

TITLE FTP - Chaos/TCP Gateway FTP

```
.lstof
.decsav
```

```
;16-Feb-84,GZ,v1 - released preliminary version
;17-Feb-84,GZ,v2 - polished up some more, page mode still doesn't work
;19-Feb-84,GZ,v3 - added a bunch of non-data comands (DELETE,RENAME, etc)
;25-Feb-84,GZ,v4 - filjfn wasn't getting 0'ed when released on reparse.
;29-Feb-84,GZ,v5 - cleaned up ^G and reply handling.
;30-Mar-84,GZ,v6 - removed ML from gateway list (only MC left, sigh)
;17-Jun-84,GZ,v7 - ML is back!
;23-Aug-84,GZ,v8 - Fetch the CLS packet for error message
;                      Default anonymous login in lowercase, for unix hosts.
;                      Print version # in decimal.
; 5-Feb-85,GZ,v9 - More room for hosts3
; 4-Mar-85,GZ,v10 - Type out version blurb first thing, before any errors
;                      can happen. Add CHANGE-PORTS option, on by default.
;                      Add a HELP command (but not by-topics yet).
;                      More complete handling of telnet negotiations.
;                      Don't consider telnet stuff as gratuitous replies.
;17-Mar-85,GZ,v11 - Accept internet host numbers. Fix up telnet stuff some
;                      more.
; 4-May-85,GZ,v12 - Better TAKE file eof handling. Allow ^G aborting out of
;                      subcommand parses. Add AI to gateways list (really should
;                      start accepting arbitrary gateways). Made page mode work
;                      and made it the default for tnx. (to do: don't flush data
;                      conn after each xfer, FDB preserve options). Made "MODE" in
;                      "SET MODE" optional, made status show mode in "local options
;                      instead of separate line.
; 2-Jul-85,GZ,v13 - more room for hosts3
vers==:13.                      ;Minor version #, should correspond to the
                                ;version of [MIT-OZ] SLOC:FTP.MID
```

```
if1,[
.insrt mid:hosts-format.mid      ;Host table definitions
NE%FMT==:17_32.
```

```
.insrt mid:chsdef.mid           ;Chaosnet definitions
```

```
.insrt mid:macsym.mid           ;Useful macros
```

```
define conc a1,a2,a3,a4,a5,a6,a7
a1!a2!a3!a4!a5!a6!a7!termin
```

```
define cmsg &msg
    call terpri
ifsn [msg] [] [tmsg msg]
termin
```

```
define jwarn &msg
    call [call $jwarn
        ifse [msg] [] 0
        .else [asciz msg]]
termin
```

```
define jerror &msg
    call [call $jerror
        ifse [msg] [] 0
        .else [asciz msg]]
termin
```

```
define fatal &msg
    jrst [call $fatal
        ifse [msg][0]
        .else [asciz msg]]
```

```
termin
```

```
define jfatal &msg
    call [call $jfatal
        ifse [msg][0] asciz "Impossible error"
        .else [asciz msg]]
```

```
termin
```

```
define noise &str
    movei 2, [flddb.(.cmnoi,,<440700,,[asciz str]>)]
    call $comnd
```

```
termin
```

```
];IF1
```

```
F==:0
```

```
f%echo==:0000000000001
f%conn==:0000000000002
f%take==:0000000000004
f%user==:0000000000010
f%pass==:0000000000020
f%acct==:0000000000040
f%rscn==:0000000000100
f%init==:0000000000200
f%dsk==:0000000000400
f%dcon==:0000000001000
f%cdis==:0000000002000
f%verb==:0000000004000
f%dbug==:0000000010000
f%alts==:0000000020000
f%nbl==:0000000040000
f.def==:f%alts
```

```
;Flags AC
```

```
;Echo is turned off
```

```
;Have a control connection
```

```
;Take file in progress
```

```
;Argb1 has user name (i.e. we are logging
```

```
;Argb1 has an unused password
```

```
;Argb2 has an unused account
```

```
;A RSCAN needs to be done
```

```
;TAKE'ing an init file
```

```
;Have an honest-to-goodness disk file
```

```
;Have a data connection (mostly for datmsg
```

```
;Display code, local in repdis
```

```
;print server replies
```

```
;print both sent commands and server repli
```

```
;Alternate ports
```

```
;Local in image mode getfil.
```

```
;Default flags
```

```
A=:5
```

```
B=:6
```

```
C=:7
```

```
D=:10
```

```
T=:11
```

```
TT=:12
```

```
;Reply code
```

```
;Transfer type during transfer
```

```
define defpg name,count=1
    name==:...pgc
    ..pgc==...pgc+count
```

```
termin
```

```
..pgc==50
```

```
;Start paged data on page 50
```

```
hstpgn==:300
defpg hstpag,hstpgn
hstbuf=:hstpag_9
```

```
;at most 300 host pages
```

```
;Map host table here
```

```
defpg filpag,2
filbuf=:filpag_9
```

```
;Window on file
```

```
defpg tmppag,2
tmpbuf=:tmppag_9
```

```
;Buffer for reading from net
```

```

pdllen==:50
    .vector pdl(pdllen)

strbsz==:100                ;COMND buffers
    .vector strbuf(strbsz),atmbuf(strbsz)

    .vector hstpmt(30)        ;Customized per-host prompt

    .vector pmtbuf(40)        ;Prompt for getlin et. al.

;Buffers for saving args to multi-arg commands.
    .vector argbf(strbsz),argbf1(strbsz),argbf2(strbsz)

    .scalar takjfn            ;Take file jfn
    .scalar filjfn            ;Data file jfn
    .scalar datjfn            ;Data connection jfn
    .scalar netjfn            ;Control connection jfn
    .scalar host              ;flags,,current host record if any
h%tnx==:1,,0                ;Host is running TNX
h%unx==:2,,0                ;Host is running Unix
h%vms==:4,,0                ;Host is running VMS
    .scalar hnmbuf(20)        ;Host current host name
    .scalar gate              ;Current gateway in lhs
    .scalar gateno            ;Gateway Internet host number
    .scalar xftype            ;Transfer type: 0=paged, 1=image,
; 7=ascii, 8=L 8, 36=L 36
    .scalar fxftype           ;Transfer type as understood by foreign h
    .scalar datcmd            ;Data command, for datopn
    .scalar datcnt            ;Count of data connection opens, for PORT
    .scalar abtpc             ;PC for ^G to jump to
    .scalar abtsp             ;SP for ^G to restore
    .vector chpkt(%cpmxw)     ;A chaos packet
    .scalar intpc             ;Interrupt PC

rskp:        aos (p)
r:           ret
retskp=:<jrst rskp>
call=:<pushj p,>              ;They are already defined, but as half-killed
ret=:<popj p,>                ;which I don't want

LEVTAB:      intpc
0
0

CHNTAB:      1,,abtint
repeat 35.,0

abtint:      exch 1,abtpc      ;^G comes here
ifn. 1                ;if enabled
    movem 1,intpc        ;new return address
    movei 1,.priou        ;Flush output
    CFOB%
    erjmp .+1
    move p,abtsp          ;restore stack
else.
    movei 1,^G            ;echo
    PBOU%
endif.
move 1,abtpc

```

```

        setzm abtpc
        DEBRK%

define setabt (addr)
        setzm abtpc                ;Avoid timing screws
        movem p,abtpc
        movei 1,addr
        movem 1,abtpc

termin

begin:      RESET%
        move p,[-pdllen,,pdl-1]
        movx f,f.def
        call terpri

radix 10.
        hrroi 1,[.ascii "Chaosnet/TCP gateway FTP, Preliminary bare-bones versi
Use ^G to abort.  Report bugs to BUG-FTP@MIT-OZ."]
radix 8.
        PSOUT%

        movx 1,gj%sht\gj%old
        hrroi 2,[asciz "SYSTEM:HOSTS3.BIN"]
        GTJFN%
        jfatal "Cannot find HOSTS3 file"
        movx 2,of%rd
        OPENF%
        jfatal
        move 2,[1,,.fbbyv]        ;Get page count
        movei 3,3
        GTFDB%
        movei 3,(3)
        caile 3,hstpgn
        fatal "HOSTS3 file too large, contact BUG-FTP@MIT-OZ"
        hrlzs 1
        move 2,[.fhslf,,hstpag]
        hrli 3,(pm%cnt\pm%cpy\pm%pld)
        PMAP%
        move 1,hstbuf+HSTSID
        came 1,[sixbit/HOSTS3/]
        fatal "Bad HOSTS3 file"

; Loop through the site table, marking all internet server sites with -1 in
; rhs of the STLFLG word.
        move T,hstbuf+SITPTR        ;Site table
        move TT,hstbuf(T)
        movei T,hstbuf+2(T)
        ifg. TT
        do.
            hllzs STLFLG(T)        ;Clear rhs
            move 2,STLFLG(T)
            tlnn 2,STFSRV        ;Server?
            ifskp.
                hrrz 1,STRADR(T)        ;Scan address list
                do.
                    jumpe 1,endlp.
                    move 2,hstbuf+ADDADR(1)
                    ifxe. 2,NE%FMT        ;Internet?
                    hllos STLFLG(T)        ;Yes, mark it
                    exit.                ;and done
                endif.
                hrrz 1,hstbuf+ADRCDR(1)

```



```

;;Field mask, for things like passwords, etc.
;; Allow everything but control chars, space and ?
fldmsk:      brmsk. filb0.,filb1.,filb2.,filb3.,[$!"#&'()+,/=@\_`{|}~]

confirm:      movei 2,[flddb. .cmcfm]
$comnd:      tdza 4,4
$comn0:      seto 4,
             movei 1,cmdblk
             COMND%
             erjmp [txne f,f%take
                   jumpn 4,cmdeof
                   jrst cmdjer]
             txne 1,cm%nop
             jrst cmderr
             movei 3,(3)                ;3/ function descriptor block
             ldb 4,[.bp cm%fnc,(3)]      ;4/ the function
             ret

txtddb:      flddb. .cmtxt,cm%sdh,,""

getlin:      movem 1,txtddb+.cmhlp
             hrroi 1,pmtbuf
             call cmdin0
getli1:      movei 2,txtddb
             call $comnd
             call echon
             callret confirm

getqli:      movem 1,txtddb+.cmhlp      ;getlin with echo off
             hrroi 1,pmtbuf
             call cmdin0
             call echoff
             jrst getli1

txtcmd:      movem 1,txtddb+.cmhlp      ;parse .cmtxt into argbf
             movei 2,txtddb
             call $comnd
             call svargb
             callret confirm

getjfn:      movem 1,cmdblk+.cmgjb
             hrroi 1,pmtbuf
             call cmdin0
             call pntfil                ;Flush jfn on reparse
             jwarn
             movei 2,[flddb. .cmfil]
             call $comnd
             movem 2,filjfn
             callret confirm

             .scalar savedp,savedr
cmdin0:      tdza 4,4
cmdini:      seto 4,
             pop p,savedr
             movem p,savedp              ;Save stack in case of reparse
             movem 1,cmdblk+.cmrty      ;Set up prompt
             movei 1,cmdblk              ;Issue the prompt
             movei 2,[flddb. .cmini]
             COMND%
             erjmp [txne f,f%take      ;Take file?

