Kermit News

Number 6
March 1995

Brazil: Election Day 1994

Brazil's October 1994 general election was almost certainly the world's largest and most complex ever.

Kermit software played a crucial role.

In Rio de Janeiro, ballots are transcribed to PCs for transmission to the tabulating center by MS-DOS Kermit. *Article on page 19.*

Kermit News Number 6, March 1995

CONTENTS

Kermit News *(ISSN 0899-9309) is published periodically free of charge by Kermit Development and Distribution, Columbia University Academic Information Systems, 612 West 115th Street, New York, NY 10025, USA. Contributed articles are welcome.*

Editor: Christine M. Gianone
E-Mail: cmg@columbia.edu

Cover: *Brazilian Presidential Election, October 1994. Using computers to count votes, Rio de Janeiro. Photo: Jorge William, Agência O Globo.*

Editor's Notes

Welcome to *Kermit News* Number 6. This issue announces new releases of our three most popular communications software programs:

1. **MS-DOS Kermit** for DOS and Windows;
2. **C-Kermit** for UNIX, OS/2, OpenVMS, Stratus VOS, and several other operating systems;
3. **IBM Mainframe Kermit** for VM/CMS, MVS/TSO, CICS, and MUSIC.

These new versions offer:

- **File transfer failure recovery** for DOS, Windows, UNIX, VMS, OS/2, OS-9, AOS/VS, VOS, Amiga, and IBM mainframes (see the article on page 12).
- **Auto-download**/upload/configuration via APC mechanism (article on p.14).
- **General improvements** in performance, script programming, client/server protocol, terminal emulation, character sets, dialing, and much more.

New and full-featured C-Kermit programs are available for **QNX** and **Stratus VOS**. And there also is new support for recent OS releases in the ever-changing UNIX market (article on p.6).

Kermit software today is a powerful, consistent, and fully interopable communications solution for the industry's most popular computers.

In the December 1994 issue of *DEC Professional* magazine, Kevin Barkes writes in his column, ''*The newest MS-Kermit and C-Kermit programs have features and speeds rivaling the leading commercial software packages.*'' Of Kermit's role in the 1994 Brazilian elections (see our article on page 19), he says ''*If Kermit can work in a difficult situation like this, you can imagine what it can do in your shop.*''

A Word to Our Sponsors

The Kermit project is entirely self-supporting, funded by book sales, mail orders, and commercial licenses. Since our last issue, there has been an explosion in the popularity of the Internet. Our Internet ftp site, KERMIT.COLUMBIA.EDU, always a popular Internet resource, is now playing host to thousands of file transfer requests each day.

The increased accessibility of Kermit software via network is good . . . and bad too. While we're delighted with its increased visibility and popularity, mail orders are down and our support burden is up.

To keep pace, we must increase our mail-order sales and the use of our documentation. Our manuals teach users, even complete novices, how to use Kermit software effectively, and are also excellent reference works for seasoned professionals and everyone in between. Please keep in mind:

- The manuals reduce the load on our help desk – and yours!
- Sales of the manuals are the primary source of funding for the Kermit project.

Second, we look to those large organizations that save huge amounts of money on software licensing and support by distributing Kermit software internally, to purchase manuals from us in bulk (at quantity discounts) and to consider making contributions to the Kermit effort so we may continue to produce and support the software that saves them so much money.

Third, we must reemphasize our policy towards commercial distribution of Kermit software. Remember, most Kermit software is not in the public domain; the copyright is held by Columbia University, and forbids redistribution by commercial enterprises without our written permission, even when not done directly for profit, and even on ''free software'' or ''shareware'' CD-ROMs (or other media). If your company wishes to distribute Kermit software to its customers or clients, please contact us – the advantages are many, our terms are easy. You can read about several successful new cooperative ventures in this issue.

While Surfing the Internet . . .

Be sure to visit our new World Wide Web home page:

http://www.columbia.edu/kermit/

Here you will find our new illustrated catalog, MS-DOS Kermit graphic screen shots, and a whole new approach (for us) to software distribution. But please do remember to purchase the relevant manuals if you haven't already done so.

And for a unique international perspective on the Internet, be sure to read *Circumnavigating the Web* on page 15.

Books, Books, Books

Digital Press Changes Hands

Digital Press, publisher of our English-language Kermit books, has been transferred from Digital Equipment Corporation to:

DIGITAL PRESS
Butterworth-Heinemann
225 Wildwood Street
Woburn, MA 01801, USA
Voice: +1 800 366-2665, Fax: +1 617 933-6333

(a member of the Reed Elsevier group).

Kermit books can, as always, be obtained directly from Columbia University (see our order form), but orders to the publisher (e.g. from bookstores) must now go to Butterworth-Heinemann.

Overseas offices of Butterworth-Heinemann can be reached at the following telephone numbers:

+44 1933 414414	Rushden, England office for Europe
+61 2 372-5511	Chatswood, NSW office for Australia and New Zealand
+65 220-3684	Singapore office for Asia
+27 031 294247	Durban office for South Africa

The new arrangement has been in effect for more than a year so all the wrinkles should be ironed out. But if your bookstore is having difficulty obtaining English-language Kermit books, please refer them to the new publisher.

Und immer noch ein neues Buch!

Always again another book! *Using C-Kermit*, the C-Kermit 5A user manual published by Digital Press, has been translated into German by our good friend, Gisbert W. Selke of the WIdO (Wissenschaftliches Institut der Ortskrankenkassen) in Bonn (a small University town in Germany), and published in deluxe hardcover edition by Verlag Heinz Heise of Hannover. The proper citation is:

Frank da Cruz and Christine M. Gianone, *C-Kermit – Einführung und Referenz* (1994), ISBN 3-88229-023-4. Deutsch von Gisbert W. Selke. Price: DM 88,00. Verlag Heinz Heise GmbH & Co. KG, Helstorfer Straße 7, D-30625 Hannover, Germany.
Tel. +49 (05 11) 53 52-0, Fax. +49 (05 11) 53 53-1 29.

Readers may recall that Gisbert also translated, and Heise also published, a German edition of *Using MS-DOS Kermit* for German-speaking Kermit users.

& en Français?

The French translation of *Using MS-DOS Kermit*, by another good friend, Jean Dutertre of Digital France, remains available not only in France, but can also be ordered from us on our order form, as a special convenience to Francophones outside France:

Christine M. Gianone, *Kermit MS-DOS mode d'emploi* (1993), ISBN 2-901143-20-2. Adaption française: Jean Dutertre. Heinz Schiefer & Cie., 45 rue Henri de Regnier, F-78000 Versailles. Tel. +33 39 53 95 26,
Fax. +33 39 02 39 71.

Our neighbors to the north are heartily encouraged to indulge their bilingual inclinations by keeping both French and English editions close by at all times! (We also hope that French authorities will kindly overlook the comments on page 46 . . .)

Here's Proof!

James Huggins
Department of EE and CS
University of Michigan
Ann Arbor, MI, USA

In his preface to Frank da Cruz's book *Kermit: A File Transfer Protocol*, Don Knuth wrote:

> I hope that many readers of this book will be challenged to find high-level concepts and invariant relations by which various versions of the Kermit protocol can be proved correct in a mathematical sense.

I'm pleased to announce that such a proof has recently been completed. It shows that Kermit is both safe (if you receive a file using Kermit, it will be the same one that was sent) and live (if you send a file using Kermit, and the network doesn't behave hideously, it will eventually get to the recipient).

The proof appears in *Kermit: Specification and Verification*, to be published in the Oxford University Press book ''Specification and Validation Methods,'' edited by Egon Börger, due in April 1995. After publication, an electronic version of the paper will be placed on the Kermit archives at Columbia.

If you're interested in a preprint, or if you have questions or comments about the paper, feel free to contact me at huggins@umich.edu.

Announcing MS-DOS Kermit 3.14

Yes, another new Kermit for your PC!

The PC marketplace grows and changes constantly, and MS-DOS Kermit is changing with it. MS-DOS Kermit 3.14 for PCs with DOS or Windows was released in January 1995. It was prepared by Professor Joe R. Doupnik of the Center for Atmospheric and Space Sciences and Department of Electrical Engineering of Utah State University in Logan, Utah, USA, in cooperation with Columbia University in New York City and Dr. Hirofumi Fujii of the Japan National Laboratory for High Energy Physics.

The new release includes an incredible number of enhancements in every area designed to keep MS-DOS Kermit up-to-date with new PC, modem, and networking technology. The big ones are:

- **File transfer recovery** allows interrupted binary-mode transfers to be continued from the point of failure, for use with C-Kermit 5A(190) and IBM Mainframe Kermit 4.3.1 (see article on page 12).
- **ANSI and Wyse-50** terminal emulation add two popular terminal types to MS-DOS Kermit's repertoire. ANSI emulation is ideal for accessing BBSs.
- The greatest Kermit protocol implementation is now available for **BBSs** themselves (see article on page 4).
- The new Kermit diskette includes complete packages for **Hebrew** and **Cyrillic** terminal emulation, including fonts and keyboard drivers. Read those Russian newsgroups! Access online Hebrew databases!
- **Japanese Kanji** and **Chinese** terminal emulations are now fully supported, even on ordinary US-model IBM PCs and compatibles. See article on page 15.
- The **script programming** language now includes built-in functions just like C-Kermit's.
- New **smaller versions** are provided for those who don't need (or can't fit) all the features of the full version, and for use as an external protocol or script execution engine in BBSs and other applications.

The New MS-DOS Kermit Diskette

MS-DOS Kermit now comes on a high-density 1.44-MB 3.5-inch DOS-format directory-structured diskette that contains everything you need, including three different executables: the full-functioned version, a smaller ''Medium'' version, and a tiny ''Lite'' version. The medium version can be used on PCs with small memories, e.g. on old XTs, where the full-featured version might not fit. It can also be used if you simply do not need Kermit's networking or graphics terminal emulation capabilities; this lets you run bigger programs ''under'' Kermit in the extra free conventional memory.

The ''Lite'' version has no network support or terminal emulation at all, including no CONNECT command. It still includes serial communications, the full Kermit protocol implementation, and the complete script programming language. Weighing in at only 105K, it is perfect for use as an external protocol and script execution engine in BBSs, in embedded applications, and behind custom menus.

Diskette subdirectories include:

- Dialing scripts for many types of modems.
- Network shims, drivers, and documentation.
- Key mapping setups and keyboard drivers.
- File transfer and printer utilities.
- Windows files.
- PC fonts and utilities.
- A complete Cyrillic support package.
- A complete Hebrew support package.

All the utilities you need are on this disk, so there is no longer a separate ''utilities'' disk. Our deepest thanks, as always, to Joe Doupnik for bringing another new version of MS-DOS Kermit to us, and special thanks to Dr. Fujii for his work on the Kanji features, and to Yossi Gil at the Technion in Israel for the fonts, to Dimitri Vulis in Brooklyn for the Cyrillic keyboard drivers, and to many others (too numerous to list) for testing and other contributions.

MS-DOS Kermit and SAS/GRAPH

Users of SAS/GRAPH® can now view Sixel-based **color graphics** with MS-DOS Kermit. Starting with release 6.10 of SAS software, which is available now on several popular UNIX platforms, two new device drivers can be used to generate color graphics directly on your Kermit display.

To view a graph from an interactive SAS/GRAPH session, use this syntax at the beginning of your graphics program:

```
GOPTIONS DEVICE=KRMTxxx;
```

where *xxx* is VGA or AUTO, corresponding to Kermit's SET TERMINAL GRAPHICS setting.

For more information on these device drivers, contact SAS Institute Technical Support at +1 919 677-8008 and ask for the Graphics group.

To view sample full-color Kermit SAS graphics screens, point your graphics Web browser at URL:

http://www.columbia.edu/kermit/shots.html

MS-DOS Kermit Meets the BBS

Most Bulletin Board Systems (BBSs) today either lack support for Kermit protocol, or supply a poor implementation. Now BBSs can use the ''Real'' MS-DOS Kermit itself, as an external protocol. Why should a BBS add or upgrade Kermit file transfer?

1. To make your BBS accessible to Kermit software programs. The new releases of MS-DOS Kermit and OS/2 C-Kermit include ANSI terminal emulation, required for accessing BBSs.
2. Kermit file transfer is robust. It works well even when connections are noisy. And, properly implemented and configured, it is just as fast or faster than other protocols (see benchmarks in *Kermit News #5*).
3. Kermit protocol survives 7-bit connections; most other protocols do not. This is important when callers arrive over public data networks and other non-direct paths. In most cases, Kermit can even sense 7-bit connections automatically.
4. Kermit protocol can be used by Internet-accessible BBSs and other services. It works well over TELNET connections, even 7-bit ones, and Kermit TELNET clients are available for most popular operating systems: DOS, Windows, OS/2, UNIX, VMS, VOS, etc.
5. Only Kermit protocol is capable of converting from one text character set to another during text-file transfer. This is vital as the BBS world becomes more international, and BBS clients more diverse. Remember, not all computers use IBM Code Pages to to represent non-English text!
6. MS-DOS Kermit can ''autoconfigure'' callers for maximum performance, and can initiate ''autoupload'' and ''autodownload'' operations without user intervention, provided the client software is MS-DOS Kermit 3.13 or later, or C-Kermit 5A(190) or later for UNIX, OS/2, VMS, or OS-9 (see article on page 14).

New features ideal for use in BBS systems:

- Support for ''Fossil drivers'' used on BBSs.
- Support for high interface speeds (up to 115200 bps) and hardware flow control for use with high-speed data-compressing modems.
- Ability to release the connection immediately when the carrier signal drops.
- File transfer operations may be logged to a file.
- Uploads can be restricted to a particular directory.
- Downloads can be driven by a file list.

The new ''Kermit Lite'' program, KERLITE.EXE, developed in part with the financial support of XAP Company (see page 5), is especially suited to BBSs. Features not needed on BBSs (such as a terminal emulator and a TCP/IP network stack) are stripped away to minimize its disk footprint and its memory requirements, so it can easily coexist in conventional memory with your BBS program.

The KERMIT.UPD file that comes with MS-DOS Kermit 3.14 includes a new BBS Operators Guide to show you how it's done.

Bob Mahoney, President of Exec-PC, Inc., the World's largest BBS (''The Business Knowledge Exchange, serving the business community since 1983''), says ''we had not supported Kermit since 1989 when our old version of PC-Kermit became obsolete when it was unable to support the higher DTE speeds of the new modems. We took Kermit offline and have been looking for a replacement ever since. We finally came across KERLITE – just what we were looking for. It is now installed on Exec-PC and has proven to be very popular on our Telnet-in nodes of the BBS that are available via our Internet connection. *The other protocols we have online were troublesome for many of the Telnet users. Kermit has done a good job on fixing the download problems for many of them.*''

Mike Robertson in Sweden says, ''We have been using the KERLITE program to provide an external Kermit protocol under the MBBS system (a BBS system found mostly in Scandinavia). . . . We run two of the BBS lines as direct connect to a Portmaster box acting as a Telnet server to let people get to the BBS over the Internet (our address is bbs.gar.no for those who'd like to try it). Because the Telnet server only gives us a short carrier interruption before it's ready to accept a new login, it's very important that external protocols terminate quickly when carrier is dropped so that the BBS can sense the drop. KERLITE fulfils this requirement. Other options set here make sure that Kermit tidies up after failed uploads, and also forbids access to files other than those specified by the BBS in the command line, both of which are important requirements for a BBS. *Experience so far has shown KERLITE to be extremely reliable as an external protocol.*''

MS-DOS Kermit may be used by BBS *operators* without any special arrangements. Simply order it, install it according to the instructions in the BBS Operators section of the KERMIT.UPD file, and then announce it to your users.

Makers and vendors of BBS software packages may contact us if they would like to make arrangements to package MS-DOS Kermit with their products.

MS-DOS Kermit Means Business

In recent years, MS-DOS Kermit has seen increasing use within business software as a communications and scripting program and/or file transfer engine. This phenonenon is especially evident in electronic claims submission and EDI (Electronic Data Interchange) applications.

Much of this new software relies on MS-DOS Kermit to make the connections and transfer the data behind the scenes.

Health care providers and insurers are turning to PCs and modems to eliminate costly paperwork in order to reduce health-care costs and speed up the reimbursement process. Claim forms are filled out on the computer screen in the doctor's office, hospital, or pharmacy, and submitted at the touch of a button to a medical claims clearinghouse or directly to the insurance company.

Over the years every insurer and clearinghouse developed proprietary and incompatible formats and software. The recent growth in electronic claims submission has spurred a movement towards standardization, allowing (for example) a doctor's office to submit different types of claims in a uniform manner. The American National Standards Institute (ANSI) recently approved EDI formats for Insurance Claims Submission and Electronic Remittance, and this will only hasten the end of the ''mountains-of-paper'' system, with all its confusion and delays, and will enable not only a standardized method for electronic submission of claims, but also for their payment.

A similar movement is stirring in the consumer electronics market, where warranty claims can now be filed electronically. Likewise, companies that use computers to keep track of inventory can now refresh their stocks automatically when items run low, using software that sends orders to their suppliers by modem. Retail distributors of standardized parts and components can now have pricing updates loaded automatically into their business computer systems by their suppliers.

Much of this new software relies on MS-DOS Kermit to make the connections and transfer the data behind the scenes. Because of its unobtrusive user interface, MS-DOS Kermit can be invisible to the user, and because of its powerful script language, it can easily be programmed to do automatically anything that might be done interactively, such as dialing up, logging in, handling various error conditions, and (of course) transferring data swiftly and accurately.

Student Financial Aid by Modem

XAP Company of Los Angeles, California, is putting together a package to allow admissions and financial aid applications for higher education institutions throughout the United States to be created on PCs and then submitted either on diskette or by modem.

According to Allen Firstenberg, XAP President, ''through the development of information management technologies − data collection, hard-copy printing, multimedia interactive information display, database management, and electronic data exchange − XAP Company is making the task of filling out complex forms simple for students and their parents and also more efficient to process.

