# STATUS AND PLANNING REPORT
# for
# OCS INTERACTIVE SERVICES

April 1969

DIRECTORATE OF SCIENCE AND TECHNOLOGY

Office of Computer Services

# STATUS AND PLANNING REPORT

# FOR

# OCS INTERACTIVE SERVICES

April 1969

## Table of Contents

Page

# STATUS AND PLANNING REPORT

# FOR

# OCS INTERACTIVE SERVICES

## I. INTRODUCTION

This report provides the background, objectives, and characteristics of the IBM 360/67 Interactive System and the plan for transition to a versatile and user-oriented interactive* environment over a 15-month period between 1 October 1968 and 31 December 1969. OCS has been planning to offer interactive services to its customers as part of its third generation system plan. A plan, which called for an interactive facility with a comprehensive array of conversational languages by April 1967, has not been realized. Even though the hardware manufacturer defaulted on its software promises, OCS, by its own efforts, did develop a modest interactive service by April 1967 and implemented this system TSMON (Time Sharing Monitor System) in its Computer Center on an IBM 360/40 in the third quarter of 1967. Subsequently, this 360/40 was replaced by a larger machine, a 360/50; and in March 1969 it was replaced by an IBM 360/67. This 360/67 (Mod 67) hardware has been designed specifically for time-sharing, thus easing the implementation of an interactive system. Although the Mod 67 was stipulated in the original 1965 plan, the base software which OCS will be using will not be IBM's standard Mod 67 software, TSS (Time-Sharing System). OCS plans to use software developed jointly by Lincoln Labs of the Massachusetts Institute of Technology and by the IBM Cambridge Scientific Center, modifying this software and integrating into the Mod 67 System additional software capability such as TSMON.

Since the acquisition of the Mod 67 is a major increase in Agency commitment to interactive use of computers, this report describes the plan of transition from the IBM 360/50 with TSMON to the Mod 67. For background information, a list of other hardware/software alternatives considered for the transition is given in Appendix A.

* This term and several others used in this report are defined in the Glossary, Appendix C.

The general objective of the service planned for the OCS interactive system is a "quick response time to user demands." Even though the planned system can offer all services which are available on a standard IBM 360, services which react to a human at a remote terminal have priority at this time. Other services such as RJE (Remote Job Entry), which answer requests for significant amounts of computational time, are not covered in this report.

The definition of "quick response" time is not easy. What may be quick to one user may be slow to another. Support capabilities of the system to individual users will vary according to the number of users on-line and the amount of central processor power the aggregate of users requires. However, with 20 to 30 users active, reasonable objectives as to maximum times or rates seem to be a response within four seconds for individual searches on large customized data bases; a response within two minutes for simple individual searches on small files using a general purpose query language; and a processing rate no less than that of the same program running in isolation on an IBM 360/30.

## II. GENERAL DISCUSSION OF TECHNOLOGY

### A. The Evolution of Interactive Computer Systems

The first computers were operated directly by the programmer. He wrote his program, requested time on the machine, ran the program, made on-the-spot changes by manipulating buttons and switches, took his results back to his desk for analysis and change; and, after several iterations of this cycle, he produced a functioning program. In this early era, he had "interactive service"; he commanded the machine and received instant response.

By the mid-1950's programmers were not interested in "running" machines. Computer operators, who were efficient at handling cards, changing magnetic tapes, and pushing the increasingly more complex pattern of buttons, now ran the programs on the machine. Normally, the programmer submitted his card deck and, upon run completion, received his printout. He touched or directly interacted with the computer only when he requested a "special time," for an emergency diagnostic session. Quantity of processing per machine-hour had now increased, since a specialist (the operator), rather than the programmer, manipulated the buttons, the cards, and the magnetic tape.

In the late 1950's with the introduction of more automatic control of machine operations (through software called "operating systems"), the programmer was further separated from contact with the computer. These operating systems processed many more programs per machine-hour, since such slow manual operations as card handling, print handling, and the program loading were automated by transferring them to small peripheral machines. The general relationship of the programmer to the machine did not change; however, in effect, he touched the machine less. The increased productivity of the machine per hour made his "hands-on" use less and less economical, since his hands-on use could not exploit (and, indeed, was at odds with) the efficient features of the operating system. Striving for increased efficiency produced another phenomenon: most computer installations began to stress "most jobs per unit-time" as the objective. Jobs were stacked up in large queues so that the machine would never be idle. The objective was met, but because of the long queues, turnaround time of the job to the programmer suffered.

In the 1960's two major changes were made to improve service to the computer programmer. However, in both changes, the previous objective of efficient machine utilization was still primary. SPOOLing (Simultaneous

Peripheral Operations On-Line), was introduced first, where the operating system did input/output functions concurrently with the main program. Several queues and some manual handling were eliminated; the machine produced yet more output; and turnaround time to the programmer was somewhat improved. In the second change, the system could run several jobs concurrently (multiprogramming). Quantity of processing per machine unit-time was optimum, but since the machine had the capability of processing several jobs concurrently, it became economically prohibitive to allow one programmer to have "hands-on" use.

About this same time systems research in multiprogramming had developed techniques for sharing resources such as devices, memory, and time, and modified implementations of these techniques could be used as the basic building block of a system focused on optimum service to users--both programmers and others who could benefit from direct access to computer facilities. Many systems which provide nearly instant service to users at remote locations have now been designed and implemented. Thus, the evolution of systems as of this time has diverged into two types, the traditional or batch system in which the hardware produces the maximum quantity of output, and the interactive system in which many concurrent users have frequent and quick interaction with the system resources.