```

```

                jumpn 4,cmdeof ;Yes, if called as cmdini, process eof
                jrst cmdjerl   ;else random jsys error
repars:         move p,savedp   ;Restore stack to cmdini point
                push p,savedr
                ret             ;and return (as if) from cmdini

```

```

svargb:         movei 1,argbf
savarg:         movei 2,atmbuf
strcpy:         tlce 1,-1
                tlcn 1,-1
                hrli 1,(ascptr)
                tlce 2,-1
                tlcn 2,-1
                hrli 2,(ascptr)
do.
    ildb 3,2
    idpb 3,1
    jumpn 3,top.
enddo.
seto 3,
adjbp 3,1
move 1,3
ret

```

```

;Main command loop
cmddb:         flddb. .cmkey,,cmdtab,"command",,hstdb
hstdb:         flddb. .cmkey,,<0>,"name of host to connect to",,hbrodb
hbrodb:        flddb. .cmtok,cm%sdh,<ascptr lbrack>,,hnumdb
hnumdb:        flddb. .cmnum,cm%sdh,10., "Internet host number, n.n.n.n"

lbrack:        asciz "["
rbrack:        asciz "]"

cmdtab:        cmdlen,,cmdlen
[cm%inv\cm%fw ? asciz/BYE/],,.bye
[asciz/CONNECT/],,.conn
[asciz/CWD/],,.cwd
[asciz/DELETE/],,.delete
[asciz/DIRECTORY/],,.dir
[asciz/DISCONNECT/],,.disc           ;Same as .bye, just different noise
[asciz/EXIT/],,.exit
[asciz/GET/],,.get
[asciz/HELP/],,.help
[asciz/INFORMATION/],,.inform        ;Same as .status
[asciz/LIST/],,.list                 ;like .dir, but just lists names
[cm%fw\cm%abr\cm%inv ? asciz/LOG/],,.clogi
clogi::        [asciz/LOGIN/],,.login
[cm%fw\cm%inv ? asciz/LOGOUT/],,.disc
[cm%inv\cm%fw ? asciz/PUT/],,.send    ;For symmetry with GET
[cm%inv\cm%abr\cm%fw ? asciz/Q/],,.cquit
cquit::        [asciz/QUIT/],,.quit
[asciz/QUOTE/],,.quot
[asciz/RENAME/],,.renam
[asciz/SEND/],,.send
[asciz/SET/],,.set
[asciz/SOAK/],,.soak
[cm%inv\cm%fw ? asciz/STATUS/],,.status ;Same as .inform
[asciz/TAKE/],,.take
[asciz/TYPE/],,.typ

```

cmdlen==:.-cmdtab-1

```
cmdlup:      call gratis                      ;Check for gratuitous "replies"
             txzn f,f%rscn                    ;If need to hack jcl
             ifskp.
             movei 1,.rsini
             RSCAN%
             jfatal
             skipg 3,1
             anskp.
             movei 4,40                      ;Flush jname
             dmove 1,[.priin ? -1,,atmbuf]
             SIN%
             andg. 3
             hrroi 1,[asciz//]               ;Use no prompt
             else.                           ;Otherwise
             hrroi 1,[asciz "FTP>"]          ;default prompt
             txne f,f%conn                   ;but if connected,
             hrroi 1,hstpmt                 ; use host-name prompt
             endif.
             call cmdini                     ;Prompt
             call echon
             call pntfil                     ;Flush file jfn on reparse
             jwarn
             txz f,f%user\f%pass\f%acct\f%dcon ;Clear assorted flags
             movx 1,cm%sdh                   ;Don't list hosts in command line help
             iorm 1,hstddb+.cmflg
             movei 2,cmddb
             call $comn0
             cain 3,cmddb                   ;Get routine to call
             skipa 1,(2)                    ;From cmdtab
             movei 1,.host                  ;or lazy connect
             call (1)
             jrst cmder1
             jrst cmdlup

cmdeof:      movei 1,.fhself
             GETER%
             movei 2,(2)
             caie 2,iox4                    ;If error is EOF
             ifskp.
             ldb 1,[350700,,strbuf]          ;and nothing read on this command line
             ande. 1                        ;then terminate normally
             hrroi 1,[asciz "End of command file "]
             call takend
             txze f,f%init                   ;If was doing init file
             txo f,f%rscn                   ;time to do rscan now
             jrst cmdlup                    ;And continue normally
             endif.

cmdjer:      jerror
             jrst cmder1

cmderr:      call cerstr
             ;jrst cmder1

;All errors must end up here sooner or later
cmder1:      setzm abtpc
             move p,[-pdllen,,pdl-1]        ;Reset, since non-local error exits
             call echon                     ;Get echo back if was off
             ifxn. f,f%take                 ;Take file in progress?
```

```

lddb 1,[350700,,strbuf]          ;Anything to show?
ifn. 1
    cmsg "Line error occured in: " ;Yes, give some more context
    hrroi 1,strbuf
    PSOUT%
endif.
hrroi 1,[asciz "%Aborting command file "] ;And flush it
call takend
txz f,f%init                      ;No longer doing init file. Forget jcl t
endif.
movei 1,datjfn                    ;We never keep data conns around, flush i
call pntjfn
    jwarn
ifxe. f,f%conn
    movei 1,netjfn
    call pntjfn
    jwarn
endif.
seto 1,                            ;Flush file page in case still mapped
move 2,[.fhslf,,filpag]
setz 3,
PMAP%
call pntfil
    jwarn
jrst cmdlup

■
takgjb:      gj%old
    .nulio,,.nulio
    0                      ; No default device
    0                      ; or directory
    0                      ; or name
    440700,,[asciz/CMD/]    ; but default extension
repeat .gjatr-.gjpro+1,0    ; Nothing else interesting

.take:      noise "commands from"
    movei 1,takgjb
    movem 1,cmdblk+.cmgjb
    movei 2,[flddb. .cmfil]
    call $comnd
    movem 2,filjfn
    call confirm
    move 1,filjfn
    movx 2,fld(7,of%bsz)\of%rd
    OPENF%
    callret jerstr
ifxn. f,f%take                ;No nesting allowed
    hrroi 1,[asciz "Closing command file "] ;But ok to chain into next
    call takend                ;Note we leave f%init unchanged
endif.
move 1,filjfn
setzm filjfn
movem 1,takjfn
hrlm 1,cmdblk+.cmioj
movei 1,.nulio
hrrm 1,cmdblk+.cmioj
txo f,f%take
retskp

takend:      call terpri
    PSOUT%

```

```

movei 1,.priou
move 2,takjfn
movx 3,js%spc
JFNS%
movei 1,takjfn
call pntjfn
jwarn
move 1,[.priin,,.priou]
movem 1,cmdblk+.cmioj
txz f,f%take
ret

```

```

.help:      noise "with program"
call confrm
movx 1,gj%sht\gj%old
hrroi 2,[asciz "HLP:FTP.HLP"]
GTJFN%
jrst [jerror "Can't find HLP:FTP.HLP"
ret]
movem 1,filjfn
movx 2,fld(7,of%bsz)\of%rd
OPENF%
jrst [jerror "Can't open HLP:FTP.HLP"
ret]
do.
move 1,filjfn
hrroi 2,tmpbuf
movx 3,5000
setz 4,
SIN%
hrroi 1,tmpbuf
PSOUT%
jumpe 3,top.
enddo.
move 1,filjfn
CLOSF%
jwarn
setzm filjfn
retskp

```

```

settab:      setlen,,setlen
[asciz/CHANGE-PORTS/],,.salts
[asciz/DEBUG/],,.sdebug
[asciz/MODE/],,.smode
[asciz/NO/],,.sno
[asciz/VERBOSE/],,.sverb
setlen==:.-settab-1

```

```

notab:      nolen,,nolen
[asciz/CHANGE-PORTS/],,.nalts
[asciz/DEBUG/],,.ndbug
[asciz/VERBOSE/],,.nverb
nolen==:.-notab-1

```

```

.sno:      movei 2,[flddb. .cmkey,,notab]
skipa
.set:      movei 2,[flddb. .cmkey,,settab,,, [
flddb. .cmkey,,modtab,"MODE of transfers,"]]
call $comnd
caile 2,modtab

