

会計とパソコン通信

藤田 芳夫* 狩野 芳文** 林 毅***

パソコン通信を会計システムで、どう利用するか、殊に、東京情報大学のパソコン通信ネットワーク（Tuis ネットワーク）での応用方法を具体的に示すのが本稿の目的である。

そのため、Xmodemを使用する、いはば最も常識的な方法と、会計データ量が増加した場合、経済性を高めるため、会計データを圧縮して送受信するにはどうすればよいかという方法、及び第三に、公開されている通信ソフト「wterm」を利用して、会計パソコン通信を自動的に実行する方法の三つを、学生諸君が実習できるよう、資料と共に具体的に示したのである。

1 会計とパソコン通信

会計にとって企業組織は常に前提されており、企業で発生した会計データが会計担当部署へ送られる方法は常識的なものであり、ことさら詳細に論ずる必要はなかった。

しかし、コンピューター・ネットワークの現状は、会計システムにとって、LAN (Local Area Network) やパソコン通信の積極利用を可能にし、また必要なものになっている。そこで、本稿では、東京情報大学のパソコン通信ネットワーク（Tuis ネットワーク）を使用して、パソコン通信を会計へ応用する方法を明らかにすることにする。

2 会計記録と Xmodem 送受信

コンピューター・ネットワークを会計で利用しようとする、まず問題になるのは会計データをまとめてバッチ・ファイルとして送受信するか、個々の取引ごとにリアル・タイムで送受信するかという点であるが、パソコン通信を利用する場合には、まとめて処理するバッチ処理の方法が適当である。以下、本稿ではバッチ処理の場合について議論することにする。

パソコン通信の主流はテキスト・ファイルの送受信であるが、会計データはどんなに単純化しても、

(借方) 現金 100 (貸方) 売上 100

という形式か、またはこれに年月日、取引ナンバーが付いたレコード形式をとる。

```

type sw=record
    YMD:string [6];      ...年月日
    TNO:integer;          ...取引ナンバー
    TA:real;              ...取引金額
    ID, IC:integer;       ...借方、貸方勘定コード
end;

```

そこで、テキスト・ファイルの送受信というパソコン通信のもっとも一般的な方法ではなく、Xmodem 送受信という方法を利用する。この Xmodem 送受信は 1 ブロック (128 バイト) 単位で送受信を行なうため、ファイルの末尾に余りが発生し、そこへ 16 進の 1A がうめられる。この Xmodem 送受信に伴う末尾の処理のため、二つの方法がある。

第一の方法は末尾レコードであることを示す特殊なレコードを初めから付けて送信する方法であり、第二の方法は“ISH.COM”という非テキスト・ファイルを形式上テキスト・ファイルに見せかけて処理するユーティリティソフトを利用する方法である。

本稿では、まず、末尾に特殊なレコードを付けて、Xmodem で送信する方法を示し、次に、送受信すべき会計データの量が大きくなったとき送受信の時間を短縮し、能率をあげるとともに通信費用を低減するためのファイルの圧縮と復元の技法についてのべ、第三に、“Wterm”と言う通信ソフトを使用して、自動的に処理する方法を解説することにする。

3 末尾レコードを用意する方法

会計データをバッチ処理用のファイルとして送信するとき、ファイルの末尾に付けるレコードは、

1). 仕訳金額として、

```
Const endTA=-10.E34
```

を使用することにする。

2). さらに、支店または営業所から本社へ Xmodem で送信するデータ・ファイルは TLYYMM-DD.YET というファイル名を付けることにする。即ち

```
TL    ... パソコン通信用会計データ・
      ファイルを示す。
```

```
YY    ... データファイルの年
```

```
MM    ... データファイルの月
```

```
DD    ... データファイルの日
```

```
YET   ... 未処理を示すファイル拡張子
```

```
DON   ... 処理済みを示すファイル拡張子
```

である。例えば、91年の1月31日付けの未処理の会計データ・ファイルであれば、TL910131.YET のように年月日を付け、処理済みか否かを示すため、ファイル拡張子を未処理ファイルについては YET とし、処理済みのものについては TL910131.DON とするのである。

