COPE

(Console Operator Proficiency Examination)*

David Farbman and Richard Ketover

Abstract: Each year electronic computers become more sophisticated, and the programs they must process become more complex. Because of this, dependence of those in computing on the skill and experience of operators is increasing.

At the same time, selection and training of qualified operators grows more difficult. To meet the need for a quick, accurate, uniform operator test and training aid, the authors have developed Cope (Console Operator Proficiency Examination), outlined below. While this examination is programmed specifically for the IBM 705 Model II with two Tape Record Coordinators, similar programs could be developed for other computers.

Introduction

There was a time, not very long ago, when a qualified computer console operator needed few skills beyond those involving the manipulation of a few buttons. That time is gone. With it has gone the old ease with which operators could be trained and their proficiency tested.

Today, computer installations are vastly more complex and costly than they were seven years ago, and the programs they process are more sophisticated and difficult to run. Programmers no longer can be expected to have sufficient technical knowledge to make unaided diagnoses of all conditions that arise during program testing and debugging; nor can they be "on call" to trouble-shoot stoppages during normal "three shift" operation.

The natural result is a growing reliance on the operator for technical assistance. This means an increasingly high standard for the operator, whose repertoire of skills must include a general knowledge of programming and a natural "feeling" for the computer and its potential.

An obvious corollary of newer, higher standards is greater difficulty in recruiting and training. The answer to this problem lies in developing accurate training aids and tests of operators. The major difficulties are two-fold—whether the subjects are trainees or applicants claiming experience:

(1) Development of a test which is comprehensive, accurate and uniform. (The problems here are those standard with any achievement test.)

(2) Minimization of computer time required for the test. (The latter is increasingly critical as computers become more expensive and machine time consequently more valuable.)

The authors feel that the test outlined below, which doubles as a training aid, answers both these difficulties. The test has been administered to several senior staff members of the authors' company, as well as to others, and has proved extremely accurate as a measure of 705 console operating ability. Approximately one and one-half hours of machine time are required for the test; this is felt by the authors to be the practical minimum for a comprehensive operator test.

Although this program was developed specifically for the IBM 705 Model II with two Tape Record Coordinators (TRC's), similar programs could be developed for other computers.

Approach

The test is designed to achieve two aims: first, to determine the level of proficiency of an experienced IBM 705 operator; and second, to serve as a teaching and evaluating aid in the training of new personnel.

The basic idea is simple. The individual is given a program deck and a set of instructions, and is told "run it." The program itself acts as the examiner, asking questions of the applicant and determining his level of proficiency.

Because the test is relatively complex and asks the applicant to take the appropriate actions required in virtually all operating conditions, it should be administered to an inexperienced operator only under the supervision of a skilled operator serving as a proctor. The presence of a proctor is valuable under any circumstances in securing the most realistic evaluation possible of an applicant's skill and potential.

Although the test should not prove overly frustrating to a relatively inexperienced operator, it is written so that even the most skilled operator will find portions challenging and interesting.

Cope is composed of three parts, which may be used independently. These are in increasing order of difficulty.

PHASE I (Ten test situations)
PHASE II (Twelve test situations)
PHASE III (Eleven test situations)

Phase I requires the execution of simple input-output commands and elementary storing and displaying in response to typewriter directions.

Phase II requires a response to more sophisticated input-output commands, as well as storing and displaying. It also tests a knowledge of the function and location of the various indicators and registers of the 705.

Phase III contains a program which deliberately causes eleven different error conditions, each of which the console operator must identify and act upon. There are, for ex-

* Copies of the COPE test program, including instructions and card deck, may be obtained by writing the authors % The Computer Usage Company, Inc., 18 East 41st Street, New York 17, N. Y.

ample, tape buffer overflows, invalid instructions, redundancies, and timed loops.

Numerical scoring has not been applied to the examination. An analysis of the printer scoring (which is automatically produced at the end of each phase) plus the opinion of the proctor observing the examinee yield a realistic appraisal of the examinee's skill and experience.

Average running times for the examination are:

Phase I —15 minutes
Phase II —20 minutes
Phase III—30–45 minutes contingent upon the number of restarts allowed by the proctor.

Phases

At the start of each test within Phases I and II, a typewriter message appears, containing the number of the test within the phase, and a description of the action to be taken by the console operator.

After completion of the last test within a phase, the typewriter indicates the numbers of those tests within the phase that were incorrectly executed.

For Phase III, the console operator is provided with a chart describing fourteen possible error conditions. When an error is detected by the operator (i.e., the machine "hangs up" because of program, machine or input error), he must analyze the situation and then find the corresponding error diagnosis on the chart provided.

The error is identified on the error chart by a code letter. The operator is instructed to place this code letter in a designated memory location and to follow additional instructions. He then is graded automatically on his ability both to identify the error and to follow the instructions relative to that error.

Following the last test of Phase III, a typewriter summary of the operator's actions appears. If the operator has correctly diagnosed an error, placed the proper code in memory, and followed the required instructions, the summary shows "correct." If the diagnosis of the error was correct but the execution of the respective instructions was incorrect, the summary shows "correct" diagnosis but "incorrect" action. In all cases the summary indicates the correct error code identification.

Examples of the error conditions programmed in Phase III include:

Attempt to do an input-output instruction on a unit that can never execute this instruction.
A tape record larger than buffer size is read or written.
Loop caused by a tape unit that is on-line but is not in ready status.
Machine check caused by a redundancy in memory.

Conclusion

The frequency of machine or program errors occurring during normal production and testing is small. The abilities of an operator cannot be determined accurately by watching him at the console for a short period of time, nor can he be properly trained to handle various types of "stoppages" without having previously experienced them. The Cope program will, within a relatively short period of time, cause the operator to take various actions. These range from displaying memory—a frequent occurrence—to correcting internal parity errors—a rarity.

The cost of one and one-half hours of machine time is negligible compared to the cost of repeating a series of production runs, or recreating a tape file because of errors caused by an unskilled operator.

(CONTINUED FROM PAGE 660)

was obtained for results from the two methods. Computing time for a 200-minute sample varied from 25 to 50 minutes, depending on the traffic density. About 300 words were required for the system representation and the program filled the remainder of the 2,000-word drum.

In both of the examples considered, the limitations on space forced the use of programming methods that reduced the effective computing speeds. The method chosen to represent each system was not such a compromise, however; in each model it was the method of choice from considerations of both space and computing speed.

Whether or not event-sequencing would have increased the speed of the air traffic model is not known; there was no space available for any additional programming. Event-sequencing is used routinely in the graphical analog of the air traffic system: successive time-increments vary with the intervals between successive nonlinearities in system behavior. In fact, it is feasible to use simultaneously different time-increments in different parts of the system. Some less sophisticated version of this process would probably have helped the computer model, at least at the lighter traffic densities. At heavier densities, as in the package-handling model, the nonlinear "events" are so frequent that event-sequencing would be almost indistinguishable from the "clock" sequencing used in both models.

ACKNOWLEDGMENTS

The work on which this paper is based was performed under a contract with United Parcel Service and under Signal Corps Contract DA-36-039 SC-73104 for the Air Navigation and Development Board. Substantial contributions were made to these studies by G. W. Bond and V. C. Lafferty, both at Armour Research Foundation; Dennis Dodds, United Parcel Service, Inc.; and K. S. Gale, University of Arizona.

Invitation. Members who wish to comment on developments and trends in the computing field may submit "Opinions" to Dr. A. J. Perlis, Editor-in-Chief, *Communications of the ACM*, Carnegie Institute of Technology, Pittsburgh 13, Pa.