```

```

        caile 2,modtab+modlen
        skipa 1,(2)
        movei 1,smode
jrst (1)

.sdebug:      call confirm
              txo f,f%debug
              retskp
.ndbug:       call confirm
              txz f,f%debug
              retskp
.sverb:       call confirm
              txo f,f%verb
              retskp
.nverb:       call confirm
              txz f,f%debug\f%verb
              retskp
.salts:       noise "for data connections"
              call confirm
              txo f,f%alts
              retskp
.nalts:       noise "for data connections"
              call confirm
              txz f,f%alts
              retskp

modtab:       modlen,,modlen
c8bt::        [cm%abr\cm%inv\cm%fw ? asciz/8/],,c8bt
              [cm%inv\cm%fw ? asciz/8-BIT-BYTES/],,8.
              [cm%inv\cm%fw ? asciz/8BIT-BYTES/],,8.
              [asciz/ASCII/],,7.           ;A
              [asciz/EIGHT-BIT/],,8.       ;L 8
              [asciz/FULLWORD/],,36.       ;L 36
              [asciz/IMAGE/],,1.           ;I
              [asciz/PAGED/],,0.           ;Paged
modlen==:.-modtab-1

.smode:       noise "of transfers"
              movei 2,[flddb. .cmkey,,modtab]
              call $comnd
smode:        push p,(2)
              call confirm
              pop p,2
              hrrzm 2,xftype
              retskp

gwytab:       gwycnt,,gwycnt
              [asciz "AI"],,0
              [asciz "MC"],,0
              [asciz "ML"],,0
;             [asciz "XX"],,0
gwycnt==:.-gwytab-1

ftpsoc==:25
icpsoc:       ftpsoc                               ; (Patchable)

.conn:        noise "to host"
              movx 1,#cm%sdh                    ;For explicit CONNECT, list hostnames on
              andm 1,hstadb+.cmflg
              movei 2,hstadb

```

```

        call $comnd
.host:   caie 3,hstdb
        ifskp.
        hrro A,(2)
        else.
        setzb T,A
        caie 3,hbrodb
        ifskp.
        seto T,          ;Remember that have a bracket
        movei 2,[flddb. .cmnum,,10.]
        call $comnd
        endif.
        cail 2,0
        caile 2,200
        ifskp.
        repeat 3,[
            lsh 2,<3-.rpcnt>*8
            ior A,2
            movei 2,[flddb. .cmtok,,<-1,,[asciz "."]>]
            call $comnd
            movei 2,[flddb. .cmnum,,10.]
            call $comnd
            cail 2,0
            caile 2,200
        ]
        anskp.
        ior A,2
        else.
        emsg "Illegal host number"
        ret
        endif.
        ifn. T
        movei 2,[flddb. .cmtok,,<ascptr rbrack>]
        call $comnd
        endif.
    endif.
    push p,A          ;Save host entry or number
    noise "via gateway"
    movei 2,[flddb. .cmkey,,gwytab,"gateway host,","MC"]
    call $comnd
    push p,(2)        ;Save the gateway entry
    call confirm
    call bye          ;Say bye bye to current host
    nop
    pop p,gate
    setzm gateno      ;Don't know the internet number yet
    setom datcnt      ;No data connections yet
    pop p,A
    ifl. A
        movei A,(A)          ;Address of host record
        hrroi 1,hnmbuf      ;Copy official name
        hlro 2,STLNAM(A)
        addi 2,hstbuf
        call strcpy
        hlr 1,STLSYS(A)      ;Check out system type
        move 1,hstbuf(1)
        came 1,[ascii/TENEX/]
        camn 1,[ascii/TOPS-/]
        tlo A,(h%tnx)
        came 1,[ascii/TOPS1/]

```

```

    camn 1,[ascii/TOPS2/]
        tlo A,(h%tnx)
    camn 1,[ascii/FOONE/]
        tlo A,(h%tnx)
    camn 1,[asciz/VMS/]
        tlo A,(h%vms)
    camn 1,[asciz/UNIX/]
        tlo A,(h%unx)
else.
    move T,[401000,,A]
    hrroi 1,hnmbuf
    movei 3,10.
    do.
        ildb 2,T
        NOUT%
        jfatal
        tlnn T,770000
        exit.
        movei 2,".
        idpb 2,1
        loop.
    enddo.
    setz A,
endif.
movem A,host
movei 1,7.
movem 1,fxftyp
txne A,h%tnx
    movei 1,0
;
    movei 1,36.
movem 1,xftype
; Open the connection
    setabt [cmsg "%Connection attempt aborted"
        ret]
    hrroi 1,tmpbuf
    hrroi 2,[asciz "CHA:"]
    call strcpy
    hlro 2,gate
    call strcpy
    movei 2,".
    idpb 2,1
    setz 2,
    idpb 2,1
    movx 1,gj%sh
    hrroi 2,tmpbuf
    GTJFN%
        callret jerstr
    movem 1,netjfn
    move 1,[441000,,tmpbuf]
    hrroi 2,[asciz "TCP "]
    call strcpy
    hrroi 2,hnmbuf
    call strcpy
    movei 2,.chspc
    idpb 2,1
    move 2,icpsoc
    movei 3,8.
    NOUT%
    jfatal
    movei 3,-tmpbuf+1(1)
;Assume ascii mode
;That's what protocol default is
;Except for TNX
;we default to paged
;L36 for now, paged doesn't work yet
;Cons connect filename
;Get a jfn
;Chaosnet not available

```

```

imuli 3,4
lsh 1,-41
sub 3,1
hrli 3,(3)
hrri 3,tmpbuf
movx 2,fld(8,of%bsz)\fld(7,of%mod)\of%rd\of%wr      ;8 bit, do not ha
move 1,netjfn
OPENF%                                ;Send the rfc
    callret jerstr                        ;Net not up.
do.
    move 1,netjfn
    GDSTS%
    cain 2,%csopn                        ;Open?
    exit.                                ;Done then
    caie 2,%csrfs                        ;Still waiting?
    ifskp.
        movei 1,500.                    ;Wait a half sec
        DISMS%
        loop.                            ;and try again
    endif.
    callret netem1                        ;Type the error message as per state, lose
enddo.
setzm abtpc                            ;Got the connection.
call reply                              ;Get the 220. reply and display it
    ret                                  ;Connection refused or user-aborted
hrroi 1,hstpmnt                         ;Cons prompt
hlro 2,gate                             ;E.g. MC|SRI-NIC>...
call strcpy
movei 2,"|
idpb 2,1
hrroi 2,hnmbuf
call strcpy
hrroi 2,[asciz ">"]
call strcpy
txo f,f%conn                            ;We are now connected
cmsg "Assuming "
call xftdis
tmsg " transfers, do SET MODE to change transfer mode."
retskp

```

```

badhnm:    emsg "Illegal host number"
    ret

```

```

;;Here to asynchronously get a connection
getcon:    txne f,f%conn

```

```

    retskp
    ;Not implemented yet
nocon:    emsg "Not connected to any host"
    ret

```

```

.disc:    noise "from current host"
    jrst .bye1

```

```

.bye:    noise "to current host"

```

```

.bye1:    call confirm

```

```

txnn f,f%conn
callret nocon
call bye
abskip
call repdis
retskp

```

;;Logout and kill connection. Skips if there is a bye reply from foreign
;;host available for displaying.

```
bye:      txnn f,f%conn
          ret
          cmsg "Closing connection to host "
          hrroi 1,hnmbuf
          PSOUT%
          hrroi 1,[asciz/QUIT/]
          call outcmd
          call reply2
          abskip
          aos (p)
          txz f,f%conn
          movei 1,netjfn
          call pntjfn
          jwarn
          ret
```

```
.soak:      noise "a server response"
            call confirm
            txnn f,f%conn                ;Doesn't make sense to get a connection,
            callret nocon                ;since that requires reading replies..
soak:       call gtrep
            ret                          ;User abort
            call repdic
            retskp
```

```
.quot:      noise "server command"
            hrroi 1,[asciz "FTP protocol command"]
            call txtcmd
            call getcon
            ret
            hrroi 1,argbf
            call outcmd
            callret soak
```

```
.quit:      noise "FTP temporarily"
            call confirm
            movei 2,[.prkep,,0]          ;Keep fork
```

```
quit:       move 1,[.prast,,.fhslf]
            movei 3,1
            PRARG%
            call terpri
            HALTF%
            retskp
;           txnn f,f%conn
;           retskp
;           hrroi 1,[asciz/NOOP/]        ;Test for auto-logout
;           call outcmd
;           callret reply2
```

```
.exit:      noise "FTP permanently"
            call confirm
            call bye
            nop
            movei 2,[.prkil,,0]          ;Kill fork
            jrst quit
```

```
.renam:     noise "from"
```

```

hrroi 1,[asciz "old filename"]
call txtcmd
call getcon
ret
hrroi 1,[asciz/RNFR/]
hrroi 2,argbf
call sndcmd
movei A,350.
call reply3
ret
hrroi 1,pmtbuf
hrroi 2,[asciz "Rename to: "]
call strcpy
setabt [cmsg "%Command aborted"
ret]
hrroi 1,[asciz "new filename"]
call getlin
setzm abtpc
hrroi 1,[asciz/RNTO/]
hrroi 2,atmbuf
call sndcmd
callret reply ;250. reply

```

```

.delet:      noise "file"
hrroi 1,[asciz "foreign filename"]
call txtcmd
call getcon
ret
hrroi 1,[asciz/DELE/]
hrroi 2,argbf
call sndcmd
callret reply ;250. reply

```

```

.cwd:      noise "to directory"
hrroi 1,[asciz "foreign directory"]
call txtcmd
call getcon
ret
hrroi 1,[asciz/CWD/]
hrroi 2,argbf
call sndcmd
callret reply ;250. reply

```

```

.infor:      noise "about current session"
jrst .stat1

```

```

.statu:      noise "of current session"
.stat1:      call confirm
call terpri

```

```

radix 10.
hrroi 1,[.ascii "Chaosnet/TCP gateway FTP version 0.!vers
Local options: "]
radix 8.