このように考えると、パソコン通信による会計データ処理プログラムのメインプログラムは次のようになる。

```

program teLMunit; { 91, 1, 16 }
uses teLglbv,      { グローバル変数の宣言 }
    teLutil,       { setptr, dspact, setswk }
    rdtrf,         { 仕訳ファイル、通信データ・ファイルの読込 }
    teLp;          { 通信データ・ファイルの処理 }
{メイン・プログラム - - - }
begin
  slist := nil;
  no := 1;
  write(' 本日の日付 - > ');
  readln(today);
  assign(fan, 'i:actname.dat'); { 勘定名ファイル読み込み }
  reset(fan);
  for act2 := noact to eigyohi do
    with actlist[act2] do
      begin
        read(fan, actn);
        Dptr := nil;
        Cptr := nil;
      end;
    close(fan);
  iptfile('i:pswkfl.dat'); { 過去データの読み込み }
                           { パソコン通信 受信データ・ファイルの }
  TeLsyori;                { 読みだし部分 }
  write(' 仕訳 - Y/N = > '); readln(yes); { 本日分取引データ入力 }
  while yes in ['Y', 'y'] do
    begin
      with xswk.sx do
        begin
          ymd := today;
          tno := no;
          write(' TA = > '); readln(TA);
          write(' ID = > '); readln(ID);
          write(' IC = > '); readln(IC);
          no := no + 1;
        end;
      setswk(xswk);
      write(' 仕訳 - Y/N = > '); readln(yes);
    end;
  end;

```

```

xptr := slist;
assign(fx, 'i:pswkfl.dat');
rewrite(fx);
while xptr <> nil do with xptr^, sx do
  begin
    write(fx, sx);
    writeln(ymd:6, tno:3, ta:7:0, ID:2, IC:2);
    xptr := sptr;
  end;
close(fx);

  writeln(' - - - - ');
write(' 勘定出力 Y/N=> '); readln(yes);
while yes in ['Y', 'y'] do
begin
  for act2 := genkin to eigyohi do
    begin
      write(actlist[act2].actn:17, 'I', ord(act2):2);
      if (ord(act2) mod 3) = 0 then writeln;
    end;
write(' act. no=> '); readln(no);
  with actlist[acts(no)] do
    begin
      xptr := Dptr; dspact(xptr, 'D');
      writeln(' - - - - ');
      xptr := Cptr; dspact(xptr, 'C');
    end;
  write(' 勘定出力 Y/N=> '); readln(yes);
end;
end.

```

グローバル変数の宣言をするユニットは次の通りである。

```
unit telglbv; { 91, 1, 16 }
interface
  const endTa = - 10.0e34;
  type
    str79 = string[79];
    str10 = string[10];
    acts = (noact, genkin, urikake, bihin,
            kaikake, shihon, uriage, shiire,
            kyuryo, eigyohi);
    ptr = ^swk;
    sw = record
      ymd : string[6];
      tno : integer;
      Ta : real;
      ID, IC : integer;
    end;
    swk = record
      sptr : ptr;
      sx : sw;
      nxtid, nextic : ptr;
    end;
    act = record
      actn : str10;
      Dptr, Cptr : ptr;
    end;
  var
    actlist : array[acts] of act;
    act2 : acts;
    slist,
    xptr : ptr;
    xswk : swk;
    xsw : sw;
    yes : char;
    today : string[6];
    no : integer;
    fan : file of str10;
    fx, frec : file of sw;
    frecname : str79;
implementation
end.
```

