Tops-20 Kermit-20 Batch Test Battery

N.B., Tests need not be performed in the order specified

Table of Contents

K20NUL: Basic Kermit-20 Server functional checkout

1) Pseudo-terminal login
2) INPUT statement test with C escape sequences
3) OUTPUT statement test with C escape sequences
4) Basic text file transfer checking
5) Wildcard file transfer checking to NUL:
6) Wildcard file transfer checking with different checksums
7) Session log testing
8) 94 character packets (largest for basic protocol)
9) Basic SERVER command repertoire
10) Basic NUL: tests

K20NUP: Basic Kermit-20 Server functional checking using PUT

1) Like K20NUL, except uses push to remote NUL: instead of GET to local NUL:

K20PTY: Kermit-20 Server Transfer additional and post transfer checks

1) Prompt defaults (DECnet node if remote) and parsing
2) Time-out parsing
3) 120 character packets (larger then basic minimum, somewhat faster)
4) Internal transport timing checkout
5) INPUT/OUTPUT statement tests with C escape sequences
6) Pseudo-terminal login with no secondary fork
7) Basic macro definition check
8) Basic SERVER command repertoire
9) Basic NUL: tests
10) Short duration communications check: 1.62 milliseconds!
11) Basic text file transfers
12) Comparison of statistics between both sides (they're 'close'...)
13) Large text file test:

Quantity	Type
1,794,319	ASCII (7 bit) bytes
701	Tops-20 disk pages
9,000	Packet size (largest possible)
101.8023	KC/s (Highest total characters/second)
1.2186	MBd (Highest effective data rate)

14) Post transfer file comparisons (all OK)

K20NRV: Kermit-20 Server functions via Tops-20 DECnet NRT to local host

1) Same as K20PTY except 94 byte packet size only
2) DECnet NRT connection to the local Tops-20 host
3) Checks files, post transfer (all OK)

4) Does not include large file case
5) Fastest possible NRT (all communications are memory only)
6) Additional program information (memory layout)

K20NRT: Kermit-20 Server functions via Tops-20 DECnet NRT to remote host

1) Same as K20NRV except going to a different Tops-20 host
2) *Far* slower transfer speeds (suspected to be an emulator artifact)

K20NOL: Basic regression test against Kermit-20 4.2(174) [2-May-85]

1) Similar to K20NUL with reduced functionality because 4.2(174) does not have:
   a) No remote NUL: testing
   b) No PWD testing
   c) No remote statistics
   d) No large file/buffer
   e) No pseudo-efficiency simulations
2) 4.2(174) reports disk quota of 70,000 as "+Inf"
3) Compression appears to be somewhat better (could be incorrect, however)
4) Session logging test
5) N.B., 174 decimal is 256 octal

K20POL: Regression test against Kermit-20 4.2(174) [2-May-85] using GET

1) Similar to K20PTY with reduced functionality
2) Otherwise, similar to K20NOL, except for 3)
3) Post transfer file comparisons (all OK)

K20PNW: Regression test against Kermit-20 4.2(174) [2-May-85] using PUT

1) Same as K20POL, but uses PUT instead of GET

K20NUR: Basic Kermit-20 Server functional checkout with parity

1) Like K20NUL (Basic Test), but using parity on packets
2) Somewhat slower transfer, perhaps 5% impact (may be false negative)

K20POR: Regression test against Kermit-20 4.2(174) [2-May-85] with parity

1) Like K20POL with parity checking

K20B8T: Kermit-20 Server 8-bit file transfer and post transfer checks

1) Like K20PTY, except 8 bit files

K20B8P: Kermit-20 Server 8-bit file transfer with parity and post transfer checks

1) Like K20B8T, except does 8 bit files with even parity

K20B8A: Kermit-20 Server terminal 8-bit file transfer with parity and transfer checks

1) Like K20B8E, except also checks parity from Tops-20 terminal driver

K20DPD: Kermit-20 Packet Decoding Example

1) Enhancement to decode a packet instead of dumping raw data
2) PTY transfer checks for fineness of time logging
3) Tops-20 design limits time of day resolution to milliseconds

K20TCP: Kermit-20 Transmit/Capture Testing via pseudo-terminal

1) Basic transmit, no parity processing, captured to NUL:
2) Transmit, space parity processing, captured to NUL:
3) Transmit, mark parity processing, captured to NUL:
4) Transmit, odd parity processing, captured to NUL:
5) Transmit, odd parity processing, forcing a parity error
6) Transmit, full even parity, including Tops-20 terminal line generated
7) Post transmit comparison checking of large and small files

K20TCN: Kermit-20 Transmit/Capture Testing via DECnet NRT

1) Same as K20TCP, except tests are performed over a Tops-10/20 type DECnet Network Remote Terminal
2) Since DECnet NRT's do not generate parity, test 6 is performed without terminal line parity checking

K2036P: Kermit-20 36 Bit Mode via pseudo-terminal

1) Same as K20PTY, except no large files, all files being executables
2) Demonstrate correct and enhanced transaction logging
3) FILCOM/E demonstrates transfers correct to bit level
4) Double checked against EXEC directory, CHECKSUM BY-PAGES
5) Pre and post transfer file byte sizes and counts visually compared to be correct

K2036C: Kermit-20 36 Bit Mode via pseudo-terminal with parity

Note, be aware that while BATCON will properly strip parity for .LOG files, performing a DO of the same control (.CTL) file as if it were a .MIC file can result in strange output on the local terminal as Kermit-20's ECHO command will force parity. This is a design feature which is used for checkout and debugging.

Further information may be gotten from Kermit-20's extensive and up to date built-in help by typing HELP ECHO and also HELP SET PARITY.

1) Same as K2036C, but with parity sending on terminal and packets and checking being performed

2) Demonstrate correct and enhanced transaction logging which is transparent to parity
3) Demonstrate proper error handling when a file cannot be opened by the Kermit-20 server;
   a. That server state is updated,
   b. That an ‘E’ packet is sent,
   c. That the ‘E’ packet is properly interpreted by the Kermit-20 client, and
   d. That no-remnants of the file are left in the testing directory.

`K10NRT`: Kermit-10 Regression Tests via DECnet NRT

1) Remote terminal tests (`LOGIN`, `SYSTAT`, Etc.)
2) Basic server tests, `REMOTE SPACE`, `REMOTE HELP`
3) `REMOTE DIRECTORY`, `REMOTE DELETE` validation
4) Kermit-10 does not implement CWD/PWD (as Tops-10 does not have the idea of a connected directory)
5) Transfer and comparison of various files, some quite large (all OK)
6) Session logging

`K10NRP`: Kermit-10 Parity Regression Tests via DECnet NRT

1) As `K10NRT`, except `REMOTE DELETE` not validated and remote terminal assumed to work
2) Transfer and comparison of various files, none particularly large
3) Errors
   a. Correct parity not sent in response to FINISH
   b. Correct parity not sent in wildcarded directory listing