

Return-Path: <jrd>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA22678; Mon, 8 Jun 92 19:28:12 EDT
Date: Mon, 8 Jun 92 19:28:12 EDT
From: "Joe R. Doupnik" <jrd@watsun.cc.columbia.edu>
To: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
Cc: jrd, fdc
Subject: Re: Belated echoes
In-Reply-To: Your message of Mon, 1992 Jun 8 16:16 EDT
Message-Id: <CMM.0.90.0.708046092.jrd@watsun.cc.columbia.edu>

John,

As far as MSK is concerned, leading garbage is ignored but the start of a packet triggers a short timer to grab pkt characters or fail. Failure cascades upward to generate retries. So we should expect rather large quantities of retries when echos start looking like packets at their front end.

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA19686; Mon, 8 Jun 92 16:16:57 EDT
Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 6086; Mon, 08 Jun 92 16:16:45 EDT
Date: Mon, 1992 Jun 8 16:16 EDT
From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Joe Doupnik <jrd@watsun.cc.columbia.edu>
Subject: Belated echoes
Message-Id: <PEPMNT.920608.161639.RC0@CFAAMP.HARVARD.EDU>

I remember now how I got so many belated echoes in some of the early tests of DDK. It hinged on the fact that every inbound packet gets echoed twice in the CMS and TSO non-transparent environments and also the fact that echoes weren't always ignored properly for one reason or another. The result was often a train of echoes of one packet coming back to the file receiver interleaved with some valid responses. This would almost call for buffer flushing, but that is so hazardous as to be out of the question, I think. Suffice it to say that an inbound packet can be completely ok in a situation where one echo (or possibly both) can appear crunched. It doesn't happen frequently, but neither is it rare, so it is important to be able to ignore *all* echoes, both valid and crunched.

John

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA25556; Wed, 11 Mar 92 15:32:41 EST
Date: Wed, 11 Mar 92 15:32:41 EST
From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
To: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
Cc: Joe Doupnik <JRD@cc.usu.edu>,
Christine M Gianone <cmg@watsun.cc.columbia.edu>
Subject: Re: Doomsday Kermit
In-Reply-To: Your message of Wed, 1992 Mar 11 14:02 EST
Message-Id: <CMM.0.90.0.700345961.fdc@watsun.cc.columbia.edu>

> Although my scheme for 4-to-5 encoding currently calls for using all the
> printable ASCII characters, it doesn't need to. The algorithm needs
> just 87 different characters (2^{32} can be represented by five digits in
> base 85, and the encoding needs two extra characters for bookkeeping).
>

Right, but your encoding solves a different problem. The problem is, in the doomsday scenario, we can only consider uppercase letters + digits to be safe. If we allow more than that, then we have to give users the ability to modify the alphabet.

So there are really two separate issues at this point. One is a totally sure-fire, foolproof, last-resort transparency mode, and the other is a better compaction method. The latter is a much bigger question.

Maybe the doomsday encoding isn't worth it. Anyway, none of us has time for prolonged discussion of this, so let's just let it cook for a while.

- Frank

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA26054; Wed, 11 Mar 92 16:01:06 EST
Date: Wed, 11 Mar 1992 14:00 MDT
From: Joe Douppnik <JRD@cc.usu.edu>
Subject: Re: Doomsday protocol II
To: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu,
cmg@watsun.cc.columbia.edu
Message-Id: <CC8AB6F5B842D67D@CC.USU.EDU>
X-Envelope-To: cmg@watsun.cc.columbia.edu, fdc@watsun.cc.columbia.edu
X-Vms-To:
IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu", IN%"cmg@watsun.c
c.columbia.edu"
X-Vms-Cc: JRD

Frank and John and Chris,

Last comment before the lid closes on the subject. We still don't know which characters are magic for a channel so if "F" is then hex won't work either. About the minimal set I can imagine a computer vendor offering is all ten digits, and these of course are case independent. So a middle ground proposal is to send the numeric string of the packet, blocked in some convenient form I haven't the time to figure out (log base 10 stuff). The SOP and EOP bytes will be present, and the rest "could" be straight decimal digits. Less complicated schemes are difficult to imagine.

My class would say, getting in some revenge, "Heck. Send the length and the CRC and let the receiver puzzle it out."

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA24795; Wed, 11 Mar 92 14:45:30 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 1372; Wed, 11 Mar 92 14:41:12 EST

Date: Wed, 1992 Mar 11 14:02 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Joe Doupnik <JRD@cc.usu.edu>,
Christine M Gianone <cmg@watsun.cc.columbia.edu>

Subject: Re: Doomsday Kermit

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 11 Mar 92 0:27:43 EST

Message-Id: <PEPMNT.920311.140219.RC0@CFAAMP.HARVARD.EDU>

> One minor, but perhaps fatal, weakness of the Kermit protocol is that it
> requires that all ASCII printable characters can pass through the
> communication link -- not true in some cases, as with Honeywell front
> ends. A related problem is that the checksum can end with a space that
> might be trimmed off.

Space trimming can be avoided in 3270-type FULLSCREEN mode, so that's not a problem as far as I know. The question is how many characters won't pass through a Honeywell FEP (and why not)?

> Of course, we could use some base other than hex, like base 36 (0-9, A-Z),
> but it doesn't gain anything -- you still need two characters to encode one.

Although my scheme for 4-to-5 encoding currently calls for using all the printable ASCII characters, it doesn't need to. The algorithm needs just 87 different characters (2^{32} can be represented by five digits in base 85, and the encoding needs two extra characters for bookkeeping). Obviously, making an arbitrary choice of any 87 characters could require some cumbersome logic, but the increased efficiency would be worth it. Other details would hinge critically on how many and which characters can't get through. With a bit of luck, you could avoid encoding the the packets and just use the 4-to-5 on the data (e.g., if the deadly characters don't include any in the range 32-95, the packet sequence number would be ok as-is; even the length code would be ok as long as you limit packets appropriately; similarly, the packet type should be fine; and 1-byte checks are already restricted to the range 32-95).

John

Return-Path: <fdc>

Received: by watsun.cc.columbia.edu (5.59/FCB)

id AA11747; Wed, 11 Mar 92 00:27:44 EST

Date: Wed, 11 Mar 92 0:27:43 EST

From: Frank da Cruz <fdc@watsun.cc.columbia.edu>

To: Joe Doupnik <JRD@cc.usu.edu>, John Chandler <PEPMNT@CFAAMP.HARVARD.EDU>,
Christine M Gianone <cmg@watsun.cc.columbia.edu>

Subject: Doomsday Kermit

Message-Id: <CMM.0.90.0.700291663.fdc@watsun.cc.columbia.edu>

I had another awful midnight idea, but I haven't tried coding it yet. One minor, but perhaps fatal, weakness of the Kermit protocol is that it requires that all ASCII printable characters can pass through the communication link -- not true in some cases, as with Honeywell front ends. A related problem is that the checksum can end with a space that might be trimmed off.

So as a final recourse, we can encode the entire packet, excluding the packet-start (which can now be printable) and end (which is not needed anyway) in hex. Of course, this doubles the overhead, but it solves the problem. It also solves the problem with 3270 screen optimization. Obviously, we still have to make hex-encoded packets fit within the negotiated packet length.

There are two choices for how to do this: (1) do it at a low level when sending and receiving packets that have already been composed by the normal means (easy, but it's a redundant level of encoding), or (2) skip all the control- and 8th-bit encoding and render these characters directly into hex (slightly more complicated in the code, but also slightly more efficient).

In either case, compression (of the current run-length kind, or any new kind we eventually settle upon) still takes place before the hexification.

Of course, we could use some base other than hex, like base 36 (0-9, A-Z), but it doesn't gain anything -- you still need two characters to encode one.

This scheme need not be linked to the 3270 environment; it could be used anywhere as a last resort.

Probably a dumb idea. Reactions?

- Frank

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA05061; Fri, 6 Mar 92 14:25:54 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 8964; Fri, 06 Mar 92 14:21:03 EST

Date: Fri, 1992 Mar 6 13:57 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>

Subject: Re: [Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>: FULL transfer]

In-Reply-To: fdc@watsun.cc.columbia.edu message of Thu, 5 Mar 92 16:43:41 EST

Message-Id: <PEPMNT.920306.135749.RC0@CFAAMP.HARVARD.EDU>

> I was able to get a small test file from the VM/CMS system in full-screen
> and send it back.

Ok, that's good news. I've already verified that the new scheme works on my 4994 and sent the updates to the users of a 3708, a MICOM 7400, a 3174, and a 7171. I'm hoping they all still work.

> However, the date was the current date, and the format
> had changed from F to V.

That's not a surprise. No variant of K-370 has ever bugged the dates on uploaded files, even though I recently added syntax checking on the received Date Attribute. In fact, it was only last year that I found out how to re-write the MFD on CMS after fiddling with one of the FST's. I don't know if it's even possible to play that sort of game under MVS or other systems, but it might be. Anyway, I plan to add that feature someday, but you can go ahead yourself if you don't want to wait that long.

As for the RECFM, there is no mechanism for conveying it, so you got the current setting in CMS Kermit, which is how it has always worked. I could point out that I have long advocated a Kermit attribute for the storage RECFM (as opposed to the transmission RECFM), but that actually wouldn't help (without an Attribute Archiving mechanism), since RECFM is not an attribute understood by MS-DOS.

By the way, there *is* a way to make a file come back with the same attributes it had when it left, but it means setting FILE COLLISION to OVERWRITE and FILE OVERWRITE to PRESERVE and uploading it over itself (or another file with the desired attributes).

John

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA26810; Wed, 4 Mar 92 13:32:11 EST
Date: Wed, 4 Mar 92 13:32:10 EST
From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
To: Joe Doupnik <JRD@cc.usu.edu>, John Chandler <PEPMNT@CFAAMP.HARVARD.EDU>
Subject: [Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>: Test of MSK 3.12
and KERMIT CMS 4.2.3 (test)]
Message-Id: <CMM.0.90.0.699733930.fdc@watsun.cc.columbia.edu>

...

Date: Wed, 04 Mar 92 13:22:52 EST
From: Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>
Subject: Test of MSK 3.12 and KERMIT CMS 4.2.3 (test)
To: Frank da Cruz <fdc@watsun.cc.columbia.edu>

I tried running KERMIT370 with MSK 3.12 today. The protocol shows promise, but doesn't work properly as yet. There are many timeouts on the PC side.

The main difficulty I see is the numerous characters that are inserted for cursor control by the protocol converter. Also characters arriving at the IBM are echoed to the screen. Maybe it would be possible to use a non-echoing read, as is done when reading passwords.

The transactions I tried were:
REMOTE DIR TEST
and
BYE

The BYE command did log off the mainframe account.

It appears easier to send to the IBM than to receive from it.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA29057; Wed, 4 Mar 92 16:00:36 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 8023; Wed, 04 Mar 92 15:55:56 EST

Date: Wed, 1992 Mar 4 15:37 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Joe Douppnik <JRD@cc.usu.edu>,
Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>

Subject: Re: [Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>: Test of MSK
3.12 and KERMIT CMS 4.2.3 (test)]

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 4 Mar 92 13:32:10 EST
Message-Id: <PEPMNT.920304.153734.RC0@CFAAMP.HARVARD.EDU>

> I tried running KERMIT370 with MSK 3.12 today. The protocol shows promise,
> but doesn't work properly as yet. There are many timeouts on the PC side.

Unfortunately, the modified Kermit protocol is just less reliable than
the real thing. I have one idea to reduce the level of problems by
clearing the screen less often, but that won't eliminate them.

> The main difficulty I see is the numerous characters that are inserted for
> cursor control by the protocol converter. Also characters arriving at the
> IBM are echoed to the screen. Maybe it would be possible to use a non-echoing
> read, as is done when reading passwords.

I'm afraid that method would simply add all the more cursor motion
sequences to the outbound stream. On the other hand, the cursor
motion commands do not, by themselves, constitute a problem, except
that they slow the transfers down a bit. At a cost of further
slowing things down, I think the incidence of problems can be
reduced (possibly dramatically) by just shortening the packets both
ways (from 79 to 77, effectively).

> It appears easier to send to the IBM than to receive from it.

This is connected with the transition that occurs every time the
screen gets cleared.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA00806; Wed, 4 Mar 92 17:21:24 EST
Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 8075; Wed, 04 Mar 92 17:16:46 EST
Date: Wed, 1992 Mar 4 17:00 EST
From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Joe Doupnik <JRD@cc.usu.edu>,
Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>
Subject: Re: [Peter Jones <SMKERMIT%UQAM@cuvmb.cc.columbia.edu>: Files for
Kermit370 4.2.3 and MSK 3.12 beta]
In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 4 Mar 92 14:30:10 EST
Message-Id: <PEPMNT.920304.170028.RC0@CFAAMP.HARVARD.EDU>

> I forgot to send the log files as part of my test results. Here they are.