ポインターをセットしたり、勘定内容を表示したりする手続きを含むユーティリティユニット telutil は次の通りである。

```
unit telutil; { 91, 1, 16 }
interface
  uses teLglbv;
  procedure setptr(var p, q, r: ptr);
  procedure dspact(xptr: ptr; c: char);
  procedure setswk(xswk: swk);
implementation
  procedure setptr(var p, q, r: ptr);
  begin
    p:=q; q:=r;
  end;
  procedure dspact(xptr: ptr; c: char);
  var xswk: swk;
  begin
    xswk:=xptr^;
    while xptr<>nil do with xswk, sx do
      begin
        writeln(ymd:6, tno:3, TA:6:0, ID:2,IC:2);
        if(c='D')then xptr:=nxtid
          else xptr:=nxtic;
        xswk:=xptr^;
      end;
    end;
  end;
  procedure setswk(xswk: swk);
  begin
    xswk.nxtid:=nil;
    xswk.nxtic:=nil;
    new(xptr);
    xptr^:=xswk;
    setptr(xptr^.sptr, slist, xptr);
    with xptr^, sx do
      begin
        setptr(nxtid, actlist[acts(ID)].Dptr, xptr);
        setptr(nxtid, actlist[acts(IC)].Cptr, xptr);
      end;
    end;
  end;
end.
```

パソコン通信会計に直接かかわる部分は手続 Telsyori である。この手続きは unit telLp のなかに収納されている。

```
unit telLp; { 91, 1, 16 }
interface
  uses dos, tel2gLbv, rdtrf;
  procedure teLsyori;
implementation
  procedure teLsyori;
  var dirinfo: searchrec;
      body: string[8];
begin
  findfirst('i: TL *.yet', archive, dirinfo);
  if doserror=0
  then
    with dirinfo do
      begin
        iptfile('i: '+name);
        assign(fx, 'i: '+name);
        body:=copy(name, 1, pos('.', name) - 1);
        rename(fx, 'i: '+body+'.don')
      end
    else writeln('*** TeLyymmdd.yet ハナイ ***');
  end;
end.
```

この手続 Telsyori の中心部分は、まず、パソコン通信で受信した会計データファイルのなかに、未処理ファイル “TLYYMMDD.YET” が存在するか否かを、

```
findFirst('i: TL *.YET', - - - - -)
```

で探索している点である。未処理ファイルが存在している場合は DosError はゼロであるから、ファイル名 ‘name’ にドライブ指定子 ‘i:’ を付けてユニット rdtrf 中の会計データファイル処理ルーチン iptfile を呼んでいる。

```
iptfile('i: '+name);
```

会計処理ルーチンの内容はつぎに示すとおり。過去の仕訳データ・ファイルとまったく同じ処理をすればよい。

```
unit rdtrf; { 91, 1, 16 }
interface
  uses tel2glbv, tel2util;
  procedure iptfile(freename: str79);
implementation
  procedure iptfile;
  begin
```

```

writeln(frecname);
assign(fx, frecname); { 過去データ読み込み及び }
reset(fx);             { パソコン通信データの読み込み }
while (not eof(fx)) and (xsw.ta <> endTa) do
begin
  read(fx, xsw);
  with xsw do
    writeln(ymd:6, tno:3, ta:7:0, id:2, ic:2);
    xswk.sx := xsw;
    setswk(xswk);
  end;
  close(fx);
end;
end.

```

処理を終えると、再び Telsyori に戻るので、ファイル名の拡張子を

```

assign(fx, 'i:' + name);
body := copy(name, 1, pos('.', name) - 1);
rename(fx, 'i:' + body + '.don');

```

によって、処理済みファイル名の拡張子 'don' に付け替えるのである。

もし、未処理のパソコン通信会計ファイル TLYYMMDD.YET が無い場合には、その旨メッセージがでる。

4 会計データの圧縮と復元

パソコン通信を使って送受信する会計データの量が大きくなると、送受信の能率をあげ、同時に通信コストの低減を計る事が重要な問題になる。

送信に当たって、あらかじめデータ量を約50%強に圧縮し、受信するとき、圧縮されたデータを元通りに復元できる、LHARC と言うソフトウェアは、この点で極めて重要である。

例えば、第二節で示した単純な会計レコードは一レコード19バイトである。したがって10レコードの場合のファイル名を rtr10.dat とすると、その全長は190バイトになる。これを

```
A>LHARC A I:RTR10 I:RTR10.DAT
```