```

```

PSOUT%
ifxe. f,f%debug
tmsg "NO "
endif.
tmsg "DEBUG, "
ifxe. f,f%verb
tmsg "NO "

```

```

endif.
tmsg "VERBOSE, "
ifxe. f,f%alts
    tmsg "NO "
endif.
tmsg "CHANGE-PORTS"
ifxn. f,f%conn
    tmsg ", MODE "
    call xftdis
endif.
tmsg "

ifxn. f,f%take
    tmsg "Taking commands from "
    movei 1,.priou
    move 2,takjfn
    movx 3,js%spc
    JFNS%
    tmsg "

endif.
ifxe. f,f%conn
    tmsg "Not connected to any host"
    retskp
endif.
tmsg "Connected to "
hrroi 1,hnmbuf
PSOUT%
tmsg " via "
hlro 1,gate
PSOUT%
tmsg "
Foreign host status:"
hrroi 1,[asciz/STAT/]
call outcmd
call gtrep
    ret
call repdis
retskp

```

```

;Should be 211 but don't rely on it...
;User abort

```

```

unmdb:      fldbk. .cmfld,cm%sdh,, "User name",,fldmsk,unmdb1
;This is just for help, since .cmfld will parse a null field
unmdb1:     flddb. .cmcfm,cm%sdh,, "confirm for ANONYMOUS login",,0

pasdb:      fldbk. .cmfld,,, "password",,fldmsk

actdb:      fldbk. .cmfld,,, "account",,fldmsk

.login:     noise "as user"
    movei 2,unmdb
    call $comnd
    ldb 1,[350700,,atmbuf]
    ife. 1
        dmove 1,[asciz/anonymous/]
        dmovem 1,argbf
        dmove 1,[asciz/guest/]
        dmovem 1,argbfl
        txo f,f%pass
        jrst login1
    endif.

```

```

call svargb
call echoff
noise "password"
movei 2,pasdb
call $comnd
call echon
ldb 1,[350700,,atmbuf]
jumpe 1,login1
hrroi 1,argbf1
call savarg
txo f,f%pass
noise "account"
movei 2,actdb
call $comnd
ldb 1,[350700,,atmbuf]
jumpe 1,login1
hrroi 1,argbf2
call savarg
txo f,f%acct
login1:      call confirm
txo f,f%user
call getcon      ;Make sure we're connected
ret
hrroi 1,[asciz/USER/]
hrroi 2,argbf
call sndcmd
call reply      ;Show 230 greeting
ret
ifxn. f,f%acct      ;If still have an account sitting ar 'nd
hrroi 1,[asciz/ACCT/]      ;send it off, maybe it means somet. g
hrroi 2,argbf2
call sndcmd
call reply2      ;Get 230 response
ret
endif.
retskp

ingjb:      gj%old ;;+gj%ifg ;No wildcarding yet
.nulio,,.nulio
repeat .gjatr-.gjdev+1,0

.send:      noise "local file"
move D,xftype      ;Transfer type (in future, allow /switches
movei 1,ingjb      ; to one-shot modify default)
movem 1,cmdblk+.cmgjb
movei 2,[flddb. .cmifi]
call $comnd
movem 2,filjfn
noise "to foreign file"
movei 2,[flddb. .cmcfm,cm%sdh,,,[
flddb. .cmtxt,cm%sdh,,"output filename"]]
call $comnd
cain 4,.cmcfm
ifskp.
call svargb
call confirm
call getcon
ret
else.

```

```

call getcon
ret
hrroi 1,pmtbuf
hrroi 2,[asciz "To foreign file: "]
call strcpy
;;default hackery would go here
setabt [cmsg "%Command aborted"
ret]
hrroi 1,[asciz "Foreign filename"]
call getlin ;Get name into atmbuf
call svargb
setzm abtpc
endif.
movx 2,of%rd
call filopn ;Open file, set f%dsk, check out D.
ret
move 1,[asciz/STOR/]
movem 1,datcmd
call datopn
ret
callret sndfil

```

;;; Data transfer sending routines

```

fdblen==:.fbusw+1 ;Just send 0 thru .fbusw
.vector fdb(fdblen)

;Page mode header
.vector pghdr(5+1) ;1 extra for overflow on read...
pghlen==:0 ;header size (4 or 5)
pgindx==:1 ;Page number
pgdlen==:2 ;data size
pgtype==:3 ;Type of page
pt.eof==:0
pt.dat==:1
pt.fdb==:2
pt.acs==:3
pt.max==:3
pgaccs==:4 ;Access control

```

```

sndfil: setabt xfrabt ;Set up ^G to abort
txz f,f%nbl ;No nibbles yet
movei A,44 ;Byte count of a page
idivi A,(D)
imuli A,1000
move B,[444400,,filbuf] ;Canonical pointer
cain D,7
hrli B,440700
cain D,8
hrli B,441000
txne f,f%dsk ;Disk file?
jrst snddsk ;Use mapping instead of SINing
do. ;Else just simple loop
move 1,filjfn ;Read a page
move 2,B
movn 3,A
SIN%
move T,A
add T,3 ;T/ file-bytes read
jumpe T,endlp. ;Didn't read nothing, done

```

```

    move TT,B                ;Point to buffer
    call sndbuf              ;Send it off
    loop.
enddo.
move 1,filjfn
GTSTS%
jxn 2,gs%eof,sndeof        ;If end of file, won
jrst filerr                 ;Else there must have been an error on the S

;Send data from disk to net. D has 0/7/8/36 as per transfer mode.
snddsk:      hrrz 1,filjfn      ;Prefetch the fdb
movsi 2,fdblen
movei 3,fdb
GTFDB%
ife. D                ;If page mode
    movei 1,5.          ;Initialize header
    movem 1,pghdr+pghlen
    movei 1,pt.acs
    movem 1,pghdr+pgtype
else.                ;Else compute byte total byte count in C
    skipg C,fdb+.fbsiz    ;Claimed count
    jrst sndeof           ;Hmm, empty file
    ldb 1,[.bp fb%bsz,fdb+.fbbyv] ;Check if right byte size
    skipn 1
    movei 1,36.
    camn 1,D              ;Sigh, gotta recompute
    ifskp.
        movei 2,36.
        idivi 2,(1)        ;bytes per word
        exch 2,C
        addi 2,-1(C)        ;round up
        idivi 2,(C)        ;number of words
        move C,2
        movei 1,44.        ;our bytes per word
        idivi 1,(D)
        imuli C,(1)        ;A=total number of bytes
    endif.
endif.
setzm pghdr+pgindx        ;Initialize page counter
do.                        ;Loop over pages
    move 1,pghdr+pgindx
    hrl 1,filjfn
    RPACS%
    ifxe. 2,pa%pex        ;Page doesn't exit
        FFUFP%           ; so find next one
        jrst sndeop
        hrrzm 1,pghdr+pgindx ;Save
        RPACS%
    endif.
    movem 2,pghdr+pgaccs    ;Save access bits in case page mode
    move 2,[.fhs1f,,filpag] ;Map the page
    movx 3,pm%rd
    PMAP%
    erjmp filerr
    ife. D                ;If page mode,
        movei 1,1000        ;Sending 1000 bytes of data
        skipn filbuf-1(1)    ;except don't bother with trailing zeros
        sojg 1,.-1
        movem 1,pghdr+pgdlen
        movei 1,pghdr

```

```

        call sndhdr
    else.
        move T,A
        camge C,T
        move T,C
        sub C,T
        move TT,B
        call sndbuf
    endif.
    aos pghdr+pgindx
    jumpe D,top.
    jumpg C,top.
enddo.
sndeof:    move 1,datjfn
ifxn. f,f%nbl
    move 2,nibble
    lsh 2,4
    BOUT%
    erjmp xfrerr
endif.
movei 2,.monop
MTOPR%
    erjmp xfrerr
CLOSF%
    jwarn
setzm datjfn
setzm abtpc
seto 1,
move 2,[.fhslf,,filpag]
setz 3,
PMAP%
move 1,filjfn
setzm filjfn
CLOSF%
    jwarn
callret reply

;Here if FFUFP errs out
sndeop:    cain 1,FFUF3
ifn. D
    call jerstr
    jrst xfrabl
endif.
seto 1,
move 2,[.fhslf,,filpag]
setz 3,
PMAP%
move 1,[fdb,,filbuf]
blt 1,filbuf+fdblen-1
movei 1,[4 ? 0 ? fdblen ? pt.fdb]
call sndhdr
movei 1,[4 ? 0 ? 0 ? pt.eof]
call sndhdr
jrst sndeof

sndhdr:    move T,pghlen(1)
move TT,1
push p,pgdlen(1)
call sndbuf
pop p,T

;else stream
;A page worth
;Unless don't have a full page
;Update remaining count
;Buffer pointer

;Next page

;Send orphan nibble

;Clear the page

;And close the file

;Get the 226. reply and done

;random error?
;Or not in page mode?
;Show error
;and abort transfer

;Clear the page

;send the FDB

;And an eof packet

;And done

```

```

        movei TT,filbuf
sndbuf:   jumpe T,r
        caie D,0
        cain D,36.
        call sndcv1
        move 1,datjfn
        move 2,TT
        movn 3,T
        SOUT%
        erjmp xfrerr
        ret

```

```
;;T/word count, TT/buffer address
```

```

sndcv1:   move 3,TT
        movn 1,T
        hrl 3,1
        movei 4,tmpbuf
        ifxn. f,f%nbl
        soj 4,
        sub 3,[1,,1]
        move 1,nibble
        move 2,1(3)
        jrst sndcv1
        endif.
do.
        dmove 1,(3)
        movem 1,(4)
sndcv1:   lshc 1,8*4
        movem 1,1(4)
        repeat 6,[
            aobjp 3,endlp.
            lshc 1,<1+.rpcnt>*4
            move 2,1(3)
            lshc 1,<7-.rpcnt>*4
            movem 1,<2+.rpcnt>(4)
        ]
        aobjp 3,endlp.
        lshc 1,8*4
        movem 1,8(4)
        addi 4,9.
        aobjn 3,top.
        enddo.
        imuli T,9
        txze f,f%nbl
        addi T,1.
        ifxn. T,1
        move 2,T
        andi 2,7
        imuli 2,4.
        rot 1,(2)
        movem 1,nibble
        txo f,f%nbl
        endif.
        lsh T,-1
        move TT,[441000,,tmpbuf]
        ret