As users of interactive computer systems have become more sophisticated, they have requested services which are totally different from those of the traditional batch system. Although some users were merely anticipating better turnaround times for their program processing requests, others desired a new kind of service. The analyst wished to make an experimental query of a data file, see the results, reform his concept of his query, and then repeat the interaction until he had a very selective answer. The engineer wanted to have the system notify him of his spelling errors when he was keyboarding program statements. These and other user desires have shifted the original interactive systems designs yet farther from the traditional batch concepts.

There are many complex tradeoffs in choosing an interactive system versus a batch system. These two systems represent the two poles of service, the one attempting to give best service to each user and the other the best utilization of the computer hardware (in economic terms). In practice, the two poles are somewhat theoretical. To give the best possible service to the user requires an overkill of hardware power which is prohibitively expensive, and to attain the best utilization of hardware requires long input queues which is unacceptable to users. Thus, in both cases there are design compromises. Interactive systems users have instant response most times rather than all times; batch computers have reasonably short input queues which are occasionally exhausted.

In addition to the interactive service and batch service, another has evolved which is essentially a blend of the two. RJE (Remote Job Entry) is a service in which a terminal or a small computer is stationed remotely. The user data is sent via a data line to the central computer where it is queued for processing, the output is transmitted to the small remote computer, and the user receives his printout directly. This is attractive because of simpler logistics and fewer queue points. The additional cost of remote hardware is the tradeoff.

Sharing of resources (memory, peripheral storage space, tapes, terminals, time, etc.) is common to both multiprogramming systems and interactive systems. The essential difference of the two system designs is the technique by which time is shared among users. In the batch multiprogramming system, users are assigned priorities, and time is allotted to the highest priority user who can profitably use it. Thus, if a user's program has been assigned memory and devices, and is not in a "WAIT" state, it can profitably use computer time to do its computation and move its job along to completion. However, while this task is active, no other jobs will progress towards completion. In the least complex interactive design*, all remote terminal users have equal priority. An equal time allotment (or slice), which is very small (such as a millisecond), is given to each user. Each user in his turn can have as much of his allotment as he can profitably use. Thus, his program momentarily ceases to run not only when he cannot use the time but also when his allotment is exhausted, whichever is first. Under peak load conditions the user receives only his quota which is a predefined amount of time per minute. At other times he receives more frequent distributions of his predefined time allotment (slice). In practice, since the user normally has much interaction and interruption at a terminal, seldom can the user profitably use his total time slice. It is this time slicing concept that makes fast response possible and a system worthy of the term "interactive."

An interactive system requires much internal bookkeeping since all users must have resources concurrently. Specifically, the change of processing from one program to another requires software which provides intricate and timely control of the internal processes of the system. But this bookkeeping consumes much of the power of the computer. The overhead is high; however, the design objective of good response to the user is met. Measurements of the economic tradeoffs in hardware overhead versus

*No attempt is made here to describe all the variations.

user efficiency are difficult to quantify. Since the overhead is large (up to 45% in some recent systems), some evaluators claim that interactive systems are too costly. Others claim that the increase in user benefit from interactive services easily outweighs the extra costs. One formal measurement states that programmers produce software at three times the rate they can realize in a batch environment. In a multiprogramming batch system, rarely does the effective use of the processor consume more than 40% of the power. In some comparisons, the measurements of net throughput of interactive and batch systems have been nearly equal.

Thus, there are many conflicting claims as to the relative merits of batch and interactive systems. Neither type of system is performing at the anticipated efficiency levels; however, designers are still hopeful of attaining these levels. The present measurements of comparative usefulness of the two types of services to humans is based on many variables, not the least of which is personal preference. At this time, it is premature to draw conclusions; objective evaluations should be conducted after both the systems and the human use of the systems have evolved to a higher level.

As an alternative to large interactive systems, the use of small, stand-alone computers might be considered. Generally speaking, three factors are relevant here:

a. Access to a large data base is more feasible in a large interactive system. Some data sets are too large to be online on a small system at any given period of time. Sequential tape-like processing poses no problem here but random inquiries and postings do. The scope of a particular run on a small computer may also be inhibited since there are not enough input/output devices to handle the relatively large number of data sets required to be online.

b. Small computers now provide reasonably large internal memories and fast processors. They are well suited to some types of scientific computation and other problems requiring few input/output operations. The increasing power of small computers and more attractive costs will keep them competitive in these problem areas.

c. Only partial independence (in the sense of user management and control) is possible with an interactive system. On the other hand, pooling of software and hardware resources may be a significant benefit to the user, making management control of a small independent computer a less significant factor. One of the inherent design philosophies of time-sharing is that of a central software authority implemented by that of the hierarchy of system management and control.

B. Types of Interactive Systems

Conceptually, there are three types of interactive systems. Although implemented systems can generally be described as being of one of these types, nearly all provide some services which are characteristic of another type and attempts at precise categorization of a given system frequently become a semantic problem.

1. Specialized Systems. Most of the early systems were of this type and their use continues to grow. They can usually be characterized as single-purpose. For example, airline reservation systems, credit check systems, systems for dynamic inventory control, and systems providing one specialized programming language (say for engineers) fall into this category.

Users are non-programmers, although frequently a background batch service is available for applications programmers. A single integrated operating system and applications program is used. The applications program, which may contain many independent modules, is the direct servicer of users. All resource and data management is provided and is totally transparent to the user. Facilities provided to him relate only to specific application involved.

2. Dedicated Systems. In this type of system, a wider variety of users are serviced through a number of general purpose facilities. Most systems built for university use and most of those offered by system manufacturers are of this type. Examples of dedicated systems include TSS for the IBM 360/67, TSMON for a large standard IBM 360, MIT's CTSS, the GE-635 system, etc.

Users in these systems include programmers and non-programmers. They work with the system through only those languages and input/output and data management services provided directly by the system. These modules are an integral part of a single operating system which exercises overall control. Two layers of partitioning protection are provided to the user: his program and the operating system.