Indeed, I see that there is a timeout every time the screen gets cleared (every other packet) for the simple reason that the protocol converter squeezes one byte out of the data stream at that point (the packet character, natch). There is, sadly, no general-purpose defense against an overzealous screen optimizer, but I might be able to attack this specific one. First, there is the item of clearing the screen less often. Second, I could probably interleave the screen display. (Unfortunately, that would downgrade one of the recovery mechanisms for tn3270.) By the way, I take it this is the STRTC network in operation. If there is any way of suppressing some of the screen optimization, that would be a big boost.

John

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA04217; Wed, 4 Mar 92 22:37:19 EST
Date: Wed, 4 Mar 1992 20:36 MDT
From: Joe Doupnik <JRD@cc.usu.edu>
Subject: Today's IBM vs PC Kermit packet log
To: smkermit%ugam@cuvmb.cc.columbia.edu
Cc: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu
Message-Id: <83B36AC06441ACB5@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"smkermit%ugam@cuvmb.cc.columbia.edu"
X-Vms-Cc: IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu", JRD

Peter and Frank and John,

That example log is so incredible that I'm duplicating it for my class. Upon reading it this evening I can't see any mistakes by MSK. The IBM host is exceptionally chatty with cursor motions however.

If worse were to come to worse, and ALL we had to contend with was ANSI compatible control sequences, then under great duress I could parse away the things so the packet recognizer never saw them. This would definitely be a performance loser on the PC side. Closer inspection of the PC log shows that the mainframe Start of Packet character is sometimes removed by the cursor stuff (that's a real data loss and I'm surprized it occurs unless the new field happens to overlap a similar old field).

My short form name for this whole thing is Two Dimensional Packets: the PC would need the same 3270 screen as the host to know what the host is thinking. Hmumum, there's a thought.

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
 Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
 id AA05642; Thu, 5 Mar 92 00:34:46 EST
 Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
 ith BSMTP id 8258; Thu, 05 Mar 92 00:30:39 EST
 Date: Thu, 1992 Mar 5 00:01 EST
 From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
 To: Joe Doupnik <JRD@cc.usu.edu>,
 smkermit <smkermit%ugam@cuvmb.cc.columbia.edu>,
 fdc <fdc@watsun.cc.columbia.edu>
 Subject: Re: Today's IBM vs PC Kermit packet log
 In-Reply-To: JRD@cc.usu.edu message of Wed, 4 Mar 1992 20:36 MDT
 Message-Id: <PEPMNT.920305.000102.RC0@CFAAMP.HARVARD.EDU>

> Upon reading it this evening I can't see any mistakes by MSK.

That's right.

> IBM host is exceptionally chatty with cursor motions however.

Not actually the host, but the protocol converter. They vary.

> If worse were to come to worse, and ALL we had to contend with
 > was ANSI compatible control sequences, then under great duress I could
 > parse away the things so the packet recognizer never saw them.

No point in that, since the cursor motion stuff isn't a problem, unless
 you *really* want to get around the 80-byte limit. Because of the
 precariousness of the modified protocol, though, I don't think we
 should even think about that. Then there's tn3270, which seems to
 like using CR, LF, HT, and BS to move the cursor around (because it's
 more efficient).

> the PC log shows that the mainframe Start of Packet character is sometimes
 > removed by the cursor stuff (that's a real data loss and I'm surprized it
 > occurs unless the new field happens to overlap a similar old field).

That's what I was pointing out in my previous message. It *does*
 happen to overlap the START character from the packet that was on the
 screen before, and it's not a coincidence at all.

> My short form name for this whole thing is Two Dimensional Packets:
 > the PC would need the same 3270 screen as the host to know what the host is
 > thinking. Hmmm, there's a thought.

Are you serious? That seems to me to be a rathole, but maybe I'm just
 too pessimistic. My idea for counteracting the nasty screen optimizer
 is to make K-370 interleave packets so that nothing ever overlaps.
 It is not too hard for CMS, I think. In fact, the following is an
 update to be applied after SC92030 that should do the trick.

John

```

-----
./ R 02602500          $ 2602500 190          03/05/92 00:26:33
FCLRF EQU X'08'      Skipping two lines on screen      ???
./ R 05701300 05701700 $ 5701300 70          03/05/92 00:26:33
SR 0,0              Clear before every packet          ???
KCALL SCRNI0        ???
XI FL3,FCLRF        Flip switch for skipping          ???
TM FL3,FCLRF        Skipping now?                    ???
BZ RTRN0            Not this time                    ???
WRTERM ' '          ???
WRTERM ' '          ???
./ D 08585500          03/05/92 00:26:33

```


Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA05747; Thu, 5 Mar 92 00:51:23 EST
Date: Wed, 4 Mar 1992 22:50 MDT
From: Joe Doupnik <JRD@cc.usu.edu>
Subject: Re re: Today's IBM vs PC Kermit packet log
To: pepmnt@cfaamp.harvard.edu
Cc: fdc@watsun.cc.columbia.edu, smkermit%uqam@cuvmb.cc.columbia.edu
Message-Id: <967158DCD441C814@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"pepmnt@cfaamp.harvard.edu"
X-Vms-Cc:
IN%"fdc@watsun.cc.columbia.edu", IN%"smkermit%uqam@cuvmb.cc.columbia.edu", JRD

John and the group,

Ok, I'll breath easier by not having to run incoming packets through a 3270 screen to undo the front end processor. I thought that was the basic solution if the host was going to send things this way. Stateful stuff, and I'd be stuck pondering how to represent an inverse video packet amongst the blinking and bold ones. (Sorry, it's late and I'm zonked).

So we keep packets safely within the 80 byte lines and hope that the IBM mainframe side provides safe screen areas for writing packets. At least this time we have the MSK protocol side functioning ok.

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA19782; Thu, 5 Mar 92 15:41:52 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 8571; Thu, 05 Mar 92 15:37:44 EST

Date: Thu, 1992 Mar 5 15:34 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Joe Doupnik <JRD@cc.usu.edu>,
SMKERMIT <SMKERMIT%UQAM@CUVMB.CC.COLUMBIA.EDU>,
fdc <fdc@watsun.cc.columbia.edu>

Subject: Re: Packets via 3270 systems, cont'd

In-Reply-To: JRD@cc.usu.edu message of Thu, 5 Mar 1992 13:21 MDT

Message-Id: <PEPMNT.920305.153440.RC0@CFAAMP.HARVARD.EDU>

> >Will Kermit know what to do if the screen goes into HOLDING status when
> >a message is received? I understand standard VM insists on a CLEAR
> >command to get rid of such a message.

The screen will never get into HOLDING, nor into MORE... K-370
disables inter-user messages (except, I suppose, breakin messages from
the operator saying "system down in 2 minutes -- logoff now!"). Also
K-370 clears the screen after every exchange of packets.

John

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA19478; Thu, 5 Mar 92 15:22:37 EST
Date: Thu, 5 Mar 1992 13:21 MDT
From: Joe Douppnik <JRD@cc.usu.edu>
Subject: Packets via 3270 systems, cont'd
To: smkermit@uqam.BITNET
Cc: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu
Message-Id: <100B9B542441D6FC@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"smkermit@uqam.bitnet"
X-Vms-Cc: IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu"

>On Wed, 4 Mar 1992 22:50 MDT you said:

>>John and the group,

>> So we keep packets safely within the 80 byte lines and hope that the
>>IBM mainframe side provides safe screen areas for writing packets. At least
>>this time we have the MSK protocol side functioning ok.

>> Joe D.

>

>Will Kermit know what to do if the screen goes into HOLDING status when
>a message is received? I understand standard VM insists on a CLEAR
>command to get rid of such a message. At UQAM, an ENTER character is
>sufficient to get into MORE status. That way, people who can't figure
>out how to clear the screen can just wait for one minute, and let the
>host do it.

Peter,

I'll show my complete ignorance of 3270 front end particulars by
saying I haven't the foggiest idea of what these keywords mean. The MSK
packet recognizer has no conception of this stuff, mostly because ignorance
is bliss and partly because packet recognizing is hardly the same as behaving
as a interactive 327x terminal. Your comments, however, are of interest here
for purposes of general education on the 327x objects, for other purposes.
I should add that, clearly, during packet transfers people are not in control,
and a minute's pause will kill transfers (not to mention generating calls to
systems managers about crummy service).

Joe D.

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA15971; Sat, 15 Feb 92 14:57:23 EST
Date: Sat, 15 Feb 1992 12:56 MDT
From: Joe Doupnik <JRD@cc.usu.edu>
Subject: Re: Kermit vs Procomm
To: fdc@watsun.cc.columbia.edu
Message-Id: <1E5F1456ECE052C8@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"fdc@watsun.cc.columbia.edu"
X-Vms-Cc: JRD

Frank,

Procomm. Sigh. Yes, those guys don't pay much attention, and they don't even ask behind the scenes. MSK server mode thinks it should do what the client asks, willy nilly.

Doomsday material: After digesting John C.'s logs I decided I could accomodate his problems without introducing new ones. I did add a buffer clear before starting a SEND operation; that's just a small help. Then I adopted his suggestion on the heuristic of seeing S/I/N packets and deducing the block check size. If an S/I packet is seen then do a block check of 1 on that packet alone, so a much delayed echo is accepted but does not upset the block check for all others. If a Nak arrives then it can make a permanent shift of block length. The difference between this and previously is the older code let S/I do a permanent shift.

Joe D.

1. Set Block = B

2. Make sure plsts are
~~shorter~~ \leq stated length

Return-Path: <pepmnt>

Received: by watsun.cc.columbia.edu (5.59/FCB)

id AA24936; Thu, 13 Feb 92 03:27:33 EST

Date: Thu, 13 Feb 92 3:27:31 EST

From: John Chandler <pepmnt@watsun.cc.columbia.edu>

To: fdc

Subject: Tried C-Kermit on 7171

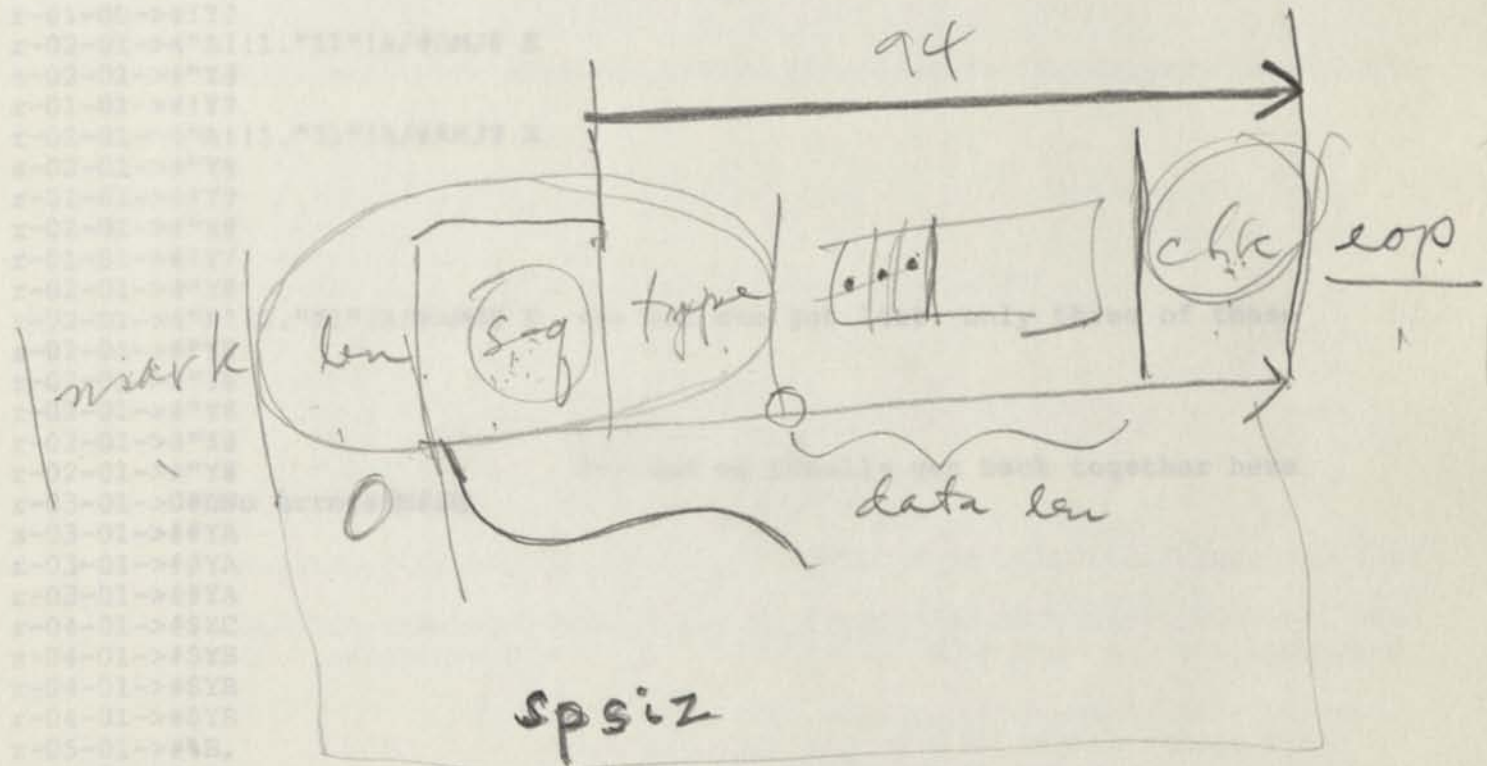
Message-Id: <CMM.0.90.0.697969651.pepmnt@watsun.cc.columbia.edu>

Frank,

This didn't work as well as expected. I could probably have gotten the uploads to work by setting the packet size a bit smaller, but I took the time to annotate and condense a packet log covering various things I noticed in this session. The log is in my directory (named packet.log).