として実行すると、I:RTR10.LZH(161バイト) が出力される。これを

```
A>LHARC X I:RTR10.LZH I:
```

で復元処理すると、iドライブにもとの RTR10.DAT(190バイト) が復元される。会計データが100レコード、250レコード、500レコード、1000レコードと量が増大するにつれ、圧縮率は大きく変わらないが、データ量そのものは比例的に減少するので、経済効果は極めて大きくなる。

ファイル圧縮効果表

ファイル名	レコード数	圧縮前バイト数	圧縮後バイト数
RTR10. DAT	10	190	161
RTR100. DAT	100	1900	1102
RTR250. DAT	250	4750	2598
RTR500. DAT	500	9500	5113
RTR1000. DAT	1000	19000	10153

5 WTERM による会計データの送受信プログラム

本節ではユーティリティソフト“WTERM. EXE”と、ファイルサイズを50～60%程度圧縮し、又、それをもとのサイズに戻すこともできるユーティリティ“LHARC. EXE”、およびバイナリファイルとテキストファイルの相互変換を行う“ISH. COM”を利用し、会計データの受渡しを支店、営業所—本店ホストコンピューター間で行う時の作業を述べる。

通信ソフトを起動して任意のファイルをホストへ電子メールとして送信する場合には、ホスト側へダイヤルし、ID 番号を入力し、パスワードを入力する等の手順をたどる。これらの定形化した手順を自動的に行うマクロ機能が WTERM には備わっている。しかし、このマクロ命令群にはバイナリファイルを送信する機能がないので、自動送信を行う場合はバイナリファイルにテキスト化作業を施してから送信することになる。

マクロ機能を使用しない場合とする場合とでデータの送受信作業をわけてみる。

1) WTERM を手動で使う場合 (マクロ非使用時)

i) 支店、営業所から本店ホストコンピューターに送信する場合

流れ

作業	ファイル名	使用するソフト名
①会計データ 圧縮作業	: TLyymmdd. yet ↓ : TLyymmdd. lzh	LHARC. EXE
②ホスト登録	: TLyymmdd. yet	WTERM. EXE

解説

①データ圧縮

A>LHARC. EXE	a	b: TLyymmdd	b: TLyymmdd. yet
	圧縮指定 スイッチ	デスティネー ションファイル	ターゲットファイル

ファイルの圧縮を行う LHARC. EXE を起動する。上記書式の場合は a 命令により、b ドライブにある TLyymmdd. yet ファイルの圧縮を行っている。

この作業により原始会計データファイル 'TLyymmdd. yet' が圧縮され、'TLyymmdd. lzh' となる。(出来上がるファイル名は書式のデスティネーションで指定したファイル名に '.lzh' を足したものである。)

②ホストに登録

A>WTERM. EXE - t タイトル省略命令

圧縮作業により出来たファイル 'TLyymmdd. lzh' は WTERM により手元のパソコン (或いはワークステーション) とホストを接続し、電子メールにして XMODEM 手順により送信する。ホストに登録する時のファイル名は自由に指定できる。ここでは 'TLyymmdd. yet' とする。パソコン・ネットにアクセスし、メールを送信する手順は資料 A を参照されたい。

ii) データ入手の方法

このようにしてホストへ蓄えられたデータを入手し、会計プログラムで利用するには、登録時と逆の作業が必要となる。

一連のデータ入手作業は、既述の telswk 内に新設するユニット frntprcs で外部の WTERM というプログラムを exec 手続きにより実行するのである。(資料 B を参照の事)

frntprcs ユニット内部の流れを要約すると、次の四段階になる。

作	業	ファイル名	使用するソフト名
①	ファイル受信前処理		
②	ホストよりデータ入手	: TLyymmdd. yet	WTERM. EXE
	↓		
③	テキスト化解除	: TLyymmdd. lzh	ish. com
	(テキスト化ファイルがないので、ここでは自動的に省略)		
	↓		
④	圧縮解除	: TLyymmdd. yet	LHARC. EXE

