By Fred Brooks

H. G. Kolsky

POUGHKEEPSIE
Department 539
South Road Laboratory

September 4, 1958

7000 Operating Techniques Memo No. 1

SUBJECT: 7000 Operating Philosophies

1. Why New Operating Philosophies are Needed for 7000 Computers.

To many computer cognoscenti who have contemplated the plans for 7000 Series computers, such machines seem to be merely super-704's. To these people there is no question but that the new machine installations will be operated in the same manner as the best of present 704 installations are. This point of view is quite natural, but it seems unrealistic in face of the magnitude of the differences between the 704-709 and the 7000 computers.

In the first place, a change in performance by a factor ten to one-hundred is alone sufficient to change the kind of problems that can be done and the optimum method of doing them. In the second place, the 7000 computers incorporate many changes of kind as well as degree: the exchange, variable field length binary operations, the connective instructions, the interruption and address monitoring systems, the interpretive console, etc.

The safest and indeed most conservative assumption is that 7000 operating techniques will be different from 700 techniques, just as the latter differ from 650 techniques. It is the purpose of this memorandum to explore the elements in which the new techniques seem most likely to differ from the old. By this exploration one can identify the difficulties, in both hardware and programming, that separate us from new and valuable operating techniques.

1.1 Efficiency

The key question to be used in evaluating any computing system and any operating procedure is that of problem-solving efficiency: How much does the computer, as operated, help people solve problems? This efficiency has often been confused with efficiency of machine utilization; but one must remember that it is the man and the machine who accomplish useful results. The object is to maximize the total output of the man-machine problem-solving system. The reasons why most efficient machine utilization is not synonymous with most effective machine utilization are explored in the second memo of this series.

Present operating technique seems, in many cases, to be based on this confusion between efficient problem-solving and efficient machine utilization. In fact, one often is amazed by the extent to which man's erstwhile tool dictates his actions. The question of "Who is boss?" does not yet quite apply, but the question is no longer whimsical.

1.2 The Computer as a Tool

Early calculators were unquestionably proper tools: extensions of the user's personality. With a desk calculator one is intimately aware of the innermost workings of the problem-solving process, and this communicativeness of the machine leads one to discover many unexpected things. Furthermore, the machine usually performs satisfactorily; i.e., it does what the user really desires.

One can say that a computing machine is obedient if it does as told. Most modern machines are satisfactory in this respect, most of the time.

The degree to which a machine is understanding depends upon the ease with which the user can specify a given task to the machine. As machines have become faster, programming systems such as FORTRAN and machine features such as indexing have made computers more understanding.

Satisfactoriness involves more than obedience and understanding, however. A 704 is less satisfactory than a desk calculator in the sense that the user is less apt to get what he wants on the first (or n^{th}) attempt. This paradox deserves a little exploration.

A 704 is as obedient as a desk calculator, or almost so. Most of the time it executes instructions as specified. A 704 is more understanding than a desk calculator: it is easier to specify an iterative process to it, for example. The cause of the difference in satisfactoriness lies not directly in the machine itself, but in the problems one specifies for the machine. But, in practice, if not in theory, the nature of the problem one attempts on a computer is itself a function of the computer. Therefore, satisfactoriness, although apparently an indirect and contrived measure of a machine, is in fact a pertinent and meaningful measure.

The essential concept in this way of looking at things is that of a reasonable problem for a given machine. By a reasonable problem we mean one which is both practical and worthy. The reasonable problem for a more powerful machine is consequently more difficult. Since the ease with which a given problem can be specified has only slightly increased as computers have evolved, and since machine power has increased rapidly, it follows that the ease with which a reasonable problem is specified has been steadily diminishing.

Speaking anthropomorphically, one might say that since our more powerful machines challenge us to attempt more and more complex problems, they should correspondingly offer more and more help in the difficult task of specifying problems fully. In more prosaic terms, one might say that this part of the computing art has not developed so rapidly as the design of arithmetic units, and that its underdevelopment is hampering our effective use of more sophisticated hardware. In any case, the problem of successfully specifying more and more complex computer tasks is a real one, and the only practical solution is for the burden to be increasingly shifted to the computer.

In part this can be accomplished by continuing to improve the understanding of computers. Better instruction sets and more sophisticated programming systems can do this. Such an improvement only helps the user to specify his wishes more simply, however, and the real difficulty is in formulating his wishes precisely.