```

```

;3/aobjn pointer to input
;4/output address
;If have a nibble
;Make like had a word before

```

```
;And jump into loop
```

```
;Number of nibbles
```

```
;If odd
```

```
;Number of nibbles in last word
```

```
;Rotate the left-out nibble to rhs
```

```
;Return number of bytes
```

```
;And pointer
```

```

outgjb:   gj%fou
        .nulio,,.nulio
        0 ? 0

```

```
;No default device/dir
```

```

0 ;No default name, but could fill in
0 ;No default type but could fill in
repeat .gjatr-.gjpro+1,0

;;Mask for setting the fdb in page mode
fdbmsk: 0 ;.fbhdr, unchangable
fb%fcf ;.fbctl (also %tmp,%del,%prm,%nod,%inv,%n
0 ;.fbexl, unchangable
0 ;.fbadr, unchangable
0 ;.fbprt (rhs only)
0 ;.fbcre
0 ;.fbaut, unchangable
0 ;.fbgen, unchangable
0 ;.fbact, unchangable
fb%bsz\fb%mod ;.fbbyv (also fb%ret?)
-1 ;.fbsiz
0 ;.fbcrv, creation date
0 ;.fbwrt, last-write date
0 ;.fbref, last-reference date
0 ;.fbcnt, writes,,reads
0 ? 0 ? 0 ;.fbbk0, .fbbk1, .fbbk2, dumper stuff
0 ;.fbbbt (ar%rar,ar%riv,ar%ndl,ar%nar,ar%exm..)
0 ;.fbnet, online expiration date
0 ;.fbusw

fdblen==:.-fdbmsk

```

```

.get: noise "foreign file"
move D,xftype ;Default xfer type
hrroi 1,[asciz "name of foreign file"]
call txtcmd
call getcon
ret
move 1,[asciz/RETR/]
movem 1,datcmd
hrroi 1,pmtbuf
hrroi 2,[asciz "To local file: "]
call strcpy
;;default hackery goes here
setabt [cmsg "%Command aborted"
ret]
movei 1,outgjb
call getjfn
setzm abtpc

$get: movx 2,of%wr
call filopn ;Open file, set f%dsk, check out D.
ret
call datopn
ret
move 1,filjfn
call getfil
ret
move 1,filjfn ;Close the file, really creating it
setzm filjfn
CLOSF%
jwarn
retskp

```

```

.typ: noise "foreign file"
hrroi 1,[asciz "name of foreign file"]
call txtcmd

```

```

call getcon
ret
move 1,[asciz/RETR/]
movem 1,datcmd
txz f,f%dsk
movei D,7
call datopn
ret
movei 1,.priou
callret getfil

```

```

.list:      skipa 1,[asciz/NLST/]
.dir:      move 1,[asciz/LIST/]
movem 1,datcmd
noise "of files"
movei 2,[fldbk. .cmfld,cm%sdh,, "foreign directory name",,fldmsk]
call $comnd
call svargb
noise "output to"
movei 1,outgjb
movem 1,cmdblk+.cmgjb
movei 2,[flddb. .cmfil,,,, "TTY:"]
call $comnd
movem 2,filjfn
call confirm
call getcon
ret
movei D,7
callret $get

```

;; Data transfer get subroutines

```

getfil: .scalar outjfn          ;Not filjfn, in case .priou
        movem 1,outjfn          ;Save the output pointer
setabt xfrabt          ;Set up ^G interrupt
txz f,f%nbl           ;No lose nibble yet
jumpe D,getpag         ;Paged transfer is different
do.
    movei TT,9*400       ;Good sized buffer (div. by 9, in case image)
    call getbuf
    cain 1,iox4          ;end of file?
    abskp
    jrst xfrerr          ;No, random error
    jumpe TT,geteof
    caie D,36.
    ifskp.
        move T,[444400,,filbuf] ;Convert to fullword
        call getimg
    endif.
    move 1,outjfn
    move 2,T
    movn 3,TT
    SOUT%
    erjmp filerr
    movem 1,outjfn      ;In case ever want to read into memory...
    loop.
enddo.

```

```

;; Paged retrieval
getpag:      seto 1,

```

```

move 2,[.fhslf,,filpag]
setz 3,
PMAP%
setzm fdb+.fbhdr          ;No fdb yet
do.
    movei T,pghdr
    movei TT,4
    call getibf
    jrst xfrerr
    skipl A,pghdr+pgtype
    caie A,pt.max
    jrst [move 2,A ? hrroi 1,[asciz "Bad page type"] ? jrst fmterr]
    move 2,pghdr+pghlen
    caie A,pt.acs
    caie 2,4.
    ifnsk.
        caie A,pt.dat
        caie 2,5.
        jrst [hrroi 1,[asciz "Bad header length"] ? jrst fmterr]
    endif.
    caie 2,5.
    ifskp.
        movei T,pghdr+pgaccs
        movei TT,1
        call getibf
    endif.
    setzm filbuf          ;Create page
    skipl TT,pghdr+pgdlen
    caie TT,1000
    jrst [move 2,TT ? hrroi 1,[asciz "Bad data length"] ? jrst fmterr]
    movei T,filbuf
    call getibf
    jrst xfrerr
    cain A,pt.eof
    exit.
    cain A,pt.fdb
    ifskp.                ;Normal page
        move 1,[.fhslf,,filpag]
        move 2,pghdr+pgindx
        hrl 2,outjfn
        movx 3,pm%wr
        PMAP%
        erjmp filerr
        caie A,pt.acs
        ifskp.
            move 2,pghdr+pgaccs
            andi 2,pa%rd\pa%wt\pa%ex\pa%cpy
            move 1,pghdr+pgindx
            hrl 1,outjfn
            SPACS%
            erjmp [jwarn "Can't set page access"
                    jrst .+1]
        endif.
    else.                ;Else FDB page
        move 1,[filbuf,,fdb]          ;Save it
        blt 1,fdb+fdblen-1
        hrrz 1,fdb+.fbhdr          ;And save length
        camle 1,pghdr+pgdlen
        move 1,pghdr+pgdlen
        caie 1,fdblen

```

```

        movei 1,fdblen
        movem 1,fdb+.fbhdr
        seto 1,                ;Flush page
        move 2,[.fhslf,,filpag]
        setz 3,
        PMAP%
    endif.
    loop.
enddo.
skipe 2,pghdr+pgdlen
    jrst [hrroi 1,[asciz "Bad data length on eof packet"] ? jrst fmterr]

movn 1,fdb+.fbhdr            ;Really want to do this AFTER close file
hrlz T,1
ifl. T
do.
    skipn 2,fdbmsk(T)
    ifskp.
        move 1,outjfn
        hrli 1,<(cf%nud)>(T)
        move 3,fdb(T)
        CHFDB%
        erjmp [jwarn "%Can't modify fdb word"
                jrst .+1]
    endif.
    aobjn T,top.
enddo.
endif.
geteof:    movei 1,datjfn                ;Flush our side of connector 'oth
call pntjfn                ;too)
    jwarn
    setzm abtpc
    callret reply                ;Get 226 reply

fmterr:    setzm abtpc
    ESOUT%
    hrroi 1,[asciz " ("]
    PSOUT%
    movei 1,.priou
    movei 3,10.
    NOUT%
    nop
    movei 1,")
    PBOUT%
    jrst xfrab1

;;Read TT bytes into tmpbuf. Exits with tmpbuf bp in T
getbuf:    move T,[441000,,tmpbuf]
    cain D,7
    hrli T,(ascptr)
    skipg 1,TT                ;See how many words will read
    retskp
    caie D,7
    idivi 1,4.
    cain D,7
    idivi 1,5.
    setzm tmpbuf                ;and clear them
    move 2,[tmpbuf,,tmpbuf+1]
    blt 2,tmpbuf+1(1)          ;One extra for good measure
    move 1,datjfn

```

```

move 2,T
movn 3,TT
SIN%
    erjmp [add TT,3
           movei 1,.fhs1f
           GETER%
           movei 1,(2)
           ret]
jumpe 3,rskp
fatal "Impossible return from SIN%"

```

.scalar nibble

```

getimg:      call getcvt
txze f,f%nbl
    skipn 1,nibble
    ret
aos 2,TT
addi 2,-1(T)      ;Count the nibble if it's non-zero
movem 1,(2)
ret

```

```

;;Read buffer and image convert. T=output address, TT=# words
getibf:      push p,T
imuli TT,9      ;# nibbles
txnn f,f%nbl
    addi TT,1
lsh TT,-1      ;# bytes
call getbuf
    jrst [pop p,T ? ret]
pop p,T
call getcvt
retskp

```

```

;;Do image conversion from tmpbuf. T has output address, TT has input byte
;;count, f%nbl on if Nibble has leftover nibble from last time.
;;Exits with TT=word count, T=fullword output bp.
;;Turns on f%nbl if only one nibble in last word, and doesn't count the word
getcvt:      skipg 1,TT

```

```

    ret
movei 3,tmpbuf
movei 4,(T)
setz TT,
txzn f,f%nbl
ifskp.
    move 2,(3)
    aoj 3,
    lsh 2,-4
    ior 2,Nibble
    movem 2,(4)
    aoj 4,
    aoj TT,
    subi 1,4.
    jumple 1,r
endif.
imuli 1,2      ;Number of input nibbles
addi 1,8.      ;Convert to output words
idivi 1,9.
skipn 2
txo f,f%nbl    ;Remember whether orphan nibble in last