以下、上記四段階の内容についてのべよう。

①ファイル前処理

ホストにあるデータを手元のパソコン (或いはワークステーション) にダウンする際、そのファイル名は自由に指定できる。ここでは登録時と同じファイル名 (TLyymmdd. yet) を使用している。ホストコンピューターからデータを入手する際に、同じファイル名が既にパソコン (或いはワークステーション) 側に存在すると思わぬトラブルを招く。そこでまず最初にファイル重複を処理する作業を行うのである。

前処理で行う作業

データ受信時に使用するファイル名として次のものがある。

TEMP.DOC : ホストよりデータを入手した時のファイル名 (マクロ使用時)

TLyymmdd.yet : 原始データファイル名

TLyymmdd.lzh : 圧縮化データファイル名

ファイル名 'TLyymmdd.yet' をホストより入手した後の、テキスト化解除 (マクロ使用時)、圧縮解除は手元のパソコン (或いはワークステーション) のファイル内部で展開される。故にパソコンのファイル名に既に同名のものが存在する場合は作業がスムーズに行われなくなる。そこで前処理として同名のファイルが存在すればこのファイル名を変更するという作業を行う。

このため、

'TEMP.DOC'、'TL *.YET'、'TL *.LZH' が既に存在するかどうか調べ、もし存在していれば削除する。

'TEMP.DOC' はマクロ非使用時には必要のないファイル名であるが、ここで削除する。

acdata [1] ← 'TL *.LZH'

acdata [2] ← 'TL *.YET'

acdata [3] ← 'TEMP.DOC' をそれぞれ代入する。

acdata [i] (i=1~3) について、該当するファイル名が既に同一ディレクトリ内に存在するかどうか調べ、もし存在していれば削除する。

前処理終了後、WTERM 等の外部プログラムを起動する前と後に swapvectors 手続を置く。これは本プログラム (telswk.EXE) で使用している割り込みベクトルをいったん待避させるためである。又、外部プログラム終了後、本プログラムに復帰するときにもこの手続を用いて待避させた割込ベクトルを戻す。

②ホストよりデータ入手

exec ('a:WTERM.EXE', '-t')	
起動する外部プログラム (含パス名)	引き渡すアーギュメント

exec 手続きによりドライブ A にある外部ユーティリティソフト WTERM.EXE を起動する。この時 -t によりオープニングタイトルの表示を省略している。

WTERM を起動してからの手順については資料 C を参照されたい。

③データテキスト化解除

WTERM を手動で使う時にはバイナリファイルをテキスト化せずにホストに登録してあるので、そのデータを入力する時はこの作業は不要となる。対象ファイルが非テキストファイルの場合、この作業は自動的にスキップされる。

④データ圧縮解除

exec ('a:LHARC.EXE', 'x b:TLyymmdd b:')	
起動する外部プログラム (含パス名)	渡すパラメータ (argument)

exec 手続により圧縮化／圧縮解除を行う LHARC.EXE を起動する。上記書式の場合は x 命令により、b ドライブにある TLyymmdd.lzh ファイル（最後 3 文字は命令では省略ができる。）の圧縮解除（解凍）を行っている。

この作業により圧縮ファイルが原始会計データファイル 'TLyymmdd.yet' に戻る。

2) WTERM 自動モード

次に、自動的に処理する場合の方法について述べよう。

資料 D に示すように、WTERM にマクロ命令群をセットしておくと、下に示すようにホスト接続、ホストとのデータ受渡し作業、回線切断までの四段階の作業を自動運転で行うことができる。

作 業	ファイル名	使用するソフト名
①原始会計データ	: TLyymmdd.yet	LHARC.EXE
↓		
②圧縮作業	: TLyymmdd.lzh	
↓		ish.com
③テキスト化作業	: TLyymmdd.ish	
↓		
④ホスト登録	: TEMP.DOC	WTERM.EXE

