AN INTRODUCTION TO THE RUSH TIME SHARING SYSTEM

Computational Services Center
Computer Sciences Division
IIT Research Institute
10 West 35th Street
Chicago, Illinois 60616
312/225-9630

7/26/67

CONTENTS

1. INTRODUCTION 1
2. RUSH CHARACTERISTICS................................ 2
3. RJE CHARACTERISTICS.................................. 7
4. SYSTEM CHARACTERISTICS.............................. 8
5. COST AND SCHEDULES................................... 11

REFERENCE.. 12

ACCESS SCHEDULE

RATE SCHEDULE

1. INTRODUCTION

IIT Research Institute has just concluded an agreement with Allen-Babcock Computing, Inc. of Los Angeles for extension of its RUSH time-sharing system to IITRI clients in the midwest. The service will be available on a subscription basis beginning September 1.

RUSH (Remote User Shared Hardware) is a conversational programming system for the IBM System/360 which has many features not available in similar existing systems. In addition to powerful algebraic capabilities, including nested loops and subscripted arrays, it has instructions for handling character strings and extensive file manipulation. The RUSH language is a subset of PL/1, a language developed recently by SHARE and IBM to combine the best and most usable features of FORTRAN, COBOL, and ALGOL and presently available in the IBM 360 line of software. RUSH has demonstrated utility in commercial as well as in scientific and engineering applications. A specially designed set of micro-instructions to make execution of RUSH programs extremely efficient on the 360 Model 50 has been incorporated in the hardware. During program input there is immediate feedback when an error is encountered and during execution recognizable program errors are flagged immediately. Both of these features contribute to getting a program into production rapidly and make RUSH a very easy language to learn.

The cost of the system installed in the Chicago area is low. After a fixed monthly charge for equipment (terminal, telephone data sets and transmission devices) the user pays only for CPU time and core memory used. There is no extra charge for time on the terminal during the hours of scheduled service. The RUSH terminal may also be used as an access device for remote job entry to conventional System/360 batch operations under OS (Operating System), using FORTRAN, COBOL, PL/1 and 360 Assembly language. Programs so entered will be run in the background during RUSH operation.

These services will be provided on a time shared basis, making the cost a fraction of that expended for leasing or buying the equipment, yet the user will have powerful computation speed and access to a growing library of tested applications programs. In order to insure adequate services, initial plans call for IITRI to limit the number of subscriptions offered to twenty-five (25).

2. RUSH CHARACTERISTICS

The RUSH statements are a subset of PL/1 (Programming Language/One), or an extension of it. PL/1 is an IBM supported general-purpose programming language meant to express either scientific or business problems in a single, compact system. It is the emerging programming language of the computer industry, combining in certain respects the roles of COBOL, FORTRAN, ALGOL and assembly languages, while exceeding any one in it's flexibility and ease of translation from or to English or mathematical notation.

The RUSH system contains a strikingly user oriented, conversationally interactive (between user and system) compiler/interpreter. By means of the RUSH PL/1 language the user is linquistically convenienced. RUSH language components are particularly useful to time-sharing, truly enabling it to be conversational. Programming and input error diagnostics are provided immediately. During data input the system asks the terminal user for the required values in sequential order. In addition, programs can read data from an OS/360 data set (e.g. disk files, data cell drives, etc.).

The RUSH system provides other on-line programming facilities. In addition to the instant diagnostics, the system provides automatic line number resequencing. A byproduct of resequencing is the ability to "block move" statements within a program. During program creation the entire program or any part can be re-listed. It is unnecessary to write a program manually, then enter it on the machine. It is best, in fact, to do the development completely on the terminal, using the interactive qualities of the system. Erasure of whole or partial lines or single characters is easily accomplished. A corrected or changed statement can supplant another line with the same number. Additional statements can be interjected between existing ones by assigning new line numbers with values between those of existing line numbers.

