CP/M-86 KERMIT VERSION 2.9 USER GUIDE

F. da Cruz

Columbia University Center for Computing Activities
New York, New York 10027

December 1984

1. CP/M-86 KERMIT

Authors: Bill Catchings, Columbia University; Ron Blanford, University of Washington; Richard Garland, Columbia University.
Language: Digital Research ASM86
Version: 2.9
Date: December 1984
Documentation:
Frank da Cruz, Columbia

This version of KERMIT is designed to support any CP/M-86 system. So far it supports the DEC Rainbow-100 and the NEC Advanced Personal Computer (APC). It is very similar to CP/M-80 and MS DOS KERMIT.

CP/M-86 KERMIT-86 Capabilities At A Glance:

Local operation:	Yes
Remote operation:	No
Transfers text files:	Yes
Transfers binary files:	Yes
Wildcard send:	Yes
^X/^Y interruption:	Yes
Filename collision avoidance:	Yes
Can time out:	Yes
8th-bit prefixing:	Yes
Repeat count prefixing:	No
Alternate block checks:	No
Terminal emulation:	Yes, uses PC firmware (VT100)
Communication settings:	Yes; duplex, parity
Transmit BREAK:	Yes
IBM communication:	Yes
Transaction logging:	No
Session logging (raw download):	Yes
Raw upload:	No
Act as server:	No
Talk to server:	Yes; SEND, GET, FIN, BYE
Advanced commands for servers:	No
Local file management:	Yes
Handle file attributes:	No
Command/init files:	Yes
Printer control:	No

CP/M-86 KERMIT Description

Since Kermit-86 runs on a standalone micro, it is always in control of the screen -- it is always *local*. Thus, it always keeps the screen updated with the file name and the packet number, whether sending or receiving. Kermit-86 is capable of timing out an input request, and can thus break deadlocks automatically. In most cases, however, this is not desirable because the KERMIT on the other side is most likely better able to handle the timeouts; therefore, Kermit-86's timer is normally not used.

If despite the timeout capability, the transmission appears to be stuck (and you can tell that this has happened if the screen fails to change for a long while) you can type carriage return to have the micro do what it would have done on a timeout, namely NAK the expected packet to cause to foreign host to send it again (or, if the micro is sending, to retransmit the last packet). Micro/micro or micro/IBM-mainframe transfers could require this kind of manual intervention.

File transfers may be interrupted in several ways.

Control-C
: This will return you to Kermit-86 command level immediately, so that you can connect back to the remote system, or take any other desired action.

Control-X
: When sending a file, this will terminate the sending of the current file with a signal to the KERMIT on the other side to discard what it got so far. If there are more files to be sent, KERMIT-86 will go on to the next one. When receiving a file, KERMIT-86 will send a signal to the remote KERMIT to stop sending this file. If the remote KERMIT understands this signal (not all implementations of KERMIT do), it will comply, otherwise the file will keep coming. In either case, the remote KERMIT will go on to the next file in the group, if any.

Control-Z
: Like Control-X, except if a file group is being transmitted, this will stop the transmission of the entire group. If only a single file is being transmitted, it works exactly like Control-X.

Carriage Returns
: If you type carriage return repeatedly Kermit-86 will retry the current packet up to its retry limit (somewhere between 5 and 16 times) and then, if no valid response was received, return to Kermit-86 command level.

When KERMIT-86 is started, it looks for the file `KERMIT.INI`. If found, it executes KERMIT-86 commands from it before prompting you for commands. The KERMIT-86 prompt looks like this:

```
Kermit-86 B3>
```

in which "B" is your current default disk and "3" is the current default user number.

1.1. Kermit-86 Commands

KERMIT-86 uses the DECSYSTEM-20 keyword style command language. Each keyword may be abbreviated to its minumum unique length. "?" may be typed to request a menu of the available options for the current field at any point in a command. ESC may be typed at any point in a command to fill out the current keyword or filename; if sufficient characters have not been typed to identify the current field uniquely, KERMIT-86 will sound a beep and allow you to continue from that point.

CONNECT
: Establish a "virtual terminal" connection to any host that may be connected to the serial port, i.e. pass all typein to the serial port and display all input from the serial port on the screen, using the system's own built-in support for ANSI (VT100-like) screen control. When you issue the CONNECT command, the PC will print a message telling you how to get back by typing an an escape sequence, an uncommonly-used control character, normally CTRL-backslash, followed by a single letter "command".

