

DISTANCE LEARNING OF CAMAC HARDWARE

Sergey E. Gavrilov¹, Sergey A. Kiprushkin², Sergey Yu. Kurskov³, Eugene D. Zhiganov⁴

Abstract — The idea of extending client-server technology to the types of services other than manipulating various files on a storage media and transferring files over a network is discussed. Special purpose protocol (CAMAC protocol) and server program (CAMAC server) are described and the possibilities of using the server in multi-user training systems are illustrated. Also, different approaches to teaching CAMAC programming are compared with each other. Description of usage CAMAC server in remote training system for organizing laboratory studies on various topics, including, for example, 'Studying CAMAC modules' and 'Remote control of data acquisition system' is presented as an example of our approach. According to this approach students use CAMAC server in conjunction with client programs, which are appeared to be helpful in acquiring basic skills in CAMAC hardware programming.

Index Terms — CAMAC hardware, client-server technology, distance education, teaching system.

INTRODUCTION

Client-server technology is widely used for building various software systems nowadays. This technology is based on the idea of dividing data processing task between a server program and a client program or programs. Client programs and server program can run on different machines, connected either through local or global network, such as Internet. The role of a server is to provide clients with the possibility to use some resource of arbitrary nature in accordance with the set of rules comprising protocol, whereas clients, using this resource by means of requests to server, perform their specific tasks. Usually, resources are of 'file' nature: file system in FTP, HTML and other files in HTTP, mail messages in SMTP and POP3, data base files in SQL, etc.

Obviously, it is easy to provide clients with resources of some other nature by developing special protocols and writing special server programs.

A number of software systems designed within the client-server model exist, in which the resource supplied to clients has the nature other than files on discs or another storage media. As examples one can mention well known X Window system, which is de-facto standard for graphical

output in Unix-like operating systems (resource is graphical display), various remote administration tools for Windows family operating systems (resource is entire machine and tasks run on it), an assortment of print servers in various operating systems (resource is remote printer or printers). Finally, we would like to mention MIDAS data acquisition system [1], designed in Paul Scherrer Institute (Switzerland), which provides for remote access to event-based experiments in nuclear and particle physics (resource are data taken from data acquisition hardware and also the experiment data base).

Particularly, we are interested in extending client-server paradigm to such sorts of resources as experimental setups or data acquisition equipment. We believe that this approach not only allows organizing effective experiment management but also may give the possibility to work several remote users with different constituents of experimental setup simultaneously. The last property can be used for establishing a multi-user remote training system on the base of such setup.

Although MIDAS system contains a part similar to CAMAC server described in this work, we considered it not quite relevant for our goals, since MIDAS itself is designed for use in nuclear and particle physics experiments; the CAMAC server part is deeply integrated into MIDAS and there was no need to use the entire MIDAS for us. Besides, we used to some extent different approach when implementing CAMAC server: namely, we designed our CAMAC protocol directly over the top of TCP/IP protocol family, whereas MIDAS designers used built-in HTTP protocol combined with the remote procedure call (RPC) mechanism.

CAMAC SERVER AND CLIENTS

We used the idea above when implementing special-purpose server program (CAMAC server), which operates in TCP/IP networks and allows remote users to access CAMAC crates. CAMAC is a program-control system of modular electronics created for communication of measuring devices with digital equipment of data processing. We also have developed corresponding application level protocol (CAMAC protocol) and several client programs demonstrating the possibilities of our approach.

¹ Sergey E. Gavrilov, Petrozavodsk State University, 185640, Petrozavodsk, Russia, GAS@ptrz.orw.ru

² Sergey A. Kiprushkin, Petrozavodsk State University, 185640, Petrozavodsk, Russia, skipr@ dfe3300.karelia.ru

³ Sergey Yu. Kurskov, Petrozavodsk State University, 185640, Petrozavodsk, Russia, kurskov@mainpgu.karelia.ru

⁴ Eugene D. Zhiganov, Petrozavodsk State University, 185640, Petrozavodsk, Russia, zed@dfe3300.karelia.ru

This work was supported by Russian Foundation of Basic Research (grant No. 02-07-97503), U.S. Civilian Research and Development Foundation for the Independent States of the Former Soviet Union (CRDF) (award No. PZ-013-02) and Ministry of Education of Russian Federation.

The remote access to experimental equipment in the scheme we used can be described as follows: dedicated computer running Linux operating system is used to control experimental setup which is connected to this computer through CAMAC crate (Figure 1). CAMAC server running on this computer is responsible for providing remote access to CAMAC crates for several clients concurrently. The sequence of operations to be performed over crates (experiment management) is determined by clients. Clients issue commands to server, which performs the operations requested and returns data obtained from crates as well as the information about CAMAC dataway state (for example, the presence of the X-reply from some slot) to the clients.