Regarding packet sizes: I take the view that we should be cautious about exceeding the stated size (applied to the entire packet, including the SOP, and even including the padding, if any). This means that Kermit can't really create packets with a length code of ~ within the confines of normal short-packet protocol -- the two bytes that don't count in tallying up the length code must nevertheless be included in the true packet length for the purposes of preventing buffer overrun. It's too bad, but I think allowing the user to specify an absolute length limit without having to remember some funny offset is more important than squeezing the last bit of efficiency out of short-packet protocol.

John



data len = spsiz

94 → 98

126
32
94

```

s-00-02->0 Iy* @-#&l~N! y,
r-00-02->0 Iy* @-#&l~N! y,
r-00-02->0 Iy* @-#&l~N! y,
r-00-02->- Yp% @-#&l~H(
s-00-02->% KstX
r-00-02->% KstX
r-00-02->% KstX
r-00-02->- Sp% @-#&l~H"          <-- ok so far
s-00-02->0 Yy* @-#&l~N! y<
r-00-00->0 Yy* @-#&l~N! ops, esc sequence interruption
r-xx-00-<crunched:chk1>
s-01-00->#!N4          <-- shouldn't do that, it was an echo
r-00-00->0 Yy* @-#&l~N! y<
s-00-00-># Y>          <-- nor that, another echo!
r-01-00->#!X>
s-01-00->#!Y?
r-01-00->#!N4          <-- uh oh. echoes from the extra packets...
s-01-00->#!Y?          <-- this is getting out of hand
r-00-00-># Y>
r-01-00->#!N4
s-01-00->#!Y?
r-00-00-># Y>
r-01-00->#!X>
s-01-00->#!Y?          <-- that makes four of these sent
r-01-00->#!Y?
r-01-00->#!Y?
r-02-01->4"A!!1."I1"!A/#AMJ@ E
s-02-01->#"Y@
r-01-01->#!Y?
r-02-01->4"A!!1."I1"!A/#AMJ@ E
s-02-01->#"Y@
r-01-01->#!Y?
r-02-01->#"Y@
r-01-01->#!Y?
r-02-01->#"Y@
r-02-01->4"A!!1."I1"!A/#AMJ@ E <-- but one got lost: only three of these
s-02-01->#"Y@
r-02-01->#"Y@
r-02-01->#"Y@
r-02-01->#"Y@
r-02-01->#"Y@          <-- and we finally get back together here
r-03-01->0#DNo errors#M#JQ
s-03-01->##YA
r-03-01->##YA
r-03-01->##YA
r-04-01->#$ZC
s-04-01->#$YB
r-04-01->#$YB
r-04-01->#$YB
r-05-01->##B,
s-05-01->##YC
s-00-09->0 Iy* @-#&l~N! y,
r-00-09->0 Iy* @-#&l~N! y,
r-00-09->0 Iy* @-#&l~N! y,
r-00-09->- Yp% @-#&l~H(
s-00-09->% KstX
r-00-09->% Ksr-xx-09-<crunched:chk1>          <-- uh oh. here we go again.
s-00-09-># N3
r-00-09->% KstX
s-00-09->-BoEgheexpectedepawkeddingpeMis msg.

..... <-- doing a download...
s-33-07->#AY_
r-33-07->#AY_
r-33-07->#AY_
r-34-07->gBD#@#P#P@#@#@.#NRHDTW#O~#@#@#A-#@#@#@@ ~H@#NREYTprrp&#Bcgc#O@~##@@@#@8@
8

```



```

s-34-07->#BY
r-34-07->#BY
r-34-07->#BY      <-- note that the screen optimizer is likely to
r-xx-07-<crunched:chk1>      do this every time!
s-35-07->#CNV      Shouldn't complain here.
r-35-07->gCD@#@AG&p&p#L#F#NREYTIc&#@#P#PlP&#L&#XO&X B#O'A#@#CZ#J&K#^#D#X&l#X#AA
?
s-35-07->#CY!
r-35-07->#CNV
s-35-07->#CY!
r-35-07->#CY!
r-35-07->#CNV
s-35-07->#CY!
r-35-07->#CY!
r-35-07->gCD@#@AG&p&p#L#F#NREYTIc&#@#P#PlP&#L&#XO&X B#O'A#@#CZ#J&K#^#D#X&l#X#AA
?
s-35-07->#CY!
r-35-07->#CY!
r-35-07->#CY!
r-35-07->#CY!
r-35-07->#CY!
r-36-07->gDD#P#P#J&X#[&#?#P#N#P#N#X&pZ#@&Bd&#P&p&8(P&p&8 #NREYTpPrq&#Bcgc#O@#@#@
L
s-36-07->#DY#
r-36-07->#DY#
r-35-08->#CY!
r-36-08->#DY#
r-36-08->gDD#P#P#J&X#[&#?#P#N#P#N#X&pZ#@&Bd&#P&p&8(P&p&8 #NREYTpPrq&#Bcgc#O@#@#@
L
.....

s-00-23->0 Sy* @-#&l~N! y6
r-00-23->0 Sy* @-#&l~N! y6
r-00-23->0 Sy* @-#&l~N! y6
r-00-23->- Yp% @-#&l~N! y6      <-- note: buffer size = 80
s-01-23->*!FKF.TEXT7
r-01-00->*!FKF.TEXT7
r-01-00->*!FKF.TEXT7
r-01-00->!/YKF TEXT$0 A U
s-02-00->G"A."U1""B8#119920205 18:28:10!!31$2400(
r-02-00->G"A."U1""B8#119920205 18:28:10!!31$2400(
r-02-00->G"A."U1""B8#119920205 18:28:10!!31$2400(
r-02-00->#"Y@
s-03-00->o#D#B&E&b&D~&@#@0@#@#@A#NREYTIc#O@~$#@#@#@B#N CVTTVU&@&@#@&#@&#B &@&#
@#P&#P&+(
r-03-00->o#D#B&E&b&D~&@#@0@#@#@A#NREYTIc#O@~$#@#@#@B#N CVTTVU&@&@#@&#@&#B &@&#
@#P&#P&+r-xx-00-<crunched:chk1>      <-- C-K tried to send 81, not including
EOP
      This will always get crunched on echoing,
      but that wouldn't be fatal, except that
      C-K keeps sending more copies.
S-03-00->~o£D£B;Â!â;D~;À£À0ÀÀ£À£A£NÒAYÔ£c£İÄÄ~$£ÄÀ£À£B£N ÄVÔÖVU;À;À;£Ä;£Ä;£B ;Ä;£
Ä£P;£P;+(
      <-- what happened here? Doubt about parity?
r-03-00->o#D#B&E&b&D~&@#@0@#@#@A#NREYTIc#O@~$#@#@#@B#N CVTTVU&@&@#@&#@&#B &@&#
@#P&#P&+r-xx-00-<crunched:chk1>
      <-- eventually, we get synched anyway
...
r-06-07->#&YD
s-07-07->3'D0#Q@#@Bha(8#NRENpacket here?
r-07-07->3'D@#O@#@#@h~(@#NREN      <-- is this the automatic size adjuster?
r-07-07->3'D@#O@#@#@h~(@#NREN      <-- if so, it succeeded!
r-07-07->#'YE
s-08-07->8(DYT~#pr&#BEbD#O~&@#@0@T
r-08-08->8(DYT~#pr&#BEbD#O~&@#@0@T
r-08-08->8(DYT~#pr&#BEbD#O~&@#@0@T
r-08-08->#(YF

```


s-09-08-><)D@#@G#NEYYTbGb#O@#@#@#U (@8
r-09-08-><)D@#@G#NEYYTbGb#O@#@#@#U (@8
r-09-08-><)D@#@G#NEYYTbGb#O@#@#@#U (@8
r-09-08->#) YG
s-10-08->A*D#@A#NbDbUcYF#O@#@#@#@W#P#P#@A
r-10-08->A*D#@A#NbDbUcYF#O@#@#@#@W#P#P#@A
r-10-08->A*D#@A#NbDbUcYF#O@#@#@#@W#P#P#@A
r-10-08->#*YH
s-11-08->G+D#GJ#NbITSaC#O@#@#@#@#^`@#@#@&m~ (@#NRET
r-11-08->G+D#GJ#NbITSaC#O@#@#@#@#^`@#@#@&m~ (@#NRET
r-11-08->G+D#GJ#NbITSaC#O@#@#@#@#^`@#@#@&m~ (@#NRET
r-11-08->#+YI
s-12-08->P,DYT~#ps&#BEbD#O~&@#@0@#@#@J&b&E&c~%@#@#@#_P@#@Z
r-12-08->P,DYT~#ps&#BEbD#O~&@#@0@#@#@J&b&E&c~%@#@#@#_P@#@Z
r-12-08->P,DYT~#ps&#BEbD#O~&@#@0@#@#@J&b&E&c~%@#@#@#_P@#@Z
r-12-08->#,YJ
s-13-08->Y-D#M"#NbEcCTDb#O@#@#@#@,x@#@#@H|&b&H&V&f~\$@#@A#@\$x@#@#@#@J~ (@7
r-13-08->Y-D#M"#NbEcCTDb#O@#@#@#@,x@#@#@H|&b&H&V&f~\$@#@A#@\$x@#@#@#@J~ (@7
r-13-09->Y-D#M"#NbEcCTDb#O@#@#@#@,x@#@#@H|&b&H&V&f~\$@#@A#@\$x@#@#@#@J~ (@7
r-13-09->#-YK
s-14-09->j.D#NREYT~#pt&#BEbD#O~&@#@0@#@#@#L#NbcAcdb&@#@#@#@&@&4x&@#@#@#@C#~WETbG#O
~#@?
r-14-09->j.D#NREYT~#pt&#BEbD#O~&@#@0@#@#@#L#NbcAcdb&@#@#@#@&@&4x&@#@#@#@C#~WETbG#O
~#@?
r-14-09->j.D#NREYT~#pt&#BEbD#O~&@#@0@#@#@#L#NbcAcdb&@#@#@#@&@&4x&@#@#@#@C#~WETbG#O
~#@?
r-14-09->#.YL
s-15-09->m/D#A#@7#F@#@#@#@#L&D&d&T&W~\$@#@#@8&x@#@#@C&k~ (@#NREYT~#pu&#BEbD#O~&@#@0@#@
#@#@P#N&
r-15-09->m/D#A#@7#F@#@#@#@#L&D&d&T&W~\$@#@#@8&x@#@#@C&k~ (@#NREYT~#pu&#BEbD#O~&@#@0@#@
#@#@P#N&
r-15-09->m/D#A#@7#F@#@#@#@#L&D&d&T&W~\$@#@#@8&x@#@#@C&k~ (@#NREYT~#pu&#BEbD#O~&@#@0@#@
#@#@P#N&
r-15-09->#/YM
s-16-09->m0D#NDdTWcVD&@#@#A#@&:N&@#@#@#@#@#MGIEcAB&@#@#@#@#A#@&:x&@#@#@#@#@#MVWcWRc&
@&@#@#@K
r-16-09->m0D#NDdTWcVD&@#@#A#@&:N&@#@#@#@#@#MGIEcAB&@#@#@#@#A#@&:x&@#@#@#@#@#MVWcWRc&
@&@#@#@K
r-16-09->m0D#NDdTWcVD&@#@#A#@&:N&@#@#@#@#@#MGIEcAB&@#@#@#@#A#@&:x&@#@#@#@#@#MVWcWRc&
@&@#@#@K
r-16-09->#0YN
s-17-09->n1D&#@&<h&@#@#@#@h~ (@#REYT~#pv&#BEbD#O~&@#@0@#@#@#@P#O#NGEUCTD&@#@#@#@&@&
=P&@#@#@\$
r-17-10->n1D&#@&<h&@#@#@#@h~ (@#REYT~#pv&#BEbD#O~&@#@0@#@#@#@P#O#NGEUCTD&@#@#@#@&@&
=P&@#@#@\$
r-17-10->n1D&#@&<h&@#@#@#@h~ (@#REYT~#pv&#BEbD#O~&@#@0@#@#@#@P#O#NGEUCTD&@#@#@#@&@&
=P&@#@#@\$
r-17-10->#1YO
s-18-10->o2D&#C' cBSbEc#O@#@#@#@Ax@#@#@I!#NcBSAcc&@#@#@#@#A#@&B#V#O@#@#@#@P#P~ (@#NREYT
~#pw&#BE^
r-18-10->o2D&#C' cBSbEc#O@#@#@#@Ax@#@#@I!#NcBSAcc&@#@#@#@#A#@&B#V#O@#@#@#@P#P~ (@#NREYT
~#pw&#BEr-xx-10-<crunched:chk1> <-- but we eventually get back to 81 aga
in...