3. General Purpose Systems. These systems are rather new. Here an attempt is made to provide a full range of services to the user through a system architecture which recognizes several "virtual machines." The notion of an operating system that provides the basic services remains, but the

change is that more than one operating system can be active concurrently, each with its own hardware and services. The system becomes a set of systems of the dedicated type or perhaps several independent copies of one dedicated system.

An example of this type of system is the CP-67 software on the IBM 360/67. The user has his choice of operating systems (and accompanying services) such as OS, DOS, CMS, TSMON, or any other that works on a stand-alone basis on the IBM 360. He (or his manager) chooses a hardware configuration and the operating system on the basis of the desired services. The fact that other systems are operating on the same physical computer is transparent to him. In this environment, three layers of partitioning protection are provided: the user program, his operating system, and the overall control program.

### C. State of the Art in Interactive Systems

The very first concepts for "sharing of time" on a computer are less than 10 years old. The earliest experimental systems used vacuum tube and transistor hardware, but with revolutionary conceptual hardware changes and had software written to utilize these hardware changes effectively. As a result of the advances in technology from these experiments, third generation hardware (with integrated circuitry) either was built with these hardware features or designed so that they could be added easily.

Two major computer manufacturers, IBM and GE, included interactive systems as part of their original third generation announcements. In both cases, the first prototype systems were dramatic failures, chiefly because of deficiencies in software. In spite of these failures many successful interactive systems are running today. Several have been written for small computers. Nearly all interactive systems which have advanced beyond the initial research stage have been operating less than two years.

Looking into the future, better hardware techniques to service program and hardware interrupts, and more efficient algorithms for utilizing such hardware will be designed. Virtual memory and paging (memory exchange) concepts will probably continue in use until a major cost reduction in large memory occurs. Terminal improvements will include the following: they will cost less, the keyboards will be tailored to interactive use, there will be a more flexible alpha-numeric display, CRT displays will have better resolution, they will have less audio and electronic emanations, and weight and size will be reduced. Many data communication functions now accomplished with software will be hard-wired into standard interface devices. Direct access storage devices which hold many times the storage capacity of present devices are already running at development laboratories. No really significant changes in hardware or software architecture related to interactive system concepts appear imminent. The designers and implementers in the near future will probably concentrate on repackaging existing software and hardware components.

## III. GENERAL REQUIREMENTS FOR THE OCS INTERACTIVE SYSTEM

The need and desire for interactive facilities come from a number of Agency components and from a number of related motivations. In some cases, the potential user wants to reap the benefits of the technology per se, but not knowing quite how to do so. In other cases, he might believe that it offers a way of circumventing some of the difficulties inherent in the more conventional services provided by the OCS Computer Center. Or, having looked deeper into the matter, he may have found that an interactive facility offers a complementary set of functions to those with which he is already familiar and which he is actively utilizing.

There is little evidence thus far that increased interactive capabilities will eliminate or replace the more conventional batch processing procedures of long standing value. Rather, we expect that it will add a new dimension to the Agency's ADP capabilities, expanding the population of ADP users and the scope of OCS activities. The possibilities of interactive computer systems are or will be attractive in two new areas: the use of the computer to assist in more clerical-type activities and, its potential use in small scale data manipulation where conventional computer assistance is too expensive or cumbersome. However, some existing applications may shift away from batch processing. Conventional reports from medium-size files may be replaced by ad hoc browsing through an on-line file, for example.

The translation of ideas or desires into hard requirements which will justify the significant cost to develop and operate the interactive facilities is perhaps the most difficult problem the user and the computer service manager face. The costs are high; approximately [redacted] of the OCS budget for rented equipment in FY-70 is allocated to the Mod 67. This is a high risk investment, particularly in light of problems that continue to demand attention in other aspects of OCS service. This risk is recognized but the benefits already realized appear to justify further investment.

25X9

OCS has in hand a number of "requirements" for interactive services that are being at least partially met or are within reach. At this point, however, we view such requirements with some caution; we interpret them as a mutual desire to experiment with the facilities, but where the ultimate payoff cannot be clearly predicted. These requirements are summarized below under three categories.

1. File-Oriented Services. For the most part, these requirements relate to the need to solve analytical or management problems through interactive file access ("browsing"). Limited success here has focused attention on the other side of the coin--the problem of creating and maintaining the file being queried. As we begin to understand how to do the latter from a

user terminal, we would expect to see more applications where the emphasis is on maintaining very active and perhaps perishable files on-line. The need for file-oriented interactive services that are being partially met (or at least the need made known) are:

a. The use of on-line query for several files already operational in the batch processing mode. These include a file of extracts from DeGaulle's speeches, data from a Soviet Foreign Trade Handbook, various files of interest to SAVA, the Target Oriented Display application from NIPE, several FMSAC files, and the DDS&T Contract Information System.

b. File-oriented applications with little or no past activity in the batch processing mode. These include on-line access to the RID name grouping file, the OCS Project and Resources Information System, files of medical records, and an OCS personnel file.

c. The Security Automated Name Check Activity (SANCA), which is of special importance in the file-oriented category. This requirement has progressed beyond the "desirable" stage in that the Security Records Division has already begun to depend heavily on the remote access facilities provided by OCS.

2. Processing-Oriented or Problem Solving Services. This class of requirements usually involves some file activity also, but the emphasis is shifted from selecting and formatting data to more complex manipulation of smaller quantities of data. Requirements or potential requirements include interactive computation through a high level programming language that a scientist or engineer can use directly. OEL and FMSAC are now using this tool. Another major area is on-line graphic displays where the combination of alphanumeric and line drawing capabilities are of potential interest to the cartographer, the reconnaissance mission planner, and the signal analyst. Third, statistical data reduction is now being performed on-line, particularly for OMS and TSD. A fourth potential application area is computer aided instruction.

