Command: **kermit – transfer a file using the kermit protocol**
Syntax: **kermit**
Flags: (many)
Example: kermit # Start kermit

This is a slightly lobotomized *kermit*. The help command, the script facility, and the automatic dial support have been removed. The ? and ESC commands still work, so there is still reasonable built-in help. The only V7 *kermit* feature that does not work is the ability to see whether there are input characters waiting. This means that you will not be able to ask for status during a file transfer (though this is not critical, because *kermit* prints a dot every so often and other special characters whenever there is an error or timeout).

Start *kermit*, and then type the following to open a 2400 baud session, for example:

```
set line /dev/tty1
set speed 2400
connect
```

(It is more convenient if you put these commands in *.kermrc* in your home directory, so that they get done automatically whenever you run *kermit*.) This will connect you to the modem or whatever on the serial port. Now log into the other system.

When you want to transfer files, run *kermit* on the other system. To it, type

```
server
```

This puts its *kermit* into a sort of ''slave mode'' where it expects commands from the *kermit* running on your MINIX system. Now come back to the command level on MINIX *kermit*, by typing the escape character followed by *c*. (*Kermit* will tell you the current escape character when you do the connect command.) At this point you can issue various commands. Your *kermit* will coordinate things with *kermit* on the other machine so that you only have to type commands at one end. Common commands are

get *filename*
put *filename*
remote *dir*

Filenames can include wildcards. By default, *kermit* works in a system-independent, text mode. (In effect it assumes that the whole world is MS-DOS and converts end of line and file names accordingly.) To send binary files, you will want to type

```
set file type bin
```

on both ends before starting any transfers. This disables CR LF to newline conversion. If both of your systems are some flavor of UNIX, you might as well put this in *.kermrc* on both ends and run in binary mode all the time. Also, if both systems are UNIX it is recommended that you use

```
set file name lit
```

on both ends. This causes it to keep file names unchanged, rather than mapping to legal MS-DOS names.

Here is a typical *.kermrc* for use on MINIX:

```
set line /dev/tty1
set speed 1200
set esc 29
set file type bin
set file name lit
set retry 90
set prompt MINIX kermit>
connect
```

On the other end of the line, for example, the host at your local computer center to which you want to transfer files, a typical profile might be:

```
set rec packet 1000
set fil name lit
set fil type bin
server
```

Kermit has many other options and features. For a pleasant and highly readable description of it, see the following book:

Title: Kermit: A File Transfer Protocol
Author: Frank da Cruz
Publisher: Digital Press
Date: 1987
ISBN: 0-932376-88

For information about recent *kermit* developments, versions for other systems, and so forth, please contact:

Christine M. Gianone
Manager, Kermit Development and Distribution
University Center for Computing Activities
Columbia University
612 West 115th Street
New York, N.Y. 10025

Over 400 versions of *kermit* are available, so it is likely there is one for any computer your MINIX system might want to talk to. Columbia University also publishes a newsletter about *kermit* that can be requested from the above address.