Kermit Lite, driven by a custom script program, dials to the XAP facility and, acting as the EDI transport, submits the completed applications automatically and reliably.

''Over 2 million students annually are expected to utilize XAP's software by 1996, submitting their completed applications electronically. As the availability of modems for students continues to increase, XAP Company is planning to receive the predominance of its applications via EDI.''

Embedded within the XAP software will be MS-DOS Kermit ''Lite,'' the special small-size version for PCs, chosen for its flexible scripting capabilities, ease of use, and high file-transfer efficiency. When modem transmission is elected, Kermit Lite, driven by a custom script program, dials to the XAP facility and, acting as the EDI transport, submits the completed applications automatically and reliably.

According to an article in *Bank Systems Technology*, October 1994, this system not only replaces the burdensome and ''notoriously lengthy and complicated'' paper-based financial aid disbursement process with a more streamlined and efficient one, but, through the newly formed Educational Loan Management (ELM) Resources, aims to create a universal standard for all students, schools, and banks that will vastly simplify the financial aid process.

C-Kermit 5A(190) for UNIX, VMS, OS/2, VOS, . . .

Frank da Cruz

Version 5A(190) of C-Kermit, the world's most portable communications software program, was released in October 1994 for:

- UNIX, all varieties
- IBM OS/2 and Warp
- Digital VMS and OpenVMS
- Stratus VOS (new! – see p.10)
- Data General AOS/VS
- Microware OS-9
- The Commodore Amiga

The short list of new features is:

- File-transfer recovery from point of failure (binary-mode transfers only): UNIX, VMS, OS/2, AOS/VS, VOS, OS-9, Amiga (see article on p.12).
- Auto-upload/download/configuration/anything-else via APC mechanism: UNIX, VMS, OS/2, OS-9 (see the article on p.14).
- Command history, recall, retry, and typeahead.
- Automatic directory creation for incoming files (UNIX, OS/2, VMS, OS-9, . . .)
- New, faster initialization file, containing many new macros, including a FAST macro to let you set up and try all the performance features conveniently.
- Massive improvements for OS/2 (see page 7).
- The fullscreen file transfer display now includes a ''percent done'' thermometer and continuous display of the transfer rate in characters per second.
- Support for many new OS releases: Solaris 2.3 and 2.4, AIX 4.1, Unixware 1.1, IRIX 5.3, new releases of Linux, {Free,Net,etc}BSD, OpenVMS 6.x, etc . . .
- Numerous improvements in performance, error recovery, script programming, client / server protocol, character sets, dialing, etc, and over twenty new commands and numerous new built-in variables and functions. The full list of changes since the previous release would include about 400 entries.

UNIX C-Kermit

As the role of UNIX-based servers becomes ever more important in all spheres of the economy, so does communication with the outside world, whether by dialup or network. Even when your UNIX system is on the Internet, C-Kermit offers features lacking from traditional Internet applications, particularly automation via script programming and conversion of incompatible character sets.

UNIX C-Kermit 5A(190) runs on 16-bit, 32-bit, and 64-bit architectures under 2.xBSD, 4.2–4.4BSD; AT&T UNIX System III and System V R2, 3, and 4; POSIX; OSF/1. Plus many and varied releases and incarnations of: 386BSD; AT&T 3Bx systems and UNIX PCs and workstations; Amdahl UTS; Apple A/UX; BSDI; COHERENT; Convex/OS; Cray UNICOS and CSOS; Data General DG/UX; DEC OSF/1 and ULTRIX; ESIX; Encore UMAX; FreeBSD; Harris CX/UX; Hewlett Packard HP-UX (all versions); IBM AIX on RS/6000, PC RT, 370 mainframe, and PS/2; ICL DRS/NX; Interactive UNIX; Intergraph CLIX; Linux; Lynx; MIPS RISC/OS; Motorola System V/68 R3 and System V/88; NCR UNIX; NeXTSTEP; NetBSD; Olivetti X/OS; Pyramid OSx; QNX; SCO ODT, UNIX, and XENIX; Sequent DYNIX and DYNIX/ptx; Silicon Graphics IRIX; Sun Solaris and SunOS; Sony NEWS-OS; Stratus FTX; Tandy XENIX; Trusted XENIX; UnixWare; and many more.

In addition to file transfer recovery and auto up- and download, version 5A(190) includes lots of UNIX-specific improvements too: a faster CONNECT mode, more reliable signal handling, an option for a system-wide initialization file, TELNET screen-size negotiation. Hardware flow control (RTS/CTS) support, since it is so important for using today's high-speed modems, has been added to many of the UNIX versions, and so has support for higher serial speeds.

C-Kermit for HP-UX 10.0

We are pleased to announce that Hewlett Packard Company has chosen C-Kermit 5A(190) as a standard component of the HP-UX 10.0 operating system, supporting both serial and TCP/IP connections. In partnership with HP, Columbia University developed, tested, and documented the new HP-UX 10.0 version, which is fully aware of the new HP-UX file system, device names and locking conventions, serial speeds and flow-control options, and so on. After you install your HP-UX 10.0 package or upgrade, just type ''kermit'' to start the program, or ''man kermit'' for further information. And if you don't already have the manual, *Using C-Kermit*, please be sure to fill out and send in the order form that came in your HP-UX 10.0 box.

C-Kermit for QNX 4.21

As part of the US Postal Service Carrier Sequence Bar Code Sorter project (see article on page 23), Columbia University, with assistance from QNX Software Systems, Ltd., Kanata, Ontario, prepared an entirely new, full-featured implementation of C-Kermit for the QNX operating system, which plays a vital role in that project. Supporting both TCP/IP and high-speed serial connections, QNX C-Kermit is an important addition to the QNX software toolbox, and is available in binary form, ready to run, on the QNX C-Kermit diskette on our order form. In case you have not heard of QNX before, here's a product profile:

As the role of UNIX-based servers becomes ever more important in all spheres of the economy, so does communication with the outside world, whether by dialup or network.

''With hundreds of thousands of installations worldwide, QNX is the leading realtime operating system for the PC. You'll find this POSIX-certified OS at work in everything from process-control applications to financial systems, point-of-sale systems, communications, and medical instrumentation.

''QNX consists of a team of optional cooperating processes that interact with each other through a tiny microkernel, just 10K. As a result, QNX can be scaled down for embedded systems, scaled up for large development workstations running X and TCP/IP, or scaled out for vast fault-tolerant networks. QNX's networking facility integrates the entire network into a single, logical computer. So, for example, an embedded system can inherit the disk, database, and other resources of the entire network, and the rest of the network can communicate with processes on the embedded system.''

As Dan Hildebrand, Senior R&D Staff Member at QNX Software Systems Ltd., says, ''The QNX network transparently supports several standard protocols while simultaneously carrying its own high-speed FTL protocol. Kermit extends QNX's connectivity by providing a broad-spectrum communications solution that lets users make connections to, and transfer files with, virtually any other kind of computer, over either serial connections or TCP/IP.''

For more info on QNX, phone QNX Software Systems at +1 800 676-0566 or +1 613 591-0931. Internet: info@qnx.com.

VMS C-Kermit

Like UNIX C-Kermit, the (Open)VMS version of C-Kermit 5A(190) for Digital Equipment Corporation VAX and Alpha AXP computers incorporates major new features like file transfer recovery and auto up- and download during CONNECT mode, described in separate articles, as well as (user-controllable) automatic directory creation for incoming files that include directory names.

However, of even greater interest to some VMS shops might be the fact that the new release is able to run in batch jobs and when SPAWNed from other programs such as ALL-IN-ONE or VMS MAIL. This was our number-one wish-list item from VMS users for the new release.

Other important new features include support for the CMU/Tektronix TCP/IP package and new, automatic compensation for small system buffers during file transfer: now long packets can generally be used even when big buffers have not been installed in VMS at SYSGEN time.

VMS C-Kermit binaries are available for all combinations of VAX and Alpha AXP hardware with the following TCP/IP products: Digital (UCX); TGV MultiNet; Wollongong WIN/TCP or PathWay; Process Software TCPware; and (VAX only) CMU/Tek. And, of course, no TCP/IP at all. A VAX binary is also available for VMS version 4.4. The TK50 BACKUP-format distribution now includes .EXE files for all of these. The 9-track ANSI C-Kermit tapes include them all in ''hex'' format.

Thanks to Mark Berryman, Terry Kennedy, William Bader, James Sturdevant, Mike O'Malley, Hunter Goatley, and Tarjei Jensen for assistance with the new release of VMS C-Kermit.

OS/2 C-Kermit, Take One

With the announcement of Warp in October 1994, OS/2 suddenly became a formidable combatant in the desktop operating system wars – if it wasn't already. C-Kermit 5A(190) for OS/2 was released simultaneously with Warp, and bears about as much resemblence to earlier OS/2 C-Kermit releases as Warp does to OS/2 1.0. Among the many big improvements, you will find:

- Certification by IBM for OS/2 and for LAN Server.
- VT220 and ANSI terminal emulation added, along with numerous other terminal-emulation improvements.
- File transfer recovery (see article on p.12)

- A new file-transfer mode that transfers OS/2 files along with all their attributes and extended attributes, either directly to another OS/2 system, or for archival on a non-OS/2 system, plus the ability to transfer and replicate entire directory trees between two OS/2 systems.
- A REXX interface, so C-Kermit scripts can contain REXX commands, and REXX programs executed from within C-Kermit may contain C-Kermit commands.
- NETBIOS and Named Pipe task-to-task communication support, for peer-to-peer local area networking.
- Support for several TCP/IP packages in addition to IBM's: FTP Software, Essex Systems, and IP-Switch.
- Improved Digital PATHWORKS networking support.

OS/2 C-Kermit's comprehensive array of communications methods is rivaled only by MS-DOS Kermit's, and it illustrates one of the greatest advantages of Kermit software: it is a single, common solution for many communications needs. You don't have to learn one application for dialing up BBSs, another one for dialing up your corporate mainframe, yet another one for the Internet, and still another one for peer-to-peer LAN connections. C-Kermit does it all, in a uniform, consistent way. You only have to learn one application, not a big pile of them. You only need one compact application on your disk, occupying about a megabyte, not three or four using up tens of megabytes. This saves you not only lots of disk space, but money too.

Kermit's user interface and script programming language are portable and consistent over dialup and network connections for hundreds of different computers and operating systems.

When you learn C-Kermit's script programming language, you can use it to make all kinds of connections, not just serial ones. You can also use the same script programs (with minor alterations, e.g. for device or directory names) on other operating systems where C-Kermit runs: UNIX, VMS, and so on. You can even make the same scripts portable to DOS and Windows, since MS-DOS Kermit's script programming language is very similar to C-Kermit's.

This is a significant leveraging of our most precious asset: people-time – time spent in training, studying, figuring things out. Make the investment once, rather than over and over again. Even afterwards, once you've become an expert, you can benefit from not having to retrain your fingers every time you access a different host or service – you can set up consistent key maps for all your online connections.

Terminal Emulation

Let's take a quick look at some of the improvements OS/2 C-Kermit's terminal emulator. First, there is **VT220 emulation**, which allows us to take advantage of many host-based applications, or features of them, primarily on the VMS and UNIX operating systems, that were not accessible to us before. The new emulation includes a complete repertoire of ''keyboard verbs'' that can be assigned to the keys of your choice, and which are compatible with those of MS-DOS Kermit. These represent not only the common program-control actions (reset emulator, send BREAK, hang up, return to prompt, etc), but also all of the keys of the VT220 terminal, including arrow, function, keypad, and editing keys, which faithfully follow the host-directed ''modes'' for these keys.

Then there is **ANSI terminal emulation** for full-color access to BBSs with all their special effects. Hopefully (see page 4) more BBSs will be offering a good Kermit protocol from now on, but even when they don't, OS/2 C-Kermit is set up to let you run external protocols easily.

In all types of terminal emulation, there are new options for **screen rollback**, as well as increased capacity – up to about two million lines!

There is also **Hebrew terminal emulation** for use with host-based Hebrew applications such as ALEPH software (see article on page 15).

The new terminal emulator also supports **auto-download** and **auto-upload** – automatic switching to file transfer mode, as well as automatic configuration by the host (see article on p.14).

You can now use the **mouse** during terminal emulation: copy and paste to and from other applications; use the mouse to move the cursor by simulating arrow-key strokes. The latter is useful with full-screen host applications that support arrow keys – one touch of the mouse button sends the arrow-key codes necessary to move the cursor from its present location to wherever the mouse pointer is.

When using a Western European language in your online session, you now have a special **Compose Key** (Alt-c) for composing accented and special characters mnemonically; for example, Alt-c, ^, and u sends û (u-circumflex), just as in MS-DOS Kermit, and as on a real VT220 terminal.

All sorts of improvements have been made to the on-screen **helpers and prompters** – the status line indicates exactly what is going on; context-sensitive **pop-up help screens** are available in all different modes: online, rolled-back, compose-key sequences, and so on. There are no sacred keys in C-Kermit's terminal emulator; all functions can be remapped to other keys, and the help screens and status line keep track automatically. And of course you have total control over the **colors** used in all types of screen elements: the status line, the pop-up screens, the terminal screen, and so on.

Printer functions are expanded and improved – host-directed and user-initiated printing of online screens and sessions is now available, as well as redirection of printer material to a file or device.

Finally, a powerful **session-debugging** capability has been added, similar to a professional-quality, expensive line monitor. It shows control characters, eight-bit characters, escape sequences, and even TELNET option negotiations symbolically, but readably, in different colors and renditions for easy problem diagnosis and, perhaps more importantly, to help you with your INPUT and OUTPUT commands when you are constructing script programs, so you can see exactly what the host is sending.

File Transfer Improvements

File transfer recovery is discussed on page 12. Automatic directory creation is available for incoming files. Automatic parity detection during file transfer was also added in this version, and along with it the ability to transfer files with IBM mainframes thru non-transparent 3270 protocol converters (the ''Doomsday Kermit'' protocol discussed in *Kermit News #5*).

In addition, some exciting new OS/2-specific features were added. OS/2 C-Kermit has always supported the long file names of the High Performance File System (HPFS), and has always been able to distinguish between FAT (DOS-like) and HPFS volumes for the purposes of file access and creation. In version 5A(190), however, OS/2 C-Kermit is able to preserve an incoming file's original long name, even when creating the file on a FAT volume, by storing the long name in the file's Extended Attributes. A file, thus created, can later be copied to an HPFS system and its long name will magically reappear.

Of even greater significance is a new method for transferring OS/2 files along with all of their regular and extended attributes, including desktop information, icons, and so on. We call this ''labeled'' file transfer; previously it was available only in VMS C-Kermit. Labeled transfers can be done directly between two OS/2 systems, in which case each file will arrive at its destination with all of its attributes intact. Labeled transfers can also take place between an OS/2 system and some other kind of system, in which case the file will be ''archived'' together with its attributes, for later restoral to (perhaps another) OS/2 system, again with all its attributes intact.

Finally, OS/2 C-Kermit comes with a procedure for sending entire directory trees and their contents, preserving the directory structure. When used in conjunction with labeled mode, this lets entire directory trees, and even entire file systems, be cloned from one OS/2 system to another. In non-labeled mode, it also facilitates the movement of directory trees between PCs running similar, but different, systems such as OS/2 and DOS or Windows.

And beyond finally: OS/2 C-Kermit 5A(190) comes with a complete repertoire of macros for easy access to external protocols, for the hopefully rare occasions when you must transfer files with a host or service that does not support Kermit protocol.

Documentation

All the new features of version 5A(190) are documented in the accompanying CKERMIT.INF file, an online supplement to (but not a substitute for) *Using C-Kermit*. You can browse the CKERMIT.INF file with the OS/2 VIEW program; this is an indexed, hypertext document that you can click your way through, search for particular material, and so on, just like the regular OS/2 help facility, and very similar to using a World Wide Web browser.

Acknowledgements

Special thanks to Jeff Altman for massive contributions to the OS/2-specific portions of OS/2 C-Kermit 5A(190), to Kai Uwe Rommel for his sage advice, and to the OS/2 Developers and Testers group for cheerfully fielding an endless barrage of Alphas and Betas.

OS/2 C-Kermit, Take Two Already!

But we didn't stop there. C-Kermit 5A(191) – an OS/2-only C-Kermit release (5A(190) is still current for UNIX, VMS, et al) – adds the following improvements and new features:

- The terminal emulator is about 500 percent faster than before, scrolling text (even in an OS/2 Window) at about 200 full-width lines per second. On TELNET connections, it is about four times faster than the system TELNET program. With speeds like this, screen updates and scrolling will never be a bottleneck on serial connections, not even the fastest ones.
- 132-column mode is available in Full Screen sessions on video adapters that support it. Under Warp, arbitrary screen dimensions are supported in an OS/2 Window. Terminal emulation screen width and height can be set by new C-Kermit commands, and

TELNET screen size negotiations (NAWS) are now supported, as well as host-directed 80/132 column-mode switching.

- Hebrew and Cyrillic EGA/VGA/SVGA fonts are supplied and can be loaded in a Full Screen session. This means that you can have Hebrew and Cyrillic terminal sessions without having to install special National Language Support versions of OS/2.
- C-Kermit can now be used to make SLIP connections, coordinating with the SLIP driver over ownership of the port, and replacing the system's SLIPTERM program, which has limited emulation and scripting capabilities.
- System Sounds are now used when installed. C-Kermit uses the ''Information,'' ''Warning,'' and ''Error'' sounds. If System Sounds are not enabled, the regular beep is used.
- Incoming TCP/IP connections are now supported for Kermit protocol operations and for ''chat mode.''

Version 5A(191) is 32-bit only and works only on OS/2 2.00 and above. The 16-bit version of OS/2 C-Kermit (for OS/2 1.x) is frozen at 5A(190). Thanks to Jeff Altman for *all* of the work that went into OS/2 C-Kermit 5A(191).

