041872

Technical Memorandum T1
April 19, 1972

Research Project Description

Project Title: Use of CMS for Updating LARSYS Programs

Experimenters: S. K. Hunt, Applications Programming Group,
H. L. Grams, T. L. Phillips

Specific Objectives:

1. Evaluate use of CMS on the LARS computer for storage and retrieval of LARS' programs.
2. Evaluate use of CMS on the LARS computer for upgrading and maintaining programs via the CMS commands.
3. Develop and test conventions for initiating steps 1 and 2.
4. Develop an orientation short course of instruction on the use and conventions for steps 1 and 2.

Background Information:

The CP-67/CMS time-sharing system consists of two independent components: the Control Program (CP-67) and the Cambridge Monitor System (CMS). The Control Program creates the time-sharing part of the system to allow many users to access the computer simultaneously. The Cambridge Monitor System provides a conversational part of the system to allow a user to monitor his work from a remote terminal.

CMS is that portion of the system that directly applies to the LARSYS system. CMS gives the user a full range of capabilities -- creating and managing files, compiling and executing problem programs, and debugging -- using only his remote terminal. One of the main purposes of CMS is to provide the user with various file-handling facilities. Files to be used under CMS may be stored on disk, cards, or magnetic tape. However, most CMS commands assume that files are stored on disk. This means that files stored on media other than disk must be transferred to disk before many of the CMS commands can be issued for them.

A system for updating and maintaining the LARSYS programs using this CMS capability would be very beneficial. Our particular need is a way for more than one person at a time to be able to access and alter subroutines without interferring with other programmers. A method for implementing and using this capability is described below.

Facilities Required:

Since the mode of most CMS operations requires disk operations, it appears that storing the LARSYS programs on an A-disk would be the most suitable. Information would be stored on disk with each individual subroutine organized into a separate file. This requirement is made so that a programmer need only access the particular data needed rather than entire programs. Also many of the subroutines are used in several of the programs and in this way only need to be stored once.

Each file requires a unique identifier which is composed of a file name, a file type, and a file mode. Since file names themselves do not have any special implications in CMS, user files may be assigned any file name the user wishes. In the LARSYS system, then, the file name used will be the subroutine name as listed in the relocatable library. File types may be any combination of one to eight nonblank EBCDIC characters. For our use we will specify the following conventions for file types:

FORTRAN	for FORTRAN source programs
SYSIN	for ASSEMBLER source programs
LNKEDT	for link edit decks
TEXT	for compiler or assembler output decks
MODULE	same as TEXT except preceded by a MODULE card
44PSJOB	44PS job deck with all control cards, ready to run
TESTDECK	for LARSYS test decks
EXEC	for CMS executive files

The file mode consists of two characters. The first character is a letter indicating the disk area on which the file resides and the second character is a number from 0-9. The file type for LARSYS programs will be A2 indicating residence on an A-disk and read-only access.

This A-disk which will be used for permanent storage of all LARSYS subroutines will have two tapes (System tapes W and Z) and source cards to be used as back up in case of loss of disk files for any reason. This tape will be created by issuing the "TAPE DUMP" command. Tape records written in this manner are 805 bytes long. The first character is a binary 2, followed by the characters CMS and a blank, followed by 800 bytes of file data packed without regard for logical record length. In the final record, the character N replaces the blank after CMS, and the data area contains directory information.

Programs will be stored on some disk other than CMS190 because of storage requirements. It would be preferable to have this additional disk pack always mounted and ready. However, this is not necessary if disk drives are at a premium. Disk storage was chosen so that programmers would not need access to a tape drive for retrieving source language code.