<-- this keeps cycling...

.....
s-00-47->0 Sy* @-#&1~N! y6
r-00-47->0 Sy* @-#&1~N! y6
r-32-47->F@D&#BEbD#O~&@#@0@#@#@#@J&b&E&U&D~\$@#@#@@,
s-00-47@e\$ Ehmsemah anythingmbpec181?

Return-Path: <JRD@cc.usu.edu>

Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA16390; Fri, 14 Feb 92 01:10:20 EST

Date: Thu, 13 Feb 1992 23:10 MDT

From: Joe Doupnik <JRD@cc.usu.edu>

Subject: What John's commentary was about

To: fdc@watsun.cc.columbia.edu

Message-Id: <E1D4F0807CE01645@CC.USU.EDU>

X-Envelope-To: fdc@watsun.cc.columbia.edu

X-Vms-To: IN%"fdc@watsun.cc.columbia.edu"

From: IN%"PEPMNT@CFAAMP.HARVARD.EDU" "John F. Chandler" 13-FEB-1992 22:32:02.53

To: jrd <jrd@CC.USU.EDU>

CC:

Subj: Strange happenings with 3.12

Return-path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by CC.USU.EDU with PMDF#10716; Thu, 13 Feb 1992 22:31 MDT

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 8769; Fri, 14 Feb 92 00:27:34 EST

Date: Thu, 1992 Feb 13 23:54 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

Subject: Strange happenings with 3.12

To: jrd <jrd@CC.USU.EDU>

Message-id: <PEPMNT.920213.235402.RC0@CFAAMP.HARVARD.EDU>

X-Envelope-to: jrd

Joe,

Here is an annotated log that illustrates some odd things I saw in attempting to do an upload with printable SOP. The salient points are two:

- 1) it would be nice if MSK could flush any stacked up packets from a server before starting to communicate with the server, and
- 2) it seems that MSK suddenly changed its mind from 3-byte checksums back to 1-byte after already having sent an F-packet.

I admit this log is ugly because of all the echoed packets, but I think the annotations should help focus on the salient points. The thing to remember is that every packet going into the mainframe gets echoed twice. When the mainframe is ready for the next input packet, it usually, but not quite always, writes RUNNING and then VM READ at the lower right-hand corner of the screen. The mainframe Kermit is actually trying to send escapes for the EOP (per request from MSK), but the full-screen environment quietly changes them all to blanks before they reach the terminal.

John

.. end of previous transaction (a download) -- note packet char is ">" and MSK is expecting ^[as the EOP (the protocol converter is always tossing around escape sequences, so this works well).

Rpack: [23;1H^[[K^[[24;61HRUNNING^[[1;1H>% 'B#6K

Spack: >% 'Y)OY^M

Ok, that transaction ended, and I then told MSK to upload...

Spack: >0 S~(@;#Y3~N"*R ^M

Rpack: !^[[2;1H^[[K^[[24;61HVM READ^[[23;1H^[[0m>% 'Y)OY^[

<Crunched packet>

This was a left-over echo still on the line from the final ACK of the previous transfer. It would be nice if it could have been flushed.


```

Spack: >0 S~( @;#Y3~N"*R ^M          <-- second S-init wasn't wanted
Rpack: [24;80H^[[2;1H^[[0m>%Y)OY^[[    <-- The 2nd left-over echo
<Crunched packet>                        <-- also could have been flushed.
Spack: >0 S~( @;#Y3~N"*R ^M          <-- third S-init packet
                                         Now, there will be *six* echoes.
Rpack: [23;1H^[[K^[[24;61HRUNNING^[[24;61HVM READ^[[23;1H^[[0m>0 S~( @
;#Y3~N"*R ^[[
<Echo of sent packet>
Rpack: [24;80H^[[4;1H^[[0m>0 S~( @;#Y3~N"*R^[[
<Crunched packet>
<Echo of sent packet>
Rpack: [23;1H^[[0m>0 S~( @;#Y3~      ^[[    <-- well, something ate one echo
<Crunched packet>
<Echo of sent packet>
Rpack: [1;1H^[[0m>- Yp% @-#&3~H*          <-- ACK to 1st S
Spack: >+!FKF.TEX 2^^M                    <-- note the 3-byte checksum
Rpack: !^[[2;1H^[[K>0 S~( @;#Y3~N"*R^[[
<Crunched packet>
Spack: >)!FKF.TEX!^M                      <-- oops, it changed to 1-byte
Rpack: [3;1H>- Yp% @-#&3~H*                <-- ACK to 2nd S
Spack: >)!FKF.TEX!^M
Rpack: !^[[4;1H^[[K>0 S~( @;#Y3~N"*R^[[
<Crunched packet>
Spack: >)!FKF.TEX!^M
Rpack: [5;1H>- Yp% @-#&3~H*                <-- ACK to 3rd S
Spack: >)!FKF.TEX!^M                      <-- 5th F packet: will be *ten* echoes
Rpack: !^[[23;1H^[[K^[[24;61HVM READ^[[23;1H^[[0m>+!FKF.TEX 2^^[[
<Crunched packet>                        <-- perfect echo, but checksum changed
<Echo of sent packet>
Rpack: [24;80H^[[6;1H^[[0m>+!FKF.TEX 2^^[[
<Crunched packet>
<Echo of sent packet>
Rpack: [23;1H^[[K^[[0m>)!^[[1;1H^[[0m>.!YKF TEX A /DR !^[[2;1H^[[K>)!F
KF.TEX!^[[3;1H^[[K>.!YKF TEX A /DR !^[[4;1H^[[K^[[5;1H^[[K^[[6;1H^[[K^
[[23;1H^[[0m>)!FKF.TEX!>)!FKF.TEX!^[[24;61H^[[0mVM READ^[[4;1H>)!FKF.T
EX!>)!FKF.TEX!^[[5;1H>.!YKF TEX A /DR !^[[6;1H>)!FKF.TEX!^[[7;1H>.!YKF
TEX A /DR !^[[23;1H^[[K^[[24;61HVM READ^[[23;1H
                                         Why didn't MSK detect all the packets and echoes in
                                         this batch?
<Timeout>
Spack: >)!FKF.TEX!^M
Rpack: ^[[0m>)!FKF.TEX!^[[
<Echo of sent packet>
Rpack: [24;80H^[[8;1H^[[0m>)!FKF.TEX!^[[
<Echo of sent packet>
Rpack: [23;1H^[[K^[[9;1H>.!YKF TEX A /DR <-- What was wrong with this one?
<Crunched packet>
    I guess it's the checksum problem.
Spack: >3!EToo many retries ^M

```


Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA28474; Fri, 14 Feb 92 14:01:21 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 9075; Fri, 14 Feb 92 13:57:37 EST

Date: Fri, 1992 Feb 14 13:22 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Joe Doupnik <JRD@cc.usu.edu>, fdc <fdc@watsun.cc.columbia.edu>

Subject: Re: Echoing galore

In-Reply-To: JRD@cc.usu.edu message of Thu, 13 Feb 1992 22:59 MDT

Message-Id: <PEPMNT.920214.132206.RC0@CFAAMP.HARVARD.EDU>

> the use of ESC as an end of packet code and as a stray item
> from the screen formatter is pushing things rather hard.

Ok, I had thought it would be useful to have MSK find the next byte after each echoed packet "happen" to be the expected EOP, but I see now by experimentation that it doesn't seem to make any difference. Indeed, I get exactly the same problems without setting the EOP to ESC. The packet log looks virtually identical (except that the S-init packets ask for CR instead of ESC). MSK just grabs one more character after the checksum and writes it into the packet log as if it were the expected EOP, no matter what it is.

> try to discover the checksum length by spying
> on S, I, and NAK packets. This section is really quite old but was awakened by
> all the ESC + echo clutter.

In fact, it gets triggered by the echoes from the extra copies of the S-init packet, which MSK is sending because it didn't flush the comm buffer before doing a SEND. I played around awhile and found that MSK *does* seem to flush the buffer each time before sending an I-packet, but never before an S-packet. Is this by design? I haven't thought through the issues, but it seems to me that it should be safe to flush the buffer before starting any fresh transaction. At any rate, though, there are two workarounds, so I'm not going to push the thesis that the heuristic should be modified somehow to avoid permanently re-setting the checksum type because of a packet that is going to be ignored (making such a change might be tricky, to say the least!).

The first workaround, of course, is to set BLOCK to 1, so that the heuristic doesn't cause any actual change in the checksum. As it happens (and as the packet log shows), there is a tendency for the extra packets running back and forth to get eaten up (on a 7171, at least) because the 7171 is being "overworked" and tries to catch up by dropping a byte or several through optimizing. Thus, all Kermit needs is to get through the first five packets, and the transfer settles down to normal behavior. That's actually very encouraging.

The second workaround is to issue a no-response REMOTE command before any upload to a server -- the REMOTE command flushes the comm buffer quite nicely.

Meanwhile, I have been following the progress of the 3708 tester. He has now installed both K-370 4.2.3 and MSK 3.1.2 and gotten as far as failing to download a file (sigh). I'll let you know how it turns out.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA07371; Mon, 10 Feb 92 19:24:03 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 6891; Mon, 10 Feb 92 19:20:22 EST

Date: Mon, 1992 Feb 10 19:01 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Joe Doupnik <JRD@cc.usu.edu>

Subject: Re: New C-Kermit

In-Reply-To: fdc@watsun.cc.columbia.edu message of Mon, 10 Feb 92 13:05:20 EST

Message-Id: <PEPMNT.920210.190131.RC0@CFAAMP.HARVARD.EDU>

> > Where do we stand on releasing this stuff? I think I have a 3708 tester
> > lined up. Should I send him a copy of the Feb 4 version of MSK 3.12?
> >
> Sure, but there's an even newer one in ~jrd/msk312.exe.

I just heard that CK 5A(179) is actually released and calls itself BETA.
Excellent news! If some form of MSK 3.12 were similarly available, even
if described as ALPHA, it would make testing the new feature simpler.
I have sent one copy of the Feb 9 version of 3.12 to the 3708 guy, along
with the necessary CMS Kermit updates, but it would also be nice to put
some form of test release out on KERMSRV and WATSUN. I'm thinking of
two different forms: an incremental set of updates called b/ik*ker.up3
and a cumulative set called test/ik*ker.upd. I could make a case for
doing both. What do you think?

> So your connection from MSK to the terminal server is serial, right?

Right.

> must have local flow control turned on, and it has Xoff'd MSK, but never
> sends the Xon, at least not within MSK's (rather lengthy) timeout period
> for recovering from flow-control deadlocks.