How can the computer be improved so that it will assist in this task? At present, the most obvious answer is to increase its communicativeness, i.e., the ease with which it can give information to the user in the form he most readily can assimilate. Since it is from observation of the results of faulty problem specification that one formulates more precise specification, increased awareness of the inner workings of a problem is the obvious need.

As machine power and problem complexity increase, satisfactoriness diminishes; as satisfactoriness diminishes, the need for communicativeness on the machine's part increases. The user must progressively substitute knowledge of what is happening for his diminishing faith that all is well. In spite of this need, satisfactoriness and communicativeness have diminished together in the evolution of computers.

At first glance, the diminution of communicativeness seems to be inexorably linked to the evolution of increased computer power. Communication implies reaction by the man, but the reaction time of the man is long in comparison to machine time. Therefore, the cost of waiting for human reaction has increased with increasing machine power. As a result, there has been a tendency to limit communication to a small number of occasions which are separated by considerable unmonitored computer activity.

The trend to faster computer speeds itself is not apt to be reversed. As speeds go up, cost per operation comes down. As speeds go up, the size of problems that can be solved in a reasonable elapsed time also increases. Because of these facts, machine and human reaction speeds will continue to move apart.

The separation of machine and human reaction times means that communication costs more, if and only if the machine waits idly for the human to react. Communicativeness can become cheap again if a machine is sufficiently powerful so that it can do something else while the human user reacts. There is reason to believe that the 7000 computers are this powerful.

1.3 Why Communicate?

There are several reasons why the super-704 operating philosophy is inadequate for 7000 computer systems. All stem from the lack of communicativeness of the computer when so operated.

The first reason why present philosophy cannot be successfully extrapolated to fill all 7000 needs is that it is already inadequate in many 704 applications. This inadequacy is felt most keenly in applications that consist largely of debugging. Here the lack of contact between user and machine deprives the user of valuable knowledge as to what happened and causes the machine to be operated at length on worthless tasks. This is true of both code debugging and of mathematical process debugging. Almost all programmers would agree that debugging progress is much more a function of the <u>number</u> of computer runs made (i.e., communicative interactions) than of the computer <u>time</u> used. Similarly, the most common reason engineers give for preferring analog computers to digital computers is the extent to which communicative interaction is possible in analog calculation.

The second reason is that many tasks now done satisfactorily with limited communication will demand sharply more communication when the tasks are extrapolated. Extrapolation of present tasks will consist in part of simple expansion of the number of points computed, number of cases evaluated, etc. This expansion will mean more voluminous outputs that must be comprehended. Extrapolation will also mean that more complex tasks will be undertaken, extra dimensions considered, etc., so that the difficulty of reconstructing a dynamic event from a static output print will be magnified. Both of these effects will make the need for dynamic observation (i.e., frequent communication) of mathematical processes more acute. Both will make the need for graphical and pictoral output more pressing.

The third reason why extrapolated 700 series operating philosophy will be inadequate is that whole new areas of application will bring problems not satisfactorily solved in 700 series computers. The most obvious of such new applications are those generally called "real time" and "integrated data processing".

These applications are new largely because the 700 series lack the communicativeness they inherently require. The 700 series machines are designed to constitute complete systems with rudimentary facilities for communicating with the rest of the world. The central concept of real time operation and of integrated data processing is that of a computer as part of a larger system, with frequent and voluminous communication between it and other system components.

Since the 7000 series hardware has been explicitly planned to suffice for such new applications, it remains to develop new operating techniques to meet the needs of the applications and to take advantage of the new machine features.

2. Modes of Operation - How and Why

One can distinguish at least six modes of operating a computer such as the 7000S. Each of these is useful and desirable in some applications. Each makes distinct demands upon supervisory programs and upon input-output facilities. We shall classify them by whether the machine is attended by users, by servants (such as operators), or by no one. We further divide the meaning of attended into two classes: attention at the central computer and attention at some remote input-output device. Any configuration of input-output devices at a single remote operating station or location will be called an apparent machine.