3. Data Entry Oriented Services. The remote terminal as a data entry device (with the full power of the computer to at least partially validate inputs in real time) is attractive. Applications here include partial substitution for conventional key punching of formatted data, entry of text directly into the computer as an adjunct to the EPIC system, input of source program statements by the OCS computer programmer followed by limited program testing at the terminal, and conventional administrative typing of reports,

memoranda, etc. In addition to the advantages of real time validation, the data in machine language--even if perishable--can be changed, added to, or deleted with relative ease and on a demand basis.

Predicting the direction, number, and scope of requirements for interactive services over the next few years is as difficult as predicting the degree of success of those applications already underway. However, it is expected that, for the most part, "more of the same" will be requested. For example, we would expect increased activity in small or medium sized intelligence and management files, the proliferation of on-line computational interests, and a gradual increase in interest in on-line programming, text editing and so forth.

A real danger lies in the experimental application becoming an operational necessity to a user component without the change noticed by the managers concerned--i.e., without explicit attention given at the appropriate time to analyzing the benefits and limitations of the service--and thus becoming a de facto requirement. To a limited extent, conventional management approvals at project initiation time can counteract such pitfalls (for example, see section V.B. of this report describing procedures for requesting terminal installations). Monitoring progress of individual applications is a joint OCS/user task requiring continuous attention. Clearly, this is difficult but must become a matter of regular interest if the system is to progress in a rational fashion.

## IV. OCS INTERACTIVE SERVICES AND OBJECTIVES

### A. TSMON (Time-Sharing Monitor) Services

OCS has developed TSMON, an interactive (time-sharing) system. This system is a partitioned system in the sense that programs assigned to specific terminals are allocated fixed blocks of memory.

#### 1. User Services *

a. SOLVE. A language created to provide a computational capability. SOLVE is a subset of PL/1 and allows conversational programming. Error messages are given after each line entry. The program is interpretive (i.e., statements are interpreted one at a time; a complete set of computational statements are not compiled into a machine program), so programs execute relatively slowly. Although SOLVE computational capability is significantly less than that of PL/1 it has the advantage of being conversational.

b. DESKCAL. A simple computational program which can be used as a substitute for an office desk calculator. It is interactive or conversational and basically performs all of the functions of the normal desk calculator and other functions such as square root and trigonometric functions. It is conversational in that if the user provides the statement, "C = A + B; it will ask, "What is A?", wait for an answer, and then ask, "What is B?" and wait for an answer. As soon as the two variables are defined, the answer will be computed and returned to the user.

c. TSAR. A language which allows rapid remote querying of direct access files and outputting of information from these files in a variety of formats and on a number of devices. Two types of commands are used: search and output. The first type specifies the information to be selected from the file; the second specifies how the selected information is to be displayed.

d. LINUS. A language for creating files of individual "lines" of data, maintaining files, and retrieving single lines of information (e.g., text or program statements) at a terminal. Files created with

*A more complete description of these services is given in "OCS Interactive Services," November 1968.

LINUS can be investigated by a variety of LINUS commands. SOLVE files (consisting of lines of SOLVE statements) can also be queried with LINUS. As the user changes line entries, previous generations of his file are automatically saved for backup.

e. TRUMP. A language to locate, display and change fixed fields in TSAR files via the terminal.

f. On-Line Batch. A method for requesting execution of a program in the batch mode. The user can compile, linkedit, and execute programs in Assembly Language, PL/1, and FORTRAN. He can enter data directly or call the data in from a specific file. The output (including source program listing) can be printed on the on-line printer in the computer center, or the source program diagnostics and output routed to the terminal.

2. Special System Features. CAM(Common Access Method) allows programmers concerned with building new interactive services to address any type of terminal with one access method. These programmers also use PAM (Paging Access Method) which is an efficient method for storing and retrieving data on direct access devices for users. Similarly, he uses a program called RINUS which efficiently utilizes PAM pages by writing records contiguously and by overlapping records between pages. RINUS also provides indexing capabilities.

A package of data compression and record addressing techniques is available for customizing large files for fast direct access in which the primary search criteria is on one field. By using these techniques, a programmer can quickly put a user file into on-line query operation.

3. Deficiencies. The TSMON system limits each user to 15,000 bytes of core memory.* The system was so designed to circumvent paging problems. However, the constraint of 15,000 bytes per program many times produces incidental benefits. Programming users must program within tight constraints and thus most times produce very efficient code. Also, such code transferred to a paging environment uses only four pages (page = 4096 bytes) which keeps processing overhead time to a minimum. However, the four-page constraint eliminates interactive use of most procedural language programs and eliminates complex programs which exceed 15,000 bytes.

* This 15,000 byte limitation applies only to those programs operating in the foreground. The on-line batch programs which operate in the background (batch mode) can use approximately 100,000 bytes of core.

User Services are also in need of improvement.

a. SOLVE. The number of statements is limited to a maximum of 150 (the number varies according to the contents of the statements). On-line techniques to overcome this limitation are available but the limitation is a deterrent to smooth general usage. Other undesirable aspects are that the programs are limited to 30 different symbols (variables and labels), and only a small subset of the usual mathematical functions is available. Subscripting of variables is not allowed.

b. TSAR. Information items that have multiple values cannot be queried effectively. The information that is to be listed cannot be sorted. The output is available only on a terminal. Permanent files cannot be created from the collection of hit records. Other features which allow sophisticated search queries have not been implemented.

c. TRUMP. Only the fixed portion of a TSAR record can be modified on-line.