```

```

add TT,1
movns 1
hrl 4,1
do.
    dmove 1,(3)
    lsh 1,-4
    repeat 7,[
        lshc 1,<1+.rpcnt>*4
        movem 1,(4)
        aobjp 4,endlp.
        lshc 1,<7-.rpcnt>*4
        move 2,.rpcnt+2(3)
    ]
    lshc 1,8*4
    movem 1,(4)
    addi 3,9.
    aobjn 4,top.
enddo.
movem 1,nibble
txne f,f%nbl
soj TT,
ret
;Make output aobjn pointer
;Don't count the nibble

```

```

;;Open file, 2 has open bits, D transfer mode.
filopn:    move 1,filjfn
    caie D,1
    dpb D,[.bp of%bsz,2]
    OPENF%
    callret jerstr
    DVCHR%
    ldb 2,[.bp dv%typ,2]
    cain 2,.dvdisk
    txoa f,f%dsk
    txz f,f%dsk
;    caie 2,.dvtty
;    cain 2,.dvpty
;    txoa f,f%tty
;    txz f,f%tty
    txnn f,f%dsk
    jumpe D,[emsg "PAGED transfers are for disk devices only"
        ret]
    retskp
;Set device flags

```

```

xftdis:    move 2,xftype
    cain 2,0
    hrroi 1,[asciz "PAGED"]
    cain 2,1
    hrroi 1,[asciz "IMAGE"]
    cain 2,7
    hrroi 1,[asciz "ASCII"]
    cain 2,8
    hrroi 1,[asciz "EIGHT-BIT"]
    cain 2,36.
    hrroi 1,[asciz "FULLWORD"]
    PSOUT%
    ret

```

```

xftsnd:    move T,fxftyp
    camn D,T
    retskp
;Already in right mode

```

```

ife. T                                ;If was paged, must want file now
    hrroi 1,[asciz/STRU F/]
    call outcmd
    call reply2                        ;200.
    ret
    movei T,36.                        ;Must be in L36/F now (since paged=L36/P)
    movem T,fxftyp
    cain D,36.
    retskp
endif.
cain T,36.(D)                          ;Unless D=0,T=36, in which case TYPE is ok
ifskp.                                ;set up the type
    hrroi 1,[asciz/TYPE/]
    hrroi 2,[asciz/L 36/]
    cain D,1
    hrroi 2,[asciz/I/]
    cain D,7
    hrroi 2,[asciz/A/]
    cain D,8
    hrroi 2,[asciz/L 8/]
    call sndcmd
    call reply2                        ;200.
    ret
endif.
ife. D                                ;If want paged
    movei T,36.                        ;then so far have L36,F
    movem T,fxftyp
    hrroi 1,[asciz/STRU P/]
    call outcmd
    call reply2                        ;200.
    ret
endif.
movem D,fxftyp
retskp

```

```

;;Set up for data transfer. Argbf has filename, datcmd has command
datopn:    call xftsnd                ;Negotiate the desired transfer mode
    ret
    cain D,1                          ;Image type?
    movei D,36.                        ;Same as L 36 as far as we're concerned
GJINF%
    move T,3                           ;job #
    hrroi 1,tmpbuf                     ;Init jfn name
    hrroi 2,[asciz "CHA:"]
    call strcpy
    hlro 2,gate
    call strcpy
    hrroi 2,[asciz ".FTP"]
    call strcpy
    move 2,[.byte 8 ? "F ? "T ? "P]    ;Init packet
    movem 2,chkpt+%cpkdt
    move 2,[141000,,chkpt+%cpkdt]
    move 3,[110300,,T]                 ;Output 3 digits of job #
repeat 3,[
    ildb 4,3
    addi 4,"0"
    idpb 4,1                           ;To jfn name
    idpb 4,2                           ;and packet
    move 3,[060300,,netjfn]            ;And 2 digits of jfn, for fork uniquizat
repeat 2,[

```

```

ildb 4,3
addi 4,"0
idpb 4,1           ;To jfn name
idpb 4,2           ]       ;and packet
setz 4,
idpb 4,1           ;Terminate jfn name
setabt [cmsg "%Attempt to open data connection aborted."
        ret]
movx 1,gj%sh      ;Get a jfn
hrroi 2,tmpbuf
GTJFN%
    callret jerstr
movem 1,datjfn
movx 2,fld(8,of%bsz)\fld(7,of%mod)\of%rd\of%wr      ;Issue the rfc, n
OPENF%
    callret jerstr
move 1,icpsoc      ;Put the socket # in chaos packet
soj 1,
lsh 1,4.
movem 1,chkpt+%cpkdt+2
movei 1,12.        ;Length of packet
dpb 1,[chkpt+$CPKNB]
movei 1,212        ;Opcode=212
txne f,f%alts      ;Unless not going for port hacks
    skipge datcnt   ;Or this is the first one
    movei 1,210     ;then opcode=210
dpb 1,[chkpt+$CPKOP]
move 1,netjfn      ;Clear any buffered up stream output
movei 2,.mosnd
MTOPR%
    erjmp neterr
;;    txne f,f%alts
;;    call gratis    ;Minimize probability of lossage
move 1,netjfn
movei 2,.mopks     ;Tell gateway to start listening
movei 3,chkpt
MTOPR%
    erjmp neterr
movei 2,.monop     ;Wait for him to get it - this might hang
MTOPR%
    erjmp neterr
call port          ;Maybe get port number (should really retry
ret               ;without port if fails...)
setzm abtpc
hrroi 1,datcmd     ;Send the STOR/RETR
hrroi 2,argbf      ;Filename in argbf
call sndcmd
call replyl        ;Expect 150. reply
ret               ;Doesn't want to do it, tough
call repdis        ;Show him that we're starting
setabt xfrabt
setz 1,            ;Tell gateway to open
dpb 1,[chkpt+$CPKNB]
movei 1,211
dpb 1,[chkpt+$CPKOP]
move 1,netjfn
movei 2,.mosnd     ;Clear stream output
MTOPR%
    erjmp neterr
movei 2,.mopks

```

```

movei 3, chpkt
MTOPR%
    erjmp neterr
movei 2, .monop
MTOPR%
    erjmp neterr
do.
    move 1, datjfn
    GDSTS%
    cain 2, %csopn
    exit.
    caie 2, %csrfs
    jrst [call datem1
          callret xfrab1]
    movei 1, 200.
    DISMS%
    loop.
enddo.
setzm abtpc
call terpri
txo f, f%dcon
retskp

port:
    aose datcnt
    txnn f, f%alts
    retskp
    move 1, netjfn
    movei 2, .mopkr
    movei 3, chpkt
    MTOPR%
    erjmp neterr
    ldb 1, [$CPKOP+chkpt]
    cain 1, 300
    ifskp.
        ; Probably a gratuitous reply from server.
        ; Should check if it's a data packet, and if so, attempt
        ; to buffer it... ** Fix this **
        emsg "Couldn't get port # from gateway - please report to BUG-FTP@MIT"
        ret
    endif.
    ldb 1, [$CPKNB+chkpt]
    caige 1, 6
    ifskp.
        dmove 1, %CPKDT+chkpt
        lsh 1, -4
        lshc 1, -20
        lsh 2, -4
        skipe 2
        movem 2, gateno
    endif.
    skipn gateno
    call lkgate
    skipn gateno
    jrst [emsg "Cannot find gateway host number"
          ret]
    hrroi 1, tmpbuf
    move TT, [401000, , gateno]
    movei 3, 10.
    do.
        ildb 2, TT

```

;Make sure no confusion

;Wait for connection to open

;Open?

;Done then

;Still waiting?

;No, show reason

;and punt

;Wait a 5th of a sec

;Have a data connection now

;Don't do it for the first transfer

;Get back packet with port number

;Is it our packet?

; Shit!

;check length

;Includes his host number?

;wunnerful

;Get host number from packet

;Flush the padding

;Move the first 2 bytes into ac2

;Right-justify

;Have one?

;No, try to get it

;Have one now?

;Lose lose

;Build the port argument

;4 bytes of host number

;In base 10

```

NOUT%
    jfatal
    movei 2,"
    idpb 2,1
    tlne TT,770000
    loop.
enddo.
ldb 2,[341000,,chpkt+%cpkdt]      ;Now 2 bytes of port number
NOUT%
    jfatal
    movei 2,"
    idpb 2,1
    ldb 2,[241000,,chpkt+%cpkdt]
NOUT%
    jfatal
    setz 2,                        ;Terminate
    idpb 2,1
    hrroi 1,[asciz "PORT"]         ;Do it!
    hrroi 2,tmpbuf
    call sndcmd
    callret reply2

;;Find an internet address of gateway host... Try to match it to
;;the target host address.
lkgate:    move 1,hstbuf+NAMPTR      ;Lookup the gateway host
    hlro 2,gate
    TBLUK%
    jxe 2,t1%nom,r
    move 2,(2)
    hrrz 1,STRADR(2)                ;Go thru the address entries
    do.
        jumpe 1,r                    ;No more
        move 2,hstbuf+ADDADR(1)      ;Get the address
        ifxe. 2,NE%FMT              ;Can only use internet entries
        movem 2,gateno
        txnn 2,bit(5)                ;Extract the gateway network number
        txza 2,177777                ;Class B
        txz 2,377                    ;Class C
        txnn 2,bit(4)                ;Class A
        txz 2,77777777               ;Class A
        skipn 3,host                 ;Do we have the foreign host record?
        ret                          ;no, any old internet address will have to do
        hrrz 3,STRADR(3)             ;Else look for an address on same network
        do.
            jumpe 3,endlp.
            move 4,hstbuf+ADDADR(3)
            txnn 4,bit(5)
            txza 4,177777
            txz 4,377
            txnn 4,bit(4)
            txz 4,77777777
            camn 2,4                  ;Same?
            ret                       ;Yes, done
            hrrz 3,hstbuf+ADRCDR(3)   ;No, try the next host address
            loop.
        enddo.
    endif.
    hrrz 1,hstbuf+ADRCDR(1)          ;No luck, next gateway address
    loop.
enddo.