マクロ命令には XMODEM 方式でホストへ送信する機能が用意されていない。そこでバイナリファイルを一旦テキスト化する作業を③で行う。

①データ圧縮

WTERM 手動時に同じ

②データテキスト化

A>ish.com	b:TLyymmdd.lzh	/mb:	/S8
	ターゲット ファイル	格納先	テキスト化指定 スイッチ

テキスト化及びテキスト化解除を行う ish.com を起動する。上記書式の場合は b ドライブにある TLyymmdd.yet というファイルをテキスト化したファイル 'TLyymmdd.ish' を作成し、b にしよう、という命令になる。

③ホストに登録

```
A>WTERM -T +A1
```

タイトル省略
指定スイッチ

ダイヤルリストの1番目にある電話番号へ
自動接続する指令

上記作業により出来たファイル‘TLyymmdd.ish’をホストに登録する。ダイヤルリスト1番目にはホスト接続から回線切断までを自動的に行ってくれるマクロ命令群がセットしてある。

ホストに登録する時のファイル名は本来自由に指定できるが、マクロ命令では任意の日付をタイトルの一部に組み込むことができない。(TL910131.YET 等のように) そこで、ここでは‘TEMP.DOC’というタイトルでホストに登録する。

このようにしてホストへ蓄えられたデータを手し、会計プログラムで利用するには、登録時と逆の作業が必要となる。以下に流れを記す。

frntprcs ユニット内部の流れは WTERM 手動時と基本的に同じで、下のようになる。

①ファイル受信前処理

②ホストよりデータ入手 : TEMP.DOC WTERM.EXE

↓

③テキスト化解除 : TLyymmdd.lzh LHARC.EXE

↓

④圧縮解除 : TLyymmdd.yet ish.com

次に、各段階の内容に付いて概要をのべよう。

①ファイル前処理

WTERM 手動時と同じ

②ホストよりデータ入手

```
exec ('a:WTERM.EXE', '-t +a2')
```

起動する外部プログラム
(含パス名)

引き渡すアーギュメント

exec 手続きにより外部プログラムを起動する。この場合はドライブ a にある WTERM.EXE という通信ソフトを起動している。この時 -t によりオープニングタイトルの表示を省略、+a2 により WTERM 中にあるダイヤルリストの2番目に自動接続を指令している。

ダイヤルリスト2番目に登録されているマクロ命令(資料E)によりホストにアクセスし、‘TEMP.DOC’を手し、ホストとの回線を切断し、プログラムを終了する一連の作業を自動で行わせている。よって、この WTERM.EXE プログラムを終了し、本プログラムに復帰する時点で、手元のパソコン(或いはワークステーション)にはデータが存在する。

③データテキスト化解除

```
exec ('a:ish.com', 'b:TEMP.DOC /mb: /r')
```

起動する外部プログラム
(含パス名)

渡すパラメータ
(argument)

exec 手続によりテキスト化／テキスト化解除を行う ish.com を起動する。上記書式の場合は b ドライブにある TEMP.DOC というファイルのテキスト化を解除し、b にしまう、という命令になる。

この作業によりテキスト化する以前のファイル 'TLyymmdd.lzh' が吐き出される。

④データ圧縮解除

手動時に同じ

謝 辞

本稿は表記三者の共同研究の産物である。とくに四節は、林、五節は狩野の努力に負うところが大きい。

なお、パソコン通信のパブリック領域で公開されている下記三氏のソフトを使用した。三氏に謝意を表すものである。

WTERM.EXE…井上氏

LHARC.EXE…Haruyasu Yoshizaki 氏

ISH.COM………石塚匡哉氏

また、パソコン通信の特性として、学生諸君の演習にとり操作の手順を明示することが必要と考え、詳しい資料を添付した。

資料 A WTERM による送信手続

※” ”がついている場所が入力箇所

WTERMを起動したら、F・5を押し、ダイヤルリストを画面に出す。
その中の1番を選択するとダイヤリング開始、接続する。

user-ID ? tju01020 ID入力
password ? PW入力

東京情報ネットワークへようこそ!!