C-Kermit for Stratus VOS

David Lane, Stratus Computer Inc

Stratus Computer, Inc. (NYSE SRA) is the second-largest provider of fault-tolerant hardware systems. VOS is the proprietary operating system for Stratus hardware. Stratus systems are designed to reach (and often attain!) uptimes greater than 99.99%. They are used for applications where downtime is extremely costly, such as stock exchanges, banking, and airline reservations; and where downtime can be deadly, such as public safety systems.

Because of the markets into which Stratus sells, VOS is primarily optimized for transaction-processing, rather than for general terminal users[1]. This means that, unlike UNIX, VOS does not come with a dial-out program, though some can be found. There are several programs available, but they either require specific software at the remote end to perform file transfers, do not perform file transfers at all, or are quite costly.

I had previously developed a small communications program for our support group to use to dial out to our customers for remote maintainance. I had been getting many requests to add file transfer capability to this program. However, I wanted to have a better program than what I was prepared to write from scratch, so I started to look around at what was available and soon discovered a ''portable, full-featured implementation of Kermit written in C.''

> *Great care has been taken by all the people involved with C-Kermit to allow it to work with older compilers.*

It only took several evenings to get the basic sources compiled under the VOS C compiler, even though it was, at the time, not an ANSI compiler. Great care has been taken by all the people involved with C-Kermit to allow it to work with older compilers. Even though the VOS compiler at the time had many ANSI features, if C-Kermit had required a full ANSI-compliant compiler, I would have had a great deal of trouble getting it to work on VOS. After the portable pieces of the C-Kermit code compiled, it took several months of evenings and weekends to fill in the system dependent modules of C-Kermit which perform serial and file I/O.

As a side benefit, C-Kermit supports X.25 and TCP/IP connections, similar to the VOS CALL_THRU and TELNET commands. There was file transfer support for X.25 on VOS, but only with modifications to a system server (x25_exchange.pm) and with unsupported software from Stratus. Of course, TCP/IP file transfers can use FTP, but using FTP through a corporate firewall or proxy server can be difficult to automate.

> *Using the ACCESS macros in C-Kermit is much easier than using different commands for each type of connection.*

Having the X.25 and TELNET support within C-Kermit has also helped to debug the option setting on remote connections, because we can have a single method that allows us to see the options for both X.25 and TCP/IP, rather than having different methods for each; and C-Kermit doesn't require privilege to debug it's own connections! Using the ACCESS macros in C-Kermit is much easier than using different commands for each type of remote host connection, and it greatly simplifies the use of proxy servers.

So now we have a full-featured C-Kermit implementation for Stratus VOS that fills a need in this area of critical computing.

[1] The mention of Stratus in Tom Clancey's *Debt of Honor* notwithstanding!

Other Recent Releases

IBM Mainframe Kermit 4.3.1

Kermit-370 version 4.3.1 for IBM mainframes with CICS, CMS, MUSIC, and TSO (and TSO's friend, ROSCOE) is now available. Its major new feature is the ability to recover interrupted Kermit transfers, when used in conjunction with MS-DOS Kermit 3.14 or C-Kermit 5A(190), described in detail in the article on page 12. Any interrupted binary-mode file transfer (even a non-Kermit one) can be restarted with this facility, and the resulting file will be identical to what would have been received in a single transfer.

Its other major new feature is the ability to initiate automatic file transfers with MS-DOS Kermit 3.14 or C-Kermit 5A(190) by sending Application Program Command (APC) escape sequences (see article on page 14). Also, the CMS and CICS variants can now set the date/time stamp for a received file to match that of the original, and the MUSIC variant has newly-added support for long names and now allows arbitrary MUSIC commands to be executed from within Kermit.

Kermit-370 4.3.1 has been verified to transfer both text and binary files flawlessly with the new 3270 terminal emulator supplied with Cisco terminal server software release 10.3, even with performance features such as long packets and 8-bit transparency enabled. This is welcome news for the many sites where Ciscos provide dialup access to the mainframe.

Thanks, as always, to John Chandler of the Harvard / Smithsonian Astronomical Observatory for the new release! January 1995, *Tape B.*

EMACS (Yes, EMACS)

An implementation of Kermit was written (on a dare) in EMACS Lisp by Bob Manson of MIT and Ben Mesander of the US Geographical Survey in June 1994. It works with GNU EMACS versions 18 and 19. It's a bare-bones implementation that should be portable to any machine where EMACS runs (UNIX, VMS, etc). It lets you transfer files in text or binary mode into and out of an EMACS buffer; for example, when using MS-DOS Kermit as a terminal emulator into a UNIX or VMS system where you are editing with EMACS. *Tape B.*

Digital PDP-11 RT-11 and TSX+

Version V03.62-8 of Kermit-11 for the Digital PDP-11 with the RT-11 or TSX-Plus operating system, and for Pro-350/380 systems with Pro/RT or TSX-Plus was contributed by Billy Youdelman on behalf of DECUS, the Digital Equipment Computer Users Society.

This program runs under RT-11 from V4 and TSX from V5. A special minimum version for floppy-disk-based systems is included, especially handy on systems having no line time clock, or for getting files from small systems often found in older image-processing equipment. File creation date, time (TSX only), protection, and length attributes are now supported and work with C-Kermit and MS-DOS Kermit. And: smaller program size, bigger packets, improved communications and modem control, faster CONNECT sessions, and command-line arguments are now supported. September 1993. *Tape B.*

Apple Macintosh

No significant progress has been on Macintosh Kermit since *Kermit News #5* (volunteers?). *Except!* The bug that was causing Mac Kermit to crash during downloads under System 7.1 or later has been fixed. Mac Kermit 0.991(190) has the fix.

PLUS . . .

Alpha Micro
From Bob Rubendunst, V2.0 of Alpha Micro Kermit, replacing version 1.0 from Feb 1985, for AMOS/L 1.3 and above and AMOS/32 systems. New features include 8-bit terminal support, auto-send/receive, batch sends with random file bypass, parity checking, CRC error-checking, statistics, AM3000 compatibility. March 94. *Tape C.*

Burroughs (UNISYS) B6800
Written in Algol by Tony Appelget, Plymouth, MN. September 94. *Tape D.*

HP-3000 MPE
Two versions from Tony Appelget, one in SPL and one in C. September 94. *Tape D.*

Nicolet 80
A brand-new Kermit program for the British Nicolet 80 series of laboratory computers from Peter McClintock, University of Lancaster, UK. July 94. *Tape C.*

File Transfer Recovery

Frank da Cruz

Seasoned modem users know well the aggravation of long file transfers interrupted by broken phone connections: The screams of agony, the torn-out clumps of hair, the rent garments . . . only to relive those awful moments when the monthly phone bill arrives.

The problem is not confined to modem connections, either. Internet users know only too well the heartbreak of the broken ftp connection, especially that excruciatingly slow trans-oceanic one. Connections break — all kinds of connections — and the laws of the eponymous Murphy dictate this will happen at the worst possible time; for example, when you are 9.8 megabytes into a 10-megabyte transfer on a sloooooow connection.

Our three new Kermit releases to the rescue: MS-DOS Kermit 3.14, IBM Mainframe Kermit 4.3.1, and C-Kermit 5A(190). From now on, no more worries about broken connections, at least not during binary-mode file transfers, on all of the following platforms:

- DOS (MS, PC, DR, etc)
- Microsoft Windows
- IBM OS/2
- UNIX (all varieties)
- Digital VMS and OpenVMS
- IBM Mainframe VM/CMS, MVS/TSO, CICS, MUSIC
- Stratus VOS
- Data General AOS/VS
- Commodore Amiga
- Microware OS-9

When a file is transferred in binary mode, its size does not change; it is sent literally, without any kind of format or character-set conversion. Thus, if a file is partially transferred, we know exactly where in the original file to resume the transmission.

For the new round of releases, the Kermit programs were changed to keep partially received files by default, rather than discard them (which was the default action previously), and the Kermit protocol was extended to support recovery of binary-mode transfers. When a recovery operation is requested, the two Kermit programs negotiate this capability; if successful, the file receiver tells the sender the position in the source file from which to start sending, and then the receiver appends the new material to the end of the partially received file. Only one new command is needed:

RESEND *filename*

The same file can be recovered in this way more than once; for example, if the phone connection is broken several times.

Another interesting property of the RESEND feature is that it can also be used to recover interrupted non-Kermit transfers, such as with Ymodem-G or FTP. As long as you have a partial file that was transferred in binary mode, by whatever means, you can continue the transfer from the point of failure using Kermit's new RESEND feature.

When you combine Kermit's automation features with its new recovery ability, you can create script programs (like the one opposite) that are virtually guaranteed to transfer a file, even under the worst conditions, by automatically redialing and RESENDing each time there is a failure, until the file is completely transferred. Once started, such a script can run totally unattended; read the newspaper, go out to dinner, take a nap — relax, don't worry — barring total failure of the telephone network or destruction of one of the computers, the file will get through.

What about text-mode transfers? These can't be recovered automatically because there is no reliable correspondence between the original file and the transferred file: many operating mark lines of text differently: CR (carriage return) and LF (linefeed) at the end, as in DOS; LF-only as in UNIX; CR-only as on the Mac; via length fields or other mechanisms on record-oriented file systems, and so on.

But all is not lost. Here is a useful hint: if you are transferring text files between computers that have like file systems (e.g. DOS to OS/2, or HP-UX to Solaris), and you don't need character-set conversion, then use binary mode. This is somewhat more efficient than text mode because all conversions are skipped, and you can recover interrupted transfers.

Even when a text-mode transfer is interrupted, it's now possible to recovery ''manually'' by telling MS-DOS Kermit or C-Kermit to ''PSEND'' (partially send) the file from the point of interruption (which you must determine by inspection), and to tell the receiving Kermit program to append the incoming partial file to the existing file, via SET FILE COLLISION APPEND if it is supported. If not, you can receive the partial file into a separate file and then join the two afterwards. In any case, the part that was successfully transferred need not be transferred again.

File Transfer Recovery Demonstration Script

```
ask \%u { username: }
askq \%p { \%u's password: }

; Settings for entire session.
;
define \%s 20             ; Seconds to pause between each try
define \%n 7654321        ; Phone number
set port com1             ; Communication port
set modem pp14400         ; Modem type (dial with PP14400.SCR)

set file type binary      ; File transfer mode must be binary
set input timeout quit    ; This is just to keep the script program short...
set count 50              ; Try up to 50 times to send the file
goto nomsg                ; Skip message the first time

:LOOP                     ; Come here to redial
hangup                    ; Give the phone line a rest
echo CONNECTION BROKEN.
echo Pausing for \%s seconds...
sleep \%s
Echo redialing...

:NOMSG
dial \%n                  ; Dial the phone number
if fail goto AGAIN        ; Keep trying...
output \13                ; System answered, send a carriage return
input 15 login:           ; Get UNIX login prompt
output \%u\13             ; Send user ID
input 8 Password:         ; Get UNIX password prompt
output \%p\13             ; Send password
input 60 {$ }             ; Get UNIX system prompt
cd \budget                ; CD to desired local source directory
output cd budget\13       ; and remote destination directory
input 8 {$ }              ; Get system prompt
out kermit -r\13          ; kermit -r(eceive) on remote system
input 10 KERMIT READY     ; Wait for READY message
pause 1                   ; Plus a second for safety
resend fy9495.wks         ; RESEND the file
if success goto done      ; Success means file is completely transferred

:AGAIN
if count goto LOOP        ; Otherwise, try again.
Stop 1 Too many tries.    ; Too many tries, give up.

:DONE
echo File transferred OK  ; Success, give message
output exit\13            ; Log out from remote computer
pause 5                   ; Give it time...
hangup                    ; Hang up
stop 0 Script succeeded   ; Finished, the end.
```

(This MS-DOS Kermit 3.14 script dials into a UNIX host and assumes UNIX prompts and other conventions. It can easily be modified to run under C-Kermit and/or to access non-UNIX host computers. A copy of this script can be found on the MS-DOS Kermit diskette as RECOVER.SCR in the UTILS subdirectory.)

Auto-Upload, Auto-Download, Auto-Anything

Kermit users often ask, ''Why is it so hard to transfer files? Why are there so many steps? Why do I have to escape back, give a SEND or RECEIVE command, and then give another CONNECT command? If I give a file transfer command to one Kermit program, why can't it just take care of everything itself?''

Stop asking so many questions! You wanted it to be easier, now it can be. MS-DOS Kermit 3.13 (announced in our last issue) added a new capability called ''APC'', which stands for Application Program Command (see box).

When an APC-capable Kermit program receives an APC while in CONNECT mode, it executes the commands contained in the APC and then returns automatically to the CONNECT screen. This allows the host application to initiate file transfers in either direction (auto-upload and -download).

It also lets the host application operate and configure Kermit in all sorts of other ways, such as setting protocol and file parameters, or loading custom keymaps. This makes it easy for administrators of central corporate computers, dialup information services, and BBSs to create canned procedures for their user communities, especially novices.

Security

Kermit's APC feature is disabled by default (that is, unless you tell it otherwise). This is because we want each user to read about the potential risks and the corresponding safeguards, and *then* turn it on. The command is:

SET TERMINAL APC { OFF, ON, UNCHECKED }

Turning it ON enables operations that are nominally safe, such as file transfer, protocol settings, and so on, without opening up operations that are intrinsically dangerous, such as system access or deleting files. Please read the APC section of the update notes carefully, and then put:

SET TERMINAL APC ON

in your Kermit startup file.

It's So Easy

C-Kermit comes with an APC command for sending APCs and some predefined macros that you can use at the C-Kermit prompt:

PCSEND *filespec*	Send a file or files to the PC
PCGET *filespec*	Get a file or files from the PC

What Is an APC?

An APC is an escape sequence defined for VT320 terminals, which allows the host to pass a command (in the form of a text string) to a terminal emulator; the string is embedded inside the escape sequence, like so:

<ESC>_*string*<ESC>

where ''<ESC>'' is the ASCII control character, Escape (27).

When the terminal emulator is a Kermit program, the string can be any Kermit command, even a list of commands separated by commas, for example:

<ESC>_set file typ bin, s oofa.zip<ESC>

PCSEND is a single command that takes the place of:

1. Give a SEND filename command to C-Kermit.
2. Escape back to the MS-DOS Kermit prompt.
3. Tell MS-DOS Kermit to RECEIVE.
4. When the transfer is complete, tell MS-DOS Kermit to CONNECT again.

Now, steps 2-4 happen automatically. The same is true in the reverse direction with PCGET.

APC All Around

The new release of C-Kermit (OS/2, UNIX, VMS, and OS-9 versions) can not only send APCs but also respond to them while in CONNECT mode, and the new MS-DOS Kermit can not only respond to them but also send them. The new Kermit-370 can send them too. So now you can choose ''one from column A and one from column B'' and start automating!

A: Local	**B:** Remote
MS-DOS Kermit	MS-DOS Kermit
OS/2 C-Kermit	UNIX C-Kermit
UNIX C-Kermit	VMS C-Kermit
VMS C-Kermit	OS-9 C-Kermit
OS-9 C-Kermit	IBM Mainframe Kermit

The ability of the remote application to initiate file transfers automatically in either direction is a big step forward in the ease-of-use department, especially when the host application is menu-driven. Since it is so easy to hide Kermit software behind menus, it's only a matter of time until ''one-touch'' file transfer becomes the norm in Kermit land.

Circumnavigating the Web with MS-DOS Kermit

Frank da Cruz

Thanks to the Internet, the world is becoming smaller every day. MS-DOS Kermit might not be a WWW navigator or newsreader, but it has a distinct advantage: it can handle a wide variety of character sets and writing systems simply and easily.

It's practically compulsory nowadays to ''surf the Information Superhighway'' using high-powered graphical navigation tools, but once you leave the confines of your own country, or — more often — attempt to access information that is not written in English, you are very likely to run into trouble. A quick tour of the World Wide Web (WWW) using NCSA Mosaic on a Hewlett Packard workstation turned up only garbage in place of accented or non-Roman characters when trying to access Web pages in just about every country we visited: Costa Rica, Belgium, the Czech Republic, France, Germany, Hungary, Iceland, Israel, Italy, Norway, Poland, Russia, . . .

Much time was wasted trying to rectify the problem, involving adding hundreds of cryptic lines to the .Xresources file, repeatedly shutting down and reloading the X Windows system, logging out and back in, all to no avail. German still came out looking like Icelandic; Russian and Hebrew were hopeless. (A solution was found eventually, but it only works for the Western European ''Latin-1'' languages; HyperText Markup Language, or HTML, the language of the Web, does not allow for any other writing systems.)

Similar problems occur when using graphical newsreaders like NewsGrazer and the ones built in to Mosaic and Netscape; they don't give you easy (or *any*) control over character sets.

MS-DOS Kermit might not be a WWW navigator or newsreader, but it has a distinct advantage: it can handle a wide variety of character sets and writing systems simply and easily. MS-DOS Kermit 3.14 comes with all the pieces you need — fonts, keyboard drivers, key mappings, instructions, and macros for quick switching — for five important writing systems: West European, East European, Cyrillic, Hebrew, and Japanese. And, by a stroke of good luck, Chinese too (Japanese and Chinese require special versions of DOS; more about that later).

The Cyrillic Alphabet

Languages like Russian, Bielorussian, Ukranian, and Bulgarian are written in the Cyrillic alphabet, named for Saint Cyril (827–868, Feast Day February 14), Apostle (with his brother Saint Methodius) to the Slavs. Saints Cyril and Methodius were Greek missionaries to Moravia, a powerful empire in the ninth century and now part of the Czech Republic, where they and their followers adapted the Greek alphabet to the Slavic languages, dropping some Greek letters and adding some new, unique ones.