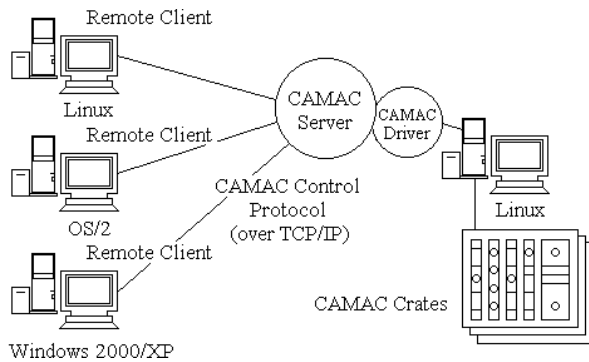


FIGURE 1
CLIENT-SERVER ARCHITECTURE.

CAMAC server is implemented for the Linux operating system in C programming language with the use of TCP sockets technology and is based on the CAMAC protocol developed with the purpose to provide a way for server and client to speak with each other [2]. According to this protocol server is a passive part of the system and the clients are active (client-driven scheme). After client establishes a connection, server sends a textual 'greeting' string, which can be used by the client for the purpose of server identification. Further the data exchange in query-answer mode between the client and the server occurs. Both query from client to server and answer from server to client are six bytes long. The former contains crate number, slot number, address within a module, command code and three bytes of data. Reply from the server includes contents of the control and status register of crate controller, error code and data. Among the commands available in the protocol one can mention 'capture slot', 'release slot' and 'execute standard CAMAC operation' commands.

Typical CAMAC session looks like this: client establishes connection with the server, captures a set of slots needed for its work, performs data acquisition and then closes the connection. Slot capturing and releasing can occur at any moment of the session. In order to be able to serve several clients simultaneously server is made parallel: each client is served by a separate server process, which is forked

by CAMAC daemon after establishing connection. Server processes communicate and synchronize with each other through the System V shared memory and semaphores IPC mechanisms, available in various Unix clones and, particularly, in Linux. CAMAC hardware is accessible for server processes through calls to CAMAC interface card driver.

It should be noted, that CAMAC protocol is designed in such way that users can perform CAMAC operations not earlier than they reserve a slot or set of slots for their purposes. It is especially important in multi-user training system. Only one client can exploit particular slot, until this client wishes to release the slot when it is not needed for the client anymore. This prevents users from disturbing each other.

We also implemented several simple client programs, which are used for various purposes, for example, testing CAMAC server. Among these one can mention client for Linux operating system (CAMAC shell) with a command line user interface and client with a GUI interface for Windows family operating systems. Both these programs allow users (for example, students) to connect to CAMAC server, capture and release CAMAC slots and perform low level CAMAC operations, both addressable, such as reading data from modules and writing data to modules, and nonaddressable, such as Z (initialize) and C (clear) commands.

USING THE SOFTWARE IN TEACHING TASKS

In order to familiarize students with basic principles of CAMAC hardware programming one can use several approaches. First, manual crate controller combined with dataway display and word generator can be used. When using such a controller, teachable persons program CAMAC modules by pressing buttons on the front panel of the controller and word generator. This method has the advantage that such a way of programming provides the closest possible contact with hardware. The main disadvantage of this method is that it is not quite convenient: in order to set, e.g. the value of data word to be written to a CAMAC module, one have to press corresponding buttons on the word generator, which is not very convenient in case when binary code is used. Second, one can use the program library, which allow interacting with CAMAC hardware. In this case students program CAMAC modules using this library and some high level programming language, such as C or Pascal. This way of teaching has the drawback, that students must be familiar well with the language. If this is not the case, they sometimes have to pay too much attention to programming itself and cannot focus on programming CAMAC. Third way to teach CAMAC programming is to use ready software, which allow to perform low-level CAMAC operations. The CAMAC clients mentioned above and CAMAC server can be classified as such sort of

software. Using this approach has several advantages by comparison with the previous two methods. First, it is quite close to the hardware, almost as the using manual controller, except that in this case CAMAC cycle cannot be performed in step-by-step mode. At the same time it is much more convenient and faster. Second, it does not require knowledge in programming languages (this does not mean, of course, that students should not know any programming languages at all, but if the goal is to make students acquainted with CAMAC, such a way of teaching seems to be quite reasonable and relevant for this purpose). Third, it provides for remote access to CAMAC hardware.

Using the last approach, we have elaborated training system on the base of the CAMAC server whose purpose is to introduce students to the basics of management of the CAMAC equipment as well as to the methods of organization of remote access to automatic experimental setup.

