PRESENT STATUS OF THE SOFTWARE FOR COMPUTER CONTROL IN THE CERN ISR PROJECT

D. Kemp, R. Keyser and L. Tausch
CERN, Geneva, Switzerland

Summary

A multi-programming system has been written to schedule the various application programs and to exploit the hardware attached to the CERN ISR control computer system. This paper describes certain features of the system, in particular those that concern its operation, as well as a synopsis of the applications.

Hardware

A dual ARGUS 500 computer system has been installed at CERN for the control and monitoring of the Intersecting Storage Rings and the associated beam transfer system[1]. Briefly, each computer has a CP with eight sets of seven accumulators, a non-priority interrupt system, paper tape input/output, 20K of 1 μs (24 bit) core and a fixed head disc (revolution time 34 ms, capacity 640,000 words). Switchable between the two machines are a card reader and a fast (100 ch/sec) printer.

Only one computer is attached to the instrumentation of the ISR[2,7]. This, the on-line computer has a digital scanner up-dating core memory, a 16 channel analog scanning system which samples variables on request and a buffered interface for outputting requests and values to various items of hardware. Also attached to the on-line computer are 4 teleprinters/keyboards and a CRT display system with its own 4K core that drives 4 CRT screens.

The System

Although the manufacturer's executive system with FORTRAN was available, it was decided to use an alternative approach on the following grounds:

(i) The design of the executive was not considered suitable for our real time requirements. In particular the communication with the operator was poor.

(ii) A tailored system is more efficient in its use of the available facilities such as core.

(iii) Much CERN designed equipment was to be attached to the computer. Detailed knowledge of the working of the executive is necessary to incorporate not only the drivers of this equipment but also the necessary checks and fail-soft procedures for equipment malfunctions. We shall return to this point later.

(iv) The manufacturer was able to supply a small organiser (for a machine without a disc), re-entrant library routines and an off-line assembler.

Using these as a basis a multi-programming system has been written which drives all the hardware, schedules the core store and display store etc. The disc scheduler organises the storage of files which the application programs access using a four character name.

The hardware dictates a maximum of seven application programs in core at a given instance. In practice, one of these levels is occupied by the command interpreter (the interface between the operator and the system) and one level is given over to alarm monitoring and allied work.

The assembler has been modified to run on-line and to produce a file of relocatable code of a complete program[3]. The system will on demand load a program from the disc, relocate and execute it.

The frequently used re-entrant routines e.g. for floating point arithmetic, input/output, are stored permanently in core for use by application programs. In particular a set of routines has been written to allow an application program to use normal printer output routines to write on the CRT screen. Within physical limitations a printer and a screen are the same to an application program.

While the second computer is used exclusively for hardware and software development it is still necessary to do some testing in the on-line machine. For this purpose an on-line program has been developed to alter files on the disc and to monitor the execution of an application program.

The system at present occupies about 10K of core. Programming for the project started mid-1969 and to date three man-years of programming effort have been invested in the operating system and a further two years in application programs.

Operator Communication

An operator's terminal consists of a keyboard together with a printer and/or CRT screen. From the point of view of operations each terminal is associated with some principle function of the ISR. The software system accepts commands typed in by an operator at any terminal. It is an operational responsibility that two operators do not give mutually conflicting commands.

One teleprinter has the special function that any detectable system fault, hardware or software will be signalled at that terminal. For example the system will discover that an interrupt has been lost, will inform the operator and render the offending hardware unavailable to application programs until commanded otherwise by the operator.

Programs have been written to perform certain defined tasks. They may be initiated by an operator typing in a line of text which, in its simplest form, is the four character name of the program together with a line terminating character (e.g. ESC). The general form of a command is:

label: program name, clauses (parameter list)

The functions of the clauses and parameter list will now be discussed.

Clauses are decoded by the command interpreter. They are used when the input (or output) for the given program should be obtained from a device other than the terminal where the command is typed. Clauses are also used when a program should be activated at a given time or in synchronisation with the PS cycle. Clauses allow an operator to activate a program at regular intervals for a certain period. On each activation the program is read from disc and relocated in core.