- C Close Connection, return to `Kermit-86>` command level.
- `?` List available single-character commands.
- B Send a BREAK signal.
- Q Quit logging the remote session.
- R Resume logging the remote session.
- L Toggle logging.
- `^\` (or whatever - a second copy of the escape character) Send the escape character itself to the remote host.

SEND *filespec*
: Send file(s) specified by *filespec* to the remote Kermit, using the prevailing file mode (ASCII or BINARY; see SET). The *filespec* may contain CP/M wildcards.

RECEIVE
: Receive file(s) from the remote Kermit. Store them under the names provided in the file headers supplied by the remote host. If the names aren't legal, use as many legal characters from the name as possible (see the description of SET FILE-WARNING below). If there's a conflict, and FILE-WARNING is ON, warn the user and try to build a unique name for the file by adding "&" characters to the name. You may also provide an optional file name in the RECEIVE command; if you do, the incoming file will be stored under the name you specify. If more than one file arrives, only the first will be stored under the given name, unless you included wildcard characters in the RECEIVE filespec; in that case, the filespec will be used as a mask for incoming filenames. For instance, you told the remote Kermit to send `*.ASM`, you could tell KERMIT-86 to "receive `*.A86`", thereby changing the filetype of all the incoming files.

GET *filespec*
When Kermit-86 is talking to a Kermit Server on the host, you should use the GET command to request the server to send files to you, for example: `get hlp:k*.hlp`

BYE When talking to a remote Kermit Server, this command shuts down the server and logs it out, and also exits from Kermit-86 to CP/M command level.

LOGOUT
Like BYE, but leaves you at Kermit-86 command level.

FINISH Like LOGOUT, but shuts down the remote server without logging it out. Leaves you at Kermit-86 command level; a subsequent CONNECT command should put you back at host system command level.

EXIT Exit from KERMIT-86 back to CP/M.

QUIT Synonym for EXIT.

SET *parameter* [*value*]
Set the specified parameter to the specified value. Possible settings:

BAUD Change the baud rate of the communications port. This command only works on some systems, and its actual operation can vary from system to system. Type SET BAUD followed by a question mark, and follow the directions. On systems that do not support this command, you must set the port baud rate from CP/M or other setup mechanism outside of KERMIT-86.

DEBUG ON or OFF. If ON, displays incoming and outbound packets during file transfer. OFF by default.

DEFAULT-DISK *disk/user*
Specify default disk and user number for subsequent file reception and transmission. The specification following the command must be in one of the following forms:

`d:` = go to drive d (A through P) without changing user
`u:` = go to user u (0 through 15) without changing drive
`du:` = go to drive d and user u
`:` = go to the defaults when Kermit was loaded

Whenever a drive is specified, even if it is the same as the current default drive, the drive is logged in so that disks can be swapped without exiting Kermit to type control-C. Kermit restores the original drive and user upon termination.

ESCAPE Change the escape character for virtual terminal connections. Select a character in the control range that you will not be likely to need at the remote host; type the new character literally. Certain characters, like Control-X, cannot be specified.

FILE-TYPE
Tells KERMIT-86 what kind of file it is sending, so that KERMIT can correctly determine the end of the file. SET FILE BINARY means to send all the 128-byte blocks of the file, including the last block in its entirety; SET FILE ASCII is used for text files, and transmission stops when the first Control-Z is encountered anywhere in the file (this is the CP/M convention for marking the end of a text file). If binary transmission is used on a text file, some extraneous characters (up to 127 of them) may appear at the end of the file on the target system. If ASCII transmission is used on a binary file, the entire file will not be sent if it happens to contain any data bytes that correspond to Control-Z. ASCII is the default.

FLOW-CONTROL
Select the desired type of flow control to be used on the communication line. The choices are NONE and XON/XOFF. XON/XOFF is the default. If the remote system is not full duplex or cannot do XON/XOFF, you should use NONE.