The six modes of operating that are here distinguished are shown in the following table:

Mode	Attending at Machine	Attending at Apparent Machine
1	Servants	None
2	None	None
3	Users	None
4	Servants, users	None
5	Servants	Users
6	None	Users

The first mode is that used in many 704 installations. As such it is described in detail in Operating Techniques Memo No. 2, and equipment implications are discussed there. There is no doubt that this sort of operation will continue to be standard practice for production runs in scientific computing centers and off-line data processing installations. It is questionable whether this mode will continue to be standard for debugging runs.

The second mode is an extension of the first in which sufficient facilities are provided to permit the machine to be operated on production problems for extended periods without any human attention. This was discussed in detail in Stretch Memo No. 47 of October 12, 1956, which points out the necessity for a real-time clock, an interruption system, address protection, and special aids for supervisory programs. These have accordingly been provided in the machine and the largest remaining task is the development of the supervisory programming techniques themselves.

The third mode is widely used for machines smaller than the 700 series. The user has full access to the power of the machine, and he does the manual tasks required in the entry and removal of data. This type of operation will be most useful in small 7000 installations where a production problem shares the machine with a problem in whose solution the user participates.

The fourth mode will be more common. Here there will be operators to do the tasks required by the production-type problems and there will also be user communication with the machine on his problem.

The fifth mode calls for the users to be removed from the actual machine to an apparent machine. This has several advantages over the fourth mode, even when the users are removed only to an adjacent room. One is the elimination of the mutual interference of users and servants with each others' work. Another is the freeing of the user from the psychological pressures caused by the physical presence of a big (and expensive) computing machine.

More important, the fifth and sixth modes of operation are those suitable for real-time control applications, for program debugging procedures in which the machine is used dynamically, for scientific computation in which the user and the machine jointly find solutions, and for integrated business applications which operate on-line.

3. Technical Problems in Developing Operating Modes

3.1 Equipment Problems

The computer features that will be needed for the several foreseeable modes of operation have been specified already and are included in the computer. These include the clock, the non-specialized exchange, the interruption system, the address protection system, the execution instructions, etc.

The equipment features whose specifications are not yet complete are all input-output devices, even though not all operate through the exchange. These are:

1. The programmer's console and inquiry station. These have been fully specified electrically. What remains to be done is the physical layout, which demands some unconventional human engineering. Here we must consider the characteristics of the operator not only as a trained human being, but also as a computer user whose own brain is directing the whole system.

2. The set of switches and lights that actually control the computer, such as POWER ON and POWER OFF.

3. A monitoring loudspeaker to convert pulse patterns to audible signals.

4. A device for graphically displaying data on-line and in real time.

5. A high-capacity, medium speed disk unit.

3.2 Programming Problems

The programming techniques necessary for the several operating modes are being developed very slowly. Most IBM programming effort is now going into the achievement of the routine things we know how to do, such as assembly programs, rather than the solution of the major new problems. These problems seem to be:

1. Planning a system for the efficient performance of a supervisory program which will not burden the coder of subject programs. This is the "unisupervisor" problem, and it has been extensively and productively considered by E. F. Codd and E. McDonough. It is not yet fully solved.

2. Planning a system for on-machine debugging using the programmer's console. This includes specification and writing of a typical console-defining routine. This problem is being attacked by R. F. Arnold and F. P. Brooks.

3. Planning a system for operating several programs at once in an interlaced manner. This is the "multiprogramming" problem. In its simplest form it includes a system for controlling "peripheral" input-output while an unrelated problem runs.

4. Planning a system for recompiling programs so as to change the location and amount of memory used. This is the "memory allocation" problem, and it is the most serious component problem in the satisfactory solution of the multiprogramming problem.

5. Planning an editing system which will convert raw data into properly edited reports and graphical displays upon receipt of a minimum of request information. This is necessary for all modes of operation, other than debugging, that involve interactions between the user and the problem.

6. Planning a system for scheduling and allocating input-output units among several programs, both for those modes in which actual problem computation is interlaced and those modes in which the computational parts of problems are completely sequential.

All of these problems, and indeed most problems of operating technique, could be considered to lie outside the province of Product Planning and Product Development. However, the operating technique is as much a part of a new computer product as is the hardware itself. In the present case, development of new techniques promises to increase the usefulness and applicability of our 7000 hardware by a large factor. Until this is more widely appreciated, it is incumbent upon those who understand the philosophy of 7000 systems to explore these areas.

F. P. Brooks, Jr.

F. P. Brooks, Jr.
Advisory Engineer
Project 7000 Planning

FPB/pkb