B. System Objectives

1. General. When a decision was made in early 1968 to upgrade interactive facilities and move into an operational mode, the primary objectives for the OCS interactive system were stated as follows:

a. The system will be used principally in the interactive mode. Background processing for efficient computer utilization is desirable but not mandatory.

b. Compatibility with the software and hardware used in the Computer Center is desirable.

c. It is desired that by the time the interactive computer system is installed, customer requirements and OCS experience with time-sharing will be such that a production time-sharing environment will be feasible and desirable.

d. Movement toward a production environment will make system experimentation and extension more difficult. Through judicious planning (and perhaps the use of Information Processing Research and Development Laboratory (IPRD) facilities), experimentation should be able to continue at a reasonable pace.

e. Keep the costs within current budget estimates. Supplementary funding could be justified only if major new requirements were surfaced.

f. Conversion to new equipment should be as painless as possible.

g. Since no firm basis exists for projecting load other than what is known about existing applications and informal requirements, it should be assumed that known requirements will constitute the minimum load to be expected.

2. Software. Objectives have been established for existing and anticipated interactive programs. Program revision will be made to effect software compatibility and also to provide more efficient system operation and better service to users.

a. SOLVE. This service is being revised to increase the efficiency of a few commands. Ultimately, SOLVE will be an independent mathematically-oriented language.

b. COINS. This is a USIB-sponsored experimental system devised to link the intelligence community together in a network of computers. When implemented, intercommunity remote querying of selected files will be possible through use of interactive terminals. TORQUE II is directed toward this objective, among others.

c. APL-360. This is a system designed to execute system commands or mathematical statements entered on a typewriter terminal. The System will be installed for test and evaluation several months after installation of the 360/67 interactive system.

d. TORQUE II. This language will replace TSAR as the basic file retrieval service. It will provide for the following: an integrated creation, maintenance, and retrieval package; on-line or off-line symbol table creation; inverted file (and hash) indexing; advanced algebraic retrieval syntax; n-level hierarchical data structure; saved hit files; variable formatting of output; sort capability; full on-line up-date; computational and extended function capability; and off-line printing from on-line queues.

e. LINUS. This service is being revised to allow file changes beyond ten generations and to incorporate a file recovery technique to prevent lost files should the system go down in the middle of creating a file.

3. Response Times to Users.

a. Customized Files. Our goal is to provide a four-second response time per query to a single target record for customized data sets. When multiple records must be accessed to collate data or to collect a data set of multiple hits, then obviously the response time will be longer. No attempt is being made to establish a maximum response time objective for this type of query. More on-line experience is required to develop an estimating algorithm.

b. TORQUE Files. The response time objective for TORQUE searched linear files is being set at about two minutes. The search time is dependent upon data set size; thus unusually large files (tens of thousands of records) would have search times which exceed the two-minute objective.

c. SOLVE. A two-second response time has been set as an objective for entering each SOLVE statement. Since SOLVE program execution is interpretive, an objective for execution must remain vague. If response time for a SOLVE program execution is inadequate for the customer's requirements (e.g., many computational iterations), he will have the option of re-entering his program as a PL/1 program and receiving the faster execution time of the non-interpretive execution.

d. User Programs Under CMS. The Mod 67 hardware has a finite number of machine cycles of central processor power which are shared with remote users. Since it is difficult to estimate an average number of active users, any objective must be qualified. With the above remarks in mind, the following objective is given: programs will execute on the Mod 67 at a speed no less than that of the speed of the same programs on a Mod 30.

4. Types of Terminals Supported. Each terminal must have available the pertinent code in both the CP (Control Program) software and the operating system under which it is to be used. Target dates for support per terminal and per operating system are as follows:

| Terminal | TSMON | CMS | OS | DOS | TDMS | APL-360 |
|---|---|---|---|---|---|---|
| IBM 2741 | now | now | now | now | now | now |
| IBM 1050 | " | " | " | -- | " | " |
| IBM 2260 | " | " | " | -- | -- | -- |
| KSR-33 | " | now | -- | -- | -- | -- |
| ASR-28 | May 69 | -- | -- | -- | -- | -- |
| IBM 2250 | now | -- | now | -- | -- | -- |
| BR-90 | " | -- | -- | -- | -- | -- |
| RJE (remote job entry) | -- | -- | May 69 | -- | -- | -- |
| COINS Switch | now | -- | -- | -- | -- | -- |

5. Number of Users Supported. We believe it is possible for the Mod 67 to meet the above response time objectives with about 55 concurrent users for file query type interaction with at least two users entering limited computational type programs. At this time it is difficult to anticipate whether this many concurrent users and TSMON and OS users will significantly degrade the total system performance. However, as many as 55 CMS users are successfully operating concurrently on-line at other installations. Upon customer demand, the number of control units or terminals can be changed.

6. Support to COINS. Initially COINS has use of the system for a period of two hours each day. Other system users having specially compartmented data will not utilize the system concurrently with non-Agency users such as COINS. COINS files will be available to Agency users at all times.

7. Schedule for Interactive Services. Generally, the system will be available seven days a week, 24 hours a day except for periods of maintenance and COINS operation.

8. System Backup Facilities. There is no immediate plan for a backup to the interactive system and all interactive users should be aware of this fact. Many time-sharing "requirements" are unfocused; a cleaner picture of need vs. nice-to-have must emerge. However, for planning purposes, a second 360/67 was included in the Program Call for FY 1973.

One alternative could be to replace one of the presently installed IBM 360/65's with another Mod 67. This Mod 67 could normally be run in Mod 65 mode, but switched to CP-67 in an emergency. This Mod 67 would be equivalent to a Mod 65 in power but would cost approximately $5,000 per month more. Section V.A. describes this and other alternatives related to backup and general expansion of system hardware facilities.

