

K E R M I Tを利用したファイル転送について

教養部 田井村 明博

1. はじめに

本学計算機システムには、ファイル転送用のプロトコルとしてデータ転送が高速で信頼性の高いFTP (File Transfer Protocol) コマンドがUTSに用意されている。しかしながら、この機能を使えるのはEthernetに接続された端末あるいはワークステーションに限られており、現在ではセンター内の端末か工学部内の端末からしか利用できない¹⁾。もちろんFTPの他にもFEXPORT, FIMPORTコマンドを利用したファイル転送が、各学部設置されているFMR-60HDから行えるようになっている。また、研究室のパソコンから電話回線を介してファイル転送可能なターミナルプログラムがユーザーの手によって開発されている²⁾。学内ネットワークが整備されれば、転送の信頼性、スピードともにすぐれたFTPが利用できるようになるであろうが、ネットワーク構想が発案されたばかりで実現にはまだ少なくとも2, 3年要するだろう。

一方、パソコン通信では、ファイル転送にXMODEM, YMODEM, ZMODEM, KERMIT, TransIt, B-puls, QuickVanなど様々なプロトコルが使われており、プログラムやデータの送受信が行われている。このうちTransItはASCIIネット、B-pulsはNiftyServe, QuickVanはPC-VAN専用のプロトコルである。大学などにあるメインフレームとのファイル転送にはコロンビア大学で開発されたKERMITがよく使われているようである。KERMITはそれ自体が通信プログラムでもあるし、大きな特徴としてサーバー機能があるので、遠隔操作も可能である。本学ではこのKERMITがUTS上で利用できる。そこで、このKERMITの利用例と問題点についての検討を行う。FTPやFIMPORT, FEXPORTコマンドが使われるわけであるが、ファイル転送を行うためだけにわざわざセンターに出向いたり、部局に設置された端末まで出向くのは結構面倒である。さらにMS-DOSの5インチのフロッピーしか利用できないので、最近主流になっている3.5インチのフロッピーや私のようにMacintoshに乗り換えたユーザーは利用できないし、面倒でもMS-DOSの5インチに変換してからでないと思えないのである。

2. KERMITによるファイル転送³⁾

(1) KERMITプログラム

KERMIT (KL-10 Error Free Reciprocal Micro Interconnect over TTY lines) ファイル転送プロトコルは先にも述べたように、コロンビア大学で開発されたもので、それをサポートするプログラムもKERMITと呼ばれている。無料で配布されていることもあってパソコンからスーパーコンピュータまで各機種用専用のKERMITプログラムが作成されている^{4)・10)}。PC98, Macintosh用のプログラムはPC-VANのPDSからオンラインで、またPC98用のプログラムは九州大学大型計算機センターからオフラインあるいはオンラインで⁵⁾、長崎大学総合情報処理センターではオフラインで入手可能である。

(2) パソコンからUTSへの転送 (アップロード)

1. UTSに接続する
2. % kermit ; UTS側のKERMITを起動する。
3. C-Kermit> receive ファイル名 ; UTSで受信のための命令とファイル名
4. (ここでパソコン側のKERMITに切り替える。)
(PC98では、'CTRL'キーを押しながら']'キーを押した後、
'c'をタイプするとKermit-MS>のプロンプトが現れる。)
5. Kermit-MS> send ファイル名 ; PC98から送信の命令とファイル名。
(転送の様子が画面上に示される。)
(終了したら画面にKermit-MS>のプロンプトが現れる。)
6. Kermit-MS> c ; UTS側に移る。
(これでUTS側に移る)
7. リターンキー
8. C-Kermit> q ; UTS側のKERMITを終了する。
9. %

*Macintoshでは4-7の操作はFileメニューからSend fileを選択し、ダイアログに従って送信したいファイルとファイルタイプを指定するだけでよい。

(3) UTSからパソコンへ転送 (ダウンロード)

1. UTSに接続する
2. % kermit ; UTS側のKERMITを起動する.
3. C-Kermit> send ファイル名 ; UTSで送信のための命令とファイル名.
4. (ここでパソコン側のKERMITに切り替える.)
(PC98では、'CTRL'キーを押しながら']'キーを押した後、
'c'をタイプするとKermit-MS>のプロンプトが現れる.)
5. Kermit-MS> receive ファイル名 ; PC98から受信の命令とファイル名.
(転送の様子が画面上に示される。)
(終了したら画面にKermit-MS>のプロンプトが現れる.)
6. Kermit-MS> c ; UTS側に移る.
(これでUTS側に移る.)
7. リターンキー
8. C-Kermit> q ; UTS側のKERMITを終了する.
9. %

*Macintoshでは4-7の操作はFileメニューからReceive fileを選択し、ダイアログに従って受信したいファイル名とファイルタイプを指定するだけでよい。

(4) サーバー機能を利用したファイル転送⁴⁾・⁶⁾

KERMITのサーバー機能を利用すると、UTS側を外部記憶装置(ハードディスク)の感覚で操作することができる。UTS側がリモートホスト、パソコン側がローカルホストとなる。

