Sezione 1: programmi eseguibili o comandi interni di shell

86.1 os32: aaa(1)

NOME

‘**aaa**’, ‘**bbb**’, ‘**ccc**’ - programmi elementari avviabili direttamente dal kernel

SINTASSI

```
aaa
```

```
bbb
```

```
ccc
```

DESCRIZIONE

‘**aaa**’ e ‘**bbb**’ si limitano a visualizzare una lettera, rispettivamente «a» e «b», attraverso lo standard output, a intervalli regolari. Precisamente, ‘**aaa**’ lo fa a intervalli di un secondo, mentre ‘**bbb**’ a intervalli di due secondi. Il lavoro di ‘**aaa**’ e di ‘**bbb**’ si conclude dopo l’emissione, rispettivamente, di 60 e 30 caratteri, pertanto nel giro di un minuto di tempo si esaurisce il loro compito.

Il programma ‘**ccc**’ è diverso, ma nasce per lo stesso scopo: controllare la gestione dei processi di os32. Questo programma si

limita ad avviare, ‘**aaa**’ e ‘**bbb**’, come propri processi-figli, rimanendo in funzione, senza fare nulla. Pertanto, se si usa ‘**ccc**’, il suo processo deve essere eliminato in modo esplicito, perché da solo non si concluderebbe mai.

Questi programmi sono indicati soprattutto per l’uso di os32 nella modalità interattiva che precede il funzionamento normale del sistema operativo, per la verifica della gestione dei processi.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/aaa.c`’ [96.1.2]

‘`applic/bbb.c`’ [96.1.5]

‘`applic/ccc.c`’ [96.1.7]

86.2 os32: allocated(1)

«

NOME

‘**allocated**’ - mappa della memoria allocata

SINTASSI

```
allocated
```

DESCRIZIONE

Il programma ‘**allocated**’ si limita a leggere dal file di dispositivo ‘`/dev/kmem_map`’, producendo un elenco delle zone di memoria continue già in uso.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/allocated.c`’ [96.1.3]

VEDERE ANCHE

mmcheck(1) [86.18].

86.3 os32: bbb(1)

Vedere *aaa(1)* [86.1].

86.4 os32: cat(1)

NOME

‘**`cat`**’ - emissione del contenuto di uno o più file attraverso lo standard output

SINTASSI

```
cat [file]...
```

DESCRIZIONE

‘**`cat`**’ legge il contenuto dei file indicati come argomento e li emette attraverso lo standard output, concatenati assieme in un unico flusso.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/cat.c`’ [96.1.6]

VEDERE ANCHE

more(1) [86.19], *ed(1)* [86.11].

86.5 os32: ccc(1)

«

Vedere *aaa(1)* [86.1].

86.6 os32: chgrp(1)

«

NOME

‘**chgrp**’ - cambiamento del gruppo proprietario di un file

SINTASSI

```
chgrp nome_gruppo file...
```

```
chgrp gid file...
```

DESCRIZIONE

‘**chgrp**’ cambia l’utente proprietario dei file indicati. Il nuovo proprietario da attribuire può essere indicato per nome o per numero.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/chgrp.c`’ [96.1.8]

VEDERE ANCHE

chgrp(1) [86.8].

chmod(1) [86.7].

86.7 os32: chmod(1)

«

NOME

‘**chmod**’ - cambiamento della modalità dei permessi dei file

SINTASSI

```
chmod mod_ottale file...
```

DESCRIZIONE

‘**chmod**’ cambia la modalità dei permessi associati ai file indicati, in base al numero ottale indicato come primo argomento.

NOTE

Questa versione di ‘**chmod**’ non permette di indicare la modalità dei permessi in forma simbolica.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/chmod.c`’ [96.1.9]

VEDERE ANCHE

chown(1) [86.8].

86.8 os32: chown(1)

«

NOME

‘**chown**’ - cambiamento del proprietario di un file

SINTASSI

```
chown nome_utente file...
```

```
chown uid file...
```

DESCRIZIONE

'**chown**' cambia l'utente proprietario dei file indicati. Il nuovo proprietario da attribuire può essere indicato per nome o per numero.

FILE SORGENTI

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

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

'applic/chown.c' [96.1.10]

VEDERE ANCHE

chgrp(1) [86.6].

chmod(1) [86.7].