## V. THE IBM 360/67 SYSTEM

### A. Current and Planned Configuration

The IBM 360/67 computer with Drum Storage (4, 090, 000 bytes) has replaced the IBM 360/50 computer. Eight additional Display Stations and ten additional Communications Terminals have been ordered in consideration of potential requirements and lengthy manufacturing schedules (for a total of 24 Display Stations and 16 Communication Terminals either on-hand or ordered).

Overall control of the Model 67 system will be vested in the Lincoln Lab/IBM developed Control Program-67 (CP-67). TSMON, the Cambridge Monitor System (CMS), several versions of OS/360, other monitors, and CMS monitor, also developed by Lincoln Lab, and IBM includes language processors and file building programs.

The hardware system consists of a Model 67 Central Processing Unit, 2-memory boxes of 260, 000 bytes each, a 4-million byte high speed drum, a 220-million byte high speed direct access storage device, a 400-million byte direct access strip storage (data cell) device, 3-magnetic tape drives, a card reader/punch, and a printer. A diagram along with a complete listing of the planned configuration of system are included in Appendix B.

There is no fixed plan for upgrading which includes both hardware components and dates. However, it is assumed that users will react favorably to the interactive system and require expanded services such as additional on-line storage, remote terminals, and processing time. The following are several upgrading alternatives. These will be reviewed and a firm plan formulated by September 1969.

a. More direct access storage on the Mod 67. As a minimum, the second IBM 2314 disk will be added to the system early in 1969. At the current rate of adding files, the present 2314 will be saturated shortly after installation on the Mod 67.

b. A second Mod 67. In order to provide backup to the Mod 67, one of the three Mod 65's could be converted to a Mod 67. This machine could be used as a Mod 67 or Mod 65 as the need occurred. Since the existing Mod 65's are saturated, they would have to be upgraded with more memory and possibly with the addition of a drum. Manual switching of storage devices would be required. An extension of this plan would be the acquisition of a second Mod 67 (in addition to the converted Mod 65) which would give the desired backup and additional power. This would be a major decision in that the budget commitment to interactive systems would be double that of present. As stated previously, a second Mod 67 was included in the FY-73 CPC submission.

c. Replacement of Mod 67 with a Mod 87. Several users have requested that IBM add dynamic relocation hardware to the Mod 85 to produce the Mod 87. Unofficially, IBM is experimenting with such an approach. For a long range plan, this alternative has merit. Also, it is possible that a pair of 87's might produce enough power to allow the creation of virtual machines equivalent to the Mod 65's now being used by OCS.

## B. Terminals

### 1. Terminal Placement

The following is a list of current and planned terminals and a brief description of their use (as known to OCS in April 1969).

| Components | No. and Type | Active Pending | Use |
|---|---|---|---|
| SCIENCE AND TECHNOLOGY: | | | |
| DD/S&T | 1-2741 | A | Information storage and retrieval |
| O/D/OCS | 1-2260 | P | Program development, information storage and retrieval |
| OCS/APS | 2-2260 | P | Interactive system development, demonstrations and random use |
| | 1-2741 | P | |
| OCS/ATS | 2-2260 | P | Programmer training |
| | 1-2741 | P | " |
| OCS/ISD | 1-2260 | A | Application development |
| | 1-2741 | P | " |
| OCS/SAD | 1-2260 | A | Application development |
| | 1-2260 | P | " |
| | 1-2741 | P | " |
| | 1-1130/2250 | P | System development and graphic support |
| OCS/MSD | 2-2260 | P | Application development |
| OCS/OPS | 4-2260 | A | Technical Staff activities and general user support |
| | 2-2741 | A | |
| OEL/ASD | 1-2260 | A | Engineering calculations; scientific applications |
| OEL/GSD | 1-2741 | P | " |
| OEL/SSOC | 1-2260 | P | Project oriented information storage, retrieval and computations |
| OEL/SSOC | 1-ASR-28 | P | Computer I/O of communications paper tape |

| Components | No. and Type | Active<br>Pending | Use |
|---|---|---|---|
| FMSAC/AID | 1-2260 | A | Missile/space event analysis |
| | 1-2260 | P | Scientific computations |
| | 1-2260 | P | " |
| OSI | 1-2260 | P | Information storage and retrieval; computations |
| | 1-2741 | P | |
| INTELLIGENCE: | | | |
| CIA Ops Ctr | 1-2741 | A | COINS, information storage and retrieval |
| CRS | 1-2741 | A | " |
| OSR | 1-2741 | P | " |
| PLANS: | | | |
| RID | 2-2260 | A | File retrieval |
| | 2-2260 | P | " |
| SUPPORT: | | | |
| OS/SR&CD | 3-2260 | A | Security name checking |

2. Requesting a Terminal. This discussion on the general criteria and procedures used in requesting and installing terminals is inserted at this point in the report in order to assist prospective users of the time-sharing system.

a. General. Before requesting a terminal, the user should be familiar with the criteria for eligibility for interactive services, which does not necessarily substitute for batch processing service. It is intended for:

--Applications requiring a reasonable degree of interaction between the user at the terminal and the central computer system.

--Applications requiring extremely short response time; times that must be measured in seconds (or in some cases minutes) instead of hours.

--Applications requiring relatively little direct user input and low volume output going directly to the user.

It is not intended for:

--Applications involving large input/output at the terminal.

--ADP Systems involving numerous processing steps and unique outputs. For example, payroll.

--Applications which would benefit from a remote job entry capability, but which do not necessarily require quick response time.

--Requirements which fall into the "nice to have" category, but which have not been subjected to ADP systems analysis to determine feasibility, cost, and alternative ADP methods.

