Date: Thu, 11 Nov 1999 12:26:19 -0800
From: George Irwin <gmieg@EGNEXT1.SLAC.STANFORD.EDU>
Subject: documentation for Brookhaven E871 data dispatcher

I have placed the "User's Guide" to our Brookhaven dispatcher at:

http://www.slac.stanford.edu/~gmieg/dispatcher/KHFSVC.html

The server and client code packages can also be instructive. These are in:

1) http://www.slac.stanford.edu/~gmieg/dispatcher/skhfsvc.bld

2) http://www.slac.stanford.edu/~gmieg/dispatcher/ckhfsvc_init.bld

3) http://www.slac.stanford.edu/~gmieg/dispatcher/ckhfsvc_query.bld

4) http://www.slac.stanford.edu/~gmieg/dispatcher/ckhfsvc_rdevt.bld

Each of these .bld files is an sh script which creates a few files of source code and tries to compile and link them into an executable. The purpose of each executable is summarized below. See source comments and "User's Guide" for further information." You can edit the .bld files to read the source code directly.

1)
 * skhfsvc is the server program called by clients ckhfsvc_init,
 * ckhfsvc_query, and ckhfsvc_rdevt. Its function is to read
 * event data out of a file and send it over the network in

 * response to RPC calls. The first instance of this process
 * is the "master server." It's function is to spawn a server
 * "thread" or "child server" to handle traffic with each client.

2)
 * ckhfsvc_init requests a new server "thread" to be created, and
 * calls the routines to set the data file to be read and the masque
 * to screen the events. Each new client uses this command to ask
 * the master server to spawn a "thread" process to handle this
 * client's requests. The "thread" self-destructs after 1 hour of
 * inactivity.

3)
 * ckhfsvc_query requests information on the status of a server
 * "thread." The information is printed to stdout.

4)
 * ckhfsvc_rdevt continuously requests event data from a server
 * "thread" (also called a child server), after the server "thread"
 * has been started by ckhfsvc_init. The data is written to stdout
 * in ASCII-readable format. This is used for testing. The offline
 * I/O subroutine uses similar code to request events from the
 * server.

Because the linking requires the Cheetah I/O library, the .bld script will fail to create the executable. But the source code will be created. The most relevant element of this is the RPC interface file khfsvc.x which defines the RPC protocol between client and server. This protocol is directly motivated by the use-case scenarios of people using the facility. RPC (Remote Procedure Call) is a UNIX utility which wraps socket communication

in a user-friendly package. Unfortunately, it is no longer considered secure and is now blocked by many firewalls. The UNIX rpcgen command reads the khfsvc.x file and creates server and client stub routines to actually do the socket communication. Machine independent format conversion is automatic and transparent in RPC communications.