Approximately 1100 years later, in the former Soviet Union and elsewhere, several encodings were devised for computer representation of Cyrillic letters, each one incompatible with the others: KOI-8, KOI-7 (Short KOI), DKOI, ISO 8859-5 Latin/Cyrillic, Alternative Cyrillic, PC Code Page 866, Mainframe Code Page 880, and so on.

Although MS-DOS Kermit has been capable of Cyrillic text-file transfer since version 3.11, version 3.14 is the first to also incorporate a complete Cyrillic terminal emulation package, including a font that everybody can use, keyboard drivers, and complete key and screen mappings for accessing host-based applications using KOI-8, Short KOI, or Latin/Cyrillic.

As a quick demonstration, if you have the *rn* or *trn* newsreader on your UNIX host, or similar programs on other hosts, you can use MS-DOS Kermit 3.14 (under DOS, not Windows) to read the Russian relcom.* newsgroups, such as relcom.ads, relcom.commerce, or relcom.currency. Just type:

```
MS-Kermit> cyrillic
```

beforehand to load the Cyrillic font (code page 866) into your PC and set up the translations between CP866 and the KOI8 encoding used in the newsgroups. Now you can get the latest news on Russian business, commodities, investment opportunities, and currency.

If you also want to write in Russian during your terminal session (or, for that matter, in DOS on your PC), you must load a Russian keyboard driver, two of which are provided on your Kermit diskette. The term ''Russian'' is used advisedly here, since these drivers do not support the special characters unique to Serbocroation, Macedonian, Ukranian, and Bielorussian. One driver conforms to the USSR-model PC keyboard layout, the other assigns Cyrillic characters to Roman letter keys ''by sound'' for easy use by QWERTY typists; pick the one that is most natural for you. In both cases, a hot key switches the keyboard between English and Roman modes.

For e-mail, which is predominantly 7-bit, you should use Short KOI, which is understood by most Russian computer users:

```
MS-Kermit> cyrillic shortkoi
```

Some applications might also require the new 8-bit ISO standard:

```
MS-Kermit> cyrillic latinc
```

As always, you can also transfer Cyrillic text files between your PC and any computer that is running C-Kermit 5A or IBM Mainframe Kermit 4.2 or later, with full character-set translation, as explained in the appropriate Kermit manuals.

Now back to the World Wide Web . . . Switching to a text-based Web browser, *Lynx* from the University of Kansas, on a UNIX host, and using MS-DOS Kermit as your terminal emulator, you can surf the Russian Web *in Russian* — for example, starting at URL http://www.ac.msk.su.

Roman Alphabet

For West European languages using the Roman alphabet but with accented letters and sometimes a few special letters (like German ß), a wide variety of character sets — both 7-bit and 8-bit — is in use. Most of the 8-bit sets are proprietary (HP Roman8, Data General, etc); the standard is ISO 8859-1 Latin Alphabet 1. The 7-bit sets come from the older ISO 646 standards, one for each language (or country), in which certain ASCII characters, normally:

```
[ \ ] { | } ~ '
```

are sacrificed for the special characters needed in a particular language; thus each ISO 646 version is incompatible with all the others. ISO 646 sets are commonly used in e-mail, a predominantly 7-bit medium.

For example, when you receive e-mail from ''`H}kan Sj|berg`'' in Sweden, it is probably from Håkan Sjöberg (fictitious name), encoded in ISO 646. Tell Kermit to SET TERMINAL CHARACTER-SET SWEDISH and it will look right. On the other hand, if the same person's return address looks like ''`Hekan Sjvberg`'', it was probably encoded originally in 8-bit Latin-1 and had its 8th bit chopped off on its way to you, an irritating trait of many e-mail systems.

Newsgroups, on the other hand, use an 8-bit transport and one can often find Latin-1 encoding in them. In one newsgroup recently, swnet.svenska, there was a debate (in Swedish) over the relative merits of Swedish ISO 646 and ISO Latin-1, with the encoding of each message reflecting the preference of its author. This was handled quite nicely by setting up ''hot keys'' in MS-DOS Kermit to switch between the two character sets without leaving CONNECT mode:

```
def swedish set term char swedish, c
def latin1 set term char latin1, c
set key \315 {\Kswedish}
set key \316 {\Klatin1}
```

Here we assign one macro to F1 and the other to F2. When a message comes up that is not readable, we push the ''other'' hot key and then ask our newsreader to redisplay the message.

Hunting through the newsgroups that we receive locally, we find Latin-1 in use on Norwegian groups such as no.film and no.sport.diverse and on German groups like de.etc.finanz.boerse (daily listings from the Frankfurt and Berlin financial markets) and de.rec.motorrad (motorcycles). But in other European newsgroups (French, Dutch, etc) we found only ASCII — no true accented letters in any encoding at all. We suspect the Finnish newsgroups, such as:

```
finet.freenet.oppimiskeskus.-
akvaariokoulut.opetussuunnitelma
```

(line continued due to lack of horizontal space) use Latin-1 in line with the other Scandinavian countries, but alas, the Finnish groups are not delivered here (possibly due to some kind of buffer overflow . . .).

Some newsgroups we would have liked to look at were the Brazilian groups in Portuguese, and the pl.* groups from Poland, in Polish, but these feeds do not arrive here either. If these newsgroups use Latin-1 and Latin-2, respectively, Kermit would handle them just fine: MS-DOS Kermit 3.14 comes with ROMAN and EASTERN macros which load Western and Eastern European fonts, respectively, and support character-set conversion for all major Western and Eastern European languages.

Not only that, for Western European languages, we also provide a Compose Key for entering accented letters ''mnemonically'', with complete independence from the character sets used on the PC and on the host. For example, no matter what code page is loaded on your PC or what character-set is used on the host, you can always enter u-circumflex the same way: Alt-c (for Compose), then circumflex (^), then u.

Hebrew

In Israel, of course, as well as up the street at the Jewish Theological Seminary of America (JTSA) – and elsewhere – people need to access Hebrew applications on the host: text editors such as HEDT on VMS and vi.iv or Mule or Hebrew Pico on UNIX, Hebrew Pine and other e-mail, and especially Hebrew University's ALEPH library catalog software, which runs at all major universities in Israel as well as at JTSA, and which accepts Hebrew queries and also can display the results in Hebrew.

Here again we have the classic problem – different computers use different encodings for Hebrew letters: 7-bit ''Hebrew-7'', ISO Latin/Hebrew; IBM PC Code Page 862; IBM Mainframe CECP 424; each incompatible with the others. Confounding the situation further is the intrinsically bidirectional nature of the Hebrew writing system: Hebrew letters right to left, digits and Roman letters left to right.

> ''*The only communications software I know of that handles all this smoothly is Kermit.*''

Since version 3.13, MS-DOS Kermit has included full Hebrew character-set translation as well as Hebrew VT terminal emulation at the VT420 level (which includes host-directed screen-writing direction). Version 3.14, however, is the first release to come with a complete Hebrew package: a Hebrew font that anybody can load (under DOS, not Windows), a key map for entering Hebrew letters on the keyboard, hot keys for switching the keyboard between English and Hebrew modes – all you need for accessing Hebrew applications online. The HEBREW macro sets it all up for you. Just type ''hebrew'' at the MS-Kermit> prompt and off you go.

David de Leeuw, Head of Computing Services at Ben Gurion University of the Negev, Faculty of Health Sciences says, ''Our range of computers and applications is very wide. To complicate things even more, many of our applications run in a variety of Hebrew setups – three different character sets and various concepts of screen orientation (mixed left-to-right and right-to-left). The only communications software I know of that handles all this smoothly is Kermit. The new 3.14 release even takes care of automatic translation between different character-sets when transferring text files from one system to the other. Our users working on PC's accessing UNIX, VAX, and IBM Mainframe can't tell the difference between different 'code-pages' and now they don't have to!''

Japanese

MS-DOS Kermit has been capable of converting Japanese character sets during file transfer since version 3.12. Version 3.14 adds Japanese terminal emulation for ordinary IBM PCs and compatibles running the DOS/V operating system – no special hardware is required. Now you can use MS-DOS Kermit for Japanese e-mail, for the Japanese fj.* newsgroups (such as fj.kermit), and for accessing the Nikkei Telecom Database, a comprehensive online service offering full text of most articles appearing in all major Japanese newspapers, plus Japanese equivalents of Readers Guide to Periodical Literature and Books in Print, patent registrations, and all sorts of financial and corporate data. Many Nikkei Telecom users prefer MS-DOS Kermit over the Nikkei-supplied access software because Kermit offers additional essential capabilities such as session logging, screen capture, and scripting.

The Japanese fonts are combined into Code Page 982, also known as Shift-JIS, supplied with DOS/V along with the Japanese keyboard input driver. CP982 includes ''half-width'' Roman, a 7-bit character set identical to ASCII except in two positions; ''half-width'' (Hankaku) Katakana, a 7-bit phonetic character set, and then a double-byte ''full-width'' Kanji set consisting of thousands of symbols. Shifting among these three character sets, which are all essential in Japanese writing, is a challenge for the terminal emulator during both keyboard input and screen display, but MS-DOS Kermit 3.14 does it all.

It is now commonplace in Japan to make use of all Kermit's Kanji features: compose long documents ''offline'' in the native PC environment, transfer them with Kermit to a UNIX host, translating them in the process from Shift-JIS to JIS X 0208 or other encoding for printing or e-mail; read and send e-mail and netnews online in Japanese, and finally download new material from UNIX to the PC, translating back to Shift-JIS.

You can also use text-based Web browsers such as Lynx-2.3jp (the Japanese version from Chiba University) to access Japanese information on the World Wide Web; no matter whether the Japanese Web server uses JIS7, EUC, or Shift-JIS, MS-DOS Kermit can display the Kanji text correctly.

If you are on the World Wide Web and have a Web viewer that can display GIF (graphics) files, you can view an illustration of MS-DOS Kermit's Kanji capacity in:

`http://www.columbia.edu/kermit/kanji.http/`

Chinese

MS-DOS Kermit can be used on regular IBM PCs and compatibles equipped with USA keyboard and VGA video adapter, running special Chinese extensions to DOS, two of which are available via anonymous ftp from host cnd.org (China News Digest):

- ZWDOS – GB 2312-80 encoding (Peoples Republic of China), shareware (free to students) by Ya-Gui Wei, Eden Prairie, MN.
- KCDOS – Big5 encoding (Republic of China), Kuo Chiau Business Computer Company Ltd, Taiwan.

or from ftp.ifcss.org (Independent Federation of Chinese Students and Scholars in the US). Other versions of Chinese DOS are available elsewhere, which fit these models, such as CC-DOS (described in *Kermit News* Number 5), ETen, and others.

Those who can read the German, Japanese, and Russian financial notices will have a definite edge. Those who can respond have a sharper edge still.

These two encodings, GB and Big5, are used on most Chinese host computers and services. So you would run the appropriate DOS extension or version, and then simply tell Kermit to:

```
set parity none
set terminal bytesize 8
set term character-set transparent
```

This allows Chinese characters to be received, viewed, typed, and transmitted during terminal emulation. No translation is necessary because the PC and the host are using the same codes. Input is according to the input method supplied in the particular DOS version or extension: BoPoMoFo, Chang-Jie, etc. In this way, MS-DOS Kermit can be used as a terminal to (say) a UNIX host, where Lynx is used to view Chinese newsgroups (such as alt.chinese.text and tw.*) or to access Chinese Web servers.

Adding Other Languages

Cyrillic terminal emulation is done externally to Kermit via keyboard and screen translation tables (such as in the KOI8.INI file) and by loading a font, which is supplied on the Kermit diskette. Other languages that have 8-bit single-byte character sets can be done in exactly the same way: Greek, Armenian, Georgian, etc, using readily available fonts, and constructing the necessary mappings in the same way that the Cyrillic ones were done – a series of SET TRANSLATION INPUT and SET KEY commands collected into a Kermit command file.

Webbing with C-Kermit

MS-DOS Kermit is not the only Kermit program that can handle character sets. C-Kermit (all versions), IBM mainframe Kermit (all versions) can also convert among diverse encodings for Roman-alphabet and Cyrillic-alphabet languages, Hebrew, Japanese, and other languages as an integral part of text-file transfer. *This is a unique capability of Kermit protocol and software.*

Most C-Kermit implementations can also handle character-set conversion during CONNECT mode, just as MS-DOS Kermit does (IBM Mainframe Kermit does not have a CONNECT mode). The difference is that most C-Kermit programs do not contain actual terminal emulators, but instead provide a semi-transparent ''pipe'' to a terminal, terminal window, or terminal emulator. Thus, loading of fonts and so on must be accomplished outside of C-Kermit.

An exception is version 5A(191) of OS/2 C-Kermit. It has a terminal emulator and it comes with Cyrillic, Hebrew, and other fonts that can be loaded when C-Kermit is running in a fullscreen session (the fonts can't be loaded in an OS/2 window because then Kermit does not have access to the video adapter).

This lets you use OS/2 C-Kermit to access Hebrew applications (such as ALEPH) just as you would with MS-DOS Kermit, and to read Russian and East European newsgoups, and (most of) the rest.

Now, since most versions of C-Kermit (OS/2, UNIX, VMS, VOS), like MS-DOS Kermit itself, are full-fledged TELNET clients as well as serial communication programs, poof! – you've got international Web access from home *and* office.

Thanks to the Internet, the world is becoming smaller every day, and it becomes increasingly necessary for us in the USA to dust off our high-school Spanish, German, Italian, French, or Russian (etc) if we want to partake fully in the information revolution – just as it is becoming increasingly necessary for business communications to take place in many languages.

Those who can read the German, Japanese, and Russian financial notices will have a definite edge. Those who can *respond* have a sharper edge still.

Kermit in the Brazilian Elections

Fernando Cabral, CEO
PADRÃO iX Sistemas Abertos, Brasília, Brazil

Columbia University's Kermit software and protocol are robust enough to work dependably even when using the poorest telephone lines — and in Brazil THERE ARE poor-quality telephone lines!

KERMIT SOFTWARE PLAYED A CRUCIAL ROLE in Brazil's general election of October 3, 1994, almost certainly the world's largest and most complex election ever. At stake in this country of 180 million were the presidency, all of the 28 state governorships, two-thirds (or 56) of the Federal Senate seats, and almost 600 Federal and 1000 State Representatives.

To cope with this task, the *Tribunal Superior Eleitoral* (Superior Electoral Court), or TSE, a specialized court of law dedicated to supervising all elections in the country, decided to take on the challenge of automating the process as much as possible, and to do it with a single stroke.

Introducing automation into a nationwide election in a huge country like Brazil, the same size as the continental USA, was fraught with hazards and obstacles. First, long-established regional oligarchies of conservative landowners would resist automation as a threat to their previous control over elections; second, the state data processing bureaus, which usually operate in the black only during election years, would be open to automation only if the bureaus could provide —and profit from —the automation instead of the TSE; and finally, the TSE staff's own lack of experience and know-how could threaten the success of the project.

Numbers

While China, the USA, Russia, and India have electorates comparable to Brazil's, none of them ever had to cope with an election involving such large numbers, either because their elections are conducted differently or because their legislative and executive elections on both the state and federal level do not coincide as they did in Brazil in 1994.

VOTING IS MANDATORY in Brazil for everyone aged 18 to 65. 96 million votors, starting at age 16, elect their government officials directly, not through an electoral college as in the USA. The widely-anticipated election involved 27 states, the Federal District, 300,000 ballot boxes, eight presidential candidates, 231 Federal Senate candidates, 3164 candidates for seats in the Federal House of Representatives, 7977 to a seat in one of the 600 seats in 27 states plus the federal district; and 134 candidates for 28 governorships. Altogether we are talking of **501,456,916 votes** in the first round alone. And all of them, checked and double-checked, **were transferred with Kermit software**.

The chairman judge in charge of the TSE, Minister Sepúlveda Pertence, and the court's director-general, Alysson Mitraud, did not take these numbers lightly. Despite the risks of failure and the uncertainty of gaining widespread support for their decision, the two officials decided to proceed with the automation. The single most important factor to the venture's success was close and effective partnerships with software and hardware vendors.

Among the software providers were Kermit developer Frank da Cruz of Columbia University, and his collaborator, Joe Doupnik of Utah State University, who both worked with the TSE to make everything run as smoothly as possible.

Old-Style Elections

The Brazilian electorate has evolved since the country's first election in the mid-19th century. At that time, only the richest could vote. The richest *men*, that is — women could not vote. Eventually the standard for elegible voters was ''universalized.'' This meant that every man could vote, as long as he could read and write amd was older that 21.

Not until the early 1930s did a modified constitution give women the right to vote. Unfortunately, a dictatorship quickly took control of Brazil and no elections were held until after World War II. So in fact, women voted for the first time in 1945. However, only in 1988 did the right to vote become truly universal. Gender, property, literacy, and other excluding criteria were eliminated and the minimum voting age was lowered to sixteen.

Brazilian elections prior to 1994 were susceptible to many different kinds of manipulation and fraud.

Most have become parts of Brazilian folklore and have revealing names like the ''tip-of-the-pen vote,'' where the result desired by the local landowner was simply recorded on the document listing the tabulation of each ballot box. There was also the ''lunchbox vote'' where the local plantation *coronel*[2] would fill out the ballots before delivering them to the awaiting voters in a closed, or ''lunch,'' box. Not even the voters knew who they were voting for. The term ''corral vote'' means that the landowner kept his workers in his own corral, like cattle, and told them who to vote for. Like cattle, they obeyed. Not to be forgotten is the ''phantom vote,'' when the dead arose to cast their ballots. Of course, these ghosts existed in name only — on their voter ID cards, their polling site signatures, and on their tombstones.

The 1989 Election

In October of 1960, a military coup and subsequent military dictatorships postponed Brazil's democracy and elections for 30 years. In 1989, Brazil held its first presidential election after three decades of opression. This was the first election after the adoption of a new constitution in 1988, the first to have a second runoff election for close races, the first to have television coverage, the first to have candidates use computers to handle huge amounts of information, and the first to broadcast live debates. And it was also the first time the electoral courts would try their hands at automation.