86.9 os32: cp(1)

«

NOME

'**cp**' - copia dei file

SINTASSI

```
cp file_orig file_nuovo...
```

```
cp file... directory_dest...
```

DESCRIZIONE

'**cp**' copia uno o più file. Se l'ultimo argomento è costituito da una directory esistente, la copia produce dei file con lo stesso nome degli originali, all'interno della directory; se l'ultimo argomento non è una directory già esistente, ci può essere un solo file da copiare, intendendo che si voglia creare una copia con quel nome specificato.

DIFETTI

Non è possibile copiare oggetti diversi dai file puri e semplici; quindi, la copia ricorsiva di una directory non è ammissibile.

FILE SORGENTI

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

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

'`applic/cp.c`' [96.1.11]

VEDERE ANCHE

touch(1) [86.26], *mkdir(1)* [86.17].

86.10 os32: date(1)

NOME

'**date**' - visualizzazione o impostazione della data e dell'ora di sistema

SINTASSI

```
date [MMGGhhmm [ [SS] AA ] ]
```

DESCRIZIONE

Se si utilizza il programma '**date**' senza argomenti, si ottiene la visualizzazione della data e dell'ora attuale del sistema operativo. Se si indica una sequenza numerica come argomento, si intende invece impostare la data e l'ora del sistema. In tal caso va indicato un numero preciso di cifre, che può essere di otto, dieci o dodici. Se si immettono otto cifre, si sta specificando il mese, il giorno, l'ora e i minuti dell'anno attuale; se si indicano dieci cifre, le ultime due rappresentano l'anno del secolo attuale; se si immettono dodici cifre, l'anno è indicato per esteso nelle ultime quattro cifre.

ESEMPI

```
# date 123123592012 [Invio]
```

Imposta la data di sistema al giorno 31 dicembre 2012, alle ore 23:59.

FILE SORGENTI

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

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

'applic/date.c' [96.1.14]

VEDERE ANCHE

time(2) [87.59], *stime(2)* [87.59].

86.11 os32: ed(1)

«

NOME

‘**ed**’ - creazione e modifica di file di testo

SINTASSI

```
ed
```

DESCRIZIONE

‘**ed**’ è un programma di creazione e modifica di file di testo che consente di operare su una riga alla volta.

‘**ed**’ opera in due modalità di funzionamento: comando e inserimento. All’avvio, ‘**ed**’ si trova in modalità di comando, durante la quale ciò che si inserisce attraverso lo standard input viene interpretato come un comando da eseguire. Per esempio, il comando ‘**1i**’ richiede di passare alla modalità di inserimento, immettendo delle righe a partire dalla prima posizione, spostando quelle presenti in basso. Quando ci si trova in modalità di inserimento, per poter passare alla modalità di comando si introduce un punto isolato, all’inizio di una nuova riga.

Per il momento, in questa pagina di manuale, si omette la descrizione completa dell’utilizzo di ‘**ed**’.

DIFETTI

Il file che si intende elaborare con ‘**ed**’ viene caricato o creato completamente nella memoria centrale. Dal momento che os32 consente a ogni processo di gestire una quantità limitata di memoria, si può lavorare soltanto con file di dimensioni ridotte.

AUTORI

Questa edizione di ‘**ed**’ è stata scritta originariamente da David I. Bell, per ‘**sash**’ (una shell che include varie funzionalità, da compilare in modo statico). Successivamente, il codice è stato estrapolato da ‘**sash**’ e reso indipendente, per gli scopi del sistema operativo ELKS (una versione a 16 bit di Linux). Dalla versione estratta per ELKS è stata ottenuta quella di os16, con delle modifiche apportate da Daniele Giacomini, tra cui risulta particolarmente evidente il cambiamento dello stile di impaginazione del codice. Dalla versione per os16 è stata ottenuta quella per os32, senza ulteriori modifiche.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/ed.c`’ [96.1.15]

VEDERE ANCHE

shell(1) [86.24].

86.12 os32: kill(1)

«

NOME

‘**kill**’ - invio di un segnale ai processi

SINTASSI

```
kill -s nome_segnale pid...
```

```
kill -l
```

DESCRIZIONE

Il programma '**`kill`**' consente di inviare un segnale, indicato per nome, a uno o più processi, specificati per numero.

OPZIONI

Opzione	Descrizione
`-l`	Mostra l'elenco dei nomi dei segnali disponibili.
`-s` ***nome_segnale***	Specifica il nome del segnale da inviare ai processi.

