RODOS

UDP communication

Version: 2.0
Date: Sept. 2019
Author: S. Montenegro

1. How to begin.

Read and understand first the RODOS-Tutorial and then the RODOS-UDP tutorial.

use

```
% rodos-executable.sh linuxMC <list of sources>
```

to compile and

```
%tst
```

to execute e.g.

```
% rodos-executable.sh linuxMC udpsender.cpp
% tst
```

For each example program please first read and understand the code, then compile and execute

and see if it does what you expected. Then modify and continue trying.

2. Useful to know

UDP implements a simple way to communicate with the external world. UDP is a datagram transfer protocol: Using one port, one or more senders send messages to a single receiver receives

- there is no acknowledgement or handshake protocol. Messages can get lost.

One port in one computer can be opened only by one receiver process. Several senders can send to the same port. If there is a (single) receiver to that port then it will receive from all senders.

The RODOS UDP implementation works asynchronously. There are no thread suspends if data cannot be received. The receiver (using get) gets the number of bytes which were received or 0 if no message was ready.

3. LINUX sender/receiver

udpLinuxTestSimple is a directory containing a simple UDP sender and a simple UDP receiver which do not use RODOS (even if the UDP interface is just copied from RODOS). This example programme can be used to communicate with RODOS processes from normal Linux processes.

To compile use:

```
% doit.sh
```

You will get two executables: tst_sender and tst_receiver. Both use port 5001. Try to start each of them in its own xterm (or other command line like Konsole) window and type sentences in the sender.

4. RODOS UDP interface

The same function can be executed from RODOS threads. They are udpsender.cc and udpreceiver.cpp. Both of them use the port 5001 to send/receive.

Compile using

```
% rodos-executable.sh linuxMC udpsender.cpp
% tst
```

In another window start the Linux receiver and see.

```
% rodos-executable.sh linuxMC udpreceiver.cpp
% tst
```

In another window start the Linux sender and see.

Or compile both in different windows and execute both RODOS UDP sender and UDP receiver.

Warning: `rodos-executable.sh linuxMC` always generates tst. You will overwrite the first compilation.

udpreceiver_checker.cpp checks with a very high frequency if data has arrived. This check is done in a time event, which is more effective than resuming threads. Comment out the time event and see the difference.