```

replen==:3000

.vector repbuf(replen)

;;Read a reply. Returns +1 if user aborted, otherwise +2 with code in C and
;; reply in repbuf (displayed if being verbose).

;;Enter at gtrep0 with pre-red char in C.

gtrep: seto C,

;Mark that no char

gtrep0: push p,abtpc

;Preserve these, even though right now

push p,abtsp

;always called with interrupts off.

move T,[440700,,repbuf]

movei TT,replen*5

setabt repabt

ifge. C

idpb C,T

soj TT,

endif.

call gtrep1

ifskp.

txzn C,.minfin

;Multi-line?

anskp.

push p,C

;Eat it all up

do.

call gtrep1

loop.

jumpl C,top.

came C,(p)

loop.

enddo.

pop p,C

endif.

movei 1,^J

;Terminate (in case ran out of buffer)

dpb 1,T

setz 1,

idpb 1,T

setzm abtpc

ifxn. f,f%verb\f%debug

;If verbose or debugging

txo f,f%cdis

call repdsl

;Then always show the reply, including code

endif.

pop p,abtsp

;Restore abort context

pop p,abtpc

retskp

repabt: cain TT,replen*5

ifskp.

;Maybe should SIBE, and soak up anything that's there anyhow?

cmsg "%Aborted while reading server reply, reply so far:"

movei 1,^J

dpb 1,T

setz 1,

idpb 1,T

txo f,f%cdis

call repdsl

else.

cmsg "%Aborted while awaiting server reply."

endif.

adjsp p,-2

;Flush old abtpc,abtsp

ret

;Return with ^G interrupt cleared

```
;Routine to read in one line of reply, into T/TT
;Skip-returns with code in C, sign bit set if multi-line.
gtrepl:      seto C,
```

```
      move 1,netjfn
      do.
```

```
      BIN%
```

```
      erjmp neterr
```

```
      jumpe 2,top.
```

```
      caie 2,177
```

```
      cain 2,^G
```

```
      loop.
```

```
      caie 2,IAC
```

```
      ifskp.
```

```
      call doteln
```

```
      loop.
```

```
      endif.
```

```
      cail 2,"0
```

```
      caile 2,"9
```

```
      exit. — if not #
```

```
      skipge C
```

```
      tdza C,C
```

```
      imuli C,10.
```

```
      addi C,"0(2) cool addr modes or what?!
```

```
      sosle TT
```

```
      idpb 2,T
```

```
      loop.
```

```
      enddo. — 2 = char we exited on
```

```
      skipl C
```

```
      aos (p)
```

```
      cain 2,"- cont'd?
```

```
      txo C,.minfin
```

```
      do.
```

```
      sosle TT
```

```
      idpb 2,T
```

```
      cain 2,^J
```

```
      ret
```

```
      BIN%
```

```
      erjmp neterr
```

```
      loop.
```

```
      enddo.
```

```
;; Handle telnet negotiations
```

```
iac==:255.
```

```
iacwill==:251.
```

```
iacwont==:252.
```

```
iacdo==:253.
```

```
iacdont==:254.
```

```
;Name table for pretty printing in debug mode.
```

```
iacmin==:239.
```

```
iactab:      offset iacmin-.
```

```
      asciz "EOR"
```

```
      ;239 End Of Record
```

```
      asciz "SE"
```

```
      ;240 Subnegotiation End
```

```
      asciz "NOP"
```

```
      ;241 Nop
```

```
      asciz "DM"
```

```
      ;242 Data Mark
```

```
      asciz "BRK"
```

```
      ;243 Break
```

```
      asciz "IP"
```

```
      ;244 Interrupt Process
```

```
      asciz "AO"
```

```
      ;245 Abort Output
```

```
      asciz "AYT"
```

```
      ;246 Are You There
```

```

        asciz "EC"
        asciz "EL"
        asciz "GA"
        asciz "SUB"
iacwil::asciz "WILL"
iacwon::asciz "WONT"
iacdo::      asciz "DO"
iacdon::asciz "DONT"
iac::        asciz "IAC"
        offset 0

doteln: IAC hacking
        ifxn. f,f%debug
        call terpri
        tmsg "<< [IAC] ["
        move 1,netjfn
    endif.
    BIN%
    erjmp neterr
    ifxn. f,f%debug
    cail 2,iacmin
    caile 2,iac
    ifskp.
        hrroi 1,iactab-iacmin(2)
        PSOUT%
    else.
        movei 1,.priou
        movei 3,10.
        NOUT%
        nop
    endif.
    tmsg "]"
    move 1,netjfn
    endif.
    move 3,2
    cail 3,251.
    caile 3,254.
    ret
    BIN%
    erjmp neterr
    ifxn. f,f%debug
    push p,3
    tmsg " ["
    movei 1,.priou
    movei 3,10.
    NOUT%
    nop
    tmsg "]"
    pop p,3
    move 1,netjfn
    endif.
    caie 3,iacwont
    cain 3,iacdont
    ret
    push p,2
    movei 2,iac
    BOUT%
    erjmp neterr
    movei 2,iacwont
    caie 3,iacdo
;247 Erase Character
;248 Erase Line
;249 Go Ahead
;250 Subnegotiation
;251 Will
;252 Won't
;253 Do
;254 Don't
;255 Interpret As Command

ignore all but will + do

will x → IAC don't x
do x → IAC won't x

; >>>

; Save command
; Random?

; Just ignore
; Get option

; Negative?

; Ignore...

; Else positive, refuse

; Reverse the command

```

```

    movei 2,iacdont
BOUT%
    erjmp neterr
    pop p,2
BOUT%
    erjmp neterr
    movei 2,.mosnd
MTOPR%
    erjmp neterr
    ret

;;Get a reply and do some post processing
getrep:    call gtrep
    ret
    caill C,100.
    ifskp.
        call repdis
        jrst getrep
    endif.
    cain C,331.
        jrst rep3pa
    cain C,332.
        jrst rep3ac
    caill C,400.
        callret repdis
    retskp

    ;User abort: lose
    ;No code, or 0xx:
    ;display
    ;and loop for more
    ;331: Send password and loop for more
    ;332: Send account and loop for more
    ;4xx and up: display and lose
    ;1xx,2xx,3xx: win

rep3pa:    txze f,f%pass
    ifskp.
        hrroi 1,pmtbuf
        hrroi 2,[asciz "Password"]
        call strcpy
        ifxn. f,f%user
            ;if we are logging in, make a useful prompt
            hrroi 2,[asciz " for logging in as "]
            call strcpy
            hrroi 2,argbf
            call strcpy
            hrroi 2,[asciz " to "]
            call strcpy
            skipn 2,host
        ifskp.
            hlro 2,STLNAM(2)
            addi 2,hstbuf
        else.
            hrroi 2,hnmbuf
        endif.
        call strcpy
    endif.
    hrroi 2,[asciz ": "]
    call strcpy
    hrroi 1,[asciz "password"]
    call getqli
    hrroi 2,atmbuf
else.
    hrroi 2,argbf1
endif.
hrroi 1,[asciz/PASS/]
call sndcmd
jrst getrep

```

```

rep3ac:      txze f,f%acct
            ifskp.
            hrroi 1,pmtbuf
            hrroi 2,[asciz "Account for "]
            call strcpy
            ifxn. f,f%user
            hrroi 2,argbf
            call strcpy
            hrroi 2,[asciz " on "]
            call strcpy
            endif.                ;Else should print last command name
            hrroi 2,hnmbuf
            call strcpy
            hrroi 2,[asciz ": "]
            call strcpy
            hrroi 1,[asciz "account"]
            call getlin
            hrroi 2,atmbuf
        else.
            hrroi 2,argbf2
        endif.
        hrroi 1,[asciz/ACCT/]
        call sndcmd
        jrst getrep

;;Read a 2xx reply.
reply2:      call getrep
            ret
            cail C,200.
            ifskp.                ;1xx:
            call repdis           ;display
            jrst reply            ;and loop
            endif.
            cail C,300.
            callret repdis        ;3xx: lose
            retskp                ;2xx: win

;;Normal reply (2xx and display)
reply:      call reply2
            ret
            call repdis
            retskp

;;Read a 1xx reply
reply1:      call getrep
            ret
            caige C,200.
            retskp                ;1xx: win
replyx:      cail C,300.
            callret repdis        ;3xx: lose
            cmsg "%Protocol violation, unexpected '2xx' reply" ;2xx: lose bi
            callret repdic

;;Read a specific 3xx reply, in A
reply3:      call getrep
            ret
            cail C,200.
            ifskp.                ;1xx: wait for more
            call repdis
            jrst reply3

```

```

endif.
camn C,A
    retskp                ;As expected.
jrst replyx

```

```

gratis:      txnn f,f%conn
             ret
             call chkcon
             move 1,netjfn
             do.
                 SIBE%
                 nop
                 jumpe 2,R
                 BIN%
                 erjmp neterr
                 caie 2,iac
                 exit.
                 call doteln
                 loop.
             enddo.
             move C,2
             cmsg "%Gratuitous input from server:"
             do.
                 call gtrep0
                 callret gratis        ;Don't really let him abort
                 call repdic
                 seto C,                ;No more pre-read chars
                 move 1,netjfn
                 SIBE%
                 nop
                 jumpn 2,top.
             enddo.
             ret

```

```

repdis:      txza f,f%cdis           ;Temp flag = off means flush error
repdic:      txo f,f%cdis           ;On means show it
             txne f,f%verb\f%debug  ;If being verbose or debugging
             ret                    ;then return, already showed it.

repds1:      call terpri.
             move 2,[440700,,repbuf]
             do.
                 tmsg "<< "          ;">>"
                 ifxe. f,f%cdis
                 do.
                     ildb 1,2
                     cail 1,"0
                     caile 1,"9
                     exit.
                 loop.
             enddo.
             caie 1,"-
             cain 1,40
             ifskp.
                 seto 1,
                 ibp 1,2
                 move 2,1
             endif.
             endif.
             movei 1,.priou
             movx 3,.infin

```

```

    movei 4,^J
    SOUT%
    move 1,2
    ildb 1,1
    jumpn 1,top.
enddo.
ret

```

;See if there is another line

```

outcmd:      setz 2,
sndcmd:      push p,2
            push p,1
            call gratis
            ifxn. f,f%debug
            cmsg ">> "
            move 1,(p)
            PSOUT%
            skipn -1(p)
            ifskip.
            movei 1,40
            PBOUT%
            move 1,-1(p)
            PSOUT%
            endif.
        endif.
        move 1,netjfn
        move 2,(p)
        setz 3,
        SOUT%
        erjmp neterr
        skipn -1(p)
        ifskip.
        movei 2,40
        BOUT%
        erjmp neterr
        move 2,-1(p)
        SOUT%
        erjmp neterr
    endif.
    adjsp p,-2
    hrroi 2,[asciz "
"] ?      SOUT%
            erjmp neterr
            movei 2,.mosnd
            MTOPR%
            erjmp neterr
            ret

```

;Maybe should check if it's PASS, and if so
;don't print the arg?