NOTE

Non è possibile indicare il segnale per numero, perché lo standard definisce i nomi di un insieme di segnali necessari, ma non stabilisce il numero, il quale può essere attribuito liberamente in fase realizzativa.

DIFETTI

os32 non consente ai processi di attribuire azioni alternative ai segnali; pertanto, si possono ottenere solo quelle predefinite. Tutto quello che si può fare è, eventualmente, bloccare i segnali, esclusi però quelli che non sono mascherabili per loro natura.

FILE SORGENTI

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

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

'`applic/kill.c`' [96.1.20]

VEDERE ANCHE

kill(2) [87.29], *signal(2)* [87.52].

86.13 os32: ln(1)

«

NOME

‘**ln**’ - collegamento dei file

SINTASSI

```
ln file_orig file_nuovo...
```

```
ln file... directory_dest...
```

DESCRIZIONE

‘**ln**’ crea il collegamento fisico di uno o più file. Se l’ultimo argomento è costituito da una directory esistente, si producono collegamenti con gli stessi nomi degli originali, all’interno della directory; se l’ultimo argomento non è una directory già esistente, ci può essere un solo file da collegare, intendendo che si voglia creare un collegamento con quel nome specificato.

Essendo disponibile soltanto la creazione di collegamenti fisici, questi collegamenti possono essere collocati soltanto all’interno del file system di quelli originali, senza contare eventuali innesti ulteriori.

DIFETTI

Non è possibile creare dei collegamenti simbolici, perché os32 non sa gestirli.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/ln.c`’ [96.1.21]

VEDERE ANCHE

cp(1) [86.9], *link(2)* [87.30].

86.14 os32: login(1)

NOME

‘**`login`**’ - inizio di una sessione presso un terminale

SINTASSI

```
login
```

DESCRIZIONE

‘**`login`**’ richiede l’inserimento di un nominativo-utente e di una parola d’ordine. Questa coppia viene verificata consultando il file ‘`/etc/passwd`’ e se coincide: vengono cambiati i permessi e la proprietà del file di dispositivo del terminale di controllo; viene cambiata la directory corrente in modo da farla coincidere con la directory personale dell’utente; viene avviata la shell, indicata sempre nel file ‘`/etc/passwd`’ per quel tale utente, con i privilegi di questo. La shell, avviata così, va a rimpiazzare il processo di ‘**`login`**’.

Il programma ‘**`login`**’ è fatto per essere avviato da ‘**`getty`**’, non avendo altri utilizzi pratici.

FILE

File	Descrizione
‘/etc/passwd’	Contiene l’elenco degli utenti, con l’associazione al numero UID, alla parola d’ordine necessaria per accedere, alla shell dell’utente. Le altre informazioni eventuali contenute nel file, non sono usate da ‘**login**’.

FILE SORGENTI

‘applic/crt0.mer.s’ [96.1.12]

‘applic/crt0.sep.s’ [96.1.13]

‘applic/login.c’ [96.1.22]

VEDERE ANCHE

getty(8) [92.2], *console(4)* [89.1].

86.15 os32: ls(1)

«

NOME

‘**ls**’ - elenco del contenuto delle directory

SINTASSI

```
ls [opzioni] [file]...
```

DESCRIZIONE

‘**ls**’ elenca i file e le directory indicati come argomenti della chiamata. Se non vengono indicati file o directory da visualizzare, si ottiene l’elenco del contenuto della directory corrente; inoltre,

questa realizzazione particolare di '**ls**', se si indica come argomento solo una directory, ne mostra il contenuto, altrimenti, se gli argomenti sono più di uno, mostra solo i nomi richiesti, eventualmente con le rispettive caratteristiche se è stata usata l'opzione '**-l**'.

OPZIONI

Opzione	Descrizione
`-a`	Quando si richiede di mostrare il contenuto di una directory (quella corrente o quella specificata espressamente come primo e unico argomento), con questa opzione si ottiene la visualizzazione anche dei nomi che iniziano con un punto, inclusi '.' e '..'.
`-l`	Con questa opzione si ottiene la visualizzazione di più informazioni sui file e sulle directory elencati.

NOTE

Dal momento che os32 non considera i gruppi, quando si usa l'opzione '**-l**', il nome del gruppo a cui appartiene un file o una directory, non viene visualizzato.

