

DCMTK - Patch #157

dulfsm kann nicht mit non-blocking sockets arbeiten

2001-10-08 00:00 - Marco Eichelberg

Status:	New	Start date:	
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Target version:		Compiler:	
Module:	dcmnet		
Operating System:			

Description

Posting in comp.protocols.dicom von alpha(at)beta.rad.mgh.mcgill.ca
Der Patch (unten) löst ein Problem, das wir bislang nicht haben. Dcmnet kennt zwar einen "blocking mode" und einen "nonblocking mode", beide betreiben den Socket aber im Blocking mode, d.h. die Optionen O_NDELAY und O_NONBLOCK werden nie gesetzt. Wenn dies vorkäme, hätte das Upper Layer ein Problem, denn dann können read()/write() auch mal weniger Bytes verarbeiten als vorgesehen, oder EAGAIN melden und gar keine Bytes verarbeiten. Beides würde zu einer Fehlermeldung führen. Die Änderung ist eher ein Workaround als eine echte Lösung, denn non-blocking I/O macht keinen Sinn, wenn man in einer Schleife immer wieder versucht, zu lesen/schreiben, ohne "zwischen durch" etwas anderes zu tun.

```
--- dulfsm.corig Wed Aug 29 16:45:27 2001
+++ dulfsm.cnew Thu Aug 30 09:50:49 2001
(at) (at) -2973,15 +2973,68 (at) (at)
    return COND_PushCondition(DUL_TCPIOERROR,
        DUL_Message(DUL_TCPIOERROR), strerror(errno),
        "writeDataPDU");
+
+/* START of modifications to correct error when compiled on RH6.2 */
+ { /* -->modified */
+     ulong sendSize, sendOffset, packetSize;
+     int counter, limit, count;
+
+     packetSize = pdu->presentationDataValue.length - 2;
+     sendSize = packetSize; /* initialize */
+     sendOffset = 0; /* initialize */
+     nbytes = 0; /* clear */
+     counter = 0; /* do not infinite loop */
+     limit = 3; /* check against this limit */
+     count = 0; /* number of times in loop */
+
+     while ( ((ulong) nbytes != sendSize)
+         && (nbytes >= 0)
+         && (counter < limit) )
+     {
+         count++ ;
+         sendOffset = sendOffset + nbytes ;
+         sendSize = sendSize - nbytes ;
+         if ( sendOffset > 0 )
+         {
+             fprintf(stderr, "\n MIR-CTN::facilities::dulprotocol::dulfsm.c::writeDataPDU()
\n");
+             fprintf(stderr, "\t EINPROGRESS + %d \n", count);
+         }
+     }
#ifdef _MSC_VER
- nbytes = send ((*association)->networkSpecific.TCP.socket,
- pdu->presentationDataValue.data,
- pdu->presentationDataValue.length - 2, 0);
```

```

+         nbytes = send((*association)->networkSpecific.TCP.socket,
+             pdu->presentationDataValue.data + sendOffset,
+             sendSize, 0);
#else
- nbytes = write((*association)->networkSpecific.TCP.socket,
-     pdu->presentationDataValue.data,
-     pdu->presentationDataValue.length - 2);
+         nbytes = write((*association)->networkSpecific.TCP.socket,
+             pdu->presentationDataValue.data + sendOffset,
+             sendSize);
#endif
+         if ( nbytes == 0 )
+         { /* frustrated send */
+             ++counter;
+         }
+         else /* nbytes != 0 */
+         {
+             counter = 0;
+         }
+     }
+     if ( ((ulong) nbytes == sendSize) && (nbytes >= 0) )
+     { /* SUCCESS! */
+         nbytes = packetSize;
+         if ( sendOffset > 0 )
+         { /* RETRIES SUCCESSFUL - modified algorithm */
+             fprintf(stderr, "\n MIR-CTN::facilities::dulprotocol::dulfs.c::writeDataPDU()
\n");
+             fprintf(stderr, "\t succeeded \n");
+         }
+     }
+     else /* FAILURE */
+     {
+         fprintf(stderr, "\n MIR-CTN::facilities::dulprotocol::dulfs.c::writeDataPDU() \n"
);
+         fprintf(stderr, "\t tried %d times, failure conditions in %d \n",
+             count, counter);
+     }
+ } /* <-- modified */
+/* END of modifications to correct error when compiled on RH6.2 */
+
if ((unsigned long) nbytes != pdu->presentationDataValue.length - 2)
    return COND_PushCondition(DUL_TCPIOERROR,
        DUL_Message(DUL_TCPIOERROR), strerror(errno),

```

History

#1 - 2012-04-26 11:54 - OFFIS DICOM Team

- Tracker changed from Feature to Patch

#2 - 2014-09-22 17:01 - Michael Onken

- Private changed from No to Yes

#3 - 2015-01-15 13:50 - Michael Onken

- Private changed from Yes to No