IBM ON (or OFF)
Allow the transfer of files to and from an IBM mainframe computer. This makes Kermit-86 wait for the IBM turnaround character (XON), ignore parity on input, add appropriate parity to output, and use local echoing during CONNECT. As distributed, KERMIT-86 uses MARK parity for IBM communication. If you don't give this command, IBM mode is OFF. Since IBM VM/CMS KERMIT does not have timeout capability, SET IBM ON also turns on the timeout facility automatically, as if you had typed "SET TIMER ON".

LOCAL-ECHO ON (or OFF)
: When you CONNECT to a remote host, you must set LOCAL-ECHO ON if the host is half duplex, OFF if full duplex. OFF by default.

LOG
: Specify a log file on the current CP/M disk into which to record incoming characters during CONNECT. If the remote host can do XON/XOFF, then the log file will normally capture every character shown on the screen. When connected to the remote system, several single-character arguments to the connect escape character can be used to control logging -- Q (quit), R (resume), L (toggle). If you use R or L during connect without having previously specified a log file name, then `KERMIT.LOG` is used. An open log is closed when you escape back to the PC.

PARITY
: Sets parity for outgoing characters to one of the following: NONE, SPACE, MARK, EVEN, or ODD. On input, if parity is NONE, then the 8th bit is kept (as data), otherwise it is stripped and ignored. The parity setting applies to both terminal connection and file transfer. If you set parity to anything other than NONE, Kermit-86 will attempt to use "8th bit prefixing" to transfer binary files. If the other KERMIT is also capable of 8th bit prefixing, then binary files can be transferred successfully; if not, the 8th bit of each data byte will be lost (you will see a warning on your screen if this happens).

PORT
: Allows you to switch between different communication ports on the PC. This command is not available on all systems.

TIMER ON (or OFF)
: Enable or disable the timeout facility. The timer is off by default, because in the normal case KERMIT-86 is communicating with a mainframe KERMIT that has its own timer. Mainframe KERMIT timers tend to be more precise or adaptable to changing conditions. You should SET TIMER ON if you are communicating with another KERMIT that does not have a timer. You should SET TIMER OFF if you are communicating over a network with long delays.

WARNING ON (or OFF)
: Warn user of filename conflicts when receiving files from remote host, and attempt to generate a unique name by adding "&" characters to the given name. OFF by default.

SHOW
: Show the current settings of the SET parameters.

TAKE
: Take KERMIT-86 commands from the specified file. The file should not contain any TAKE commands; nested command files do not work.

LOCAL
: This is a prefix for local file management commands, to distinguish them from remote file management commands (which aren't implemented yet). The LOCAL prefix is optional; if left off, the commands will be performed locally.

SPACE
: Show how much space is used and remaining on the current disk.

DIRECTORY
: Provide a directory listing for the current disk, showing the name and size of each file. A filespec may be given to select only a certain file or wildcard file group.

DELETE
: Delete the specified files from the current disk.

TYPE
: A wildcard filespec is accepted and files displayed alphabetically. The display is paged in Unix fashion with "`--more--`" displayed on the last line. Typein options at that point can be obtained by hitting a '`?`'.

1.2. Installation:

CP/M-86 KERMIT is broken up into several source modules:

`C86CMD.A86`	Command parser
`C86FIL.A86`	File handler
`C86Xxx.A86`	System Dependent I/O
`C86KER.A86`	Main Program
`C86PRO.A86`	Protocol Module
`C86TRM.A86`	Terminal Emulation
`C86UTL.A86`	Utilities

The main program module, `C86KER.A86`, contains INCLUDE directives for the other files. The C86Xxx module is stored with "xx" replaced by codes denoting the machine for which the program is being built -- RB for Rainbow, AP for NEC APC, etc. The program may be built on the CP/M-86 system by obtaining all the source files listed above, storing them on the current disk with the names indicated, renaming the appropriate `C86Xxx.A86` file to be `C86XXX.A86`, and then doing:

```
ASM86 C86KER $PZ   (takes about 6 minutes on the Rainbow)
GENCMD C86KER      (takes less than a minute)
```

and, if desired,

```
REN KERMIT.CMD=C86KER.CMD
```

1.3. DEC Rainbow 100 Support

Kermit-86 runs on the DEC Rainbow 100 or 100+ under CP/M-86/80, version 1 or 2, on the 8088 side. It uses the built-in firmware to emulate a VT102 ANSI terminal during CONNECT, and runs well at speeds up to 9600 baud.