FILE SORGENTI

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

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

'`applic/ls.c`' [96.1.23]

86.16 os32: man(1)

«

NOME

‘**man**’ - visualizzazione delle pagine di manuale

SINTASSI

```
man [sezione] pagina
```

DESCRIZIONE

‘**man**’ visualizza la pagina di manuale indicata come argomento, consentendone lo scorrimento in avanti. La «pagina» viene cercata tra le sezioni, a partire dalla prima. In caso di omonimie tra sezioni differenti, si può specificare il numero della sezione prima del nome della pagina.

Le pagine di manuale di os32 sono semplicemente dei file di testo, collocati nella directory ‘`/usr/share/man/`’, con nomi del tipo ‘***pagina*** **.** ***n***’, dove ***n*** è il numero della sezione.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/man.c`’ [96.1.24]

VEDERE ANCHE

cat(1) [86.4].

86.17 os32: mkdir(1)

NOME

‘**mkdir**’ - creazione di directory

SINTASSI

```
mkdir [-p] [-m mod_ottale] [directory] ...
```

DESCRIZIONE

‘**mkdir**’ crea una o più directory, corrispondenti ai nomi che costituiscono gli argomenti.

OPZIONI

Opzione	Descrizione
`-p`	Se la directory che si vuole creare, può richiedere prima la creazione di altre directory, con questa opzione (*parents*) si generano tutte le directory genitrici necessarie, purché quei nomi non siano già usati per dei file.
`-m` ***mod_ottale***	Quando si crea una directory, senza specificare questa opzione, si ottengono i permessi 0777_8 meno quanto contenuto nella maschera di creazione dei file e delle directory. Con l’opzione ‘**-m**’ si vanno invece a specificare i permessi desiderati in modo esplicito.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘applic/mkdir.c’ [96.1.25]

VEDERE ANCHE

mkdir(2) [87.34], *rmdir(2)* [87.41].

86.18 os32: mmcheck(1)

«

NOME

‘**mmcheck**’ - verifica della coerenza tra processi e mappa della memoria allocata

SINTASSI

```
mmcheck
```

DESCRIZIONE

Il programma ‘**mmcheck**’ si limita a leggere dai file di dispositivo ‘/dev/kmem_map’ e ‘/dev/kmem_ps’, per verificare che le informazioni sull’allocazione della memoria relative ai processi in corso siano coerenti con la mappa effettiva.

FILE SORGENTI

‘applic/crt0.mer.s’ [96.1.12]

‘applic/crt0.sep.s’ [96.1.13]

‘applic/mmcheck.c’ [96.1.26]

VEDERE ANCHE

allocated(1) [86.2].

86.19 os32: more(1)

«

NOME

‘**more**’ - visualizzazione di file sullo schermo, permettendo il controllo dello scorrimento dei dati, ma in un solo verso

SINTASSI

```
more file...
```

DESCRIZIONE

‘**more**’ visualizza i file indicati come argomenti della chiamata, sospendendo lo scorrimento del testo dopo alcune righe, consentendo all’utente di decidere come proseguire.

COMANDI

Quando ‘**more**’ sospende lo scorrimento del testo, è possibile introdurre un comando, composto da un solo carattere seguito da [*Invio*], tenendo conto che ciò che non è previsto fa comunque proseguire lo scorrimento:

Comando	Descrizione
`n`	si richiede di saltare al file successivo, ammesso che ce ne sia un altro;
`q`	si richiede di interrompere lo scorrimento e di concludere il funzionamento del programma;
`Spazio`	si richiede di proseguire nella visualizzazione progressiva dei file.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘applic/crt0.sep.s’ [96.1.13]

‘applic/more.c’ [96.1.27]

VEDERE ANCHE

cat(1) [86.4].

86.20 os32: nc(1)

«

NOME

‘**nc**’ - gestione di connessioni arbitrarie di tipo UDP o TCP

SINTASSI

```
nc [-u] [-l] indirizzo porta
```

DESCRIZIONE

Il programma ‘**nc**’ (noto come «netcat») consente di gestire delle connessioni UDP o TCP, utilizzando il flusso di standard input in trasmissione ed emettendo il flusso ricevuto dalla controparte attraverso lo standard output.

OPZIONI

Opzione	Descrizione
`-u`	Utilizza il protocollo UDP invece di TCP.
`-l`	Si mette in ascolto e attende una richiesta di connessione.