STRINGS and FILES were the first extension to the original RUSH PL/1 implementation. STRINGS refer to the character data type and the functions and operations necessary to manipulate character strings. FILES refer to implementation of record I/O facilities, a necessary companion to STRINGS for a broad class of applications concerned with non-mathematical processing (i.e., commercial data processing as opposed to scientific data processing). Further, the record I/O implementation gives RUSH capabilities in communicating with other processors across disk files. FILES supports both sequential and direct access file methods utilizing the Data Management interface routines available in OS/360. STRINGS and FILES allow the conversational programmer to build programs for such applications as syntax-directed compilers, information retrieval, context text editors, freight rate billing, inventory control, simulation, and business management activities.

Current RUSH language development includes the CALL, PROCEDURE, EXTERNAL and RETURN statements to give the user subroutine capability exceeding that of FORTRAN and COBOL. Parts and pieces of large programs may be constructed, debugged, edited and cataloged onto disk. Different portions then may be collected into a single parcel for the required solutions. Programs too large for resident core may be overlayed, using external, resident data blocks. This capability in RUSH obviates the need for other languages or large batch core runs in order to achieve full programming needs. Most data processing problems may now reside in the full, interactive environment.

There are two major categories of RUSH statements -- "direct" and "collect". Direct statements are executed immediately upon entry, while collect statements are translated, gathered, and executed only when a direct statement to execute is given. The two kinds of statements are distinguished by the fact that collect statements must always begin with a line number and direct statements begin with a per cent sign (%). The value of the line number indicates the position of the collect statement in the program.

There are four types of collect statements in RUSH -- declarative, assignment, control and input-output. Declarative statements give information about program variables.

This may include the size of arrays, the type, if not numeric (character, label and file types are presently implemented), the length of character strings, the precision of numeric data and the use of files.

Assignment statements allow values to be assigned to program variables. All normal arithmetic, relational and logical operations are allowed, as well as concatenation of character strings. A number of mathematical and character string functions are also available.

Control statements include facilities for conditional and unconditional transfers and for automatic iteration of a range of statements. Statements which are being transferred to must be labeled with alphameric names. These names can be assigned to label variables. Loops must also be labeled.

There are facilities for performing input and output on the terminal as well as on external files. Terminal I/O can be an unformatted stream of values or characters or can be in a defined format.

Two non-PL/1 statements have been included in RUSH to make programming at the terminal easier. One allows definition of a one-statement function with substitutable arguments which can then be used in the program. This is called the LET statement. The other allows defining the format of terminal input or output by a combination of characters, dots and dashes. This is called the IMAGE statement.

Direct statements include assignment and input-output statements, which are also collect. In addition, many terminal operations are carried out by direct statements. These operations include saving and loading programs, listing, erasing, and resequencing entire programs or segments of programs, scheduling and checking on jobs submitted for background processing, listing programs saved and erasing saved programs from the user's file.

3. RJE CHARACTERISTICS

The ability to reside in a multiprogramming environment has been very important to the success of RUSH. In addition to the user conveniences of the RUSH language Allen-Babcock has developed an extension of the machine scope and versatility called RJE (Remote Job Entry). The RJE capability enables a remote time-sharing terminal, via RUSH to command OS/360 to initiate the processing of a program (in FORTRAN, COBOL, PL/1 or other IBM-supported language) for execution in the background batch processing mode. By utilizing the "core partition" option of OS/360, this processing occurs at the same time that the RUSH system is operating. The multiprogramming capability of OS/360 is an important asset to the success of RJE. The system accommodates numerous RJE users by a queueing type of scheduling during the simultaneous execution of regular RUSH programs. RJE through RUSH enables the user to update files, to interrogate OS/360 output concerning the status of execution, and to receive the print-out at a terminal or on a high speed printer. Of particular importance to the user is the ability to store OS/360 list output on an external file and subsequently, through RUSH, examine and list the important parts. Under OS/360 control, data storage and retrieval can be achieved using the data cell drive.