Cautiously, the TSE opted not to dive directly into automation. Instead, they contracted state-owned data processing bureaus to do the data entry of each state's votes. Then in Brasília, SERPRO, the federal data processing bureau, was contracted and regally paid to tabulate this data. It was a timid but important first step into the realm of automation, and there was no turning back.

New times, new ways to commit fraud. The computer introduced new potential and real ways to manipulate election results, such as a variation on the the ''tip-of-the pen'' scheme: simply alter the numbers during the transcription of the official ballot box results from paper to computer. The easiest way to do this without attracting too much attention is to turn blank or invalidated ballots into ''valid'' ballots.

[2]In Brazil, originally a title of honor which could be awarded by − or bought from − the federal government. Eventually the title took a derogatory meaning when used to identify landowners, industry barons, and other rich and powerful people who used their money and influence to force common people and lesser politicians to do what they wanted. When refering to elections, the term always means the rich, influential, and conservative persons who use their power and money to allure or coerce poor voters.

The 1994 Election

For the 1994 election, the TSE was ready to fully accept any challenge posed by total automation — it wanted to take computer automation as far as possible. This included automating the voter and candidate registry and verification, data transfer among regional election courts supervising the elections and tabulating stations, public access to voting regulations, and dissemination of the results. The only phase not automated was the tabulation of individual ballot boxes — not surprisingly, the only phase to suffer fraud in the 1994 election, primarily in Rio.

MS-DOS Kermit software was installed at 2,000 data entry and transmission sites in all parts of Brazil, some of them so remote that they could only be reached by boat or small plane.

The elements of election automation include hardware, operating system, networking software, database software, transmission lines, security software, terminal emulation and file transfer software. For each of these elements, the Electoral Court found a working partner. Hewlett Packard Company (and its Brazilian distributor Mito), for example, supplied servers, operating system, and networking software. Trusted Information Systems (TIS) and PADRÃO iX supplied security software and consulting services. And Columbia University furnished Kermit for terminal emulation and file transfers.

The Electoral Network

The electoral computer network was composed of 33 HP RISC servers whose size varies from state to state according to population. Each machine runs HP-UX and includes both TCP/IP and X.25 networking. TCP/IP would suffice save that the only public network available in Brazil, RENPAC, is X.25-based. In fact, it is only the bare bones of a network, providing no services, not even transport. So having TCP/IP and being able to make it run on top of X.25 was a distinct advantage. The available X.25 infrastructure permitted TSE to build a virtual network connecting all the regional courts within just a few weeks, embodying functionality that TCP/IP users were familiar with.

The RISC servers installed at each regional electoral court ran HP-UX, Oracle database software (supplied by Oracle's Brazilian distributor, UNIMIX), Gauntlet security software from TIS, and Columbia University's **C-Kermit** communications software. Each machine was responsible for tallying all state ballots, including those for state and federal representatives and senators, and for transferring the results of the presidential race from each tabulating station to the Superior Electoral Court in Brasília, and at the same time, offered any interested party, particularly the press, all information concerning the election, especially the numbers coming out of the ballots boxes.

Meanwhile 3,800 Digital Equipment Corporation DECpc personal computers with modems, special data entry software, and Columbia University's **MS-DOS Kermit** software were installed at 2,000 data entry and transmission sites in all parts of Brazil, some of them so remote that they could only be reached by boat or small plane.

Thus Kermit software linked together the two worlds: the world outside the network and the world inside it. In more than one sense Kermit was the bridge connecting the external, unprotected world to the internal, Gauntlet-protected world.

Election Day

On Election Day, the one and only day all Brazilians are equal – they each have one vote – the polls are open from 8am to 5pm. Because of the numerous races involved, the voting was conducted in two parts; the state and federal races each had a separate ballot. First the voter shows personal and voter identification and receives a white ballot. Then behind a screen, the voter chooses one presidential candidate and two federal senators, and then drops the folded ballot into the ballot box in view of the recipient committee, which includes common citizens as well as representatives of the political parties. Then the voter receives a second ballot for state races, this time yellow, and marks it behind a paper screen suspended over a counter, folds it, and deposits it in the ballot box in front of the committee. When the polls close, the ballot boxes are sealed and sent to the tabulating stations, along with an official report stating the number of people voting at that site.

The next morning, dozens of tabulating teams, under the close scrutiny of the political parties' representatives, break open the ballot boxes one by one and check the reported numbers of voters against the ballot count for the box. If there are discrepancies, or if there is any indication of tampering, the ballot box is declared invalid. If everything checks out, the tabulation proceeds.

The white and yellow ballots are separated into two piles. First the votes for the presidency and the federal senate are counted; then the votes for governor and federal and state representatives. This is a time-consuming process since each name or number has to be checked against a long list of valid numbers, names, nicknames, etc. After all the ballots are counted, an official statement is issued and signed by the committee, the parties' representatives, and the judge in charge of the regional electoral court.

Too good to be true. Without this instant response, all the election automation could have been compromised.

Then this official statement is transcribed to the PC. *This is the point where most of the fraud occurred*; blank and invalidated ballots were ''transferred'' to a chosen candidate. Cross-checking can't prevent this type of fraud; only an attentive monitor can spot it. After the transcription, a computer report is printed and checked against the original statement. If the numbers are equal, the file can be transferred.

Enter Kermit

Once the file transfer is authorized, the file is encrypted and compressed. Then Kermit assumes control, making decisions about how to connect to the remote server at the TRE (Regional Electoral Court): dial-up, TCP/IP, or an X.25 connection with or without a PAD (Packet Assembler/Disassembler).

Once the connection is established, the TIS software, Gauntlet, sends a challenge to the calling machine. Using her ''Digital Pathways' SecureNet Keys'' (token generator), the user types in her PIN and then the challenge. The generator produces a number that is sent as an answer to the server. If all is OK, the Gauntlet firewall opens and the file is transferred.

Once at the regional machine, the federal (white) votes are dispatched for tabulation at the TSE, while the state (yellow) votes are tabulated locally. Small numbers flow in, big numbers flow out. The results of each ballot box are added to the total as they arrive. An exact copy of each individual box's result is kept so if any fraud eventually turns up in any ballot box, its votes can be deducted easily from the total.

Newspapers, TV and radio stations, poll takers, and other interested parties could access partial results using a number of methods. Here again, TIS's Gauntlet ensured that only cleared information flows out and no tampering is possible. And Kermit was there too, ensuring that the information that flowed in piece by piece can now flow out in aggregate.

Kermit's update feature allowed any user with read privileges to dial in and download the latest numbers without tying up valuable telephone lines unnecessarily if no updates had occurred since last time.

Kermit's flexible scripting language eliminated the end-user contact with the file transfer mechanism: after automatically dialing, Kermit would check whether the remote file was newer than the local one, and transfer it only if it was. In any case the local application would proceed. This way no file was ever transferred twice, and no user had to control anything: Kermit took care of all this automatically.

Using Kermit's powerful scripting language, the results of each ballot box, as well as the aggregated results, were easily transferred from end to end —all complexities were hidden under Kermit's well-thought-out user interface.

Why Kermit Was Chosen

Kermit was chosen to connect the PCs at the tabulating stations to the regional courts because:

1. Columbia University's Kermit software and protocol are robust enough to work dependably even when using the poorest telephone lines —and in Brazil *THERE ARE* poor-quality telephone lines!
2. Kermit software was available for both MS-DOS and HP-UX.
3. Kermit's powerful scripting language could be used to automate most of the logon/transfer/logoff process. This was an important concern since 11,000 people would be using PCs, modems, and communication software for the first time in their lives. It was not realistic to expect them to understand and learn how to transfer files.
4. Kermit can also use TCP/IP, allowing its use in different communication environments with the same interface (and TSE would not be forced to teach FTP to some people and Kermit to others).
5. According to different local conditions, the line used could be dial-up, leased, or X.25 PAD. When an X.25 PAD comes into play, *NO PROTOCOL BUT KERMIT* does the job.
6. The Kermit team could be counted on to help out if the need arose. And it did. TSE needed screens with messages in Portuguese so any Brazilian operator could understand them. Joe Doupnik and Frank da Cruz inserted a Portuguese translation and delivered it within a day. Then, when the new Digital Equipment Corporation PCs arrived, they behaved strangely when the COMx ports were manipulated; Digital rushed a sample PC to Joe, who quickly updated MS-DOS Kermit for these new machines. The updated Kermit software was transferred to Brazil using Kermit itself via long-distance phone call. Too good to be true. Without this instant response, all the election automation could have been compromised.

People may wonder why didn't the TSE try other protocols like ZMODEM, YMODEM and akin beasts. Simple to answer in a nutshell (the long answer has been provided above): Kermit can be used with 7- or 8-bit lines, with leased, dial-up or PAD lines; the scripting language can be used to automate even the most complex operation; smooth operation in MS-DOS, MS-Windows, and HP-UX environments; and superb, unbeatable performance in all kinds of connections and line conditions. Finally, if anything bad happened, prompt and expert help was just a phone call or an e-mail away.

The Results

The election was marred by widespread fraud in Rio de Janeiro. But the automation helped detect it, allowed its extent to be assessed, and prompted measures to avoid it in round two. The time saved by the network was more than 75% in most states, the big exception being Rio, where bandits blocked entry of votes into the system (where they could not be altered) until after the ballots were forged.

But despite minor disturbances and a few major troubles, the election was considered a huge success. President-Elect Fernando Henrique Cardoso is recognized as a prudent person, an intellectual who has written dozens of books and taught sociology in the USA, England, France, and Chile. As the Economy Minister he reduced inflation from 48% per month to about 3% in less than five months. Since his election as President, inflation has dropped to under one percent per month, and the *Real* has gained value against the US dollar, which not even the wildest dreamer could have predicted a year ago. 85% of Brazilians are optimistic about the future and the economy is growing by leaps and bounds.

The Future

Today Brazilians seem to be ready and eager to have the next election in 1996 completely automated. The TSE conducted extensive studies not only of computer technology, but also of the Brazilian public's reactions to these new technologies to identify the right tools to provide a fully automated election within two years. In this upcoming election, when almost 5,000 mayors and 50,000 city representatives will be elected, 100 million Brazilians will touch a screen, not mark a piece of paper. There will be no transcription, therefore there will be no fraud. *Unless we come to know some new kind of ''cyberfraud''* . . .

About the Author

Fernando Cabral studied Philosophy, Psychology, and Mathematics but ended up involved with C, UNIX, and networking. Five years ago he founded PADRÃO iX, a consulting firm dedicated to connectivity and interoperability. He wears many hats, often playing the *agent provocateur* among mainframers, COBOLers, and MS-Windowers.

Kermit Helps Automate Mail Delivery

Did you ever wonder how U.S. Mail is delivered? Much of the process has been automated in recent years, and soon, thanks in part to Kermit software, one of the most tedious tasks will be on its way out.

Postal Bar Codes

Let's take a simplified look at how mail is delivered today. When a piece of mail enters the system at a local post office, it is sent to the nearest Processing and Distribution Center (P&DC). At each of the 270 P&DCs, mail passes through an optical character recognition (OCR) device. If the OCR machine can read the address, it applies a POSTNET bar code (if there already isn't one), representing the ZIP Code. If the address is illegible, the mail piece is rejected.

Bar codes will soon speed the delivery of mail directly to your door. Kermit software is a key component.

In many of the P&DCs (and eventually all of them), the rejected pieces are routed to a Remote Bar Code System (RBCS), where the electronic image of the front of the mail piece appears on a video display terminal, an operator keys in the ZIP Code, and then the bar code is applied. Thus all mail pieces leave these facilities with bar codes.

The Problem

From the receiving P&DC the mail is routed, in many cases by a Delivery Bar Code Sorter (DBCS) machine, either for local delivery or else to the appropriate destination P&DC, from which it is sent to the appropriate local post office for delivery by mail carriers.

It is at the destination post office that the automation process returns to manual handling of the mail. Mail carriers still have to sort mail into bins and cubbyholes, a labor-intensive and time-consuming process, before they can embark on their routes. The ever-increasing volume of mail adds to the workload of the mail carriers, and the manual sorting step can no longer keep pace.

The Solution

Now bar codes will be used to eliminate the manual sorting step too, speeding the delivery of mail directly to your door. A major contract between the United States Postal Service and Loral Federal Systems (formerly IBM Federal Systems Company) of Owego, New York, calls for the installation of 3,144 Carrier Sequence Bar Code Sorter (CSBCS) machines in approximately 1,100 of the larger local post offices.

Using technology licensed from AEG ElectroCom of Konstanz, Germany, the CSBCS machines sequence bar-coded mail for each mail carrier at 36,000 pieces per hour. The mail is sorted by mail route, and within each route according to the order in which the mail carrier visits each building. Routes can be customized on a daily basis; for example, to allow for buildings that are closed on Saturdays. By sorting mail in walk sequence without manual handling, the new systems will cut costs and enable faster mail delivery, even as volume goes up.

Both the DBCS and the CSBCS machines are partially controlled by PCs running DOS or QNX. QNX is a POSIX-compliant realtime version of UNIX from QNX Software Systems, Ltd., Kanata, Ontario (see profile on page 7). The PCs provide the user interface and the link to the outside world.

Kermit's Role

Columbia University's Kermit software − MS-DOS Kermit on DOS PCs, QNX C-Kermit on the QNX PCs − is a key component; it handles (according to the specification) ''transfer of End of Run reports, End of Period reports, density analysis data, sort plans, and software configuration updates.''

Each day, Kermit software dials up and sends machine usage, performance, diagnostic, and other reports to a regional hub of the National Directory Support System (NDSS) for postanalysis −fault detection, trend analysis, and so on. The regional hubs are equipped with Digital Equipment Corporation VAX and Alpha AXP computers running the (Open)VMS operating system and Kermit software.

Once a week, Kermit software is also used to load updated address information, the data for the ''sort plans,'' into each CSBCS [3] from the NDSS, which maintains a database of every address in the region. Thus each bar-code sorting system depends on Kermit software in order to do its job: to sort the mail.

[3] The DBCS receives the sort plan data via TCP/IP, but is also equipped with Kermit software as a fallback in case the network should fail.

Kermit and Market Research in the UK

The Role of Kermit Software in our Computer-Assisted Personal Interviewing System

Pat Molloy, Operations Director, NOP Research Group Limited
Tower House, Southampton Street, London WC2E 7HN, England

NOP is the third largest market research company in the UK. Roughly 55% of our interviewing is done face to face, conducted by some 1600 field staff located all over the country. In February 1994, we equipped 600 of our field staff with small personal computers, with which to conduct this data gathering exercise.

Naturally we required a cost-effective and reliable method for transferring data to and from these machines. It must be borne in mind that the typical market research interviewer is no PC expert and so a further prime requirement was that the system had to be pretty well idiot- and bomb-proof.

We looked at a number of commercial offerings which, as well as being expensive, also did not give us the required functionality. We decided early on that the prospect of managing a rack of fifty or so modems filled us with dread and therefore looked towards a service provider giving us a number of dial-up Points of Presence (PoPs) all over the country. Not only did this mean that the modems were avoided, but it also offered us the potential to significantly reduce our telecommunications costs.

We settled on a system called GNS (Global Network Systems) by British Telecom. It is an X.25 Packet Switching service, available all over the world, but importantly for us, with a hundred PoPs in the UK, from which 93% of our field force is just a local (i.e. cheap) phone call away.

Kermit was selected as the communications software of choice — it is extremely robust, cheap, and provides us with the all the scripting functions required. Kermit 3.12 is currently in use at the PC end, and Kermit 5A(190) on a Sun SPARCstation host at head office, running SunOS 4.1. We tried a number of other protocols, X-, Y-, and Zmodem for example, but because field agents are actually dialing into an X.25 PAD, we found that these did not work to our complete satisfaction.

Each PC in the field is associated with an interviewer who has a five-digit interviewer ID. We maintain a database on the SPARCstation that contains information concerning the interviewer, their location, various passwords, and the phone number of the local PoP. When a machine is sent into the field for the first time and booted, the boot process asks the interviewer for their ID number, and then Kermit places a call to a known PoP. Having established a link to the PoP, Kermit then initiates a transfer of the configuration file from the host to the PC. Thereafter the PC is ‘‘configured,’’ that is, it contains a file which has pointers to the local PoP.

> *The move away from traditional data collection techniques has eliminated seven million sheets of paper a year already — we expect this to rise to a savings of twenty million within two years.*

As far security is concerned, once the call has been made to the PAD the process is entirely automated and the user cannot ‘‘escape’’ to gain access either to the PAD or the SPARCstation. The autocall facility on the PAD is set to call only a single host, our SPARCstation in London. Once the call arrives at the host, a modified login process takes the call and fires up Kermit in a restricted shell, from which the end user cannot escape.

At the head office, the SPARCstation joins the GNS network via four 64K digital links, which automatically load-share and perform hand-offs to other circuits if any should fail. This gives us an inbound capacity of 256K/second, handled by SunLink X.25.

There is a single menu option available to the interviewer to ‘‘transfer data to head office’’. In all cases, a single ZIP’d file is sent from the PC to the Sun and another back from the Sun to the PC. (Actually there are other hidden menu options to put Kermit into ‘‘verbose’’ mode for error trapping).

A long (500+ lines) Kermit script file handles the entire communications process. It reads data from the configuration file and sets up a call to the local PoP. Because there are over 70 ways in which the process can fail there is extensive error checking and reporting in the script file. Having set the call up to the PoP, Kermit logs in and the PoP automatically and sets up an X.25 call to the Sun here in London (this is the GNS autocall feature). Kermit then logs into the Sun host, does some authentication and then starts C-Kermit on the Sun in server mode. The PC file is uploaded first and then, if it exists, the Sun file is downloaded.

