NAME

cu − call another UNIX system

SYNOPSIS

cu telno [**−tneoi**] [**−s** speed] [**−anytilda**] [**−tandem**]

DESCRIPTION

Cu calls up another UNIX system, a terminal, or possibly a non-UNIX system. It manages an interactive conversation with possible transfers of text files. *Telno* is either the telephone number, with **w**'s at appropriate places to wait for secondary dialtone(s), a telephone number with the appropriate *uucp*(1C) dialcode prefixed, a system name listed in the *uucp* database (*L.sys*(5) file), or a hardwired line. *Cu* distinguishes all these various possibilities by looking at the telno character string. If no slashes appear and the first character is not a number, *cu* looks in the *uucp* database for information about the name. First *cu* accuses the telno string of being a system and looks in the *L.sys*(5) file of *uucp*. That failing, it tries to identify an initial string of alphabetic characters as being a location prefix as found in the *uucp* file *L-dialcodes*(5). That failing, the string is handed to the *conns*(3C) subroutine for an attempt to connect. If a system name was matched or a telephone prefix was found, *cu* generates a new teleno string (or if a hardwired line was found for the system in *L.sys*(5) the device name string is used). If the **−n** option is specified, *cu* prints the system name (if it exists), full telephone number and the line speed and exits. This useful when trying to determine whether you have guessed the correct system name or telephone location prefix.

The telno string is then handed to the *conns*(3C). subroutine which makes the actual connection to the remote system. If it fails, *cu* prints a message and exits.

The **−t** flag is used to dial out to a terminal. The **−anytilda** flag causes *cu* to accept the escape sequences listed below anywhere on a line, not just at the beginning (this is very useful when connecting to DEC ODT). Only the *send* process interprets the escape sequences anywhere on the line; the receive process is unaffected. The **−tandem** flag designates that the TANDEMI and TANDEMO flags are to be set meaning that XON/XOFF processing should take place. This allows *cu* to take and put files at 9600 baud. The **−e** (**−o**) flag designates that even (odd) parity is to be sent. If both **−e** and **−o** are on, marked parity is sent, ie. the high order bit is always set. This is useful when talking to the dataswitch. The **−i** switch puts *cu* into an interactive mode when selecting phone numbers from the *uucp* database. Since there may be more than one entry for a system, this is the only way to select some other entry than the first one encountered. When the **−i** switch is specified, *cu* will ask whether it should use each entry it finds in the *uucp* database. When you respond with **y**<*return*>, *cu* uses that number. Any other response and *cu* will continue looking in the database for another entry for the same system. *Speed* gives the transmission speed (110, 134, 150, 300, 1200, 4800, 9600); 300 is the default value.

After making the connection with the *conns*, *cu* runs as two processes: the *send* process reads the standard input and passes most of it to the remote system; the *receive* process reads from the remote system and passes most data to the standard output. Lines beginning with '~' have special meanings.

The *send* process interprets the following:

~.	terminate the conversation.
~EOT	terminate the conversation
~<*file*	send the contents of *file* to the remote system, as though typed at the terminal.
~!	invoke an interactive shell on the local system.
~!*cmd* ...	run *cmd* on the local system (via **sh −c**).

~$*cmd* ...	run *cmd* locally and send its output to the remote system.
~%take *from* [*to*]	copy file *from* (on the remote system) to file *to* on the local system. If *to* is omitted, the *from* name is used in both places.
~%put *from* [*to*]	copy file *from* (on local system) to file *to* on remote system. If *to* is omitted, the *from* name is used in both places.
~%cd *newdir*	change directory on local system.
~%speed *newspeed*	change the speed of the remote line.
~%anytilda	change the state of the *anytilda* flag to opposite. A "was [OFF | ON]" message is printed.
~%xclude	change the state of the XCLUDE bit for the remote line. A "was [OFF | ON]" message is printed.
~%tandem	change the state of XON/XOFF processing. A "was [OFF | ON]" message is printed.
~%break	send a break to the remote system.
~%?	print a list of wiggle (~) usages.
~~...	send the line ~... .

The *receive* process normally copies data from the remote system to its standard output. Any line from the remote that begins with ~> initiates an output diversion to a file. The complete sequence is:

~> [>] [:] *file*
zero or more lines to be written to *file*
~>

In any case, output is diverted (or appended, if >> is used) to the file. If : is used, the diversion is *silent*, i.e., it is written only to the file. If : is omitted, output is written both to the file and to the standard output. The trailing ~> terminates the diversion.

The use of **~%put** requires *stty*(1) and *cat*(1) on the remote side. It also requires that the current erase and kill characters on the remote system be identical to the current ones on the local system. Backslashes are inserted at appropriate places.

The use of **~%take** requires the existence of *echo* and *tee* on the remote system. Also, **stty tabs** mode is required on the remote system if tabs are to be copied without expansion.

FILES

/dev/null

SEE ALSO

cat(1), stty(1), uucp(1C), conns(3C), dh(4), dn(4), tty(4)

DIAGNOSTICS

Exit code is zero for normal exit, non-zero (various values) otherwise.

BUGS

At speeds greater than 1200 baud, characters are likely to be lost unless the TANDEMI and TANDEMO bits are set by the option on the command line or through the wiggle escape sequence described above.
The algorithm used to send breaks is somewhat unreliable. The requirements for transfers (*stty*(1), *cat*(1), *echo*(1), and *tee*(1)) are not changeable.