Quando non si utilizza l’opzione ‘**-l**’, l’indirizzo e la porta indicati nella riga di comando, rappresentano la controparte che si intende contattare; se invece si usa l’opzione ‘**-l**’, si tratta di indirizzo e porta locali.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/nc.c`’ [96.1.29]

VEDERE ANCHE

arp(8) [92.1], *ipconfig(8)* [92.5], *route(8)* [92.9], *http(8)* [92.3], *ping(8)* [92.8].

86.21 os32: ps(1)

NOME

‘**ps**’ - visualizzazione dello stato dei processi elaborativi

SINTASSI

```
ps [-u|-g]
```

DESCRIZIONE

‘**ps**’ visualizza l’elenco dei processi, con le informazioni disponibili sul loro stato. L’elenco è provvisto di un’intestazione, come si vede nell’esempio seguente:

```
pp  p pg                              T * 0x1000 D * 0x1000 stack
id id rp  tty  uid euid suid usage s addr  size addr  size usage    name
 0  0  0 0000    0    0    0 00.02 r 00000 028e 00000 0000   0% os32 kernel
 0  1  0 0000    0    0    0 00.00 s 0051e 003a 00000 0000   9% /bin/init
 1  2  2 0500 1001 1001 1001 00.00 s 0033d 003a 00000 0000  26% /bin/shell
 1  3  3 0501    0    0    0 00.00 s 0028f 003a 00000 0000  36% /bin/login
 1  4  4 0502    0    0    0 00.00 s 002c9 003a 00000 0000  36% /bin/login
 1  5  5 0503    0    0    0 00.00 s 00303 003a 00000 0000  36% /bin/login
 2  6  2 0500 1001 1001 1001 00.00 R 003b1 003a 00000 0000  20% /bin/ps
```

La prima colonna, con la sigla «ppid», ovvero *parent pid*, riporta il numero del processo genitore; la seconda, con la sigla «pid», *process id*, indica il numero del processo preso in considerazione; la terza, con la sigla «pgrp», *process group*, indica il gruppo a cui appartiene il processo; la quarta colonna, «tty», indica il terminale associato, ammesso che ci sia, come numero di dispositivo, ma in base sedici. Le colonne «uid», «euid» e «suid», riguardano l'identità dell'utente, per conto del quale sono in funzione i processi, rappresentando, nell'ordine, l'identità reale (*real user id*), quella efficace (*effective user id*) e quella salvata precedentemente (*saved user id*).

La colonna «usage» indica il tempo di utilizzo della CPU; la colonna «s» indica lo stato del processo, il cui significato può essere interpretato con l'aiuto della tabella successiva:

Lettera	Significato	Descrizione
R	*running*	in corso di esecuzione
r	*ready*	pronto per essere messo in esecuzione
s	*sleeping*	in attesa
z	*zombie*	terminato e non più in memoria, per il quale si attende di passare lo stato di uscita al processo genitore.

Le prime due colonne «addr» e «size» indicano l'indirizzo iniziale e l'estensione dell'area codice del processo, in memoria (*text*); le altre due successive indicano l'indirizzo iniziale e l'estensione dell'area dati del processo, in memoria (*data*). Gli indirizzi e le dimensioni delle aree di memoria utilizzate appaiono divisi per 1000_{16}, ovvero sono multipli di 4096 byte. La colonna «stack

usage» indica la percentuale di utilizzo dello spazio attribuito alla pila dei dati (*stack pointer*).

L'ultima colonna indica il nome del programma, assieme al suo percorso, con il quale il processo è stato avviato.

OPZIONI

Opzione	Descrizione
`-u`	Nell'elenco mostra i numeri UID relativi al processo. Questa opzione è predefinita.
`-g`	Nell'elenco, al posto dei numeri UID, mostra i numeri GID.

NOTE

L'elenco dei processi include anche il kernel, il quale occupa correttamente la prima posizione (processo zero).

FILE

'**`ps`**' trae le informazioni sullo stato dei processi da un file di dispositivo speciale: '`/dev/kmem_ps`'.

FILE SORGENTI

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

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

'`applic/ps.c`' [96.1.31]

86.22 os32: rm(1)

«

NOME

'**`rm`**' - cancellazione di file

SINTASSI

```
rm file...
```

DESCRIZIONE

‘**rm**’ consente di cancellare i file indicati come argomento.