前回利用 91年 1月28日 17時48分 4秒

おはようございます !!

未読みのメールが 4 通あります。

Tuis-Net トップメニュー TOP

1. サービス内容紹介

2. 電子メール

3. 電子会議室

4. 電子掲示板

5. フォーラム

OFF.終了

> 2

メール(1.受信(全メール) 2.受信(未読み) 3.送信 4.消去 5.消去(全メール)
6.受信簿管理 7.発信簿管理 T.トップメニュー)

> 3

メッセージの作成方法は

1 エディタで作成 2 アップロード 3 XMODEM 4 キャビネット

番号を選択して下さい? 3

XMODEM 受信ファイルの準備ができました。 ファイルを送信してください。

ここでF・9を押し、D (XMODEM送信命令)を押し、ファイル名を入力する。

MODEM 受信を終了します。(882H"ト)受信しました。

宛先の指定方法は

1 直接指定 2 一覧選択 3 CUG選択 4 同報ファイル

番号を選択して下さい? 1

宛先識別子を入力して下さい? TJU00108

宛先識別子を入力して下さい? www

1 TJU00108

課題を入力して下さい? TL910129.YET

選択して下さい(R:実行, L:リスト, C:キャンセル, 番号)? R

受付番号 1572 で受付ました

メール(1.受信(全メール) 2.受信(未読み) 3.送信 4.消去 5.消去(全メール)
6.受信簿管理 7.発信簿管理 T.トップメニュー)

> OFF

利用状況をお知らせします。

接続開始時間 09時21分26秒

接続終了時間 09時23分09秒

利用時間 00時間01分43秒

またの御利用をお待ちしております。

ここでF・8 (終了命令) -Yを押し。とWTERMモードを抜ける。

資料 B WTERM を使用するプログラム frntpres

```
unit frntpres;
interface
uses dos;
type str12=string[12];
```

```

var
  acdata:array[1..3] of str12;
  i:integer;
implementation

procedure check1(letters:str12);
  var dirinfo:searchrec;
      fx:file;
  begin
    findfirst(letters,anyfile,dirinfo);
    while doserror = 0 do
      begin
        assign(fx,'b:'+dirinfo.name);
        erase(fx);
        findnext(dirinfo);
      end
    end;
  end;

begin
  acdata[1]:='b:¥TL*.LZH';           /* ファイル 前処理 */
  acdata[2]:='b:¥TL*.VET';
  acdata[3]:='b:¥TEMP.DOC';

  for i:=1 to 3 do
    check1(acdata[i]);

  swapvectors;
  exec('a:wterm.exe','-t +a16');     /* ホスト よりデータ入手 */
  swapvectors;
  exec('a:ish.com','b:TEMP.DOC /mb: /r'); /* テキスト化解除 */
  swapvectors;
  exec('a:larc.exe','x b:tl*.lzh b:'); /* 圧縮解除 */
  swapvectors;
end.

```

資料 C EXEC ('a:WTERM.EXE', '-t') 実行中の処理

※" _ " がついている場所が入力箇所

W T E R Mを起動したら、F・5を押し、ダイヤリストを画面に出す。
その中の1番を選択するとダイヤリング開始、接続する。

```

user-ID ? TJU00108 ..... I D入力
password ? ..... P W入力

```

東京情報大ネットワークへようこそ!!

前回利用 91年 1月29日 9時23分 9秒

おはようございます !!

未読みのメールが 5 通あります.

