The Ames Laboratory HOLLY Programming System

W. J. Higby and Arthur H. Klein
Ames Laboratory, USAEC-Iowa State University
Ames, Iowa 50010

ABSTRACT

This paper presents some of the program development that we have done in connection with our automatic "POLLY-like" measuring system. Some features of our program overlay supervisor for a small computer are discussed. The operator CRT display system includes a control interface designed and built at Iowa State University and a computer controlled display file refresh logic. The bulk of software event measuring logic consists of the POLLY FORTRAN programming done by the staff at Argonne National Laboratory.

INTRODUCTION

The major programming tasks that we faced during our development of HOLLY were generated by "making do" with our present computer hardware. Financial conditions also required that we could not buy the entire CRT system "off the shelf".

In this report we will cover some of the major problems that we have encountered so far and the solutions which we have implemented or propose to implement. After the system is completely in operation we expect to prepare a more complete description of the supervisor logic.

The first part of this paper is a general description of some of the features of our system. The second part is a more detailed discussion of system concepts. No track following or measuring logic will be presented. We are not far enough along in the production development to add anything substantial to the state of the art on these points.

General Description of ISU System

Computer Configuration

The computer we are using is an EMR 6050 with a memory of 16K 24 BIT words. Figure 1 represents the channel layout of our external devices. The dashed lines indicate the channel communication to HOLLY. Both channels are completely buffered with each other and the CPU.

Program Overlays

Unfortunately, the standard EMR program overlay structure was not adequate for the type of system environment that we desired. Our main objectives in the design of a new overlay system were the following:

1. The conversion of a large programming system, such as the ANL POLLY programs ⁽¹⁾, to our overlay structure should be as simple as possible.
2. The system should have multiple job capabilities
3. System overhead must be kept to a minimum

Multi-Job Processing

We have been quite pleased with the operation of our current computer controlled on-line measuring system ⁽²⁾. So naturally, one of the design requirements of the system was to include the option of on-line measuring jobs running concurrently with HOLLY. With this requirement in mind, we have designed a general multi-job processing supervisor. There are many job configurations available with such a system. One option could be to structure the programming of HOLLY so that it looked like more than one job. This would allow the time sharing of the fiducial measurement in one view with the track following in another view.

Operator Display CRT

Cost factors made it a necessity that we both design and build the display CRT interface, rather than buying such a system off the shelf ⁽³⁾. As a consequence, we were able to structure the hardware design to meet the programming requirements as economically and efficiently as possible.

The display interface is designed as another external device that has access to a standard buffered word channel. There are three commands that the display will recognize.

1. Position beam to a point
2. Display character string
3. Display vector string

Since the interface logic has no automatic refresh capabilities, it is necessary to "slave" the CPU for a particular file display. Refresh logic is controlled by a 33 millisecond clock interrupt. For most displays, this is quite adequate to produce no visible flicker with a P7 phosphor.

OVERALL VIEW OF THE SUPERVISOR

The supervisor is composed of the following sections:

1. Segment CALL and RETURN
2. Memory allocation processor
3. Execution control and job selector
4. KUBER, the controller for conventional on-line measuring machines
5. Drum I/O stack handler and processor
6. HOLLY I/O processor
7. Display CRT command processor
8. Display file refresh processor
9. Teletype, tape, and card reader I/O processors

System control information exists in many data areas used by the supervisor only. We will list a few of the main data control blocks. The Segment Table of Contents contains information describing the status of all program segments at any time in the system. The status of a segment includes:

- a) Which job (if any) is using the segment
- b) Where in memory the segment is loaded
- c) Logical checksum, name, length, drum address of segment, etc.

Each job in the system has a current execution status. This status could be

- a) Job has called a segment and memory has been allocated for the segment load
- b) A segment load is complete

- c) A segment load is in progress
- d) No memory is currently available for the segment load
- e) A segment is being executed
- f) A segment has executed a RETURN

The EXECUTION SCHEDULER is the main controller of the system. It checks for any I/O requests pending in the respective stacks, and will start the selected request if conditions will permit this. By using the execution status of each job, memory conflicts (which will be discussed later) are resolved.