DIFETTI

Non è possibile eseguire la cancellazione ricorsiva di una directory.

FILE SORGENTI

‘applic/crt0.mer.s’ [96.1.12]

‘applic/crt0.sep.s’ [96.1.13]

‘applic/rm.c’ [96.1.32]

VEDERE ANCHE

unlink(2) [87.62].

86.23 os32: rmdir(1)

«

NOME

‘**rmdir**’ - cancellazione di directory vuote

SINTASSI

```
rmdir directory...
```

DESCRIZIONE

Il programma ‘**rmdir**’ consente di cancellare le directory indicate come argomento, purché queste siano vuote.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/rmdir.c`’ [96.1.33]

VEDERE ANCHE

unlink(2) [87.62].

86.24 os32: shell(1)

NOME

‘**`shell`**’ - interprete dei comandi

SINTASSI

```
shell
```

DESCRIZIONE

‘**`shell`**’ è l’interprete dei comandi di os32. Di norma viene avviato da ‘**`login`**’, in base alla configurazione contenuta nel file ‘`/etc/passwd`’.

‘**`shell`**’ interpreta i comandi inseriti; se si tratta di un comando interno lo esegue direttamente, altrimenti cerca e avvia un programma con il nome corrispondente, rimanendo in attesa fino alla conclusione del processo relativo, per riprendere poi il controllo.

DIFETTI

L’interpretazione della riga di comando è letterale, pertanto non c’è alcuna espansione di caratteri speciali, variabili di ambiente o altro; inoltre, non è possibile eseguire script.

A volte, quando un processo avviato da '**shell**' termina di funzionare, il processo di '**shell**' non viene risvegliato correttamente, rendendo inutilizzabile il terminale. Per ovviare all'inconveniente, si può premere la combinazione [*Ctrl c*], con la quale viene inviato il segnale '**SIGINT**' a tutti i processi del gruppo associato al terminale.

Anche il fatto che un segnale generato con una combinazione di tasti si trasmetta a tutti i processi del gruppo associato al terminale è un'anomalia, tuttavia fa parte delle particolarità dovute alla semplificazione di os32.

FILE SORGENTI

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

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

'`applic/shell.c`' [96.1.35]

VEDERE ANCHE

input_line(3) [88.68].

86.25 os32: t_(1)

«

NOME

'**t_**...' - programmi di prova

DESCRIZIONE

I programmi con prefisso '**t_**...' sono delle prove per verificare alcune funzionalità di os32.

86.26 os32: touch(1)

«

NOME

‘**touch**’ - creazione di un file vuoto oppure aggiornamento della data di modifica

SINTASSI

```
touch file...
```

DESCRIZIONE

‘**touch**’ crea dei file vuoti, se quelli indicati come argomento non sono esistenti; altrimenti, aggiorna le date di accesso e di modifica, sulla base dello stato dell’orologio di sistema.

DIFETTI

Non è possibile attribuire una data arbitraria; inoltre, a causa della limitazione del tipo di file system utilizzato, non è possibile distinguere tra date di accesso e modifica dei file.

FILE SORGENTI

‘`applic/crt0.mer.s`’ [96.1.12]

‘`applic/crt0.sep.s`’ [96.1.13]

‘`applic/touch.c`’ [96.1.51]

86.27 os32: tty(1)

«

NOME

‘**tty**’ - nome del file di dispositivo del terminale associato allo standard input

SINTASSI

```
tty
```

DESCRIZIONE

Il programma '**tty**' individua il dispositivo del terminale associato allo standard input e lo traduce in un percorso che descrive il file di dispositivo corrispondente (ovvero il file di dispositivo che dovrebbe corrispondergli)

FILE SORGENTI

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

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

'applic/tty.c' [96.1.52]

86.28 os32: yes(1)

«

NOME

'**yes**' - visualizzazione ripetuta di un testo

SINTASSI

```
yes [testo]
```

DESCRIZIONE

Il programma '**yes**' emette sullo schermo, all'infinito, il contenuto del testo fornito come argomento, oppure la lettera '**y**'. In tutti i casi, il testo o la lettera sono seguiti dal codice di interruzione di riga (*new line*).

FILE SORGENTI

‘applic/crt0.mer.s’ [96.1.12]

‘applic/crt0.sep.s’ [96.1.13]

‘applic/yes.c’ [96.1.54]