Yes, that seems consistent with the symptoms. I'll try turning flow
control off. Thanks for the tip.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA07965; Wed, 5 Feb 92 18:53:16 EST
Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 4825; Wed, 05 Feb 92 18:49:43 EST
Date: Wed, 1992 Feb 5 18:47 EST
From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
To: Frank da Cruz <fdc@watsun.cc.columbia.edu>
Subject: New KF module
Message-Id: <PEPMNT.920205.184730.RC0@CFAAMP.HARVARD.EDU>

Frank, I've regenerated KF. It now accepts CONTROLLER=FULL as an
excuse to allow non-control settings for packet and handshake chars.
I also checked on the 4994's handling of trailing blanks and found
them to be chopped off pretty thoroughly.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA24529; Thu, 6 Feb 92 15:23:49 EST
Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 5216; Thu, 06 Feb 92 15:20:14 EST
Date: Thu, 1992 Feb 6 14:23 EST
From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
To: Frank da Cruz <fdc@watsun.cc.columbia.edu>
Subject: 7171 and Doomsday Kermit
Message-Id: <PEPMNT.920206.142342.RC0@CFAAMP.HARVARD.EDU>

Frank,

It just struck me why the 7171 emitted the trailing blank that so puzzled me before. It's not something you can count on -- it just happens that every packet gets written in the same place on the screen, and the previous packet had a non-blank in the position where the new blank was to appear. Rather than actually CLEAR the whole screen (as Kermit instructed), the 7171 decided to write one more blank to cover up the last trace of evidence that the screen had *not* been CLEAR'ed. It makes sense when you think about it. The upshot is that it really is necessary to SET HAND 33 in K-370 for FULLSCREEN mode.

I got a report on the 3708 -- it seems to make no attempt to optimize away imbedded blanks. Therefore, the new mode should work. I broadcast a request for reports on any and all other protocol converters. We'll see what comes up.

I tested Joe's latest MSK against my latest KF module, and there seem to be no clashes in transmitting MSBPCT.EXE back and forth. Speaking of KF, I updated FULL TAKE on JCHBN 191 to take advantage of the special dispensation for CONT=FULL.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA06031; Wed, 5 Feb 92 16:56:35 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 4777; Wed, 05 Feb 92 16:52:59 EST

Date: Wed, 1992 Feb 5 16:16 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>

Subject: Re: Success with C-Kermit and 7171

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 5 Feb 92 14:32:14 EST

Message-Id: <PEPMNT.920205.161625.RC0@CFAAMP.HARVARD.EDU>

> C-Kermit actually received the trailing blank. It's in my packet and
> debug logs.

Wow. The 7171 is certainly literal!

> Like don't
> require SET TEST ON / OFF around packet-start selection (at least not if
> CONTROLLER is FULL),

Now, that's a good idea. I'll put that in right away.

> maybe make SET ETOA 74 92 the default for this too?

No, that's front-end-specific. The 3708, in particular, probably doesn't have that problem.

> Then we'll give them some TAKE files and/or macros to make it easy to set up.

Sounds good. I guess the thing to do is collect all the post-4.2.2 updates into a file called something like test/ikcker.up3 -- that's simpler than figuring out for sure whether there are any dependencies of the new feature on intervening updates. I'll make a new KF presently with the SET TEST requirement eased.

John

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA03633; Wed, 5 Feb 92 14:32:15 EST
Date: Wed, 5 Feb 92 14:32:14 EST
From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
To: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
Cc: Joe Doupnik <JRD@cc.usu.edu>,
Christine M Gianone <cmg@watsun.cc.columbia.edu>
Subject: Re: Success with C-Kermit and 7171
In-Reply-To: Your message of Wed, 1992 Feb 5 13:14 EST
Message-Id: <CMM.0.90.0.697318334.fdc@watsun.cc.columbia.edu>

> > set test on ; To allow printable SOP.
> > set controller full ; Don't use transparent mode.
>
> Not to mention SET TEST OFF! Also, there is no need to set the packet
> length to K-370, since it automatically limits that to 79 with CON FULL.
>

Right, I didn't mention it, but I did it...

> Were they actually inserted into echoed packets, as you noted before, or
> has that problem gone away with the other adjustments?
>

The problem went away. If they had been in the packets, C-Kermit would have gagged.

> > r-08-01-:?(DCP SET EMSG ON'#M#J'CP SPOOL
>
> This is puzzling. I would have expected a problem here from the screen
> optimizer in the 7171. Note that the packet ends with a blank checksum...
> ... I would have expected the 7171 to omit the trailing blank as well. Did
> C-K somehow restore the blank, or did the 7171 leave it in?
>
C-Kermit actually received the trailing blank. It's in my packet and debug logs.

> > So Doomsday Kermit works, so far. Now on to the 3708? - Frank
>

> I see two avenues here. One is to release new test versions of K-370,
> MSK, and CK to make them available through normal channels to
> potential testers.
>

This is not out of the question. I think both C-K 5A BETA and MSK 3.12 can be ready to go within a few days (Joe ???), at least for public testing. However, there are so many SET commands required to make this stuff work that we might want to make it a little easier to use in real life. Like don't require SET TEST ON / OFF around packet-start selection (at least not if CONTROLLER is FULL), and maybe make SET ETOA 74 92 the default for this too? Then we'll give them some TAKE files and/or macros to make it easy to set up.

- Frank

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA03427; Wed, 5 Feb 92 14:19:40 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 4684; Wed, 05 Feb 92 14:16:05 EST

Date: Wed, 1992 Feb 5 13:14 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,

Joe Doupnik <JRD@cc.usu.edu>,

Christine M Gianone <cmg@watsun.cc.columbia.edu>

Subject: Re: Success with C-Kermit and 7171

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 5 Feb 92 10:58:34 EST

Message-Id: <PEPMNT.920205.131424.RC0@CFAAMP.HARVARD.EDU>

> And to John's test version of CMS Kermit, I said:

>

> set test on ; To allow printable SOP.
> set send start-of-packet 58 ; Use colon as SOP
> set receive start-of-packet 58 ; in both directions.
> set receive packet-length 70 ; Short packets.
> set etoa 74 92 ; Translate cent sign to backslash.
> set controller full ; Don't use transparent mode.

Not to mention SET TEST OFF! Also, there is no need to set the packet length to K-370, since it automatically limits that to 79 with CON FULL.

> don't show up here, there were many VT100 escape sequences (cursor
> positioning, set graphic rendition) intermingled with the packets, which were
> successfully ignored.

Were they actually inserted into echoed packets, as you noted before, or has that problem gone away with the other adjustments?

> r-08-01-:?(DCP SET EMSG ON'#M#J'CP SPOOL

This is puzzling. I would have expected a problem here from the screen optimizer in the 7171. Note that the packet ends with a blank checksum which is, in turn, followed in K-370's output buffer by a CR and an XON. The latter two are filtered out, so I would have expected the 7171 to omit the trailing blank as well. Did C-K somehow restore the blank, or did the 7171 leave it in? To be safe, you can tell K-370 to use a printable "handshake" instead of the XON, and then trailing blanks are protected.

> So Doomsday Kermit works, so far. Now on to the 3708? - Frank

I see two avenues here. One is to release new test versions of K-370, MSK, and CK to make them available through normal channels to potential testers. The other is to line up testers and feed them special copies of whichever ones they want. I've already asked one potential 3708 tester to run a feasibility test to see how zealous the 3708 screen optimizer is. When (and if) he responds, I'll be clearer on the likelihood that this will work for the 3708. For that matter, there are lots of other boxes out there, like pre-B 3174's and the infamous Micom 7400, that might benefit from this mode.

John

Return-Path: <cmg>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA02779; Wed, 5 Feb 92 13:51:02 EST
Date: Wed, 5 Feb 92 13:51:00 EST
From: Christine M Gianone <cmg@watsun.cc.columbia.edu>
To: John Chandler <pepmnt@cfaamp.bitnet>
Subject: 3708 Messages
Message-Id: <CMM.0.90.0.697315860.cmg@watsun.cc.columbia.edu>

Hi John, thought you might want to respond to the following two messages
since you were looking for people to test Kermit with the 3708..... /Chris

Date: Tue, 4 Feb 1992 13:39:46 EST
Sender: IBM Mainframe Discussion list <IBM-MAIN@RICEVM1.BITNET>
From: Lee Garner <GARNER@FSUAVM.BITNET>
Subject: Kermit over IBM 3708 attachment?
To: George Giraldi <GEOCU@CUVMB.BITNET>

Hi,

Can anyone tell me if CMS Kermit can transfer files to/from
a PC that is attached via an IBM 3708 Network Conversion Unit (just
a fancy name for "protocol convertor")?

I understand that the 3708 does have a data transparency feature,
but I'm not sure how close it is to 7171 transparency, or whether or
not Kermit knows how to take advantage of it.

Thanks for any help,

- Lee

| | | |
|---|------------------------------|--|
| Lee Garner (Sys prog & IBM systems manager) | BITNet: garner@fsuavm | |
| Computing Center, Florida State University | I'net: garner@avm.cc.fsu.edu | |
| Tallahassee, Florida, USA | ATTnet: (904) 644-2591 | |

Date: Tue, 4 Feb 1992 14:40:00 -00
Sender: IBM Mainframe Discussion list <IBM-MAIN@RICEVM1.BITNET>
From: 'Gene Mangum' <H198@HOSP.MED.UMICH.EDU>
Subject: Kermit over IBM 3708 attachme

*** Original Author: GARNER @ FSUAVM - ** Remote User **: 02/04/92 01:49pm

>Date: Tue, 4 Feb 1992 13:39:46 EST
>Sender: IBM Mainframe Discussion List
> <IBM-MAIN@AKRONVM.BITNET@VM1.CC.UAKRON.EDU>
>From: Lee Garner <GARNER@FSUAVM.BITNET@VM1.CC.UAKRON.EDU>
>Subject: Kermit over IBM 3708 attachment?

>
>Hi,

>
> Can anyone tell me if CMS Kermit can transfer files to/from
>a PC that is attached via an IBM 3708 Network Conversion Unit (just
>a fancy name for "protocol convertor")?

>
> I understand that the 3708 does have a data transparency feature,
>but I'm not sure how close it is to 7171 transparency, or whether or
>not Kermit knows how to take advantage of it.

>
>Thanks for any help,

>

> - Lee

>

>

> Lee Garner (Sys prog & IBM systems manager) | BITNet: garner@fsuavm |
> Computing Center, Florida State University | I'net: garner@avm.cc.fsu.edu |
> Tallahassee, Florida, USA | ATTnet: (904) 644-2591 |

>

>

*** Comments from H198 - Mangum, Gene; 02/04/92 02:27pm:

I can't say about VM, but for MVS Kermit through the 3708, we had
to:

- Configure ports for line mode (no protocol conversion).

- Change the Kermit start-of-packet character - as I remember,
I couldn't get the 3708 to pass a X'01', the default start-
of-packet char, so I changed it to x'0C'.

If you can't get it going, EMail me and I'll try to dig up more
details.

Good Luck,
Gene Mangum
Univ. of Michigan Hospitals

Return-Path: <fdc>
 Received: by watsun.cc.columbia.edu (5.59/FCB)
 id AA29433; Wed, 5 Feb 92 10:58:35 EST
 Date: Wed, 5 Feb 92 10:58:34 EST
 From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
 To: John Chandler <PEPMNT@CFAAMP.HARVARD.EDU>
 Cc: Joe Doupnik <JRD@cc.usu.edu>,
 Christine M Gianone <cmg@watsun.cc.columbia.edu>
 Subject: Success with C-Kermit and 7171
 Message-Id: <CMM.0.90.0.697305514.fdc@watsun.cc.columbia.edu>

Thanks for the hints, John. I just transferred a file between C-Kermit and CMS through the 7171 in full-screen mode (no transparent mode!).

