys/os32.h`' [95.21]

'`applic/MAKEDEV.c`' [96.1.1]

92.7 os32: mount(8)

NOME

'**mount**', '**umount**' - innesto e distacco di un file system

SINTASSI

```
mount dispositivo dir_innesto [opzioni]
```

```
umount directory
```

DESCRIZIONE

‘**mount**’ innesta il file system contenuto nell’unità di memorizzazione rappresentata dal file di dispositivo che va indicato come primo argomento, nella directory che appare come secondo argomento. Eventualmente si possono specificare delle opzioni di innesto, come terzo argomento.

‘**umount**’ stacca il file system innestato precedentemente nella directory indicata come unico argomento del comando.

OPZIONI DI INNESTO

Opzione	Descrizione
`ro`	Innesta il file system in sola lettura.
`rw`	Innesta il file system in lettura e scrittura. Si tratta comunque del comportamento predefinito, in mancanza di un’opzione contraria.

DIFETTI

Non viene preso in considerazione un eventuale file ‘`/etc/fstab`’; inoltre, l’utente non può conoscere lo stato degli innesti già in essere e, a questo proposito, l’uso di ‘**mount**’ senza argomenti produce semplicemente un errore.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/mount.c`’ [96.1.28]

‘`applic/umount.c`’ [96.1.53]

92.8 os32: ping(8)

«

NOME

'**ping**' - invio di richieste di eco: ICMP ECHO_REQUEST.

SINTASSI

```
ping indirizzo_ipv4
```

DESCRIZIONE

Il programma '**ping**' invia una richiesta di eco (ICMP ECHO_REQUEST) alla destinazione indicata attraverso un indirizzo IPv4. Se si ottiene risposta, il programma termina con successo, altrimenti si ottiene un errore; in ogni caso, viene eseguito un solo tentativo.

Per utilizzare '**ping**' è necessario operare in qualità di utente '**root**'.

FILE SORGENTI

'applic/crt0.mer.s' [96.1.12]

'applic/crt0.sep.s' [96.1.13]

'applic/ping.c' [96.1.30]

VEDERE ANCHE

arp(8) [92.1], *ipconfig(8)* [92.9], *route(8)* [92.9], *http(8)* [92.3], *nc(8)* [86.20].

92.9 os32: route(8)

«

NOME

‘**route**’ - mostra la tabella degli instradamenti

SINTASSI

```
route
```

DESCRIZIONE

Il programma ‘**route**’ consente di visualizzare la tabella degli instradamenti, ma senza la possibilità di potervi intervenire per modificarla (la configurazione delle interfacce di rete e degli instradamenti avviene esclusivamente attraverso le opzioni di avvio del kernel). Pertanto, l’uso del programma è ammissibile per qualunque utente.

os32 fornisce l’accesso alla tabella degli instradamenti attraverso il file di dispositivo ‘`/dev/kmem_route`’, da cui attinge il programma ‘**route**’.

FILE

‘`/dev/kmem_route`’
: È il file di dispositivo attraverso il quale è possibile leggere la tabella degli instradamenti del kernel di os32.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/route.c`’ [96.1.34]

VEDERE ANCHE

kmem_route(4) [89.9], *arp(8)* [92.1], *ipconfig(8)* [92.5].

92.10 os32: umount(8)

Vedere *mount(8)* [92.7].