1. UTSに接続する
2. % kermit
3. C-Kermit> server ; UTS側をサーバーモードにする。
4. (ここでパソコン側のKERMITに切り替える。)
(PC98では、'CTRL'キーを押しながら']'キーを押した後、
'c'をタイプするとKermit-MS>のプロンプトが現れる。)
5. Kermit-MS> get ファイル名 ; ファイルの受信
(Kermit-MS> send ファイル名 ; ファイルの送信)
6. Kermit-MS>finish
Kermit-MS> c
7. (これでUTS側に移る)
8. リターンキー
9. C-Kermit> q
10. %

*Macintoshでは4-8の操作はFileメニュー、remoteメニューから目的とするコマンドを選択し、ダイアログに従って操作するだけでよい。

*その他の主なコマンド

- | | |
|---------------|---------------------|
| delete | ; ローカルホストのファイル削除 |
| remote delete | ; リモートホストのファイル削除 |
| dir | ; ローカルホストのファイル一覧 |
| remote dir | ; リモートホストのファイル一覧 |
| cwd | ; ローカルホストのディレクトリー変更 |
| remote cwd | ; リモートホストのディレクトリー変更 |

(5) 転送速度について

KERMITプログラムは受信時のパケット長がデフォルトで90あるいは94となっているのでこのままだと転送にかなり時間がかかるので、KERMITを起動後にset コマンドで1000程度に設定するとよい。送信時のパケット長は受信側の受信速度に応じて自動的に調整されるので、設定する必要はない。

但し、デジタルフォンから接続している場合この設定値でも大丈夫であるが、通常のアナログ回線ではノイズが頻繁に入りやすく、Packet-Lengthを長くすることでかえって転送の速度が落ちたりすることがあるので注意が必要である。

* U T S では起動時に以下のオプションを付ける。

```
% kermit -e 1000
```

* P C 9 8 では、Kermitを起動したあと

```
Kermit-MS> set receive packet-length 1000
```

* M a c i n t o s h では、

settingsメニューからprotocolを選択し、Packet-Length を1000に設定し直す。

これらの設定を接続する度に行うのはかなり面倒であるが、KERMITプログラムでは、各種のパラメータを書き込んだファイルを起動時に読み込む機能があるので心配はいらない。

* U T S

ファイル名 .kermrc に以下の一行を入力してセーブしておく。

```
set receive packet-length 1000
```

* P C 9 8

ファイル名 MSKERMIT.INI に以下の行を入力してセーブしておく。

```
set receive packet-length 1000
```

```
set terminal kanji-code shift-jis ; (漢字が表示できるように  
する。)
```

* M a c i n t o s h では、

settingsメニューでPacket-Lengthを1000に設定したあとで、FileメニューからSave Settings を選びパラメータファイルを適当なファイル名でセーブする。そして、次回からは、パラメータをセーブしたファイルのアイ

コンをダブルクリックするだけでよい。

(6) かな漢字を含んだファイルの転送について^{1)・7)}

UTS側では自動的に漢字コードが判別されるので、パソコンで使われているシフトJISコードのまま送受信が出来る。ただし、UTSではEUCコードが標準であるので、日本語エディターjviを使ったり、MSPに再転送する場合にはシフトJIS→EUCの変換が必要である。かな漢字コード変換用のプログラム(nkf)はUTSにインストールされているので、UTS上で変換することができる。

```
% nkf -e < shift-jis file > EUC file ; シフトJIS→EUC
```

```
% nkf -s < EUC file > shift-jis file ; EUC→シフトJIS
```

また、PC98用のかな漢字変換プログラムは九州大学のPDSに⁵⁾、Macintosh用のプログラムはNiftyServeのPDSにあるので、ダウンロードして使用すればよい。

(7) バイナリーファイルの転送について⁸⁾

通常のテキストファイルは以上の方法で送受信可能であるが、バイナリーファイルの送受信をするには、以下の設定を行う必要がある。

* UTS側

```
% jstty -n
```

```
% kermi
```

```
C-Kermit> set file type binary
```

; この設定は.kermrcに書いておいてもよい。

* PC98

設定の必要なし

*Macintosh

送受信のダイアログでファイルタイプBinaryを選択する。

(MacBinaryファイルの転送はできないようである。)

3. 問題点など

(1) 転送速度について

K E R M I Tの最大の弱点は転送速度である。他のプロトコルが利用できないので具体的には比較はできないが、50Kを越すファイルの送受信では転送にかなり時間がかかる。U T S、M S P上で計算や印刷を行うためのファイル転送では、ある程度の転送時間と場合によっては漢字コードの変換が必要である。一方、P D Sとして登録したり、他の研究者とのデータの交換を行うのであれば、ファイルを圧縮して転送すれば転送時間がかなり節約できる。(九州大学からダウンロードしたP C 9 8用のK E R M I Tプログラムはドキュメントも含んでおり、実際のサイズは約105Kであるが、圧縮されているので約55Kと約半分になっている。)