A short description of the lab work 'Studying CAMAC modules' is used here as an example of usage of this system. This lab study is fulfilled with the help of Windows CAMAC client program, which acts almost as manual CAMAC crate controller. This work is designed for the 3rd year students of the Physical Faculty studying the 'Automated systems for scientific researches' course. CAMAC modules used in the remote training system are connected to the automatic experimental setup, which is used for studying processes of atom-atom collisions by methods of optical spectroscopy.

The goal of this laboratory study is to get introduced to the principles of CAMAC system architecture and to gain some practical skills in working with CAMAC modules.

In order to be able to complete this lab study, students have to go through their lecture materials on data exchange on CAMAC bus, including timing diagram of bus signals during CAMAC cycle. In addition to that they also have to study the structure of crate controller, instructions on how to operate it, and also technical characteristics and commands of CAMAC modules according to the variants of the lab study given to every student.

The assignment includes a number of tasks, which differ in a set of functional modules. Binary-decimal counter and synchronizing timer, for example, is used in one of the tasks, ADC and DAC converters – in another and so on.

Having studied principles of module operation as well as its functional purpose and commands by which it can be controlled, students then have to develop an algorithm of how to perform the task and then, using the Windows client program, which simulates the crate controller, get done an assignment, making sure that all of the commands are being executed by particular module properly.

When accomplishing their tasks students connect to the CAMAC server from their personal computers. Having established connection each user with a special command

reserves CAMAC modules assigned for him that allows to avoid an access to those modules by other users.

It has been mentioned before that client, which is used to accomplish the lab study, allows a student to generate Z, C and NAF CAMAC commands. Most of the work is done by manipulating well-known controls such as buttons, menus, combo boxes, etc. For example, sending commands to modules include choosing slot and crate numbers, internal module address and function to be executed from a set of combo boxes and pressing 'EXECUTE' button with a mouse click. After each command server returns the content of the control and status register of crate controller to the user and, in case of the read command, three bytes of data. If a user acts incorrectly server returns error code. Analyzing contents of the control and status register, a student is able to determine values (active/nonactive) of X, Q and I signals, and also to determine bits' states set by the previous command. As a result users receive the same information as they would when using dataway display.

At the end of the work students have to answer to a number of additional test questions concerning CAMAC system. Also students are suggested to plan some physical experiment automated on the base of CAMAC system and think of what sort of CAMAC modules one would use in this experiment.

We have discussed one of the most simple laboratory studies. Senior students have to do more complicated assignments. Students with Masters degrees studying the 'Technologies of remote access to physical equipment' course have to develop protocol by themselves and to write both server and client program that would manage some experiment.

We would like to stress that the distinctive feature of the system described is that it allows active interaction of the student with real equipment. That helps not only to acquire a certain amount of knowledge as when reading books, no matter, printed or electronic, but also to verify whether knowledge acquired were deep and correct. To be able to reach the goals of the assignments it is necessary to have knowledge of both syntax of commands (system will refuse to perform illegal commands and return error message) and principles of structure and operation of the modular automated system (functioning set of modules will output true data only under proper usage). Automatic control of knowledge can be enriched with the help of communication of different electronic forms between students and a teacher, such as e-mail or ICQ. As long as the help of the teacher is needed only in very difficult cases live communication can be substituted by modern means of communication that can easily be integrated with the system described. Practice showed that written narration of the problem helps students to deeper understand the assignment and even to find a correct answer themselves.

CONCLUSION

In a conclusion we would like to say that CAMAC server developed by authors was successfully utilized for building training system, which in turn was used for organizing laboratory studies on various topics, including 'Studying CAMAC modules' and 'Remote control of data acquisition system'. In conjunction with the client programs, the server is used to help students in acquiring basic skills in CAMAC hardware programming. Notice that combination of the server with similar client programs may be useful for lecture demonstration of ideas of how to work with CAMAC based data acquisition system.

Finally, the approach discussed in this work can easily be extended to other types of modular data acquisition hardware.

ACKNOWLEDGMENTS

We would like to express gratitude to prof. A. D. Khakhaev and chief of laboratories I. P. Shibaev for support of this work.

REFERENCES

- [1] <http://midas.psi.ch>
- [2] Zhiganov E. D., Kiprushkin S. A., Kurskov S. Yu., Khakhaev A. D., CAMAC Server for Remote Access to Physical Equipment, *Learning and Teaching Science and Mathematics in Secondary and Higher Education*, Proc. of the Fifth Inter-Karelian Conf. (Petrozavodsk, Russia, 17-19 May 2000), Joensuu: Joensuu University Press, 2000, pp. 170-173.