

Sezione 7: varie

91.1 os32: environ(7)

NOME

‘**environ**’ - ambiente del processo elaborativo

SINTASSI

```
extern char **environ;
```

DESCRIZIONE

La variabile *environ*, dichiarata nel file ‘`unistd.h`’, punta a un array di stringhe, ognuna delle quali rappresenta una variabile di ambiente, con il valore a lei assegnato. Pertanto, il contenuto di queste stringhe ha una forma del tipo ‘*nome=valore*’. Per esempio ‘**HOME=/home/user**’.

In generale, l’accesso diretto ai contenuti di questo array non è conveniente, in quanto sono disponibili delle funzioni che facilitano la gestione di questi dati in forma di variabili di ambiente.

Dal momento che le funzioni di accesso alle informazioni sulle variabili di ambiente sono definite nel file ‘`stdlib.h`’, la gestione effettiva dell’array di stringhe a cui punta *environ* è inserita nei file contenuti nella directory ‘`lib/stdlib/`’ di os32. Per la precisione, nel file ‘`lib/stdlib/environment.c`’ si dichiara l’array di caratteri `_environment_table[][]` e array di puntatori a caratteri `_environment[]`:

```
char  _environment_table[ARG_MAX/32][ARG_MAX/16];  
char *_environment[ARG_MAX/32+1];
```

L'array ***_environment_table***[][] viene inizializzato con lo stato delle variabili di ambiente ereditate con l'avvio del processo; inoltre, all'array ***_environment***[] vengono assegnati i puntatori alle varie stringhe che si possono estrapolare da ***_environment_table***[][]. Questo lavoro iniziale avviene per opera della funzione ***_environment_setup***(), la quale viene avviata a sua volta dal file 'crt0.s'. Successivamente, nello stesso file 'crt0.s', viene copiato l'indirizzo dell'***_environment***[] nella variabile ***environ***, di cui sopra.

FILE SORGENTI

'lib/unistd.h' [95.30]
'lib/stdlib.h' [95.19]
'lib/unistd/environ.c' [95.30.9]
'applic/crt0.mer.s' [96.1.12]
'applic/crt0.sep.s' [96.1.13]
'lib/stdlib/environment.c' [95.19.8]
'lib/stdlib/getenv.c' [95.19.10]
'lib/stdlib/putenv.c' [95.19.15]
'lib/stdlib/setenv.c' [95.19.18]
'lib/stdlib/unsetenv.c' [95.19.21]

VEDERE ANCHE

getenv(3) [88.52], *putenv*(3) [88.94], *setenv*(3) [88.104],
unsetenv(3) [88.104].

91.2 os32: socket(7)

**NOME**

‘**socket**’ - gestione dei socket

DESCRIZIONE

Il sistema os32 gestisce esclusivamente socket di dominio Internet, con indirizzi IPv4. Per creare e gestire una connessione si usano le funzioni seguenti:

Funzione	Scopo
<code>socket ()</code>	Crea la terminazione locale di una connessione, ma senza attribuire indirizzi o porte [87.54].
<code>connect ()</code>	Connette un socket locale a un socket remoto [87.11].
<code>bind ()</code>	Attribuisce a un socket locale un indirizzo [87.4].
<code>listen ()</code>	Mette un socket locale in ascolto, attendendo una richiesta di connessione da un socket remoto [87.31].
<code>accept ()</code>	Recepisce una richiesta di connessione ricevuta da un socket locale, generando con questa un nuovo socket locale connesso a quello da cui ha avuto origine la richiesta [87.3].
<code>send ()</code>	Invia dati attraverso una connessione [87.45].
<code>recvfrom ()</code>	Riceve dati attraverso una connessione [87.40].
<code>close ()</code>	Chiude un socket locale [87.10].

Quando si crea un socket per poi contattare un socket remoto in attesa di connessioni, è possibile utilizzare le funzioni *socket()* e *connect()*, senza avvalersi di *bind()*, lasciando che le informazioni mancanti nel socket locale siano determinate o fissate automaticamente.

Quando si crea un socket da mettere in ascolto, dopo la funzione *socket()* è necessario utilizzare *bind()* per determinare i parametri locali; poi, se si tratta del protocollo TCP si usa la funzione *listen()* per attendere una richiesta di connessione da un socket remoto. Quindi, per recepire una richiesta di connessione si usa la funzione *accept()*, con cui si genera un nuovo socket locale connesso a quello remoto che ha emesso la richiesta.

Quando il socket deve essere in ascolto, ma in attesa di pacchetti UDP, dopo l'utilizzo della funzione *bind()* si può passare subito all'uso di *recvfrom()*.

VEDERE ANCHE

socket(2) [87.54], *accept(2)* [87.3], *bind(2)* [87.4], *connect(2)* [87.11], *listen(2)* [87.31].

91.3 os32: undocumented(7)

«

Questa sezione ha il solo scopo di raccogliere i riferimenti ipertestuali dei listati che, per qualche ragione, sono privi di una documentazione specifica.