4. SYSTEM CHARACTERISTICS

At present the remote terminal consists of an IBM 2741 (Communication Terminal) connected to a data set (Bell System Model 103A2, or equivalent) which is connected to a Bell System exchange facility (example, business line). Therefore direct access to the time-sharing computer system is orginated by the remote terminal by direct dialing the special (non-published, non-listed) IITRI Chicago telephone number. Upon connection, the data would then flow to and from the on-line computer upon command.

Data flow between the IITRI access location and the ABC data center is via a 2712 Remote Multiplexor which may be considered as a "transparent" medium. Therefore, the terminal may be considered as directly connected to the computer.

As mentioned before, the operating environment of RUSH/RJE utilizes the multiprogramming, multiprocessing concepts. RUSH manages and maintains it's own memory space and I/O facilities requirements for terminal users, and all RJE scheduled tasks are processed by a separate scheduler. This is accomplished by using the Option 2 scheduler of OS/360 which supervises the "partitions" of memory in the gross sense.

RUSH occupies the high priority partition (No. 1) with two other partitions used by other concurrent tasks. One such task is for an "experimental" copy of the RUSH system for the convenience of the system support staff. During the day, this copy is available to certain predetermined I/O ports to allow virtually unrestricted testing of future versions. This technique assures the user of adequately tested functions before release. Another task is the actual execution of the background batch processing.

In the implementation of the RUSH system one of the first hardware decisions was to use Large Capacity Storage (core storage) as a substitute for a high-performance, high-speed drum. The economies of these large cores are just becoming apparent in the computing field. RUSH is one of the first applications which utilizes this scheme. The LCS (Large Capacity Storage) has a cycle time of 8 microseconds (on the Model 50 each cycle includes 4 bytes). The access time to the LCS is 3 microseconds. It is this overlap of access to cycle time that makes the LCS even more attractive as a high speed secondary memory device.

The LCS was chosen to replace the traditional high speed drum normally used in time sharing for "core swapping". In most time-sharing systems, there are usually more simultaneous users active than there is core available to process them. A very large high speed core is the solution to the difficulties in those time-sharing systems that are using drums with access times on the order of 4 to 8 milliseconds and no facility for addressing at the byte level. The Allen-Babcock use of LCS forces the executed instructions into the high-speed main memory and the data into the LCS. The overlap features of LCS allow very efficient usage during execution.

Another important RUSH choice was to use interpretation as opposed to compilation prior to execution. PL/1 syntax in many cases calls for an interpretation rather than a compiling techniqte; particularly in those areas where the language is "dynamic" rather than "static".

A disadvantage of interpretation of a high level language is execution time degradation. For this reason micro-programming was employed to add instructions to the computer to increase the speed of interpretation. Instructions have been added by Allen-Babcock to the System/360 model 50H for the implementation of RUSH. They have effected an over-all increase in speed up to 50 to 1 over the same implementation with standard 360 instructions. Continual improvements in the hardware/software interface routines promise an even greater execution efficiency.

5. COST AND SCHEDULES

IITRI offers ABC's RUSH time-sharing service on either a "full-day" or "half-day" basis as indicated on the attached Access Schedule. The cost breakdown for both the full-day and half-day service is given on the attached Rate Schedule. These costs consists of fixed and variable monthly charges as indicated on the Rate Schedule. It is interesting to observe that the rate structure is not based on terminal time, but only on CPU time and additional storage requirements.

REFERENCES

1. Frank Bates and Mary L. Douglas, PROGRAMMING LANGUAGE/ONE, Prentice-Hall, Inc., 1967.

2. Gerald M. Weinberg, PL/1 PROGRAMMING PRIMER, McGraw-Hill, Inc., 1966.

3. Allen-Babcock Computing, Inc., RUSH TERMINAL USERS MANUAL, 11-15-66 (available from the IIT Research Institute).

4. IBM, A PL/1 Primer, document No. C28-6808.

5. IBM, A Guide to PL/1 for FORTRAN Users, document No. C20-1637.

6. IBM, A Guide to PL/1 for Commercial Programmers, document No. C20-1651.

Note: References 4 through 6 are available through your nearest IBM Branch Office.