;Analyze a net error. Should really check if there is any input and
;type it out, for maximum info. Should also show the error message
;unless iox4 (eof) or iox5 (data error). Maybe base that on verbosity
;and/or debugging flags.

netems: setzm abtpc

```

    movei 1,.fhslf
    GETER%
    move T,2
    move 1,netjfn
    GDSTS%
    andi 2,17
    caie 2,%csopn

```

;Save the error (in case errors in PSOUT'
; can happen if running in the background

;Check out connection state

```

ifskp.                ;Connection still open
GTSTS%
ifxn. 2,gs%eof        ;Eof received means TCP connection closed
    emsg "Connection broken by foreign host"
    ret
endif.
emsg "Error on control connection: "        ;Else random
move 2,T              ;show the actual error message
callret .erstr
endif.
netem1:    setzm abtpc
caie 2,%csinc
ifskp.                ;Connection incomplete
    emsg "Gateway host not responding"
    ret
endif.
caie 2,%cslos
ifskp.
    emsg "Lossage on control connection"
    move 1,netjfn      ;Find the reason
    movei 2,.moerr
    hrroi 3,tmpbuf
    setzm tmpbuf
    MTOPR%
    ifnje.
        skipn tmpbuf
    anskp.
        tmsg ": "
        hrroi 1,tmpbuf
        PSOUT%
    endif.
    ret
endif.
caie 2,%cscls
ifskp.
    hrroi 1,[asciz "Connection closed"]
    txnn f,f%conn      ;Were we supposed to be connected?
    hrroi 1,[asciz "Connection refused"] ;No, then CLS means refused
    ESOUT%
    move 1,netjfn
    do.
        GDSTS%
        movei 3,(3)
        jumpe 3,endlp.
        movei 2,.mopkr
        movei 3,chkpt
        MTOPR%
        erjmp endlp.
        ldb 2,[chkpt+$CPKOP]
        caie 2,%cocls
        loop.
        ldb 3,[chkpt+$CPKNB]
        jumpe 3,endlp.
        tmsg ": "
        movei 1,.priou
        move 2,[441000,,chkpt+%cpkdt]
        movns 3
        SOUT%
    enddo.
    ret

```

```

endif.
emsg "Connection entered strange state ("
movei 1,.priou
movei 3,8
NOUT%
  nop
tmsg ") "
ret

```

```

;;Here on random errors on net connection
neterr:      call netems                      ;show reason
            txz f,f%conn                      ;Not connected nowhere anymore
            jrst cmderl                      ;Go hack commands again

```

```

;;Here to make sure connection still good.
chkcon:      move 1,netjfn
            GDSTS%
            andi 2,17
            cain 2,%csopn
            ret
            call neteml                      ;Make like neterr
            txz f,f%conn
            jrst cmderl

```

```

;;Once more with feeling
datems:      setzm abtpc
            movei 1,.fhslf                      ;Save the error (in case errors in PSOUT'
            GETER%                              ; can happen if running in the background
            move T,2
            move 1,datjfn
            GDSTS%                              ;Check out connection state
            andi 2,17
            caie 2,%csopn
            ifskp.                              ;Connection still open
            GTSTS%
            ifxn. 2,gs%eof                      ;Eof received means TCP connection closed
            emsg "Data connection broken by foreign host"
            ret
            endif.
            emsg "Error on data connection: "      ;Else random
            move 2,T                              ;show the actual error message
            callret .erstr
            endif.

```

```

dateml:      setzm abtpc
            caie 2,%csinc
            ifskp.                              ;Connection incomplete
            emsg "No response on data connection"
            ret
            endif.
            caie 2,%cslos
            ifskp.
            emsg "Lossage on data connection"
            move 1,datjfn                      ;Find the reason
            movei 2,.moerr
            hrroi 3,tmpbuf
            setzm tmpbuf
            MTOPR%
            ifnje.
            skipn tmpbuf
            anskp.

```

```

    tmsg ": "
    hrroi 1,tmpbuf
    PSOUT%
endif.
ret
endif.
caie 2,%cscls
ifskp.
    hrroi 1,[asciz "Data connection closed"]
    txnn f,f%dcon ;Were we supposed to be connected?
    hrroi 1,[asciz "Data connection refused"] ;No, then CLS mean
    ESOUT%
    move 1,datjfn
    do.
        GDSTS%
        movei 3,(3)
        jumpe 3,endlp.
        movei 2,.mopkr
        movei 3,chkpt
        MTOPR%
        erjmp endlp.
        ldb 2,[chkpt+$CPKOP]
        caie 2,%cocls
        loop.
        ldb 3,[chkpt+$CPKNB]
        jumpe 3,endlp.
        tmsg ": "
        movei 1,.priou
        move 2,[441000,,chkpt+%cpkdt]
        movns 3
        SOUT%
    enddo.
    ret
endif.
emsg "Data connection entered strange state ("
movei 1,.priou
movei 3,8
NOUT%
nop
tmsg ") "
ret

```

;;Here on SOUT%/SIN% (datjfn) error

```

xfrerr:      call datems ;Show reason
            callret xfrabl

```

;;Error on local end during data transfer

```

filerr:      seto 2,
            call cerstr ;Show error
            callret xfrabl ;abort transfer and return

```

;;Here to abort a data transfer

```

xfrabt:      setzm abtpc
            cmsg "%Aborting file transfer"
xfrabl:      setzm abtpc
            movei 1,datjfn ;flush the connection, in case other end
            call pntjfn ;needs that before it replies
            jwarn
            call gtrep ;Soak up a reply
            ret ;aborted that, oh well..

```

```

callret repdis                                ;Show why he closed the connection

;Get rid of a jfn, 1/address of jfn
pntfil:      movei 1,filjfn
pntjfn:      skipn 2,(1)                      ;Anybody there?
      retskp                                  ;That was easy...
      setzm (1)                               ;Mark that flushed now
      move 1,2
      GTSTS%
      ifxe. 2,gs%opn                          ;Not open -
      RLJFN%                                  ; just release the jfn
      ret                                     ; let caller give a more informative error
      retskp
      endif.
      txo 1,cz%abt                            ;Else abort it
      CLOSF%
      ret
      retskp

echoff:      saveac [1,2]
      movei 1,.priin
      RFMOD%
      txz 2,tt%eco\tt%ecm
      SFMOD%
;      STPAR%
      txo f,f%echo
      ret

echon:      txzn f,f%echo
      ret
      saveac [1,2]
      movei 1,.priin
      RFMOD%
      txo 2,tt%eco\tt%ecm
      SFMOD%
;      STPAR%
      ret

terpri:      saveac [1,2,3]
      movei 1,.priou
      movei 2,.vtadv
      VTSOP%
      erjmp [hrroi 1,[asciz/
/] ? PSOUT% ? ret]
      ret

jerstr:      move 2,1
cerstr:      setzm abtpc
      emsg ""
.erstr:      movei 1,.priou
      hrli 2,.fhslf
      setz 3,
      ERSTR%
      jrst [tmsg "Unknown error"
      ret]
      jfatal
      ret

$fatal:      setzm abtpc
      hrro 1,(p)

```

```

        ESOUT%
die:      move 1,[.prast,,.fhslf]
        movei 2,[.prkil,,0]
        movei 3,1
        PRARG
        call terpri
        HALTF%
        fatal "Can't continue"

```

```

$jwarn:      nop
$jerror:nop
$jfatal:push p,abtpc
            push p,abtsp
            setzm abtpc
            exch 3,-2(p)
            push p,2
            push p,1
            movei 1,.fhslf
            GETER%
            hrrz 1,-1(3)
            push p,1
            caie 1,$jwarn
            ifskp.
                call terpri
                movei 1,"%
                PBOUT%
            else.
                emsg ""
            endif.
            skipn (3)
            ifskp.
                hrro 1,3
                PSOUT%
                tmsg " -- "
            endif.
            call .erstr
            pop p,1
            caie 1,$jfatal
            ifskp.
                tmsg " (PC="
                movei 1,.priou
                move 2,-5(p)
                movei 2,-2(2)
                movei 3,8
                NOUT%
                nop
                tmsg ")"
                jrst die
            endif.
            pop p,1
            pop p,2
            pop p,abtsp
            pop p,abtpc
            pop p,3
            ret

```

end begin

;really should just defer interrupts,
;not ignore them...

;Get real error first, in case of errors
;during output

NIC. SRT. COM

10.0.0.51

NETINFO: HOSTS.TXT

7391

REC952 describes format