§17. Development of CAMAC Handling Software on Windows NT

Kojima, M., LABCOM group

In LHD experiment, the CAMAC system and the Windows NT operating system are applied for the data acquisition and its server computer, respectively. For that purpose, we have developed a new CAMAC handling software, in order to manipulate the crate controller connected with the Windows NT by SCSI and also to acquire the data from CAMAC ADCs. The contents of the CAMAC control software on Windows NT are shown in Fig. 1. This software consists of the following three parts;

1) CAMAC driver: the SCSI class-driver to control SCSI crate controller
2) CAMAC library: the application programming interface (API) to the CAMAC driver
3) CAMAC list sequencer: both the application and the higher-level API to the CAMAC library

These software support Intel x86 and DEC Alpha processors. The CAMAC library is provided as DLL (dynamic link library) in Windows NT. The CAMAC list sequencer works as a list-processing server, where the CAMAC commands, such as N, A, F, etc., will be executed to control the CAMAC modules. The user application will hand over the CAMAC command list to the server, and then obtain the data. These CAMAC-related data acquisition programs would behave in accordance with so-called the client-server model. The list sequencer is implemented as a background service of Windows NT; A server process provides the API which are automatically created as the system start-up and other process can call them.

The list sequencer will hand the list to the driver sequentially after the client application have completed to send a series of command lists, so-called the list block. The client application iterates the following process for each list block; First it copies the list block to the sequencer, and then read out the data according to the returned status from the server. The block data transfer between the list sequencer and the application program will be executed efficiently through the double buffering mechanism. The schematic view around the list sequencer is shown in Fig. 2. The list sequencer can handle multiple requests from several applications simultaneously. The client application can obtain data only with describing the necessary CAMAC lists and ask the sequencer to execute them. In list processing, the arithmetic calculations toward the readout data and the iteration loops with the Q-response are also carried out within the list sequencer. The list sequencer can provide the independent operation for arbitrary modules and crates to multiple client applications. While a data acquisition program continues to collect experimental data, therefore, other application like IDL can also collect the test data independently. Network applications for the fault diagnosis of CAMAC modules can also work simultaneously. Users of CAMAC modules can easily operate them just by describing the CAMAC lists without writing any programs. The program to write the CAMAC lists could be provided with GUI, and it would help the CAMAC users to do that. Of course, CAMAC operation for a brand-new module can be done simply in the same way without any programming.