Files stored on disk are formatted into records 829 (805 bytes of file information and 24 bytes of control information) bytes long. This formatting is handled internally by CMS, and is not controlled by the user. Each file (subroutine in the LARSYS case) will require a minimum of one record. In order to determine the amount of disk storage required, the following conversions may be used:

1 record	= 805 bytes	= 10 source cards
1 track	= 7 1/2 records	= 75 source cards
1 cylinder	= 20 tracks	= 1500 source cards
1 disk pack	= 203 cylinders	

The LARSYS programs contain the following number of source cards:

	# cards	# CMS Records
LARSYSAA	13800	1380
LARSPLAY	7600	760
FIELDSEL	5450	545
DIVERG	4250	425
NSCLAS	1900	190
Library Routines	4500	450
Link edit decks	450	45
Test decks	500	50
	38450	3845

Approximately 3845 records will be required or 513 tracks, or 26 cylinders for storing the source code as program entities. In order to store each subroutine as a separate file additional records will be required. Forty cylinders should be sufficient to store this data as well as six additional EXEC files to be used for update purposes.

An additional requirement for the use of these files will be the creating of seven new user IDs. Seven are required so that each person in the Applications Programming Group will have access to private P-disk space in which to temporarily store his updates and complete his editing without disturbing other programmers. Six of these seven IDs will have seven cylinders of P-disk available on CMS190, read-only access to the permanent program storage on the A-disk, read-only access to a 44PS 2C1 disk, and a link option to a 44PS 2C0 disk. The remaining ID (which will belong to the Manager of Applications Systems) will have all of the same facilities except that he will have automatically attached to him the A-disk in read/write mode. The CMS virtual machine, then, for each of the IDs will consist of:

009	Console	
00C	Spooled reader	
00D	Spooled punch	
00E	Spooled printer	
0FF	Timer	
190	System residence disk	
191	P-disk (permanent file storage)	7 cyl.
192	T-disk (temporary file storage)	1 cyl.
19A	Read-only A-disk	
19C	Read-only LARSYS Control System Disk	

An eighth ID would also be desirable for running batch jobs. This ID would not have CMS capabilities and would be used for submitting jobs to the operator to run. The suggested ID is LARSYS with the password being HUNT. It would be preferable to have the other IDs reflect the last name of the user to which it is assigned. This provides an easy method of assigning IDs identifying computer output, determining computer usage, etc. The passwords will be something a bit more complicated to assure their secrecy.

These facilities will provide the means for ease in accessing and updating the LARSYS programs.

Methods of Maintenance:

The major purpose for implementing this CMS procedure is to allow multi-person access to the LARSYS subroutines for editing and updating. Hence the procedure outlined below will center around the EDIT command in CMS and the storage and retrieval of source programs in CMS mode. This does not include any execution or developing of programs under CMS control. All programs will be compiled and tested under 44PS control.

When updating and maintaining the LARSYS programs the programmer should follow the steps outlined below:

1. Copy the subroutine to be updated from the permanent storage area to temporary storage area by using the COPYAP command.
2. Update the temporary file by using the EDIT facilities.
3. Initiate execution of an EXEC file (CRCAL) which will set up cards for compilation or assemble and transfer those cards to the card reader and which will set up proper cards for link editing the program to be updated and transfer cards to the card reader.
4. Execute compile, assemble, and link edit job under 44PS control. (RUN44PS command)
5. Initiate execution of an EXEC file (to be defined) which will set up proper cards for testing the program and transfer cards to the card reader.
6. Execute test job under 44PS control.
7. OFFLINE punch the updated source deck.
8. Obtain module deck from source deck. (CRC command)
9. Obtain source compilations listings for notebooks. (CRC command)
10. Obtain link edit maps for notebooks. (CRL command)
11. Run MODPRO to update link edit tapes.
12. Bring source deck, module deck, listings, and maps to Manager for Applications Systems.
13. The Manager will then see that all permanent files, notebooks, and card drawers are updated with the new subroutines.

Of course the programmer may have to retrace his steps if there is a programming error at any point in this update procedure. In case of system malfunction the programmer should have a tape backup. For approximately every 10 hours of terminal time spent a new updated tape should be generated. At step 13 the program should be completely verified and checked out. Then the module may be turned over to the System Programmer for entering into the 44PS system for users to access. The Manager will simultaneously read the new source deck into the A-disk area replacing the old source. The new permanent files will also be dumped onto one of the system tapes W or Z, whichever is the least current.

The Manager will also obtain a listing of the permanent file names each time an update is made which will indicate the number of records, number of logical records and date when the files were last written.