Packet-length 以外にも転送効率をあげる方法がある。複数のパケットを連続して送るsliding windowやK E R M I Tプロトコルをサポートしたモデムを利用すればよいが、K E R M I Tのバージョンに依存している¹⁰⁾。残念ながら今回利用したバージョンのK E R M I Tではサポートされていなかった。

(2) M S Pへの転送について^{3), 9)}

U T SにはU T S < --- > M S P間のファイル転送を行うためのコマンド (ut ocp) が用意されている。U T Sに送信したファイルをM S Pへ転送する場合は、あらかじめM S P側でデータセットを作成しておかなければならない。また、M S Pで作成されるデータセット (拡張子がT E X Tの場合) のデフォルト属性は可変長で1レコード2 5 5バイト、ブロックサイズ3 1 2 0になっているので、テキストのサイズを一行2 5 1バイト以下にするか、作成するデータセットの1レコードを充分大きくしておく必要がある。

かな漢字ファイルの転送では注意が必要である。先に述べたようにパソコンから送信したシフトJ I SコードのファイルをU T S上でE U Cコードに変換してから、M S Pへ再転送する。また、M S PからU T Sに転送されたファイルのかな漢字コ

ードはEUCコードになっているので、パソコン側へ再転送する場合は、あらかじめUTS上でシフトJISコードに変換しておく必要がある。

(3) なぜKERMITを使うのか？

理由は簡単である。電話回線接続でホストを利用している研究室のパソコンから、面倒なフロッピーディスクの変換等をしないでプロトコル転送を行うにはUTSにインストールされたKERMITしかなかったからである。

パソコンからUTS、MSPにファイルを転送するもっとも簡単な方法は、エディットモードで入り、パソコン側の通信ソフトにある自動タイピング機能を使うことである。しかしながらこの方法では送受信のエラー検出ができないので、送受信の際にデータが化けたり、失われる危険性が非常に大きいのである（以前は転送したデータをANALYSTやSASで計算した単純集計の結果と、パソコン側で計算した結果とを比較してエラーの検出を行っていた）。また、バイナリーファイルの送受信もできない。ここで取り上げたKERMITや他のプロトコルではエラー検出、バイナリー転送とも可能であるから、データの送受信の信頼性に対して不安がなくなるのである。研究室のパソコンからUTS、MSPともにFTP、XMODEM、YMODEM、ZMODEM等のプロトコルでファイル転送が一日も早く使えるようになれば、その便利さは一層増大するであろう。

(4) その他

メインフレームで処理を行う以外のファイル転送の使い方としては、先にも触れたように学内あるいは学外の研究者とのデータ、プログラムの交換はもちろん、他の研究者の所有する印字品質の良いレーザープリンターに転送した文書等の出力を依頼することも可能であろう。'フロッピーディスクを持って行けばいい'と思われる方もあろうが、フロッピーのサイズや記録密度などの問題が生じるのでかなり面倒である。

4. おわりに

以上、KERMITを利用したファイル転送についてその利用方法と問題点について述べてきた。現在のところ上記の方法で問題なく転送できることを確認してい

るが、不都合な点などありましたら、ご連絡ください。電子メールでいずれかのアドレスにご連絡くだされば幸いです。

UTS :	f 0 2 0 1
MSP :	TAIMURA@KYOYO
PC-VAN :	REE74715
NiftyServe :	PGA00341

「KERMITという名前自身は、セサミストリートに出てくる蛙の名前」⁴⁾である。また、マウスの移動単位は”Mickey”と呼ばれており、両方ともその命名がユニークで楽しませてくれる。

本稿を作成するに当り、教養部の木村広先生、総合情報処理センターの内本佳彦先生には多大なるご指導をいただいた。紙面をお借りして、深く、感謝の意を表します。

参考文献

- 1) UTSとMS-DOS間のファイル転送について、
長崎大学総合情報処理センターニュース, No. 13, 4-6, 1989.
- 2) 修行 稔 : PC 98用端末エミュレータTSS.COM v. 4. 00,
長崎大学総合情報処理センターレポート, 第9号、36-49, 1990.
- 3) ファイル転送コマンドKERMITの公開について、
九州大学大型計算機センター広報, Vol. 23, No. 4, 416-417. 1990.
- 4) 村上健一郎 : KERMIT-ファイル転送プログラム,
Computer Today, Vol. 5, No. 1, 34-40, 1988.
- 5) 利用者提供通信ソフトのオンラインコピーサービス,
九州大学大型計算機センター広報, Vol. 23, No. 5, 547-551. 1990.
- 6) Frank da Cruz : KERMIT A File Transfer Protocol ,
Pp. 372, Digital Press, 1987.
- 7) UTSの利用について,
長崎大学総合情報処理センターニュース, No. 25, 6-13, 1990.
- 8) ファイル転送コマンドKERMITのレベルアップについて,
九州大学大型計算機センター広報, Vol. 23, No. 5, 543. 1990.
- 9) MSP-UTS間ファイル転送について,

長崎大学総合情報処理センターニュース, No. 15, 6-7, 1989.

- 10) Mark K. Murphy : Kermit 入門,
bit, Vol. 22, No.8, 4-15, 1990