You should be able to download the program using the old KERMIT on the Z80 side (Rainbow Kermit, VT180 Kermit, or generic CP/M-80 Kermit will do the job, but only under DEC CP/M-86/80 version 1.0), or an earlier version of Kermit-86.

If you don't have an earlier version of KERMIT, then follow the directions for installing KERMIT-80 (yes, KERMIT-80) in the KERMIT-80 section of the *Kermit User Guide*, but send the Kermit-86 hex file instead. This works because the Rainbow can run CP/M-80 programs like DDT.

Another way to get Kermit onto your Rainbow for the first time would be from a DEC VT-180 diskette. A VT-180 can use its own Kermit to load Rainbow Kermit onto its disk, which can then be read directly by a Rainbow. Also, note that VT-180 Kermit-80 can actually run on the Rainbow on the Z80 side under DEC CP/M-86/80 version 1 (but not version 2 or higher), at speeds of 1800 baud or lower.

1.4. NEC Advanced Personal Computer Support

(Contributed by Ron Blanford, University of Washington)

Currently only the standard serial port is supported, and not the H14 auxiliary port. The SET PORT command is not implemented.

While in Kermit's terminal emulation mode, local commands are initiated by a two-character sequence consisting of the "escape character" followed by one other character identifying the command. (Make the second character a '?' to see a list of the valid commands.) As distributed, the standard Kermit-86 uses the control-backslash character as the escape character in terminal mode. The trouble is that the CP/M-86 BIOS in the APC ignores a keyboard entry of Control-\ (i.e. holding down the CTRL key while striking the '\' key), making it difficult (impossible) to use this method to get out of terminal mode.

One solution is to perform a "SET ESCAPE ^" command before entering terminal mode to change the escape character to a caret (or any other character the APC keyboard will generate). This command could be placed in your `KERMIT.INI` file for automatic execution every time Kermit is started.

The simpler solution is to realize that the character code for a Control-\ is a hexadecimal 1C, and that this is the code generated by the INS key on the numeric keypad. Once you can remember that every reference to Control-\ should be interpreted as a reference to the INS key, this is actually easier to use than the two-key Control-\ sequence.

In the standard CP/M-86 BIOS, the unshifted DEL key generates a Control-X character (hexadecimal 18). This is the CP/M command to erase the current input line, and is very useful for local processing. Most mainframes do not use the Control-X character at all, so it becomes much less useful during terminal emulation. The DEL character (hexadecimal 7F), on the other hand, is often used by mainframes and can only be generated on the APC by holding down the SHIFT key while striking the DEL key (this capability is not mentioned anywhere in the documentation).

Because the Control-X character is so seldom used while the DEL character is commonly used, the initialization procedure in Kermit-86 modifies the CP/M-86 BIOS so that the DEL key generates the DEL character whether shifted or not. Control-X can still be generated if necessary by holding down the CTRL key while striking the 'X' key. The CP/M-86 BIOS is returned to its original state when Kermit terminates.

The APC uses escape sequences which have been standardized by the American National Standards Institute (ANSI) to control cursor movement, screen erasing, and character attribute manipulation. Perhaps the best-known other terminal which follows ANSI guidelines is the DEC VT100. The APC only recognizes a few of the more important ANSI commands, and not the complete set which the VT100 supports.

The ANSI/VT100 features that the NEC APC supports are:

- direct cursor addressing (by row and column)
- relative cursor addressing (up, down, left, right)
- line erasing (cursor to end, beginning to cursor, entire line)
- screen erasing (cursor to end, beginning to cursor, entire screen)
- character attributes (underline, reverse video, blink, but not bold)

In addition, the first four grey function keys (unshifted) generate the escape sequences associated with PF1 through PF4 on the VT100 keyboard. The arrow keys and numeric keypad DO NOT generate the corresponding VT100 sequences.

These functions are enough to support simple command line editing on most systems, and allow mailers or paged file display programs to clear the screen before each display. Underlining and reverse video are also useful in some applications. This is not enough to support the more sophisticated screen control required by screen editors such as EMACS or KED. In addition, due to a bug in the implementation of the CP/M-86 BIOS, the sequence ordinarily used to home the cursor (esc [H) does not work correctly; a patch for CP/M to correct this problem is distributed with APC Kermit-86.

Index

Table of Contents