Project Evaluation:

The permanent files of all LARSYS programs have been generated and are available for Applications Programmer's use. These files are also available to IBM for use during the LARSYS contract period

One of two educational sessions has been presented to the Applications Programming Group by Howard Grams. This session was rather intensive and really an orientation to CMS and its capabilities. Some on-the-machine practice is now taking place and then a second educational session will be available as a question and answer period. These learning periods are the first step in develo ing an orientation short course of instruction.

It is our intention to continue experimenting with this system until June 1, 1972, at which time a complete evaluation of its usefulness and effectiveness will be made.

Addendum
Technical Memorandum T1
February 9, 1973

Research Project Description

Project Title: Use of CMS for Updating LARSYS Programs

Experimenters: S. K. Hunt, Applications Programming Group,
H. L. Grams, T. L. Phillips

Project Evaluation:

During the course of the previously defined experiment, we were able to meet each of the specific objectives as outlined. Each of these are discussed below.

1. Evaluate the use of CMS on the LARS computer for storage and retrieval of LARS' programs. --
   One particular need at LARS is a method for more than one person at a time to be able to access LARS' programs without interferring with other programmers and using as little time and resources as possible. It is no longer necessary to handle large volumes of cards and computer paper. Decks, listings, and tapes of existing programs used for export to other LARS personnel or outside personnel may be retrieved with little effort on the Applications Programming Personnel's part. Specific examples of this are as follows:

   - Data Analysis Research has had immediate access to existing programs for research projects now being investigated.
   - The entire LARSYS Version 2 System was transferred to IBM for use on the LARSYS Software Contract.
   - Many card decks and listings have been generated using CMS.

   Each of these required minimal effort and certainly consumed much less time and resources than were previously necessary.

2. Evaluate the use of CMS on the LARS computer for upgrading and maintaining programs via the CMS commands. -- The use of CMS on the LARS computer is advantageous for program maintenance. A major advantage of CMS is the editing capability. It is no longer necessary to handle large volumes of source decks, hopefully minimizing the possibility of shuffled cards, which has plagued us in the past. Because IBM has elected to implement LARSYS Version 3 under CMS control, this experiment has provided us with the possibility of a very smooth transition from Version 2 to Version 3. Several projects were completed using this CMS capability and progressed much more smoothly and quickly. Some of these projects were:

   - Implement the "ENLARGE PICTURE" option in LARSPLAY
   - Implement the "ENTER TITLE" option in LARSPLAY
   - Implement the $PHOTO capability
   - Fix various "bugs" in the system.

3. Develop and test conventions for initiating steps 1 and 2 The method of maintenance as outlined is quite adequate but specific routines have been added throughout the course of the experiment because of deficiencies in the old ones. These new capabilities are outlined in LARSYS Abstract 801, "CMS Commands for 44PS Operations". Most of these facilities concern running under 44PS control and therefore will not be needed for Version 3 maintenance. With this exception the method of maintenance is quite adequate. This method has been tested throughout the projects listed in step 2. One point should be reemphasized - the programmer must always take care in saving information from his P-disk on a backup tape. I would recommend here that any research programming be handled similarly.

4. Develop an orientation short course of instruction on the use and conventions for steps 1 and 2. -- Two educational sessions were presented to the Applications Programming Group in April, 1972. One session was intensive and really an orientation to CMS and its capabilities. The second was a question and answer session after some on-the-machine practice. These sessions were the first step in developing an orientation short course of instruction. In September, 1972 a second course was presented to two

new staff members who were to work on LARSYS Version 3. Experimentation and development has continued since that time and a formal course will be presented to all those interested (probably in March 1973). Gradually we hope to expand the CMS capability to everyone so that there will be a complete, smooth transition when the IBM contract is complete.

There are of course, some disadvantages to using CMS. The major one at this time is, of course, having to make the transition from CMS storage and editing to 44PS control and execution. This, however, desappears when the complete conversion from 44PS to CMS execution is finished. As the EXEC file library expands, CMS will become much more useful and used. In general, CMS capability is a very powerful and useful tool.