UNIVERSITY OF EDINBURGH
Edinburgh Regional Computing Centre

User Note 78

(August 1985)

Title:

An Introduction to KERMIT

Author: Adam Albert-Recht

Contact: Advisory service

Software Support Category: n/a

Synopsis

This Note is an introduction to using the Kermit file transfer program for transferring a file over ordinary serial lines between two machines. Kermit is most commonly used to transfer files between a small micro- or mini-computer and a larger midi- or mainframe computer, or between two micro-computers.

Keywords

File transfer, KERMIT

Edinburgh Regional Computing Centre
James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ. Telephone 031-667 1081

An Introduction to KERMIT

Kermit is a file transfer protocol specifically designed for communication over ordinary serial lines. It is most commonly used to transfer files between a small micro- or mini-computer and a larger midi- or mainframe computer, or between two micro-computers.

This document is intended as an introduction to using the Kermit file transfer program, and takes the user through some of the more common uses of Kermit: transferring a *character* file between two machines. Users who wish more detailed documentation on their implementation of Kermit should consult the relevant Kermit Users' Guide. User Note 79 covers the EMAS implementation of Kermit.

In this document, the smaller machine will be referred to as "micro", and the larger machine as the "host". "KERMIT-X" will be the implementation of Kermit running on the micro, and "KERMIT-Y" will be the implementation running on the host. A sequence of characters actually typed by you at the console will be shown as underlined, for example send, and sequences of characters typed by you which are dependent upon the machine you are using or the version of Kermit you are running are shown as *underlined italics*, for example *host filename*.

This document does not cover how to physically connect the micro to the host, but it is assumed that the microcomputer is connected up to the host instead of a terminal.

Micro to host

First, run Kermit on the micro. This is usually accomplished by typing KERMIT in response to the command prompt. The micro will respond with various messages to do with the version of Kermit you are running etc, and finally with:

```
KERMIT-X>
```

which is the micro's Kermit command prompt (in these examples). The Kermit command prompt for Kermit-86 on machines running MS-DOS, for example, is:·

```
KERMIT-86>
```

At this point, you may need to set up some parameters which are dependent upon the implementation of Kermit which you are running. See Appendix I (Implementation-dependent documentation) for details of what you have to do on your machine. Having done this, type:

```
KERMIT-X> connect
```

which will connect your micro to the host computer, as if your micro was an ordinary terminal. Log into the host as usual, and run Kermit on the host (again, usually accomplished by just typing KERMIT). The host will respond with various messages, again to do with the version number of Kermit etc, and finally with:

```
KERMIT-Y>
```

which is the host's Kermit command prompt in these examples. The Kermit command prompt for Kermit-32 on a VAX running VMS, for example, is:

```
KERMIT-32>
```

Then do:

```
KERMIT-Y> receive host filename
```

followed by the sequence of control characters necessary to return control to the micro (see Appendix I for details of what this is on your machine). You will find that you are now back at the micro, still running Kermit. Type:

```
KERMIT-X> send micro filename
```

The file called *micro filename* will be sent to the host, and called *host filename* there. The transfer may take a while depending on the size of the file being transmitted. Once the transfer is complete (a message will appear on the screen to alert you to this fact), you should do:

```
KERMIT-X> connect
```

to get you back to the host, then:

```
KERMIT-Y> exit
```

Now log out of the host computer (using QUIT or whatever), and then again type the sequence of control characters to return control to the micro (see Appendix I). Finally,

```
KERMIT-X> exit
```

finishes this session of Kermit.

Host to Micro

First, run Kermit on the micro. This is usually accomplished by typing KERMIT in response to the command prompt. The micro will respond with various messages to do with the version of Kermit you are running etc, and finally with:

```
KERMIT-X>
```

which is the Kermit command prompt. At this point, you may need to set up some parameters which are dependent upon the implementation of Kermit which you are running. See Appendix I (Implementation-dependent documentation) for details of what you have to do on your machine. Having done this, type:

```
KERMIT-X> connect
```

which will connect your micro to the host computer, as if your micro was an ordinary terminal. Log into the host as usual, and run Kermit (again, usually accomplished by just typing KERMIT). Then do:

```
KERMIT-Y> send host filename
```

followed by the sequence of control characters necessary to return control to the micro (see Appendix I for details of what this is on your machine). You will find that you are now back at the micro, still running Kermit. Type:

```
KERMIT-X> receive micro filename
```

The file called *host filename* will be sent to the micro, and called *micro filename* there. The transfer may take a while depending on the size of the file being transmitted. Once the transfer is complete (a message will appear on the screen to alert you to this fact), you should do:

```
KERMIT-X> connect
```

to get you back to the host, then:

```
KERMIT-Y> exit
```

Now log out of the host computer (using QUIT or whatever), and then again type the sequence of control characters to return control to the micro (see Appendix I). Finally,

```
KERMIT-X> exit
```

finishes this session of Kermit.

Micro to Micro

Transferring files from one micro to another is considerably simpler. The micros are assumed to be connected back-to-back. Since smaller machines tend not to allow commands to be typed at anything but their own keyboard, no connecting is necessary. When you wish to type something at the other machine, just type it on its own keyboard! A summary of a typical transfer between two micros is shown below:

```
           Micro 1                          Micro 2

KERMIT-X> send filename 1

                                  KERMIT-Y> receive filename 2

KERMIT-X> exit

                                  KERMIT-Y> exit
```

APPENDIX I: Implementation-dependent documentation

On most micro implementations of Kermit, you must tell Kermit what baud rate you wish to use in talking to the host. In most cases, this will depend on the type of connection you are using. For TCPs the baud rate is usually 1200, and for PADs the baud rate is variable, but 9600 is usually best. Use the command:

```
KERMIT-X> set baud nnnn
```

where *nnnn* is the baud rate.

KERMIT-11

Kermit-11 is the implementation of Kermit for the PDP-11 or LSI-11 running RT11. Note that multi-terminal support, or the XL handler, must be present for Kermit to be able to connect to the host. Note also that the current implementation of Kermit (version T2.26) does not support the `receive` *micro filename* syntax. The filename on the PDP-11 is the same as the host filename in all cases.

Before setting the baud rate (if speed can be set on your machine), you must first define an output port. This is the physical port that the line to the host is connected to. You define it by:

```
KERMIT-11> set line n
```

where *n* is the number of the output port. If you are using the XL handler, this should become:

```
KERMIT-11> set line XL:
```

The sequence of control characters necessary to return control to the micro after connecting to the host is CTRL+\ followed by C.

KERMIT-86

Kermit-86 is the implementation of Kermit for the Sirius, Apricot, and IBM-PC microcomputers running MS-DOS or PC-DOS. On these machines, you only need to set up the baud rate before initiating the transfer.

The sequence of control characters necessary to return control to the micro after connecting to the host is CTRL+] followed by C.

BBC Kermit

The BBC Kermit command prompt is "BBC> ". This machine only requires that you set up the baud rate before initiating the transfer.

The sequence of control characters necessary to return control to the micro after connecting to the host is CTRL+f_0.

EMAS Kermit

This is described fully in User Note 79. To access it, give the once-only command:

```
Command:OPTION SEARCHDIR=MICROS.KERMDIR
```

and to run it, just type:

```
Command:KERMIT
```

Type HELP in response to the Kermit-EMAS> prompt for on-line documentation.