--Compute-bound programs; that is, programs which will saturate the system and cause extensive degradation in the service to other users.

b. Preliminary Analysis of User Requirement. As in any potential ADP application, a preliminary discussion and analysis of the prime factors in the user's requirement should be arranged between the user component and the appropriate applications division of OCS. This joint user/OCS analysis should determine if interactive services are essential or necessary to support the user requirement.

c. Formal Request for Interactive Service. If the preliminary analysis indicates that a valid need for interactive services exists, the user office should submit a formal request to OCS through the Information Processing Coordinator of the appropriate Directorate for concurrence.

The following items should be included in the request:

--Brief explanation of the application, including normal periods of terminal operation, frequency of terminal use, file size, and other pertinent factors.

--Justification for interactive support in lieu of conventional ADP support.

--Type of programming support that OCS must provide to the user.

--Security classification and special clearance category of data which the user will process.

--Requested operational date. This should be a reasonable estimate based on the physical installation task schedule, readiness of application programs, and availability of data bases.

If existing or planned terminals cannot be utilized to provide the requested service, the following items should be included in the request:

--Type of terminal desired such as IBM 2260 Display Station, IBM 2741 Communications Terminal, DATAPORT, etc. Except for expensive, special purpose terminals, the rental costs of terminals connected to OCS computers will be budgeted and funded by OCS.

--Physical location of terminal. Terminals should be shared to the extent feasible to minimize equipment and installation costs. Physical security policy requires that terminals be installed in a secure area unless (a) facilities and procedures are available for physical disconnect in the computer center and (b) the terminal area is attended while it is connected.

Most terminals require special data communications circuitry between the terminal and the central computer. The cost of secure areas and communication lines will be borne by the user office. Actual installation of a terminal is currently taking four to eight months, because of the extensive coordination and approval pattern for individual requests (see d. through h. below) and the shortage of electricians. This work cannot be initiated without specifying a firm physical location and firm terminal type, since conduit must be emplaced, wires must be pulled, and terminal equipment must be acquired.

--User office FAN NUMBER (for charging installation costs).

d. OCS Response to Interactive Services Requests. OCS will acknowledge user requests and send a memorandum through the Office of Security to the Office of Communications to obtain security approval and installation of the data communication lines. Concurrently, OCS will begin system and software development.

e. Office of Security. The Office of Security will survey the planned location for the terminal and review other pertinent aspects of the operational environment; the OCS Security Officer and the responsible OCS applications division will join in this activity. If the user and the computer systems analyst have performed the proper groundwork and security requirements as explained in para. 2. c. have been met, security approval should be fairly routine.

f. Office of Communications. After OS concurrence, the OCS memorandum will be forwarded to the Office of Communications, where 25X1 [redacted] will prepare detailed specifications for the installation of data communication lines. These specifications will be coordinated with the Communications Security Staff and then be forwarded to the Logistics Services Division, Office of Logistics.

g. Logistics Services Division. The Space Allocation and Facilities Branch/LSD/OL will take action with the GSA building maintenance personnel to install conduit and communication lines between the terminal area and central computer facility.

[redacted] 25X1

i. Status of Interactive Services Requests. OCS will act as a focal point to provide users with information concerning the progress and status of their time-sharing service.

j. Conclusions. The OCS Interactive System is in a developmental and initial operational phase. Uncontrolled expansion of the system while it remains in a dynamic developmental state could lead to processing inefficiency, difficulties in managing the integration of facilities, and dissatisfaction from users who depend on the system operationally. Special security procedures may be necessary for terminal installations. Installation of data communications lines is cumbersome, slow, and expensive. All of these factors point to the fact that requests for support in this new and promising technology should be carefully planned and evaluated.

## C. Control Program (CP-67)

1. Structure. Control Program (CP-67) is a "super" control program; it is not an operating system. Operating systems such as OS, BOS, CMS, and even the OS based TSMON system run at a lower hierarchical level. All are equal and subordinate to CP. The usual operating system performs two chief functions: resource (time, memory, device) allocation and data management. CP performs only resource allocation. Thus, multiple combinations of resources such as memory, tapes, and DASD (Direct Access Storage Device) space can be defined to CP. A package of the above resources (memory, tapes and DASD) in effect is the equivalent of a computer; the only missing ingredient is central processor time. CP allocates this time by giving time increments of CPU power to each defined combination of resources.

In effect, we see a small control program (CP) controlling multiple combinations of resources, each combination having its own software or operating system. Thus, we say that CP controls imaginary computers (virtual computers in the trade terminology). Each virtual computer can run any IBM 360 system and program. When CP must allocate more memory than it has, it uses hardware features of the Mod 67, and uses DASD space to simulate the real memory requirements.

2. Problems. The installation of the hardware and the software is an ambitious undertaking but reasonably feasible. The software task team is a well balanced team with the necessary skills to produce a successful and timely implementation. However, there is little backup of skill resources; sickness, resignations, or other priority tasking of team individuals would seriously disrupt system implementation.

There are problems to resolve, and in all cases, approaches to solutions are not clearly defined. The purely technical problems, in most cases, will be solved as part of the normal workload of the implementation team. Thus far, solutions to nearly all technical problems have been fast. For example, the initial draft of this paper listed a timing problem in the TSMON interaction with CP and the fact that PL/1 I/O did not execute properly under CMS. Both problems were very serious in that major segments of the system were not usable, and to the system user or even the pessimistic EDP manager, these two problems could easily generate an attitude of "it will never work." However, by the time of the final draft of this paper, both problems (and many others) had been solved by the implementation team.

Some obvious technical problems which are not totally solved as of this time (April 69) are as follows:

a. The IBM 2260 cathode-ray tube devices do not work correctly under some conditions.

b. There are timing inconsistencies with remote high speed controls.

c. Not all user requested features have been incorporated into TORQUE, the file query language.

d. The changing of the scheduling algorithm in CP to allot more time slices to TSMON is not completed.