To my test version of C-Kermit, I said:

```
set send start-of-packet 58      ; Use colon as SOP
set receive start-of-packet 58   ; in both directions.
set rec packet-length 40         ; Use short packets.
set parity even                  ; Even parity.
set block 1                      ; 1-byte checksum.
set window 1                     ; No sliding windows.
```

And to John's test version of CMS Kermit, I said:

```
set test on                      ; To allow printable SOP.
set send start-of-packet 58      ; Use colon as SOP
set receive start-of-packet 58   ; in both directions.
set receive packet-length 70     ; Short packets.
set etoa 74 92                  ; Translate cent sign to backslash.
set controller full              ; Don't use transparent mode.
```

Here is C-Kermit's packet log. As you can see (and as you explained), the 7171 echoes each received packet twice, once in place, and once again to put it in the echo area. C-Kermit now ignores multiple echoes. Although they don't show up here, there were many VT100 escape sequences (cursor positioning, set graphic rendition) intermingled with the packets, which were successfully ignored.

```
s-00-12--:0 IH* @-#&1~J GB
r-00-12--:0 IH* @-#&1~J GB
r-00-12--:0 IH* @-#&1~J GB
r-00-13--:- Yf% @-#&1~H!
s-00-13--:/ Rprofile exec8
r-00-13--:/ Rprofile exec8
r-00-13--:/ Rprofile exec8
r-00-13--:- Sf% @-#&1~H[
s-00-13--:0 YG* @-#&1~J GQ
r-00-00--:0 YG* @-#&1~J GQ
r-00-00--:0 YG* @-#&1~J GQ
r-01-00--:!/FPROFILE.EXEC]
s-01-00--:!/Yprofile.execN
r-01-00--:!/Yprofile.execN
r-01-00--:!/Yprofile.execN
r-02-00--:2"A!!1."I1"!A/#AMJ" <-- Hey, look at this, multiple A-packets!
s-02-00--:#"Y@ <-- C-Kermit never received these before.
r-02-00--:#"Y@ <-- Nice to see that it works!
r-02-00--:#"Y@
r-03-00--:8#A#119911122 19:28:44@ \
s-03-00--:##YA
r-03-00--:##YA
r-03-01--:##YA
r-04-01--:?$D/* */#M#J'exifdsc rdr2mail fF
s-04-01--:#$YB
r-04-01--:#$YB
r-04-01--:#$YB
```


r-05-01-:?:%Ddc cunixf' #M#J'exifdsc CP LO=
s-05-01-:?:%YC
r-05-01-:?:%YC
r-05-01-:?:%YC
r-06-01-:?:&DGOFF' #M#J'CP TERM LINESIZE 8^
s-06-01-:?:&YD
r-06-01-:?:&YD
r-06-01-:?:&YD
r-07-01-:?:'D0' #M#J'SET RDYMSG LMSG' #M#J'F
s-07-01-:?:'YE
r-07-01-:?:'YE
r-07-01-:?:'YE
r-08-01-:?:(DCP SET EMSG ON' #M#J'CP SPOOL
s-08-01-:?:(YF
r-08-01-:?:(YF
r-08-01-:?:(YF
r-09-02-:?:)D RDR CLASS *' #M#J'CP Q RDR AL
s-09-02-:?:)YG
r-09-02-:?:)YG
r-09-02-:?:)YG
r-10-02-:?:*DLL *' #M#J'CP SET PF10 RET BA>
s-10-02-:?:*YH
r-10-02-:?:*YH
r-10-02-:?:*YH
r-11-02-:?:+DCK' #M#J'CP SET PF09 RET FORW0
s-11-02-:?:+YI
r-11-02-:?:+YI
r-11-02-:?:+YI
r-12-02-:?:@,D' #M#J'CP SET PF03 IMM quit' #M0
s-12-02-:?:#,YJ
r-12-02-:?:#,YJ
r-12-02-:?:#,YJ
r-13-02-:?:-D#J'CP SET PF01 help ' #M#J'CPC
s-13-02-:?:-YK
r-13-02-:?:-YK
r-13-02-:?:-YK
r-14-02-:?:.D LINK * 191 191 MW' #M#J'CP LO
s-14-02-:?:#.YL
r-14-02-:?:#.YL
r-14-03-:?:#.YL
r-15-03-:?:/DINK * 197 197 MW' #M#J'CP LINH
s-15-03-:?:#/YM
r-15-03-:?:#/YM
r-15-03-:?:#/YM
r-16-03-:?:?0DK * 193 193 MW' #M#J'GLOBAL M2
s-16-03-:?:#0YN
r-16-03-:?:#0YN
r-16-03-:?:#0YN
r-17-03-:?:?1DACLIB DMSSP CMSLIB TSOMAC OSY
s-17-03-:?:#1YO
r-17-03-:?:#1YO
r-17-03-:?:#1YO
r-18-03-:?:?2DMACRO' #M#J'ACCESS 193 C' #M#JH
s-18-03-:?:#2YP
r-18-03-:?:#2YP
r-18-03-:?:#2YP
r-19-03-:?:?3D'ACCESS 197 B' #M#J'ACCESS 199
s-19-03-:?:#3YQ
r-19-03-:?:#3YQ
r-19-03-:?:#3YQ
r-20-03-:?:+4D1 A' #M#JY
s-20-03-:?:#4YR
r-20-03-:?:#4YR
r-20-04-:?:#4YR
r-21-04-:?:#5ZT
s-21-04-:?:#5YS

```

r-21-04-: #5YS
r-21-04-: #5YS
r-22-04-: #6B=
s-22-04-: #6YT
s-00-14-: 0 IG* @-#&1~J GA
r-00-14-: 0 IG* @-#&1~J GA
r-00-14-: 0 IG* @-#&1~J GA
r-00-14-: - Yf% @-#&1~H!
s-00-14-: $ GF4
r-00-14-: $ GF4
r-00-14-: $ GF4
r-00-14-: # Y>

```

And here is KER LOG from the mainframe. Each of the "S:" packets ends with a Ctrl-Q (not shown):

```

R::0 IH* @-#&1~J GB
S::- Yf% @-#&1~H!
R::/ Rprofile exec8
S::- Sf% @-#&1~H[
R::0 YG* @-#&1~J GQ
S::/!FPROFILE.EXEC]
R::/!Yprofile.execN
S::2"A!!1."I1"!A/#AMJ"
R::#"Y@
S::8#A#119911122 19:28:44@ \
R::##YA
S::?$D/* */#M#J'exifdsc rdr2mail fF
R::#$YB
S::?$Ddc cunixf'#M#J'exifdsc CP LO=
R::#%YC
S::?$&DGOFF'#M#J'CP TERM LINESIZE 8^
R::#&YD
S::?'D0'#M#J'SET RDYMSG LMSG'#M#J'F
R::#'YE
S::?(DCP SET EMSG ON'#M#J'CP SPOOL
R::#(YF
S::?)D RDR CLASS *'#M#J'CP Q RDR AL
R::#)YG
S::?*DLL *'#M#J'CP SET PF10 RET BA>
R::#*YH
S::?+DCK'#M#J'CP SET PF09 RET FORW0
R::#+YI
S::@,D'#M#J'CP SET PF03 IMM quit'#M0
R::#,YJ
S::?-D#J'CP SET PF01 help '#M#J'CPC
R::#-YK
S::?.D LINK * 191 191 MW'#M#J'CP LO
R::#.YL
S::?/DINK * 197 197 MW'#M#J'CP LINH
R::#/YM
S::?0DK * 193 193 MW'#M#J'GLOBAL M2
R::#0YN
S::?1DACLIB DMSSP CMSLIB TSOMAC OSY
R::#1YO
S::?2DMACRO'#M#J'ACCESS 193 C'#M#JH
R::#2YP
S::?3D'ACCESS 197 B'#M#J'ACCESS 199
R::#3YQ
S::+4D1 A'#M#JY
R::#4YR
S::#5ZT
R::#5YS
S::#6B=
R::#6YT
R::0 IG* @-#&1~J GA

```


S::- Yf% @-#&l~H!

R::\$ GF4

S::# Y>

So Doomsday Kermit works, so far. Now on to the 3708? - Frank

Return-Path: <@cuvmb.cc.columbia.edu:IBM-KERM@CUVMA.BITNET>
Received: from cuvmb.cc.columbia.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA25634; Mon, 3 Feb 92 18:55:46 EST
Message-Id: <9202032355.AA25634@watsun.cc.columbia.edu>
Received: from CUVMB.COLUMBIA.EDU by CUVMB.COLUMBIA.EDU (IBM VM SMTP R1.2.1) with BSMTP id 8813; Mon, 03 Feb 92 18:54:41 EST
Received: by CUVMB (Mailer R2.07) id 3176; Mon, 03 Feb 92 18:54:39 EST
Date: Mon, 3 Feb 1992 18:22:00 EST
Reply-To: IBM-KERM%CUVMB.CC.COLUMBIA.EDU@cuvmb.cc.columbia.edu
Sender: IBM mainframe KERMIT developers <IBM-KERM%CUVMA.BITNET@cuvmb.cc.columbia.edu>
From: "John F. Chandler" <PEPMNT%CFAAMP.HARVARD.EDU@cuvmb.cc.columbia.edu>
Subject: Re: Replying ...
Comments: To: IBM-KERM <IBM-KERM@CUVMB.CC.COLUMBIA.EDU>
To: "Frank da Cruz, Columbia U, US" <fdc@watsun.cc.columbia.edu>
In-Reply-To: MAINT@UQAM.BITNET message of Mon, 3 Feb 1992 15:31:41 EST

> >Despite all that I said about the 3708, there actually *is* a possibility
> >of getting it to support almost-Kermit file transfer in full-screen mode.
..

> there is no way to get into transparent mode. So, being able
> to transfer small files while still in full-screen mode would be useful.

The key word in my discussion above was "almost-Kermit". My tests with the new (still unreleased) MS-Kermit and CMS Kermit continue to be encouraging, but there is an important difference between this new kind of file transfer and normal Kermit protocol: no control characters are used. Thus, the transfers depend critically on the internals of the full-screen protocol converter. To be specific, this method of file transfer assumes that the host program can clear the screen and write an 80-byte packet such that all 80 bytes pass through unmangled (just translated into ASCII). There may be strings of up to three blanks imbedded in the packet, and these must not be "optimized" in any way. So far, the preliminary results seem to show that the 7171 and 4994 both conform to these restrictions, but the Unix program tn3270 does not. Obviously, all three of these can handle transparent mode anyway, so the exercise was purely for demonstration. The next step will be to test the new programs on other equipment, such as the 3708 and STRTC.

> Ability of PC Kermit to automatically change to file receive mode when
> the mainframe starts sending a file would be useful. This feature is
> available in some communication programs like ZCOMM (shareware from
> OmenTech, as I remember).

That particular feature is available in vanilla MS-Kermit as well (as long as you are willing to devote one of the TERMINAL sequences to it), but we have to start all over with non-transparent full-screen mode. That's one of the side effects of the almost-Kermit-ness required.

John

Return-Path: <JRD@cc.usu.edu>

Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA29331; Wed, 29 Jan 92 23:46:58 EST

Date: Wed, 29 Jan 1992 21:46 MDT

From: Joe Doupnik <JRD@cc.usu.edu>

Subject: Re: 3708 and printable SOP chars

To: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu,
cmg@watsun.cc.columbia.edu

Message-Id: <0CB13365AE663873@CC.USU.EDU>

X-Envelope-To: cmg@watsun.cc.columbia.edu, fdc@watsun.cc.columbia.edu

X-Vms-To:

IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu", IN%"cmg@watsun.c
c.columbia.edu"

X-Vms-Cc: JRD

Frank and John,

Arrrgh, as Linus says. Here I just explained in class all the restart-state-machine nicities and the desirability to get synchronized rapidly. MSK does indeed look for out of band characters, especially SOP and EOP, for every character read from the comms line, and it acts upon them right away. Syncing has to work this way or we giant step down the comms string and miss a perfectly good packet or more (the legit SOP char(s) was/were gobbled as part of the malformed pkt). What's worse, we can get a damaged packet which indicates Long Pkt construction and we proceed to gobble many real packets before arriving at the checksum field (or what should be that field). MSK has no way of backing up and replaying historical comms data.

Please, can we go back to the dark ages of out of band signals for framing? Pretty please?

Joe D.

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA02488; Fri, 31 Jan 92 13:56:53 EST
Date: Fri, 31 Jan 1992 11:54 MDT
From: Joe Douppnik <JRD@cc.usu.edu>
Subject: Discarding echos, other things
To: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu
Message-Id: <4C5B531134658511@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"fdc@watsun.cc.columbia.edu",IN%"pepmnt@cfaamp.harvard.edu"
X-Vms-Cc: JRD

Frank and John,

MSK does similar things in the transportation section. The trouble is, with sliding windows the concept "last sent" gets lost unless we a) look through the pile of unfinished pkts in the queue or b) simulate those which were finished in the previous window. That's too much work to do a simpler job: unknown -> discarded. Like CK, MSK gets sequenced pkts and then has a look at the TYPE field (with a few management exceptions such as Error pkts). After looking for interesting TYPES it discards pkts as the last option. I just changed the MSK code from "complain and die" to "discard quietly and keep trying." This way I don't have to remember or search for what the speaking half has said recently. A receiver filtering out accumulated NAKs from an idle server uses the same mechanism, just ignore NAKs and they will quietly go away.