The parameter list is passed on to the program and is interpreted according to the conventions of the program. For a repeated command the parameter list has to be stored in core until the command is completely exhausted, which is somewhat consuming in core space. Though, it is possible to conceive a scheme to store these parameter lists (and possibly the commands) on the disc, this has not been implemented.

An example of a command is

BTMO, IN = 1P(2)99, FP (PU 103)

which is interpreted as 'execute program BTMO on the next PS injection, repeat execution every second PS pulse for a total of 50 executions and print the output of the program on FP (fast printer)'. Program BTMO calculates the beam position at the various pick-up stations of the beam transfer that are requested in the parameter list (in this case Pick-Up 103).

A further example is

XSET(24 GV)

The program XSET will set the auxiliary power supplies to the values given in the file named 24 GV. The file 24GV may have been created by the operator from theoretical values or by executing a command that records the existing values in the file.

It is an operational responsibility that the operator will not use the name of a file that should be kept, when performing an operation that writes to a file. However, the software assists in avoiding such errors in two ways. Important files that do not change can be rendered permanent softwarewise thus preventing application programs from writing to them. Further, unknown to the operator, an extension is added by the application program to the name that he supplies. Thus the vacuum system and the auxiliary power supplies may have files of the same name which are differentiated by their extension.

To help operators who are not conversant with the 4 character mnemonics or are wanting to use an infrequently used command, a program exists to display on the CRT screen the name and short description of all the programs of the requested application system. Often this is enough to jog the operators memory, otherwise he must consult the documentation.

Occasions arise when it would be convenient to execute a command on the press of a button, such as when a command needs to be repeated frequently during a test. It is therefore planned to have a set of push buttons and a suite of programs to accept a command and associate it with a push button, to inform the operator which commands are already set up, and to initiate that command when the button is pressed.

A means of communications that has been used in the monitoring of the sputter ion pump currents is a cursor that is moved over a CRT screen by a rolling ball. The operator can request a display of points representing the currents and can see at a glance any that are abnormal. Using the cursor he can pick out such points causing a message identifying the point to appear on the screen.

Application Programs

The computer system has been used successfully since the commencement of running in tests and is proving to be an indispensable tool. Its principal applications have been the following functions[5,6].

(i) Beam transfer power supplies. These are always set up through the computer for a run. Programs are used for recording existing values, setting, checking and for optimising injection conditions. At present the operator closes the optimisation loop but it is planned this shortly should be done by the computer.

(ii) Beam observation in the beam transfer. The computer calibrates the pick-up electrodes and calculates the beam position.

(iii) Vacuum system. Displays and print-outs may be produced to monitor the sputter ion pumps, turbomolecular pumps and sector valves.

(iv) Auxiliary power supplies. The program principally used to date changes the Q of the rings. Other programs are ready that create bumps, change the amplitudes of the orbit harmonics, test the power supplies etc.

(v) Logging. Beam statistics and proton deposits in various places are calculated for safety purposes.

At present there are some 100 application programs occupying about 1/4 of the disc which are constantly being added to as the functions increase to which the computer is applied.

References

1. The dual ARGUS 500 control computer system for the ISR - R. Keyser - ISR-CO/69-45.

2. Input and Output systems for the ISR - P. Wolstenholme- ISR-CO/69-24.

3. The ISR ARGUS 500 System. On line ASTRAL programming- D. Kemp - ISR-CO, September 1970.

4. The ISR ARGUS 500 System. Description of Operations - R. Keyser - ISR-CO, June 1970.

5. Computer control for the beam transfer system of the ISR - D. Neet - ISR-CO/69-72.

6. Proposed computer control for the ISR vacuum system - O. Gröbner - ISR-CO/70-6.

7. Data transmission and computer application for the CERN ISR project - P. Wolstenholme - ISR-CO/70-42.