At any failure or exception, an external program is called to beep and display an appropriate flashing error message. Otherwise the communication session terminates normally and the machine is returned to the main menu.

We'll be upgrading to Kermit 3.14 soon to take advantage of the the new performance and recovery features which will greatly improve the overall system. We experience around 10% failure rates on the UK phone system which can be frustrating if you are 250K into a 300K transfer. We've already lab tested this feature to death and have found it to be extremely successful and will be deploying it following some more small-scale field trials with agents dotted around the country.

Since February 1994, we have transferred over 50,000 files of about 250K each on average – 12.5Gbytes roughly. Kermit has performed magnificently and the sophistication of the scripting and error reporting has meant that even when things have gone wrong the support team has easily been able to spot the problems (most often the modem not switched on!). We've taken advantage of both large packets and sliding windows to enhance performance of the file transfers – we have also disabled compression on the modems themselves, since that process was actually leading to significantly slower transfers of the ZIP files (some 20%).

The move away from traditional data collection techniques (pen and paper, primarily) has been a sea change for the market research industry. As well as the considerable technical challenge, we have faced a number of difficult management issues – scaling down our print department, elimination of our key-to-disk department, the changed lines of communication within the business itself.

Putting on a green hat, we've eliminated seven million sheets of paper a year already – we expect this to rise to a savings of twenty million within two years. Then of course there was the huge challenge of training 600 people, most of whom had never used a PC before!

Kermit has been central to the success of the project thus far, and with the continuing developments such as the newly introduced recovery facility, it is certain to remain a cornerstone of the system.

MS-DOS Kermit and Screen Reading Technology

Computer Access for Persons with Print Handicaps

Alan Cantor
West Toronto, Ontario

Personal computers equipped with adaptive technology are making it possible for thousands of individuals with disabilities to pursue independently their personal, vocational and educational goals. Examples of adaptive technologies include environmental control units; TDDs (Telecommunication Devices for the Deaf); large keyboards; input devices actuated by infra-red pointers, eye blinks, puffs of air, and head-sticks; voice recognition systems; optical character recognition systems (OCR); reading machines; text enhancement software; refreshable braille displays; and screen readers. This article is about screen readers, and why MS-DOS Kermit is an appropriate communication program for use in conjunction with this class of adaptive technology.

> *The proliferation of graphics-based computer environments threatens to wipe out many of the gains made by persons who rely on screen reading technology.*

A screen reader transforms an ordinary PC into a talking computer. The voice is heard through a headphone or the system speaker. The screen reader pronounces keystrokes or words as they are typed. Keystroke commands allow the user to ''read'' individual characters, words, lines, sentences, and paragraphs. Some screen reader systems use Alt- and Control-key combinations for their reading functions; others use a separate keypad. All screen reading systems allow the user to customize and save vocal characteristics such as reading speed, inflection and pronunciation.

The original screen readers were intended for persons who are blind or who have low-vision. In recent years it has become clear that other constituencies benefit from the technology too. For example, persons who have a learning disability that interferes with their ability to read, but not to write, successfully use screen readers.

Screen reading technology has made previously inaccessible documents available to persons with print handicaps. Because almost all books, newspapers, and journals begin as computer files, the potential exists, for the first time, for persons who have print-handicaps to read any printed document.

MS-DOS Kermit works beautifully with screen readers. Its rich macro language is an exceptional medium. Complex series of commands can be reduced to a single macro or keystroke.

Virtually all DOS, VMS and Unix applications are ''screen reader friendly.'' Screen reading technology enables users to run most DOS applications, access Bulletin boards, communicate by e-mail, and transfer files and other information using Internet services like GOPHER and FTP. The ability to use computers effectively has opened doors to new educational and employment opportunities for persons who have print-handicaps.

The Threat of the GUI

The proliferation of graphics-based computer environments threatens to wipe out many of the gains made by persons who rely on screen reading technology. DOS-based screen readers do not work with GUI (graphical user interface) applications. In character-based applications, information is written to the screen in predictable ways. The consistency of character-based applications makes it easy to program a screen reader to zoom in on the most salient area of the screen. Pressing F7 in WordPerfect, for example, might cause a screen reader to read line 25:

```
Save Document? Yes (No)_
```

The same consistencies let print-handicapped users navigate through complex programs without having to see the screen. In WordPerfect 5.1, line 25 of the document always displays the status line and lines 24 and 25 of the ''List Files'' screen always display a list of options (1 Retrieve; 2 Delete, etc.). A person with a print handicap can easily ''get lost'' in an unfamiliar program. By pressing the key that reads Line 25, the WordPerfect user can usually become reoriented.

Such consistencies do not exist in GUI environments. A Windows screen consists of a riot of icons, scroll bars, pull-down menus, dialogue boxes, and cascading windows. A window can assume different shapes and sizes. The active window may be minimized or maximized; shrunk or expanded; placed above, below, to the side, behind, in front of, or overlapping another window. Although keyboard shortcuts exist, many functions were intended to be done by positioning the screen pointer and clicking the mouse. The mouse was designed to be guided by eye. Controlling a mouse is awkward — if not impossible — if you cannot see the screen.

Efforts to develop screen reading programs for GUI environments have been underway for several years. Screen readers for GUIs are now available, but the graphic user interface itself remains relatively inaccessible. Windows access is especially problematic because many Windows applications write information to the screen in non-standard ways, making automatic detection of screen updates difficult.

The graphical user interface problem is compounded by the proliferation of ''pseudo-GUIs'' — character-based programs that emulate the look and feel of GUI software. Pseudo-GUIs are sometimes even less ''screen reader friendly'' than true-GUIs. The boundaries of a window in a real GUI application can at least be defined; determining the position of a pseudo-window in a pseudo-GUI with a character-based screen reader is problematic. The latest DOS-based screen readers can, however, handle some of the idiosyncrasies of some pseudo-GUIs applications.

There are many horror stories about persons who rely on screen readers being fired or denied promotion because of their difficulties coping with GUIs. In 1993 blind students and staff at the University of Toronto were effectively barred from using the on-line library catalogue when a new ''Windows-like'' interface was introduced. Students with disabilities and their advocates pressured the library for more than a year before the old command-line system was reinstated. Interestingly, now that both systems are readily available, many able-bodied students say they prefer the old command-line interface.

Many DOS-based communication programs are pseudo-GUIs, a development that has perilous consequences for blind computer users. Because these software packages feature dialogue boxes, pull-down menus and graphical file transfer ''thermometers,'' they are inherently more difficult to access by screen reader than text-based communication programs.

Kermit and Screen Readers

I encourage my clients who use screen readers in conjunction with communication software to try MS-DOS Kermit for the following reasons:

1. **Compatibility.** Because MS-DOS Kermit is a command line processing application, it works beautifully with screen readers. Kermit has no pull-down menus or dialog boxes. Users type commands at the prompt, and during terminal emulation they perform special functions by pressing ''hotkeys.'' Screen reading programs configured for DOS can often be used with Kermit with little or no modification.

2. **Automation.** MS-DOS Kermit's rich macro language is an exceptional medium for writing robust dial-up, log-in, and special-purpose scripts.
3. **Simplicity.** MS-DOS Kermit's macro language makes it possible to reduce a complex series of commands to a single macro or keystroke. For example, a script for uploading a file from a PC to a UNIX mainframe *[included on the MS-DOS Kermit 3.14 diskette as UTILS\UPLOAD.SCR]* was custom-designed for a blind doctoral student who had no interest in learning the complexities of using a computer.
4. **Special Commands.** MS-DOS Kermit features many special commands that make it a practical choice for use with screen readers, among those described in Chapter 15 of *Using MS-DOS Kermit*, 2nd Edition, by Christine M. Gianone.

An example Kermit's screen-reader-friendly commands is SET DISPLAY SERIAL, which causes the status of the transfer to be written to the screen as a series of dots and pluses instead of as a screen thermometer and columns of continuously changing numbers. When using a screen reader the default setting (SET DISPLAY REGULAR) produces an ''alphanumeric stream of consciousness.'' SET DISPLAY SERIAL transforms the cacophony into a useful gauge of file transfer progress.

Other settings that I would usually recommend for use with screen readers include SET TERMINAL MARGIN-BELL ON, SET INPUT ECHO OFF, SET MODE-LINE OFF, SET TERMINAL VIDEO-WRITING BIOS.

Of particular utility is the SET KEY command. With it, I can reassign application keystrokes that conflict with screen reader commands, disable online application keystrokes that have untoward effects, and bind special-purpose macros and scripts to a key. For example, UPLOAD.SCR might be activated by Alt-u:

```
define UPLOAD take upload.scr
set key \2326 {\kUpload}
```

Also handy are CLEAR, which clears the INPUT/REINPUT command buffer and the communication device buffer, and SET TERMINAL CLEAR-SCREEN for clearning the terminal screen. Together the two commands effectively wipe the display clean during script operations so the screen reader does not read irrelevant, out-of-date information.

These two commands have proven so indispensable that I have combined them into a single macro:

```
define ERASE-SCREEN clear,-
  set terminal clear-screen
```

Finally, a novel use for the SET PROMPT command, which changes MS-DOS Kermit's interactive command prompt. In some circumstances I have found it advantageous to define macros with single-character names as a means to streamline script and macro operations. Then I modify the prompt to create an audible ''menu prompt:''

```
define 1 {dial School}
define 2 {dial Library}
define 3 {dial Work}
define e {echo Exiting, hangup, exit}
set prompt -
 {1=School 2=Library 3=Work e=Exit> }
```

Conclusion

I have written MS-DOS Kermit scripts for a number of clients who use screen readers. Although I am not print-handicapped, I use virtually identical scripts for my own purposes. Many able-bodied friends and colleagues, having seen my scripts in action, have been impressed enough to request copies. This has led to an ironic situation: more able-bodied people use my ''print-handicapped'' scripts than people with print-handicaps!

The moral of this story is that improving accessibility for persons with disabilities benefits everybody else too. Kermit scripts that are suitable for persons who rely on screen reader technology are good for able-bodied computer users as well.

About the author

Alan Cantor, 171 Roxborough Street, West Toronto, Ontario M5R 1T9, Canada, is a Workplace Accommodation Consultant with a special interest in improving individuals' access to workplaces and schools. His clients include the Ontario Ministry of Labour, the Ontario Ministry of the Environment and Energy, the Office of the Employment Equity Commissioner for the Province of Ontario, the Hugh MacMillan Rehabilitation Centre (Toronto), Services for Students with Disabilities at the Ontario Institute for Studies in Education (Toronto), and the Zimbabwe-Canada General Training Facility (Ottawa). He is an Associate of Advanced Work Design (Toronto), a Research Associate with Employment Achievement Services (Scarborough), and a member of the Advisory Group on Employment Equity for Persons with Disabilities. His current project is a program of Repetitive Strain Injury (RSI) awareness and prevention for persons with disabilities who use assistive devices. He will present a paper on his work on RSI and disability at the 10th annual CSUN ''Technology and Persons with Disabilities'' Conference in Los Angeles, 14–18 March 1995.

Acknowledgments

I would like to thank Professor Betsey Doane, Mr. Tim Noonan, Mr. John O'Rouke, and Dr. Jim Thatcher for their thoughtful critiques of this article.

Ordering Information

Kermit software is distributed by Columbia University on magnetic tape, tape cartridges, and certain diskette formats. Tapes and cartridges include all source code and supporting files in machine-readable form for each Kermit implementation, and in some cases also binaries (encoded in hex or other printable format, along with suitable decoding software, if needed). Diskettes have no source code except when noted on the order form.

Kermit software programs are collected on six reel-to-reel 9-track tapes: A, B, C, D, E, and F. The programs are assigned to tapes A–F as shown in the second column of the Kermit version list as follows: Tape A has the MS-DOS, CP/M-80, and Apple II versions. **Tape F has C-Kermit.** Tape B has the IBM mainframe and DEC PDP-11 versions. Tape C has other miscellaneous microcomputer, PC, and workstation versions. Tape D has other miscellaneous minicomputer and mainframe versions. Tape E contains machine-readable copies of various manuals, articles, the Kermit Digest, newsletters, a character-set-aware text-to-PostScript printing utility, and tape utilities. Tape and cartridge formats:

ANSI: ANSI labeled ASCII, format D (variable length records), blocksize 8192. 9-track, half-inch, reel-to-reel, 1600 bpi. Readable by many computer systems, including VAX/VMS.

TAR: UNIX TAR format, blocksize 10240, 9-track, 1600 bpi.

OS: IBM OS standard labeled EBCDIC, format VB (variable length records), blocksize 8192, 9-track, 1600 bpi, for MVS, CMS, and other mainframe systems. VM/CMS users should order the OS format and use one of the included tape-reading programs to read the tape on a CMS system; instructions are included with the OS tape.

TK50: TK50 tape cartridge for the DEC MicroVAX or VAXstation. VMS BACKUP format. Also readable by TZ30, TK70, and compatible drives.

QIC: UNIX TAR-format quarter-inch tape cartridge. Readable on Sun computers, IBM RS/6000, SCO systems, and other UNIX systems equipped with QIC cartridge drives.

8MM: EXABYTE 8-millimeter cassette, UNIX TAR format.

DAT: 4mm Digital Audio Cassette, UNIX TAR format.

IMPORTANT C-KERMIT INFORMATION

All the above tape and cartridge formats, when ordered in the *C-Kermit* section of the order form (page 31), include printably encoded binaries for the AOS/VS, VOS, Amiga, and Atari versions of C-Kermit. The 9-track ANSI format also includes hex-format binaries for all VMS configurations. The C-Kermit TK50 includes all VMS binaries. No UNIX binaries are included on any tape or cartridge format: you must build UNIX C-Kermit from source code (C compiler required) or order binaries on diskette, or place a custom order. OS/2 C-Kermit should be ordered on diskette.

NEWS AND UPDATES

Our newsletter, *Kermit News*, is mailed free of charge about every year or two to all our customers to bring news of Kermit software releases and developments. Ordering any Kermit material from us automatically adds you to the subscriber list. *Kermit News* includes an order form to obtain new versions of the Kermit software, or call +1 212 854-3703 for inquiries.

TERMS AND CONDITIONS

The Kermit software — including source code — is furnished without warranty of any kind, and neither Columbia University, nor the individual authors or publishers, nor any institution that has contributed Kermit material, acknowledge any liability for any claims arising from the use of Kermit. Since source code is available, users may fix bugs and make improvements, and are encouraged to contribute their work back to Columbia for further distribution.

Kermit software may be ordered by private individuals, corporations, academic or government institutions, and other organizations for their own internal use, but may not be resold or otherwise redistributed to external clients, customers, or contractors without written permission of the Manager of Kermit Development and Distribution at Columbia University. Most Kermit software is copyrighted, and not in the public domain. Contact us for further information.

HOW TO ORDER

Fill out and return the enclosed order form. PREPAYMENT by credit card or check is encouraged; an additional ORDER PROCESSING FEE is required if we must send an invoice. Orders are shipped by delivery service or US mail, normally within 2–4 weeks of receipt, but firm delivery schedules or methods can not be guaranteed. Prices are in US dollars and include shipping costs. When two prices are shown (like $100 / $135), the first price applies to the USA, Canada, and Mexico and the second price is for shipments to other countries (exception: if you can supply your Federal Express account number, then pay the first price). Rush service is available for an extra fee. Call +1 212 854-3703 for additional ordering information. Telephone and Fax orders are accepted if payment is by Master Card or Visa. Use the order form for Fax orders, and, for payment by credit card, be sure to include your signature.