We do want to ensure that echoed pkts don't end up sitting in a buffer (forever % 64) because no one calls their sequence number. This cleanup happens naturally with the sliding windows mechanism because if it's old that fact is detected before the pkt is queued for reception (stashed for sequencing). The only decision in this case is do we send back a complementary ACK (receiver's action) for this old pkt or not. MSK's sequencer code looks at the old pkt's TYPE and if it's an ACK or NAK then the pkt is purged, else it gets the complementary ACK. A current N/ACK coming back as an echo gets purged when TYPES are examined. The sender of course has an easier time of it: old packets (previous window and back) are ignored out of hand before entering the sequenced queue, current ones are processed through the TYPE filter.

A corollary of all this is the general case, sliding windows, occurs almost naturally from specifying these detailed rules. Hind sight.

Change of topic to suggestion from Frank for MSK:

INC/DEC variable amount. Worth a try. It's a common enough situation to be desirable. Will have to see if numerical values are easily discoverable.

8003PDPK.EXE fellow. Go ahead and send the standard msg to him on this. I posted a similar one on News in the comp.protocol.tcp-ip.ibmipc group to let the technical troops know about this problem Packet Driver. Hopefully SMC will get the word and redesign it. After MSK 3.12 comes out I expect we will still receive msgs on the PD from people trying to use NetWare and sundry because TYPE 8137 is not accepted by this PD, for gosh sakes.

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>
Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA16502; Thu, 30 Jan 92 18:04:01 EST
Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 2186; Thu, 30 Jan 92 18:00:33 EST
Date: Thu, 1992 Jan 30 17:49 EST
From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
To: Joe Doupnik <JRD@cc.usu.edu>, fdc <fdc@watsun.cc.columbia.edu>,
cmg <cmg@watsun.cc.columbia.edu>
Subject: Re: Printable Start of Packet char
In-Reply-To: JRD@cc.usu.edu message of Thu, 30 Jan 1992 11:39 MDT
Message-Id: <PEPMNT.920130.174905.RC0@CFAAMP.HARVARD.EDU>

> I will meet people half way. I have rewritten the MSK packet code
> to accept a printable receive end of packet character,

>From the subject, I assume you meant to say START of packet, right? I
seem to remember that the EOP is actually ignored. Anyway, it sounds
as if you have gone ALL the way, and now it's up to me to get the K-370
end in shape. I've written the code, in fact, so now I just need to
test it out.

> The only small side effect in MSK is the SOP char must be specified
> as a \number rather than just pressing ":" ...

I can live with that. K-370 has always insisted on decimal numbers for
the SOP characters anyway, and the only way to specify a non-control
character has always been to SET TEST ON first to suppress the limit.
I'm not sure if I should remove that restriction.

John

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA17701; Thu, 30 Jan 92 19:29:09 EST
Date: Thu, 30 Jan 92 19:29:08 EST
From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
To: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>
Cc: jrd <jrd@cc.usu.edu>, Christine M Gianone <cmg@watsun.cc.columbia.edu>
Subject: Re: Printable Start of Packet char
In-Reply-To: Your message of Thu, 1992 Jan 30 18:13 EST
Message-Id: <CMM.0.90.0.696817748.fdc@watsun.cc.columbia.edu>

> There is a real problem that I didn't think of until just now, namely, that
> having different SOP's for the two directions isn't enough. If the chosen
> outbound SOP appears *anywhere* in an inbound packet, we're dead. The only
> hope would be that the MSK echo detector would recognize and suppress the
> entire echoed packet.

>
But if the send-SOP (SSOP) is the SAME as the receive-SOP (RSOP -- isn't
that some kind of cognac?) then the packet reader will read the echoed packet
correctly, pass it up to the transport layer, and the transport layer will
silently discard it. This should work right now, in C-Kermit anyway.

> Does 3.12 allow a printable SOP in *both* directions?

>
Yes. C-K too.

> The other aspect to consider is that the 3270 screen will fill up after
> a rather small number of packets have been exchanged. I think K-370
> *must* clear the screen before sending each packet, since there's
> no simple way to get the PC to sense the FULL condition and clear it.

>
Right, clear the screen after each packet. And don't allow packets longer
than the screen area. And make sure there are no repeated blanks, right?
Because we're writing them over a blank screen. Otherwise the protocol
converter might "optimize" them... That shouldn't be a problem, because all
the Kermit's involved to repeat counts.

I think we can make this work...

- Frank

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA17759; Thu, 30 Jan 92 19:33:24 EST
Date: Thu, 30 Jan 92 19:33:24 EST
From: Frank da Cruz <fdc@watsun.cc.columbia.edu>
To: Joe Doupnik <JRD@cc.usu.edu>
Cc: pepmnt@cfaamp.harvard.edu
Subject: Re: Printables
In-Reply-To: Your message of Thu, 30 Jan 1992 17:04 MDT
Message-Id: <CMM.0.90.0.696818004.fdc@watsun.cc.columbia.edu>

> The Kermit protocol says that if the main body sees a TYPE it does not
> expect then kill the transfer. Well, that needs modification to say discard
> the packet and carry on regardless. What do you think Frank?
>

I said it slightly wrong in my last message. The transport layer gets a packet, it has the right sequence number, but its type is the same as the type of the packet it just sent (or, in the s/w case, which really doesn't apply here, the same as the type of packet n that it sent), it is silently discarded. This simple heuristic has been in C-Kermit for several years and has never caused a problem.

- Frank

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA18067; Thu, 30 Jan 92 19:59:53 EST
Date: Thu, 30 Jan 1992 17:59 MDT
From: Joe Doupnik <JRD@cc.usu.edu>
Subject: Discarding echos
To: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu
Message-Id: <B621874C04664EB6@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu"
X-Vms-Cc: JRD

Frank and John,

I'm just about finishing editing MSK to totally discard packets whose Type it does not know about, period. Remembering what was sent last is a moot point with sliding windows and I choose not to make a mess of this. This has been a soft spot in the protocol for a long time: killing the transfer if it gets an unexpected pkt.

I also put in a loop suppressor so that if an old N/ACK is echoed to the receiver and it's in the previous window then it is not blindly ACK'd again to get rid of the thing.

Joe D.

Return-Path: <JRD@cc.usu.edu>
Received: from cc.usu.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA29703; Fri, 31 Jan 92 11:22:49 EST
Date: Fri, 31 Jan 1992 09:22 MDT
From: Joe Douppnik <JRD@cc.usu.edu>
Subject: Control-less Kermit packets
To: fdc@watsun.cc.columbia.edu, pepmnt@cfaamp.harvard.edu
Message-Id: <37113FCC6464AE23@CC.USU.EDU>
X-Envelope-To: fdc@watsun.cc.columbia.edu
X-Vms-To: IN%"fdc@watsun.cc.columbia.edu", IN%"pepmnt@cfaamp.harvard.edu"
X-Vms-Cc: JRD

Frank and John,

I built the special C Kermit on my Unix PC, to which I have an RS232C connection, and tried making the start of packet character printable. Banging on the keyboard to simulate timeouts and clicking the ABCD box to simulate the real world produced usable file transfers and, so far, no funny situations. It's not a strong test, nothing like what running against an IBM mainframe would produce, but it does say the two ends can recover in finite time. The MSK end had the mods to say strange packets are discarded without comment; that's the one I put on watsun ~jrd late yesterday.

Joe D.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA29272; Wed, 29 Jan 92 23:40:26 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 1714; Wed, 29 Jan 92 23:37:00 EST

Date: Wed, 1992 Jan 29 23:16 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>,
Brian Nelson <BRIAN@uoft02.utoledo.edu>

Subject: Re: [Brian Nelson <BRIAN@uoft02.utoledo.edu>: cms kermit]

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 29 Jan 92 20:36:58 EST

Message-Id: <PEPMNT.920129.231618.RC0@CFAAMP.HARVARD.EDU>

> Has anyone had success with getting cms kermit to work with a Xyplex
> terminal server with tn3270?

Failing that, there is always line mode, i.e., connect with plain
TELNET style. That should work.

> its obvious that cms kermit must do something
> special to get 7171's to pass all.

Yup. It sure does.

> Obviously, the Xyplex t/s implementation of tn3270 does not include
> transparent mode, or, if it does, it doesn't look like a Series/1 or
> 7171 to Kermit-370. Try all the different SET CONTROLLER options to
> CMS Kermit.

The most likely CONTROLLER option is SERIES1, so definitely try that
first. (Presumably, you've already tried transfers with the startup
value, which I'm guessing is GRAPHICS). There is an optional mod in
the BWR file for forcing the startup value to be SERIES1.

While I'm at it, I have a counter-question. I think I sent this last
Fall but never got any response. It concerns K11, which seems to
drop dead when it receives an Attribute packet containing an
attribute of zero length. The following is an example:

7"A!!1."I1"!A*!A/#AMJ@ R

The new "End of attributes" attribute is the "@" string at the
end of that packet. Obviously, there is a work-around (telling
Kermit-370 to suppress the new attribute), but it would be nice
if K11 didn't crash the system upon receiving a valid packet!
What are the prospects of getting that fixed? I can put you in
touch with the person who discovered this bug if that would help.

John

Return-Path: <@cuvmb.cc.columbia.edu:IBM-KERM@CUVMA.BITNET>
Received: from cuvmb.cc.columbia.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA04548; Fri, 31 Jan 92 15:33:11 EST
Message-Id: <9201312033.AA04548@watsun.cc.columbia.edu>
Received: from CUVMB.COLUMBIA.EDU by CUVMB.COLUMBIA.EDU (IBM VM SMTP R1.2.1) with
BSMTP id 4165; Fri, 31 Jan 92 15:32:05 EST
Received: by CUVMB (Mailer R2.07) id 8624; Fri, 31 Jan 92 15:32:03 EST
Date: Fri, 31 Jan 1992 15:04:00 EST
Reply-To: IBM-KERM%CUVMB.CC.COLUMBIA.EDU@cuvmb.cc.columbia.edu
Sender: IBM mainframe KERMIT developers <IBM-KERM%CUVMA.BITNET@cuvmb.cc.columbia.edu>
From: "John F. Chandler" <PEPMNT%CFAAMP.HARVARD.EDU@cuvmb.cc.columbia.edu>
Subject: Replying ...
Comments: To: IBM-KERM <IBM-KERM@CUVMB.CC.COLUMBIA.EDU>
To: "Frank da Cruz, Columbia U, US" <fdc@watsun.cc.columbia.edu>
In-Reply-To: POSTMASTER@CUVMA.BITNET message of Fri, 31 Jan 1992 10:07:33 CET

> I thank you for your answer about the problems of Kermit in
> Fullscreen in 3708. Indeed, 3708 cannot work in 8 bits
> in fullscreen. I am trying with line mode now but I have
> problems still.

Despite all that I said about the 3708, there actually *is* a possibility of getting it to support almost-Kermit file transfer in full-screen mode. The key feature of the Kermit protocol that makes it incompatible with 3708 conversion mode is the requirement of control characters only for the SOP (Start Of Packet). In principle, a printable SOP could be used, but most Kermit implementations do not allow a printable SOP (MS-Kermit, for example, does not). The author of MS-K has recently agreed to work out a mod to allow future versions to adopt a printable SOP, and a similar capability has been included in the newest C-Kermit just released (5A.178). The use of a printable SOP makes the protocol somewhat less robust, since Kermit can no longer resynchronize automatically after a mangled packet, so the number of time-outs is likely to be higher, but it seems likely that transfers will actually work. If you want to try out this sort of file transfer, the new MS Kermit should be available soon, and I will make some corresponding changes to K-370 to get past the other problems of running through a protocol converter. The initial tests of some of the new code look promising (but it hasn't been tried with a 3708). Let me know if you're interested. At the moment, I'm concentrating on the CMS variant, but the changes will also need to be tested under TSO, MUSIC, and CICS.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA05536; Fri, 31 Jan 92 16:29:03 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 2700; Fri, 31 Jan 92 16:25:35 EST

Date: Fri, 1992 Jan 31 15:40 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Joe Doupnik <JRD@cc.usu.edu>, fdc <fdc@watsun.cc.columbia.edu>

Subject: Re: Discarding echos, other things

In-Reply-To: JRD@cc.usu.edu message of Fri, 31 Jan 1992 11:54 MDT

Message-Id: <PEPMNT.920131.154002.RC0@CFAAMP.HARVARD.EDU>

Despite the equipment failure, I was able to try out the new code last night with MSK312 <-> tn3270 <-> K-370. The trick of clearing the CMS screen before sending every packet turns out not to be enough. I got an upload going, and it chugged along nicely until the 64th packet, at which point the tn3270 optimizer did us in. The way I did things, CMS issues a CLEAR, then sends a packet and gets a reply, and then repeats the cycle. Unfortunately, tn3270 has time (or makes time) to peek at the next packet sent after the CLEAR, and that causes the problem: the 64th and 65th ACKs are :#_Y> and :# Y>. tn3270 notices that it can save two bytes of transmission by keeping the "Y>" on the screen, rather than erasing and rewriting the same characters on those two locations. Thus, the 65th ACK is crunched, and MSK resends its 65th packet, but then tn3270 gets an even bigger bonus: it can reuse the *whole* ACK on the screen *plus* the repeated data packet! I think CMS Kermit can get around this problem by writing a "packet" of 80 K's and then re-erasing the screen before retransmitting any packet.