Tuis-Net トップメニュー TOP

1. サービス内容紹介

2. 電子メール

3. 電子会議室

4. 電子掲示板

5. フォーラム

OFF.終了

> 2

メール(1.受信(全メール) 2.受信(未読み) 3.送信 4.消去 5.消去(全メール)
6.受信簿管理 7.発信簿管理 T.トップメニュー)

> 2

発信者 :TJU01020

標題 :TL910128.VET

受信日 :91-01-29 09:23:03

選択して下さい(R:内容表示, N:次メッセージ, S:格納, X:XMODEM, E:終了) ? X
XMODEM ファイルの送信をします。ファイルを受信してください。

ここでF・9を押し、5(XMODEM受信命令)を押し、ファイル名を入力する。

MODEM 送信を終了します。

選択して下さい(R:内容表示, N:次メッセージ, S:格納, X:XMODEM, E:終了)? OFF
利用状況をお知らせします。

接続開始時間 09時23分56秒
接続終了時間 09時25分11秒
利用時間 00時間01分15秒

またの御利用をお待ちしております。

ここでF・8(終了命令)とYを押してWTERMモードを抜ける。

資料D <WTERM 自動実行バッチ>

```
START:
  SWITCH
    CASE("user-ID ?")
      SEND("¥$JU01020¥R") #ID入力
      GOTO(START)
    CASE("password ?")
      SEND("¥$-----¥R") #PW入力
      GOTO(START)
    CASE(">")
      SEND("¥$2¥R")
      GOTO(STP2)
  END

STP2:
  SWITCH
    CASE(">")
      SEND("¥$3¥R") #3. 送信を選ぶ
      GOTO(STP3)
    CASE("誤りがあります") #エラー処理
      GOTO(STP2)
  END

STP3:
  SWITCH
    CASE("下さい ?")
      SEND("¥$3¥R") #XMODEMを選択
      GOTO(STP3)
    CASE("ください。")
      UPLOAD(2,"B:¥TEMP.DOC") #B:TEMP.DOCを送信
      GOTO(STP3)
    CASE("指定方法は")
      GOTO(STP4)
    CASE("誤りがあります") #エラー処理
      GOTO(STP3)
  END

STP4:
  SWITCH
    CASE("選択して下さい ?")
      SEND("¥$1¥R") #直接IDを指定
      GOTO(STP4)
    CASE("入力して下さい ?")
      SEND("¥$JU00108¥R") #送信先ID入力
      GOTO(STP5)
    CASE("誤りがあります") #エラー処理
      GOTO(STP4)
  END

STP5:
  SWITCH
    CASE("入力して下さい ?")
      SEND("¥$¥R") #送信実行
      GOTO(STP6)
    CASE("誤りがあります") #エラー処理
      GOTO(STP5)
  END
```

```

STP6:
SWITCH
CASE("入力して下さい ?")
SEND("¥$データファイル¥R") #タイトル入力
GOTO(STP6)
CASE("誤りがあります") #エラー処理
GOTO(STP6)
CASE("番号) ?")
SEND("¥R¥R") #送信実行
GOTO(STP6)
CASE(">") #ログオフ
SEND("¥$OFF¥R")
EXIT
END

```

資料 E EXEC ('a:WTERM', '-t +a2') 用自動実行用マクロ

```

START:
SWITCH
CASE("user-ID ?")
SEND("¥$TJU010108¥R") #I D 入力
GOTO(START)
CASE("password ?")
send("¥$-----¥R") #P W 入力
GOTO(START)
CASE(">")
GOTO(STP2)
END

STP2:
SEND("¥$2¥R") #メールを選択

SWITCH
CASE(">")
SEND("¥$1¥R") #1. 受信(全メール)を選ぶ
GOTO(STP3)
CASE("誤りがあります") #エラー処理
GOTO(STP2)
END

STP3:
SWITCH
CASE("下さい ?")
SEND("¥$1¥R") #メール番号1を選択
GOTO(STP3)
CASE("終了) ?")
SEND("¥$R¥R") #内容表示を選択
GOTO(STP3)
CASE("<表示開始>")
LOGOPEN("B:¥temp.doc") #ログ開始
GOTO(STP4)
CASE("誤りがあります") #エラー処理
GOTO(STP3)
END

STP4:
SWITCH
CASE(" ---") #ログ終了(ISHファイルの最後を検出)
LOGCLOSE
GOTO(STP4)
CASE("終了) ?")
SEND("¥$OFF¥R")
EXIT
END
EXIT

```