Prefix	*Tape*	*Machine*	*Operating System*	*Program Language*	*Program Version*	*Released yy/mm/dd*
CK	F	*various*	4.1-4.4 BSD	C	5A190	94/10/04
CK	F	*various*	POSIX	C	5A190	94/10/04
CK	F	*various*	UNIX Sys III	C	5A190	94/10/04
CK	F	*various*	UNIX Sys V R2C		5A190	94/10/04
CK	F	*various*	UNIX Sys V R3C		5A190	94/10/04
CK	F	*various*	UNIX Sys V R4C		5A190	94/10/04
EM	B	*various*	EMACS	ELISP	1.4	94/06/04
TRI	C	*various*	TRIPOS	BCPL	-	87/07/10
TD	C	*various*	TurboDOS	ASM	-	92/10/29
FLX	C	*various 6809*	Flex 9	6809 Asm	-	86/04/17
UF	C	*various 6809*	UniFLEX	C	1.5	93/08/23
OS9	C	*various 6809*	OS-9	C	1.5	85/09/20
CK	F	*various 68000*	OS-9	C	5A190	94/10/04
CP	A	*various 808x*	CPM80 2.2	LASM	4.11	91/04/23
CP	A	*various 808x*	CPM80 3.0	LASM	4.11	91/04/23
MS	A	*various 80x86*	MS-DOS	MASM	3.14	95/01/18
CP	A	Access Matrix	CPM80 2.2	LASM	4.11	91/04/23
AR	C	Acorn Archimedes	Arthur	C	1.3	93/05/01
AR	C	Acorn Archimedes	RISC_OS	C	1.3	93/05/01
BBC	C	Acorn BBC B	OS1.20	6502 Assem.	1.45	87/05/19
BBC	C	Acorn BBC B+,128	OS 2	6502 Assem.	1.45	87/05/19
CP	A	Acorn BBC	CPM80 2.2	LASM	4.11	91/04/23
BBC	C	Acorn Compact	OS 3	6502 Assem.	1.45	87/05/19
BBC	C	Acorn Compact	Panos	C	4C 57	87/07/14
BBC	C	Acorn Master 128	OS 3	6502 Assem.	1.45	87/05/19
AC	C	Acorn Workstation	PANOS	C	-	87/07/13
MS	A	ACT Apricot	MS-DOS	MASM	2.30	88/05/12
CP	A	Action Discovery	CPM80 2.2	LASM	4.11	91/04/23
AM	C	Alpha Micro 68K	AMOS/xx	AM68K Asm	2.0	94/03/21
CK	F	Altos ACS68000	UNIX III,V	C	5A190	94/10/04
CK	F	Amdahl Mainframe	UNIX SVR2	C	5A190	94/10/04
CK	F	Amdahl Mainframe	UTSV	C	5A190	94/10/04
IK	B	Amdahl Mainframe	*Also see*	IBM 370		
CP	A	Amstrad	CPM80 2.2	LASM	4.11	91/04/23
APO	C	Apollo	Aegis	Pascal	2.9	89/05/07
CK	F	Apollo	Aegis 9.x	C	5A190	94/10/04
CK	F	Apollo/HP	SR10.0	C	5A190	94/10/04
CP	A	Apple II	CPM80 2.2	LASM	4.11	91/04/23
APP	A	Apple II	DOS,ProDOS	CROSS	3.87	90/12/05
UCA	C	Apple II	UCSD p-Sys	UCSD Pascal	1.0	86/04/08
CK	F	Apple Macintosh	AU/X	C	5A190	94/10/04
CK	F	Apple Macintosh	Mac OS	MPW C	0991	94/08/16
CK	F	Apple Macintosh	MachTen	C	5A190	94/10/04
CN8	C	Argos Pro PC	CCPM86	ASM86	2.9	86/04/10
CK	F	AT&T 3B Series	UNIX SVR3	C	5A190	94/10/04
CK	F	AT&T 6300 PLUS	UNIX SVR3	C	5A190	94/10/04
CK	F	AT&T 6386 WGS	UNIX SVR3	C	5A190	94/10/04
CK	F	AT&T 7300 UNIXPC	UNIX Sys V	C	5A190	94/10/04
ATA	C	Atari Home Comp.	DOS	Action!	-	84/01/09
UCP	C	Atari MEGA ST2	UCSD p-Sys	Pascal	1.1	90/08/05
CK	F	Atari ST	GEMDOS	C	5A189	93/06/30
CK	F	Atari ST	MINIX ST	C	5A190	94/10/04
CP	A	BigBoard II	CPM80 2.2	LASM	4.11	91/04/23
B78	D	Burroughs A-Series	MCS/AS	Algol	1.019	86/09/11
CT	C	Burroughs B20	BTOS	C	2.00	93/01/20
B68	D	Burroughs B6800	CANDE	Algol	-	85/02/15
B78	D	Burroughs B7800	Burroughs	Algol	1.019	86/09/11
B79	D	Burroughs B7900	Burroughs	Algol	5.2	85/11/27
U*xx*	D	Burroughs	*Also see*	UNISYS		
CD3	D	CDC Cyber	NOS	Fortran 5	3.4	88/05/10
CYB	D	CDC Cyber	NOS 2.2	Compass	1.0	86/04/17
NOS	D	CDC Cyber	NOS 2.4	Compass	1.30	87/05/19
CDC	D	CDC Cyber 170	NOS,NOS/BE	Fortran-77	2.2	84/09/07
CK	F	CDC Cyber 910	IRIX	C	5A190	94/10/04
CK	F	Charles River	UNOS	C	5A190	94/10/04
CIE	C	CIE 680/XX	REGULUS	C	-	87/01/26
CP	A	Cifer 1886	CPM80	LASM	4.11	91/04/23
CP	A	Comart CommunicatorCPM80 2.2		LASM	4.11	91/04/23
C64	C	Commodore 64	FORTH	FORTH	1.5	85/02/08
C64	C	Commodore 64/128	DOS	CROSS	2.27	92/09/30
CK	F	Commodore Amiga	3000UX	C	5A190	94/10/04
CK	F	Commodore Amiga	Intuition	C	5A190	94/10/04
CP	A	Compupro IF 3/4	CPM80 2.2	LASM	4.11	91/04/23
CVK	D	Computervision	CGOS	Fortran S	1.21	87/03/04
PER	D	Concurrent 3200	OS/32 MT72	Fortran	1.0	87/03/04
CK	F	Concurrent 3200	Xelos SV	C	5A190	94/10/04
CK	F	Concurrent 6xxx	RTU 4,5,6	C	5A190	94/10/04
CK	F	Convergent	CTIX	C	5A190	94/10/04
CT	C	Convergent NGEN	CTOS	C	2.00	93/01/20
CK	F	Convex	ConvexOS	C	5A190	94/10/04
CP	A	CPT-85xx	CPM80 2.2	LASM	4.11	91/04/23
CK	F	Cray C90	UNICOS 6,7,8	C	5A190	94/10/04
CK	F	Cray X/MP, Y/MP	UNICOS 6,7,8	C	5A190	94/10/04
CR	D	Cray-1,Cray-XMP	CTSS	Fortran-77	-	85/02/08
CK	F	Cray-2,Cray-3	CSOS	C	5A190	94/10/04
CP	A	Cromemco	CPM80 2.2	LASM	4.11	91/04/23
CN8	C	Daisy PCi	CCPM86	ASM86	2.9	86/04/10
CK	F	DEC Alpha AXP	OpenVMS	C	5A190	94/10/04
CK	F	DEC Alpha AXP	OSF/1	C	5A190	94/10/04
MS	A	DEC Alpha AXP	Windows NT	MASM, C	3.14	95/01/18
K12	D	DEC PDP-8	OS/8 Fam.	PAL-8	10g	90/09/13
CK	F	DEC PDP-11	2.1x BSD	C	5A190	94/10/04
K11	B	DEC PDP-11	IAS 3.1	Macro-11	3.60	89/06/13
MP	B	DEC PDP-11	MUMPS-11	MUMPS-1982	-	84/04/11
K11	B	DEC PDP-11	RSTS/E	Macro-11	3.60	89/06/13
K11	B	DEC PDP-11	RSX-11/M	Macro-11	3.60	89/06/13
K11	B	DEC PDP-11	RSX-11/M+	Macro-11	3.60	89/06/13
KRT	B	DEC PDP-11	RT-11	Macro-11	3.62	93/09/22
KRT	B	DEC PDP-11	TSX+	Macro-11	3.62	93/09/22
K12	D	DEC PDP-12	OS/12	PAL-8	10g	90/09/13
K11	B	DEC Pro-3xx	P/OS	Macro-11	3.60	89/06/13
KRT	B	DEC Pro-3xx	Pro/RT	Macro-11	3.60	93/09/22
C86	C	DEC Rainbow	CPM86	ASM86	2.9	84/12/03
MS	A	DEC Rainbow	MS-DOS	MASM	3.10	91/03/18
QNX	C	DEC Rainbow	QNX 1.x	C	1.0	85/09/23
CK	F	DEC VAX	4.xBSD	C	5A190	94/10/04
PIC	D	DEC VAX	PICK	DATA/BASIC	0.3	89/08/21
CK	F	DEC VAX	(Open)VMS	C	5A190	94/10/04
CK	F	DEC VAX	ULTRIX	C	5A190	94/10/04
CK	F	DEC VAX	UNIX SVR3	C	5A190	94/10/04
MS	A	DEC VAXmate	MS-DOS	MASM, C	3.14	95/01/18
CK	F	DEC VAXstation	(Open)VMS	C	5A190	94/10/04
CK	F	DEC VAXstation	ULTRIX	C	5A190	94/10/04
CP	A	DEC VT-180 Robin	CPM80 2.2	LASM	4.11	91/04/23
K12	D	DEC VT-78	OS/8 Fam.	PAL-8	10g	90/09/13
K12	D	DEC VT-278	OS/78 Fam.	PAL-8	10g	90/09/13
K12	D	DECmate I	OS/78 V4	PAL-8	10g	90/09/13
K12	D	DECmate II,III	OS/278 V2	PAL-8	10g	90/09/13
CP	A	DECmate-II,III	CPM80 2.2	LASM	4.11	91/04/23
MS	A	DECmate-II,III	MS-DOS	MASM	2.29	86/05/28
CK	F	DECstation	MACH	C	5A190	94/10/04
CK	F	DECstation	OSF/1	C	5A190	94/10/04
CK	F	DECstation	ULTRIX	C	5A190	94/10/04
K10	D	DECsystem-10	TOPS-10	Bliss, Macro	3.134	89/09/18
K20	D	DECSYSTEM-20	TOPS-20	MACRO-20	4.2	88/01/25
CK	F	Dell PC	Dell UNIX	C	5A190	94/10/04
CP	A	Delphi 100	CPM80 2.2	LASM	4.11	91/04/23
RD2	D	DG 800	RDOS	BASIC	-	87/03/26
CK	F	DG AViiON	DG/UX	C	5A190	94/10/04
AOS	D	DG MV	AOS,AOS/VS	SP/Pascal	-	85/02/08
CK	F	DG MV	AOS/VS	C	5A190	94/10/04
CK	F	DG MV	AOS/VS-II	C	5A190	94/10/04
DGM	D	DG MV	AOS/VSMVUX	C	-	85/11/27
RDO	D	DG Nova	RDOS	Fortran-5	-	84/09/14
AOS	D	DG S250	AOS	Fortran-5	-	84/09/14
CK	F	DIAB DS90	DNIX	C	5A190	94/10/04
CK	F	Dolphin Triton	UNIX SV/88	C	5A190	94/10/04
CK	F	Encore	UMAX 4.x	C	5A190	94/10/04
CK	F	Encore	UMAX V	C	5A190	94/10/04
CP	A	Epson PX8	CPM80 2.2	LASM	4.11	91/04/23
LUX	C	FACIT DTC/DTC2	ABC-DOS	ABC-BASIC-II	4.11	90/07/13
CN8	C	Fallon 2000	CCPM86	ASM86	2.9	86/04/10
CP	A	Ferguson BigB.I	CPM80 2.2	LASM	4.11	91/04/23
CK	F	Fortune 32:16	For:Pro2.1	C	5A190	94/10/04
CK	F	FPS 500	FPX 4.1	C	5A190	94/10/04
CN8	C	FTS PCi	CCPM86	ASM86	2.9	86/04/10
C86	C	Fujitsu Micro16s	CPM86	ASM86	2.9	85/09/23
C86	C	Future FX20/FX30	CPM86	ASM86	2.9	86/04/10
GEC	D	GEC 4000 Series	OS4000	MUM/SERC	3.9	89/05/07
OS9	C	Gimex III	OS-9	C	1.5	85/09/20
GM	D	Gould/SEL 32	MPX-32	Fortran 77+	2.3	86/12/10
MS	A	GRiD Compass II	MS-DOS	MASM	3.10	91/03/18
H1	D	Harris H100-1	VOS 4.1.1	Fortran-77	1.06	88/03/17
H8	D	Harris 800	VOS	Pascal,Asm	-	85/02/11
CK	F	Harris NightHawk	CX/UX 6.1	C	5A190	94/10/04
CP	A	Heath H8	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Heath/Zenith-89	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Heath/Zenith-100	CPM85	LASM	4.11	91/04/23
MS	A	Heath/Zenith-100	MS-DOS	MASM	3.14	95/01/18
MU	D	Honeywell	MULTICS	PL/I	2.0h	84/09/20
HD6	D	Honeywell DPS6	GCOS6	C	2.01	91/06/03
HG	D	Honeywell DPS66	GCOS3,8	C	3.0	84/10/05
HC6	D	Honeywell DPS8	CP-6	PL/6	1.00	88/01/28
HCP	D	Honeywell DPS8	CP-6	Pascal	-	85/04/04
HDP	D	Honeywell DPS8	GCOS/TSS	B	1.1	85/03/21
HG	D	Honeywell DPS8	GCOS3,8	C	3.0	84/10/05
HC6	D	Honeywell DPS90	CP-6	PL/6	1.00	88/01/28
HCP	D	Honeywell DPS90	CP-6	Pascal	-	85/04/04
HL6	C	Honeywell L6/10	MS-DOS	MASM	1.20A	84/10/05
HP8	C	HP86	HP-BASIC	HP BASIC	1.01	87/04/29
HP8	C	HP87	HP-BASIC	HP BASIC	1.01	87/04/29
MS	A	HP95, 100	MS-DOS	MASM, C	3.14	95/01/18
MS	A	HP Portable Plus	MS-DOS	MASM	3.10	91/03/18
MS	A	HP-110	MS-DOS	MASM	3.10	91/03/18
CP	A	HP-125	CPM80 2.2	LASM	4.11	91/04/23
MS	A	HP-150	MS-DOS	MASM	3.10	91/03/18
HPM	D	HP-1000	RTE6, RTEA	F77 & Asm	1.99D	90/01/09
HP2	C	HP-264x	-	8080ASM	1.2	87/10/09
HP3	D	HP-3000	MPE	SPL	-	94/07/20
HP3	D	HP-3000	MPE	C	-	94/07/20
CK	F	HP-9000	HP-UX	C	5A190	94/10/04
HPB	C	HP-9000/200,/300	HP-BASIC	HP BASIC	1.02	89/06/21
CK	F	HP-9836CU	HP-UX	C	5A190	94/10/04
HP9	C	HP-9845	BASIC/SAM	HP BASIC	1.00	86/10/07
HP9	C	HP-98xx	UCSD p-Sys	HP Pascal	-	84/01/20
IK	B	IBM 370 Series	CICS/MVS	Assembler	4.3.1	95/02/02

Prefix	*Tape*	*Machine*	*Operating System*	*Program Language*	*Program Version*	*Released yy/mm/dd*
IK	B	IBM 370 Series	CICS/VSE	Assembler	4.3.1	95/02/02
MT2	D	IBM 370 Series	MTS	PLUS	-	86/11/03
MTS	D	IBM 370 Series	MTS	Assembler	-	84/01/06
MTS	D	IBM 370 Series	MTS	Pascal	1.0	84/01/06
IK	B	IBM 370 Series	MUSIC	Assembler	4.2.3	92/09/30
GUT	D	IBM 370 Series	MVS/GUTS	Assembler	-	85/04/05
IK	B	IBM 370 Series	MVS/ROSCOE	Assembler	4.3.1	95/02/02
IK	B	IBM 370 Series	MVS/TSO	Assembler	4.3.1	95/02/02
IK	B	IBM 370 Series	MVSXA/TSOE	Assembler	4.3.1	95/02/02
IK	B	IBM 370 Series	VM/CMS	Assembler	4.3.1	95/02/02
MS	A	IBM compatibles	MS-DOS	MASM, C	3.14	95/01/18
CS9	C	IBM CS9000	CSOS	Pascal	-	92/09/10
CK	F	IBM Mainframe	AIX	C	5A190	94/10/04
UCI	C	IBM PC	UCSD p-Sys	UCSD Pascal	0.1	84/05/23
CC	C	IBM PC + clones	ChineseDOS	MASM	2.32A	91/09/09
MX	B	IBM PC family	MINIX 1.0	C	4D 61	88/05/17
TP4	C	IBM PC family	PC-DOS	Turbo Pascal	1.1a	88/04/15
QK	C	IBM PC,XT,AT	PC-DOS	Turbo Pascal	3.1	88/12/14
QNX	C	IBM PC,XT,AT	QNX 1.x	C	1.0	85/09/23
MS	A	IBM PC family	PC-DOS	MASM, C	3.14	95/01/18
MS	A	IBM PC family	MS Windows	MASM, C	3.14	95/01/18
WK	A	IBM PC family	MS Windows	Microsoft C	-	91/10/19
MS	A	IBM PC family	Windows NT	MASM, C	3.14	95/01/18
CK	A	IBM PC family	OS/2	C	5A191	95/03/01
PIC	D	IBM PC/XT,AT	PICK	DATA/BASIC	0.3	89/08/21
CK	F	IBM PS/2	AIX 1.2	C	5A190	94/10/04
MS	A	IBM PS/2 Series	PC-DOS	MASM, C	3.14	95/01/18
CK	F	IBM RS/6000	AIX 3.x	C	5A190	94/10/04
CK	F	IBM RT PC	4.3BSDReno	C	5A190	94/10/04
CK	F	IBM RT PC	ACIS 4.x	C	5A190	94/10/04
CK	F	IBM RT PC	AIX 2.x	C	5A190	94/10/04
VME	D	ICL 2900	VME	S3	1.01	87/07/14
CK	F	ICL DRS3000,6000	DRS/NX	C	5A190	94/10/04
CN8	C	ICL PC 2,Quattro	CCPM86	ASM86	2.9	87/05/17
PQ	C	ICL/Perq	Perq OS	Pascal	2.0	84/12/04
MS	A	Intel 300 Series	iRMX-286	MASM/ASM86	2.30	88/05/02
CK	F	Intel 302	Bell Tech	C	5A190	94/10/04
RMX	C	Intel 86,286	RMX 1.0	PL/M	1.0	85/10/25
I86	C	Intel 86/380	iRMX-86	PL/M	2.3	85/09/23
IRM	C	Intel 86/380	iRMX-86	PL/M	2.41	87/03/04
MD	C	Intel MDS	ISIS	PL/M	-	87/04/06
CK	F	Intergraph Clipper	CLIX	C	5A190	94/10/04
CP	A	Ithaca Intersys	CPM80 2.2	LASM	4.11	91/04/23
UCJ	C	J Loebl Magiscan2	UCSD p-Sys	UCSD Pascal	-	86/06/23
CP	A	Kaypro II	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Kaypro 4	CPM80 2.2	LASM	4.11	91/04/23
M2	C	Lilith Worksta.	Medos	Modula-2	1.0	87/05/17
LM	C	LMI Lispmachine	LMI-Lambda	ZETALISP	1.0	85/09/12
CP	A	Lobo Max-80	CPM80 2.2	LASM	4.11	91/04/23
LUX	C	Luxor ABC-80	ABC-DOS	Z80 Asm	1.0	90/07/13
LUX	C	Luxor ABC-80x	ABC-DOS	ABC-BASIC-II	4.11	90/07/13
CK	F	Luxor ABC-9000	DNIX	C	5A190	94/10/04
MBF	D	MAI Basic Four	BOSS/VS	BASIC BB86	1.0	88/04/11
CK	F	Masscomp	RTU 4.0+	C	5A190	94/10/04
CP	A	Merlin M2215	CPM80 2.2	LASM	4.11	91/04/23
PIC	D	MicroDataREALITY	PICK	DATA/BASIC	0.3	89/08/21
CP	A	Micromint SB180	CPM80 2.2	LASM	4.11	91/04/23
CK	F	MIPS System	RISC/os	C	5A190	94/10/04
CK	F	Modcomp	Real/IX	C	5A190	94/10/04
MOD	D	Modcomp Classic	MAX IV	Fortran/ASM	A.0	87/01/26
CP	A	MorrowDecisionI	CPM80 2.2	LASM	4.11	91/04/23
CP	A	MorrowMicroDecI	CPM80 2.2	LASM	4.11	91/04/23
FL	C	Motorola 6809	Flex	Assembler	-	86/02/14
FL2	C	Motorola 6809	FLEX-09	C	3.0	87/03/04
CK	F	Motorola Delta	SV/68 R3x	C	5A190	94/10/04
CK	F	Motorola Delta	SV/88 R3,4	C	5A190	94/10/04
NCR	D	NCR 9800-4	VE4.0	C	4E 72	90/07/13
CP	A	NCR Decisionmate	CPM80 2.2	LASM	4.11	91/04/23
CK	F	NCR System 3000	UNIX SVR4	C	5A190	94/10/04
CK	F	NCR Tower 1632	UNIX SVR2	C	5A190	94/10/04
CK	F	NCR Tower 32	UNIX SVR3	C	5A190	94/10/04
C86	C	NEC APC	CPM86	ASM86	2.9	84/12/03
MS	A	NEC APC	MS-DOS	MASM	2.29	86/05/28
MS	A	NEC APC III	MS-DOS	MASM	2.30	88/03/21
MS	A	NEC PC9801	MS-DOS	MASM	3.10	91/04/18
CK	F	NeXT	NeXTSTEP	C	5A190	94/10/04
NIC	C	Nicolet 80	Demon	Assembler	1.76	94/07/01
CK	F	Nixdorf Targon31	TOS	C	5A190	94/10/04
CP	A	Nokia MikroMikko	CPM80 2.2	LASM	4.11	91/04/23
ND	D	Norsk Data 10,..	SintranIII	ND-Pascal	3.1b	85/06/24
CK	F	Norsk Data 88/17	UNIX SV/88	C	5A190	94/10/04
CP	A	Northstar	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Ohio Scientific	CPM80 2.2	LASM	4.11	91/04/23
CK	F	OkiStation 7300	UNIX SVR4	C	5A190	94/10/04
CK	F	Olivetti CP 486	UNIX SVR4	C	5A190	94/10/04
CK	F	Olivetti LSX30xx	X/OS	C	5A190	94/10/04
CK	F	Olivetti PCs	SCO UNIX	C	5A190	94/10/04
MS	A	Olivetti PCs	MS-DOS	MASM, C	3.14	95/01/18
CN8	C	Orion PCi	CCPM86	ASM86	2.9	86/04/10
CP	A	Osborne 1	CPM80 2.2	LASM	4.11	91/04/23
UCM	C	PascalMicroengin	UCSD p-Sys	Pascal	III.0	84/12/03
MS	A	PC/8088 & higher	MS-DOS	MASM, C	3.14	95/01/18
CK	F	PC/286 & higher	Microport	C	5A190	94/10/04