Structure of Program Overlays

Segment Definition

The 16K memory is considered as consisting of 96 word "pages". A program "segment" is a sequence of instructions and the associated local data areas. A segment could consist of only one SUBROUTINE or it might be some main SUBROUTINE and the associated called SUBROUTINES. Figure 2 is an example of a typical segment call structure. Segments can be viewed as operating similar to SUBROUTINES. They are called via a simple call statement. They execute and then return to the calling segment.

Multi-job Operation

When only one job is considered, it is easy to assign the appropriate memory pages for each segment. The addition of other jobs to the system introduces a serious problem when a small memory computer is used. Memory pages must be shared by all jobs in the system. Figure 3 is an example of memory allocation when the system is servicing two jobs.

Supervisor Segment Call Processing

The details of executing a segment call by the supervisor are quite complicated for a multi-job system. Certain conditions must be satisfied before a requested segment can be loaded from drum and executed. First of all, memory must be available. That is, the segment cannot be loaded if another job is currently using the same space for the execution of a segment. Once memory has been allocated, a drum I/O request cannot be executed if some other device on the same channel is busy. We are using a concept which we call "non-sequential processing" to take care of these conflicting situations. When a job encounters such a conflict, no specific action is taken by the supervisor at this point to correct the conflict. At some later time the supervisor can resolve the conflict and schedule the job for execution.

Multiple Copy Segments

In general, the problem of loading a particular program section in any system consists of the task of memory address relocation. With our computer there exists no hardware for automatic base register address modification. The possible choices that we had for segment loading were the following:

- a) Design and write the necessary compiler and assembler that would produce code which made use of some index register for relocatable instructions.
- b) At segment call and load time, one could relocate the necessary instructions before the segment was executed.
- c) Assign a segment to a particular memory page starting address prior to the loading of the system. Use the standard EMR loader to load the segment, and copy the memory used by this load to drum. When a segment call is processed, check the current memory status table. If the appropriate memory block is free, schedule the segment for loading.

Choice a) might be OK, but we do not have a large enough programming staff at this time to consider such a project. Our two-man staff has been tied up for some time on just the supervisor programming. We ruled out b) because of too much system overhead time for the relocation. Choice c) was actually used, with a slight modification for "multiple copies".

A multiple copy segment is one which is assigned to more than one particular memory page for the load and execution. It becomes a task of the memory allocation processor to "pick" which copy to load depending upon the current status of memory at the time the segment is called. Using this technique, it would be ideal to assign a particular segment as having multiple copies beginning at page 1, 2, 3, 4, etc. Unfortunately, our drum unit is not quite that big. During the supervisor testing period we ran a variety of job structures. One particular structure consisted of five jobs that used the same memory area for the segment loads. The introduction of just one segment consisting of two copies instead of one greatly increased the frequency of job execution.

Segment Preparation

When the HOLLY supervisor is loaded into memory, all job segments must exist on the mass storage drum unit. The method of preparing these segments is a topic for another paper itself. In a nutshell, what basically happens is this: The programmer must decide at what page in memory each segment is to start. This is rather a burden, but it is necessary. Arthur Klein has written a very clever program called "SUBVERSION". It subverts the standard EMR monitor, job processor, loader, etc. The action of this routine is to load a segment into memory

starting at the desired memory page, copy this section of memory to drum, and go onto the next segment. After all segments have been processed in this fashion, a table of contents is also written on drum.

Programming View of Display CRT

We were quite concerned about the demand on the CPU for CRT refresh and the visible flicker rate. When a display file consists of many characters, there is some flicker. This has not been a problem so far, since most HOLLY displays do not have a large amount of text to be displayed.