The other little problems I found are probably solved by insisting that all packets be no more than 80 bytes. I tried letting inbound packets be 94, but that caused the echoed input to come out to MSK looking "crunched" by the insertion of a CRLF, and that had a bizarre side effect: the last byte of the echoed packet would not be skipped over, and, if the checksum ended with the SOP, that byte would then appear to begin a new packet and gobble up the real reply. The only other problem is that a packet could end with two blanks of data plus a one-byte blank checksum plus an unprintable-and-thus-blanked-out EOP, and I don't know if four blanks in a row like that would trigger an optimization. I can make CMS Kermit stick a printable "handshake" character after the EOP, but it may be necessary to allow a printable EOP as well. As it happens, I know that a 4994 won't optimize a string of blanks shorter than 9 bytes, but the cutoff might be different for other devices.

John

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA23432; Mon, 3 Feb 92 16:42:43 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 3527; Mon, 03 Feb 92 16:38:56 EST

Date: Mon, 1992 Feb 3 15:56 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: jrd <jrd@cc.usu.edu>, Frank da Cruz <fdc@watsun.cc.columbia.edu>

Subject: Locking shifts in MSK

Message-Id: <PEPMNT.920203.155622.RC0@CFAAMP.HARVARD.EDU>

I managed to test the new control-less protocol, but tripped over some problems with locking shifts. First, the good news: I was able to upload text files through both the 7171 and tn3270 to cuvma. I was also able to download through the 7171, but tn3270 is perhaps a too-avid optimizer. It seems willing to substitute a tab for any length string that spans column 8 or 16 or 24 (etc.) and already appears on the screen. The latest CMS Kermit is on JCHBN 191 as KF MODULE. I tested it only remotely (through long-distance telnet) and only against MS-Kermit. It would be a good idea to repeat the testing locally and against C-Kermit -- Frank, could you give it a try? Just SET CONTROLLER FULL.

*need
read
pw*

Now, the bad news. MSK312.EXE doesn't agree with K-370 about a few points in the locking shift protocol. I thought we had left things such that K-370 agreed with C5A, but it's been a while. Anyway, the discrepancies seem to me to be bugs in MS-K, so I'll describe them in that light, pending clarification. I tried uploading MSBPCT.EXE as a fairly short test of a binary file. Near the beginning is a ^O as data in the midst of some 8-bit stuff. MSK was in SHIFT state and oddly decided to encode the ^O as #P#O, rather than just &#O -- that's ok, but wasteful and perhaps an indication of a logic problem. When faced with decoding &#O in SHIFT state, MSK seems to have decided that meant unshift. There appears to be another problem in MSK when the first of a sequence of 8-bit codes is a special. The example I found was 11,01,43,43,8f,d9,e2,f8,eb,... The encoding was #Q#ACC #P#N#OY... I remember this as one of the heuristic exceptions for shifting: never shift for 8f because you wind up with a 0f that has to be quoted, i.e., #N#P#O would have been "correct", but &#O#N would have been shorter. Anyway, the #P#N#O is clearly wrong.

Now, some worse news. MSK312 had some other trouble uploading. There were some side effects, and I don't know exactly what was going on. I thought I had saved a log, but I don't see it now. Anyway, the upload in question seemed to die repeatably, and the packet.log showed that, although the receiver had advertised a 94-byte buffer, MSK suddenly logged a long data packet (a length code of BLANK and the extra three bytes indicating a length of 103 or so). That shouldn't have caused any problems, since K-370 is always willing to decode a long packet, but the packet was apparently harmful to the tn3270 connection by virtue of its length -- it simply locked up the session.

John

Return-Path: <fdc>
Received: by watsun.cc.columbia.edu (5.59/FCB)
id AA21361; Wed, 5 Feb 92 01:24:12 EST
Date: Wed, 5 Feb 92 01:24:12 EST
From: Frank da Cruz <fdc>
Message-Id: <9202050624.AA21361@watsun.cc.columbia.edu>
To: pepmnt@cfaamp.harvard.edu
Cc: jrd@cc.usu.edu
Subject: 7171 results so far

Setting the packet start character to ":" (58) in both directions, using C-Kermit on my end (my private copy, because I keep having to change it)...

I tell C-Kermit to GET PROFILE EXEC. I find that the 7171 seems to echo each packet I send not once, but twice! (I'm still using CR as "send end" -- maybe I have to use a printable EOP?) I adjust C-Kermit to soak up echoed packets indefinitely, then I get all the way up to receiving the F and A packets, ACK the A packet, but K-370 responds with an E-packet "Lost a packet".

This was with block check = 2. Tried again with block check = 1, to prevent confusion about switching block check types. This time it didn't go so well. C-Kermit gave up with too many retries, and sent an E-packet containing a message to that effect. When reading the echo back, C-Kermit found this:

:4 EToo <ESC>[24;1Hmany re

where "<ESC>" is the ESC character. It seems like the 7171 is positioning the cursor in the middle of this packet. Strange...

Here's another one: the echo back of my I-packet (when I tried to send a FINISH command):

:0 I<ESC>[24;1He* @-#&

I also noticed that when I reconnected, the 3270 screen said "NOT ACCEPTED" at the lower right.

Too sleepy to go any further tonight, more later...

- Frank

P.S. I didn't get a KER LOG on the K-370 end because I don't know how to run Kermit off your disk without making it the A disk, and if I do that, I don't have write access to it for creating KER LOG.

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)
id AA27350; Mon, 3 Feb 92 21:19:10 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) w
ith BSMTP id 3692; Mon, 03 Feb 92 21:15:38 EST

Date: Mon, 1992 Feb 3 21:12 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>

Subject: Re: KF MODULE

In-Reply-To: fdc@watsun.cc.columbia.edu message of Mon, 3 Feb 92 19:47:43 EST

Message-Id: <PEPMNT.920203.211247.RC0@CFAAMP.HARVARD.EDU>

> can't try it because I need a read password for your 191 disk.

It's NOLOG (same as yours). I presume that's some sort of
installation default that you (and Vace, too) never bothered to
change, so that's how I left mine.

John

SET CONTROLLER
FULL

link JCHBN 191 09FRR NOLOG

ACCESS 09F J/A

KF

set test on

set controller full

set send start 58

" rec " 58

set rae pack 70

set etoa 79 92

set test 076

(type FULL TAKE)
set deb on

Return-Path: <PEPMNT@CFAAMP.HARVARD.EDU>

Received: from cfaamp.harvard.edu by watsun.cc.columbia.edu (5.59/FCB)

id AA22224; Wed, 5 Feb 92 03:04:29 EST

Received: from CFAAMP.HARVARD.EDU by CFAAMP.HARVARD.EDU (IBM VM SMTP R1.2.2MX) with BSMTP id 4402; Wed, 05 Feb 92 03:00:56 EST

Date: Wed, 1992 Feb 5 01:45 EST

From: "John F. Chandler" <PEPMNT@CFAAMP.HARVARD.EDU>

To: Frank da Cruz <fdc@watsun.cc.columbia.edu>

Subject: Re: 7171 results so far

In-Reply-To: fdc@watsun.cc.columbia.edu message of Wed, 5 Feb 92 01:24:12 EST

Message-Id: <PEPMNT.920205.014521.RC0@CFAAMP.HARVARD.EDU>

> I find that the 7171 seems to echo
> each packet I send not once, but twice!

Yes, that's the way CMS works -- you type into an input area (and get echoed), and the whole thing gets copied to the scrolling area when you hit ENTER.

> (I'm still using CR as "send end"
> -- maybe I have to use a printable EOP?)

Actually, I used EOP = 27 with MSK because VT100 cursor commands are always flying around.

> I adjust C-Kermit to soak up echoed
> packets indefinitely, then I get all the way up to receiving the F and A
> packets, ACK the A packet, but K-370 responds with an E-packet "Lost a
> packet".

Let's see. Did you forget to set TEST back OFF after setting the packet characters? The sequence byte for the A packet would be ", which happens to be the default CMS linediting "escape" character, and setting TEST ON makes Kermit bypass the usual step of suppressing linediting. This one bit me the first time I tried a transfer -- that's why I put the commands into the TAKE file.

One more thing that I forgot to mention about the 7171 -- it translates ASCII Backslash into EBCDIC Cent Sign. You probably knew that but never considered its implications for Kermit because the 7171 has a transparent mode. Well, I neglected that too when I tried my second transfer, but the fix is simple: enter SET ETOA 74 92 (but only for the 7171, of course!).

> C-Kermit gave up with too many retries, and sent an E-packet containing
> a message to that effect. When reading the echo back, C-Kermit found this:
>
> :4 EToo <ESC>[24;1Hmany re

This stuff might go away when you fix the two little details I just mentioned, but it would pay to understand what's happening. The context of this operation (from your log) would probably help. Note that the [24;1H puts the cursor at the start of the input area, almost as if CK were typing too soon and getting away with messing up the screen (maybe because K370 has cleared the screen and CMS has neglected to reformat it with protected areas). This business of timing was one of the things I was worried about for a local connection.

> P.S. I didn't get a KER LOG on the K-370 end because I don't know how
> to run Kermit off your disk without making it the A disk,

Nothing special about having a CMS command on the A disk. Let's say you have linked to it as follows:

LINK JCHBN 191 091 RR NOLOG
ACCESS 091 J/A

You could just type KF to execute the KF module on my disk. As I recall, you don't have anything accessed at mode J, so that should be safe.

John

s-00-59-:0 If* @-#&2~N" eF
r-00-59-:0 If* @-#&2~N" eF
r-00-59-:0 If* @-#&2~N" eF
r-00-59-:- Yf% @-#&2~H"
s-00-59-:/ Rprofile exec8
r-00-59-:/ Rprofile exec8
r-00-59-:/ Rprofile exec8
r-00-59-:- Sf% @-#&2~H\
s-00-59-:0 Ye* @-#&2~N" eU
r-00-00-:0 Ye* @-#&2~N" eU
r-xx-00-<crunched:chk2>
s-01-00-:;!N"3
r-00-00-:0 Ye* @-#&2~N" eU
r-xx-00-<crunched:chk2>
s-01-00-:;!N"3
r-01-00-:0!FPROFILE.EXEC/[
s-01-00-:0!Yprofile.exec5N
r-01-01-:;!N"3
s-01-01-:;!Y">
r-01-01-:;!N"3
s-01-01-:;!Y">
r-01-01-:0!FPROFILE.EXEC/[
s-01-01-:;!Y">
r-01-01-:;!N"3
s-01-01-:;!Y">
r-01-01-:;!N"3
s-01-01-:;!Y">
r-01-01-:0!FPROFILE.EXEC/[
s-01-01-:;!Y">
r-01-01-:0!Yprofile.exec5N
r-01-01-:0!Yprofile.exec5N
r-02-01-:K"A!!3."I1"!A*A/#AMJ#119920115 22:43:32@ @[
s-02-01-:;\$"Y"?
r-02-01-:K"A!!3."I1"!A*A/#AMJ#119920115 22:43:32@ @[
s-02-01-:;\$"Y"?
r-01-01-:;!Y">
r-02-01-:K"A!!3."I1"!A*A/#AMJ#119920115 22:43:32@ @[
s-02-01-:;\$"Y"?
r-01-01-:;!Y">
r-01-01-:;!Y">
r-02-01-:K"A!!3."I1"!A*A/#AMJ#119920115 22:43:32@ @[
s-02-01-:;\$"Y"?
r-01-01-:;!Y">
r-01-01-:1!ELost a packet52
s-00-15-:0 Ie* @-#&2~N" eE
r-00-15-:0 Ie* @-#&2~N" eE
r-02-15-:;\$"Y"?
r-xx-15-<crunched:chk1>
S-00-15-:0 Ie* @-#&2~N" eE
r-00-15-:0 Ie* @-#&2~N" eE
r-00-15-:- Yf% @-#&2~H"
s-00-16-:;\$ GF4
r-00-16-:0 Ie* @-#&2~N" eE
s-00-16-:9 EUnexpected packet typeM