Prefix	*Tape*	*Machine*	*Operating System*	*Program Language*	*Program Version*	*Released yy/mm/dd*
CK	F	PC/286 & higher	QNX 4.2x	C	5A190	94/10/04
CK	F	PC/286 & higher	SCO Xenix	C	5A190	94/10/04
CK	F	PC/286 & higher	Trusted Xenix	C	5A190	94/10/04
CK	F	PC/386 & higher	386BSD	C	5A190	94/10/04
CK	F	PC/386 & higher	BSDI/386	C	5A190	94/10/04
CK	F	PC/386 & higher	Coherent	C	5A190	94/10/04
CK	F	PC/386 & higher	ESIX SVR3	C	5A190	94/10/04
CK	F	PC/386 & higher	FreeBSD	C	5A190	94/10/04
CK	F	PC/386 & higher	Interactive	C	5A190	94/10/04
CK	F	PC/386 & higher	Linux/386	C	5A190	94/10/04
CK	F	PC/386 & higher	Lynx	C	5A190	94/10/04
CK	F	PC/386 & higher	MINIX/386	C	5A190	94/10/04
CK	F	PC/386 & higher	MtXinuMach	C	5A190	94/10/04
CK	F	PC/386 & higher	NetBSD	C	5A190	94/10/04
CK	F	PC/486 & higher	NeXTSTEP	C	5A190	94/10/04
CK	F	PC/386 & higher	OS/2 1.x	C	5A191	95/03/01
CK	F	PC/386 & higher	OS/2 2.0	C	5A191	95/03/01
CK	F	PC/386 & higher	OS/2 Warp	C	5A191	95/03/01
CK	F	PC/386 & higher	SCO ODT	C	5A190	94/10/04
CK	F	PC/386 & higher	SCO UNIX	C	5A190	94/10/04
CK	F	PC/386 & higher	Solaris	C	5A190	94/10/04
CK	F	PC/386 & higher	UnixWare	C	5A190	94/10/04
UCP	C	Pecan	UCSD p-Sys	Pascal	1.1	90/08/05
PER	D	PerkinElmer 3200	OS/32 MT72	Fortran	1.0	87/03/04
CK	F	PerkinElmer 3200	Xelos SV	C	5A190	94/10/04
PE7	D	PerkinElmer 7000	IDRIS	C	1.1 0	86/12/08
CP	A	PMC Micromate101	CPM80 2.2	LASM	4.11	91/04/23
PRI	D	Prime	PRIMOS R2x	PL/P	8.15	93/04/19
CK	F	Pyramid 9810x(T)	OSx	C	5A190	94/10/04
CP	A	Rair Black Box	CPM80 2.2	LASM	4.11	91/04/23
CP	A	RM380ZF, ZM	CPM80 2.2	LASM	4.11	91/04/23
RM	C	RML 480Z	ROS 2.x	C	1.22	86/11/03
RM	C	RML Nimbus	MS-DOS?	C	1.22	86/11/03
CP	A	Sanyo 1100 MBC	CPM80 2.2	LASM	4.11	91/04/23
MS	A	Sanyo 550 MBC	MS-DOS	MASM	2.30	88/05/16
CP	A	ScreenTyper	CPM80 2.2	LASM	4.11	91/04/23
CK	F	Sequent	DYNIX(/ptx)	C	5A190	94/10/04
CK	F	Silicon Graphics	IRIX	C	5A190	94/10/04
QL2	C	Sinclair QL	QDOS	BCPL	-	87/05/15
QLK	C	Sinclair QL	QDOS	C	1.10	87/05/15
CK	F	Solbourne	OS/MP	C	5A190	94/10/04
CK	F	Sony NEWS	NEWS-OS	C	5A190	94/10/04
UN	D	Sperry 1100	Exec	Assembler	2.5	86/09/03
UN	D	Sperry 1100	Exec	NOSC Pascal	2.0	84/10/08
CK	F	Sperry 5000	UNIX SVR3	C	5A190	94/10/04
CK	F	Sperry 5000	UTS V	C	5A190	94/10/04
SP9	D	Sperry 90/60	VS9	Assembler	-	86/04/09
CK	F	Stardent 1520	UNIX SVR3	C	5A190	94/10/04
CK	F	Sun, all models	Solaris 1,2	C	5A190	94/10/04
CK	F	Sun, all models	SunOS 3,4,5	C	5A190	94/10/04
CP	A	Superbrain	CPM80 2.2	LASM	4.11	91/04/23
LM	C	Symbolics 36xx	Lisp	ZETALISP	1.0	85/09/12
TAN	D	Tandem Nonstop	Guardian	TAL	1.0	86/04/08
TA1	C	Tandy 100	Tandy 100	BASIC	1984	90/10/07
TA2	C	Tandy 2000	MS-DOS	MASM	1.20	84/02/16
CK	F	Tandy Model 16	Xenix 3.0	C	5A190	94/10/04
C86	C	Tektronix 4170	CPM86	ASM86	2.9	84/12/03
CK	F	Tektronix 4xxx	UTek	C	5A190	94/10/04
CK	F	Tektronix 6130	UTek	C	5A190	94/10/04
CK	F	Tektronix XD88	UTek	C	5A190	94/10/04
CP	A	Telcon Zorba	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Teletek	CPM80 2.2	LASM	4.11	91/04/23
UCT	C	Terak	UCSD p-Sys	UCSD Pascal	-	84/04/11
TI9	D	TI 990	DX10	Pascal	1.0	87/07/10
EXP	C	TI Explorer	LISP	Common Lisp	1.0	87/03/04
MS	A	TI Professional	MS-DOS	MASM	2.29	86/05/28
CP	A	Torch Unicorn 5	CPM80 2.2	LASM	4.11	91/04/23
CK	F	Tri Star Flash	ESIX	C	5A190	94/10/04
CO	C	TRS-80 CoCo	DOS	EDTASM	1.1	85/03/21
OS9	C	TRS-80 CoCo	OS-9	C	1.5	85/09/20
TRS	C	TRS-80 I and III	TRSDOS	M80	3.5	84/08/08
CP	A	TRS-80 Model II	CPM80 2.25	LASM	4.11	91/04/23
TR2	C	TRS-80 Model II	TRSDOS	Assembler	1.2	87/03/26
CP	A	TRS-80 Model 4	CPM80 2.2	LASM	4.11	91/04/23
M4	C	TRS-80 Model 4	TRSDOS	ASM	5.2	86/10/29
UM	C	UMicro U-MAN1000	CP/M-68K	C and Asm	-	86/04/10
CK	F	UNISYS 5000	UNIX SVR3	C	5A190	94/10/04
CK	F	UNISYS 5000	UTS V	C	5A190	94/10/04
UAS	D	UNISYS A-Series	MCS/AS	Algol	1.041	90/07/13
USY	D	UNISYS A-Series	MCS/AS	Algol	0.0	94/07/20
CP	A	USmicroSalesS100	CPM80 2.2	LASM	4.11	91/04/23
CP	A	Vector Graphics	CPM80 2.2	LASM	4.11	91/04/23
C86	C	Victor/Sirius 1	CPM86	ASM86	2.9	86/07/07
MS	A	Victor/Sirius 1	MS-DOS	MASM	3.14	95/01/18
CP	A	Video Genie	CPM80 2.2	LASM	4.11	91/04/23
MS	A	Wang PC, APC	MS-DOS	MASM	2.31	88/08/13
CP	A	Xerox 820	CPM80 2.2	LASM	4.11	91/04/23

Prices, terms, and items are subject to change. If this form is dated more than 12 months prior, please contact us for new information. Please order carefully since we can not refund or exchange items that were ordered incorrectly. Prices are in US dollars ($), **first price for North America / second price for shipping outside North America**.

MS-DOS KERMIT 3.14 FOR DOS AND WINDOWS

IBM PC/PS2 MS-DOS Kermit software with book *Using MS-DOS Kermit*, $36.95 / $47:

❒ 5.25-inch ❒ 3.5-inch: . . . . $____________

MS-DOS IBM PC/PS2 Kermit source code, $60 / $68:

❒ 5.25-inch ❒ 3.5-inch: . . . . $____________

Crynwr (formerly Clarkson) packet drivers. For the IBM PC family on DOS diskettes:

Binaries and Docs, $35 / $40: ❒ 5.25-inch ❒ 3.5-inch . . . . $____________

Source Code, $60 / $68: ❒ 5.25-inch ❒ 3.5-inch . . . . $____________

C-KERMIT 5A(190). Each format includes the book, *Using C-Kermit*.

Tapes and Cartridges (Tape F) − Please read IMPORTANT C-KERMIT INFO on page 28.

❒ 9-Track Tape, $135 / $170, Format: ❒ ANSI (includes VMS ‘‘hex’’ binaries), ❒ TAR . . . . $____________

❒ TK50 cartridge, DEC VMS / OpenVMS BACKUP format, includes VMS binaries, $185 / $220 . . . . $____________

❒ Quarter-Inch Cartridge (QIC), UNIX TAR format, $185 / $220 . . . . $____________

❒ 4mm DAT cassette, UNIX TAR format, $185 / $220 . . . . $____________

❒ 8mm EXABYTE cartridge, UNIX TAR format, $185 / $220 . . . . $____________

❒ C-Kermit source code on DOS-format diskettes, $100 / $115: ❒ 5.25-inch, ❒ 3.5-inch: . . . . $____________

C-Kermit Binaries on DOS-format diskettes, $45 / $55 each:

3.5"	5.25"		
❒	❒	C-Kermit for OS/2 2.00 & Warp, 32-bit	$____________
❒	❒	C-Kermit for OS/2 1.xx, 16-bit	$____________
❒	❒	C-Kermit for QNX	$____________
❒	❒	C-Kermit for SCO UNIX / ODT	$____________
❒	❒	C-Kermit for SCO Xenix	$____________
❒	❒	C-Kermit for Solaris/SPARC	$____________
❒	❒	C-Kermit for Solaris/Intel	$____________
❒	❒	C-Kermit for Commodore Amiga	$____________

Others: contact us for availability.

9-TRACK 1600-bpi MAGNETIC TAPE. Price: $100 / $135 *per tape:*

	ANSI	TAR	OS		ANSI	TAR	OS		ANSI	TAR	OS
Tape A:	❒	❒	❒	Tape C:	❒	❒	❒	Tape E:	❒	❒	❒
Tape B:	❒	❒	❒	Tape D:	❒	❒	❒	*See above for C-Kermit . . .*			

9-TRACK TAPE SUBTOTAL . . . . $____________

TAPE CARTRIDGES. $150 / $185 each.

	DAT	8MM	QIC	TK50
Contents of Tapes A, B, and E:	❒	❒	❒	❒
Contents of Tapes C, D, and E:	❒	❒	❒	❒

TAPE CARTRIDGE SUBTOTAL . . . . $____________

MACINTOSH KERMIT *(a work in progress . . .)*

❒ *Macintosh Kermit* 0.991(190) or latest prerelease, 3.5-inch, $25 / $30 . . . . $____________

LITERATURE

❒ Book: *Using MS-DOS Kermit:* $36.95 / $47 . . . . $____________

❒ Book: *Using C-Kermit* (without software): $36.95 / $47 . . . . $____________

❒ Book: *Kermit, A File Transfer Protocol* (Kermit protocol specification): $32.95 / $43 . . . . $____________

❒ Book: *Kermit MS-DOS Mode d'Emploi* (in French): $36.95 / $47 . . . . $____________

❒ Any three of these books: $85.00 / $115 . . . . $____________

Contact us for quantity discounts on books.

SIDE 1 SUBTOTAL (please complete side 2 also) . . . . $________________

Shipping by UPS or post is included in the price. Please do not add sales tax.

SUBTOTAL from Side 1: . . . . $____________

❒ Voluntary tax-deductible donation *(help support the Kermit effort) :* . . . . $____________

USA, CANADA, AND MEXICO ONLY:
❒ For PRIORITY HANDLING and NEXT-DAY SHIPMENT, add $30: . . . . $____________

A. TOTAL MATERIALS, DONATION, AND SHIPPING: . . . . $____________

Please complete ONE of the numbered sections, 1, 2, or 3, and then fill in your shipping information. Use only the payment methods indicated; please do NOT attempt to pay us with international bank transfers or postal coupons, or with credit cards that are not listed.

1. PAYMENT BY CREDIT CARD:

Check one: ❒ MasterCard ❒ Visa

Cardholder Name ____ TOTAL PAYMENT (Line A above): $_________________

Card Number ____ Expiration Date ______________________________________

Signature ____ Today's Date ____

2. PREPAYMENT BY CHECK

Please make your check *payable in US dollars* to:

Columbia University Kermit Distribution

B. If your check is *not* drawn on a US bank, please add a $35 check-cashing fee: . . . . . $____________

TOTAL AMOUNT OF YOUR CHECK (Lines A and B): . . . . $____________

3. PURCHASE ORDER, WE BILL YOU

PO Number:_________________________

C. Add $25 invoicing fee: $____________
D. If your check will *not* be drawn on a US bank, add $35 check-cashing fee: $____________

TOTAL, Lines A, C, and D. Please enclose your purchase order for this amount: . . . . $____________

SHIPPING INFORMATION

(Please do not use Post Office Box for UPS or Federal Express):

Name: ____

Company: ____

Address: ____

City:__________________________________State or Province:______________________Zip or Postal Code: ____________________

Country:____Phone: ______________________________________

E-Mail and/or Fax, if any:____

If you want us to use your Federal Express account, please provide account number and authorized signature:

Federal Express Account Number:___________________________ Signature:____

MAIL YOUR COMPLETED ORDER FORM TO:

Kermit Distribution
Columbia University Academic Information Systems
612 West 115th Street
New York, NY 10025 USA

Phone: +1 212 854-3703, FAX: +1 212 663-8202, E-Mail: kermit@columbia.edu. Sorry, we can't respond by FAX.

Thank you!

Directory

Staff:

Frank da Cruz
Christine M. Gianone
Max Evarts
Andy Newcomb

Postal Address:

Kermit Development and Distribution
Columbia University Academic Information Systems
612 West 115th Street
New York, NY 10025 USA

Telephone:

Ordering Information:	+1 212 854-3703
Technical Support:	+1 212 854-5126
Fax:	+1 212 663-8202

Networks:

E-mail: kermit@columbia.edu
News: comp.protocols.kermit.announce, .misc
Web: http://www.columbia.edu/kermit/

Kermit News

Kermit Development and Distribution
Columbia University Academic Information Systems
612 West 115th Street
New York, NY 10025 USA

Non Profit Org.
U.S. Postage
PAID
New York, N.Y.
Permit # 3593