The refresh data file does take up much memory space for certain displays. The worst one is the display inMPOINT of the digitized area for the operator. This problem has been solved by building into the refresh logic the use of multiple areas for the particular data file to be displayed.

Use of Multiple Area Files

The CRT supervisor allocates a fixed amount of area for the display file. We are presently using about 200 words. If a given program segment needs more area, it can call a supervisor routine which will then use, in addition to its own area, the area specified by the segment.

CRT Subroutines

We have essentially duplicated the types of subroutines that the ANL POLLY II software is using. In some cases we have added features of our own. In addition to the routines that build up a display file, we have added a feature called "Immediate display". For example, the "display vector immediately" routine will not add a vector data word to the display file, but instead it will execute the command to display the given vector on the tube.

This feature is quite useful when the HOLLY software is in the digitizing display mode. A refresh file is built that consists of a highly magnified view of a region of the film. While the refresh logic is displaying this file every 33 m sec, the joy ball position is displayed by calls to the "plot vector immediate" subroutine.

Current Programming Status

HOLLY FORTRAN Programming

We have not yet completely measured an entire event in one view, but we do have some bits and pieces running. The following major routines are now running with our system:

1. Fiducial measuring with FIDS
2. Track finding with FINDTKS

3. Track following with PHOLLOW

When the new supervisor is completed, we will be ready to add TVERT (the vertex calculation and track association routine) to the system.

Slice Scan Usage

From a programming point of view, the ability to specify the slice scan mode by one parameter has been a definite advantage. Fiducial and track finding are both done in the "data dump" mode, while the measuring of fiducials and tracks are done in the "center of gravity" mode. In order to keep the conversion problems of the ANL software to a minimum, we have not at present made use of the variety of slice scan possibilities available to us. Figure 4 illustrates some of the different slice scans that we can implement by simply changing a few of the specification parameters via the software.

Supervisor Status

All coding and debug is essentially complete, but no test runs have been made with the HOLLY software using the new supervisor. We hope to get the final system running by the end of January. The job structure will consist of one job assigned to HOLLY and three jobs controlling the current on-line measuring system. With this environment, we will be able to make the necessary changes in the ANL software for 35mm film without disrupting the present on-line production.

A Look at the System in Retrospect

We consider our approach to the HOLLY hardware design quite adequate for the current and future programming demands. It would, of course, be very convenient to have a larger and faster computer. Discussions relating to this question include the concept of adding another small computer to the system. Such a hybrid system might give one more power than a single large computer. Obviously this question requires a considerable amount of study.

The programming effort required for the multi-job supervisor was approximately one man year. We were fortunate to have an indirect addressing and indexing hardware feature on our computer. With clever use of this feature we were able to keep the amount of supervisor code less than 3K words.

Acknowledgements

We would like to thank the ANL High Energy Physics staff for their help during our program conversion stages. We also thank our hardware staff, in particular, W. D. Thomas, John Homer, Robert Haxby, and Charles Fullhart for their patience and understanding during our program debug.

References

1. W. W. M. Allison, F. Beck, and J. G. Loken, "POLLY II, A Complete System Incorporating Facilities for Measuring Bubble Chamber Film Without Prescanning," ANL/HEP 6916 (July, 1969)
2. W. J. Higby and A. H. Klein, "Ames Laboratory On-Line Control System for Bubble Chamber Film Measuring," IS-2228 (December, 1969)
3. William D. Thomas, "The Ames Laboratory Hardware Programmed POLLY," ANL - Proceedings of the Informal Collquium on POLLY and POLLY-like Devices (January, 1972)

Fig. 1 Hardware Configuration

Fig. 2 Single job Memory Allocation

Fig. 3 Multiple job Memory Allocation

STANDARD SLICE SCAN NOW USED

A POSSIBLE TYPE

ONE THAT MAY BE USEFUL IN TRACK FOLLOWING

Fig. 4 Slice Scan types