The most serious problems in the full implementation of time sharing will be administrative and managerial. Access, security, types of terminals permitted, number of terminals and data sets, physical environment for remotes, sizes of programs, types of operating systems allowed, allocation of data set storage, man/machine languages which are system supported, number of predominantly computational programs, control of module packaging, etc. are problems that can be partially solved by administrative control and standardization. At this time, answers or even enough facts to suggest answers are not available on these topics. The environment of interactive services is new and unfamiliar. Only with time will enough managerial experience emerge to administer the above problems with assurance. In the conversion from second to third generation systems, the list of potential problems proved to be the most valuable managerial checklist. Many of the items on the list proved of no importance; others were solved at the appropriate time; and still other unanticipated ones were uncovered. Hopefully, because new applications rather than conversions are to be implemented on the interactive system, and with sound planning, the above problems will be solved during the growth of interactive services.

3. Systems Available Under CP. OCS presently plans to implement the TSMON, CMS, OS, and APL-360 systems. In addition, we hope to test and evaluate other systems mentioned below. If these prove useful and necessary to the users, and if OCS has the resources to support these systems, a decision will be made to do so. For a detailed description of each of these systems, refer to the "OCS INTERACTIVE SERVICES" report, November 1968.

a. TSMON (Time-Sharing Monitor). This is the present OCS interactive system and is being given the highest priority for implementation.

b. CMS (Cambridge Monitor System). This is an operating system designed specifically to run in a paging environment under CP and is being given next priority. Under CMS there are additional user services available:

(1) Data Set Handling. Facilities are available in CMS for the handling of disk, card, and tape files. Most of the CMS commands, however, require that accessed files be on disk. Thus, card and tape files should be transferred to disk for effective use. Commands are available for file creation, file maintenance, and file manipulation.

(2) Execution Control. In CMS, several commands are available to the user for execution control, i.e., the loading and running of programs. Files (or programs) which are to be loaded and run must reside on disk and must be either in relocatable object code form or in core-image form.

(3) Debugging Facilities. In CMS, a debugging tool is provided to allow the user, while at his terminal, to examine and change the contents of core locations, program status words, general purpose registers, the channel status word, and the channel address word; to dump portions of core at the terminal or on an offline printer; and to stop and restart programs at any specified point or points. Two commands allow the user to trace supervisor calls (SVC instructions).

(4) FORTRAN Language Processor. In CMS, the FORTRAN command will compile files of FORTRAN source language into machine-language object code. Up to 32 files may be compiled by one command by listing the file names, and each file may contain any number of routines.

(5) Assembler Language Processor. In CMS the ASSEMBLE command creates relocatable object programs from programs written in the IBM 360 assembler language (ALC).

(6) PL/1 Language Processor. In CMS, the PL/1 command compiles files of PL/1 source language into machine language object code.

c. OS (Operating System). The same version of OS/360 used by the Computer Center will be used on the 360/67 as well. This is the largest and most comprehensive of the several operating systems furnished with the IBM 360. The use of other versions of OS will be on an ad hoc basis with no guarantee of support. Until the effects of large virtual OS machines are well understood, virtual core size for this system will be limited to 512K bytes.

d. TDMS (Time-Sharing Data Management System). This system enables individuals with large complex data files to manage their data with speed and ease. Like TORQUE II, it is oriented to the non-programmer user, who after learning some basic commands, can work directly with the computer to manage his data base. TDMS permits the user to describe entries in a data base, to load them into the machine, to ask questions about them, to perform calculations on them, to have the data displayed on a display terminal, to obtain hard copy reports, and to update and maintain the data base. Testing of this system will be done to determine whether it works well under CP and whether it can meet any user needs.

e. ATS (Administrative Terminal System). This system consists of a group of application programs and related computer equipment designed to support secretarial operations such as drafting large texts and handling data files. Data is entered into the system through the IBM-2741 terminal. This system will be available for test shortly.

f. APL-360 (A Programming Language). This is a mathematically oriented language developed by IBM which is useful to the non-programmer on a remote terminal. With a minimum of typing, the terminal user is able to perform arithmetic operations and complex mathematical functions and receive an immediate response. Operations and responses can be stored in a library for future access. This system will be available in Mid-1969.

## D. Security of OCS Interactive Facilities

The purpose of this section is to acquaint users of OCS interactive facilities with the controls and procedures which must be observed to ensure system security and integrity.

1. OCS Computer Center. The OCS Computer Center is an approved secure area. Because of the nature and volume of the material being processed in the Center, access is strictly controlled by OCS. Requests for additions to the Computer Center Access List should be submitted to the Security Officer, OCS and should include the employee's name, Agency badge number and the reason for requesting access.

2. Terminal Network. Lines, conduit, junction boxes, distribution panels and receptacles which are part of the OCS interactive facilities communications network will be installed in accordance with standards established by the Office of Communications and the Office of Security for the protection of classified transmission lines.

3. Terminal Sites. A terminal may be installed to service one or more offices. However, one office will be designated as the Responsible Office. Generally, a terminal will be installed in an approved secure area, unless provision can be made for disconnecting the terminal lines at the Computer Center when a terminal is unattended. Proposed terminal sites must be approved by OCS, the Office of Communications and the Office of Security.

A staff employee assigned to the Responsible Office will be designated as Terminal Monitor. A Terminal Monitor will be responsible for:

a. Serving as a contact for OCS on matters relating to the security and use of the terminal;

b. Providing for the protection and dissemination of password and keyword data;

c. Ensuring access control at the terminal site when the terminal is on-line;

d. Securing the terminal site when the terminal is unattended.

4. Terminal Use. Each Responsible Office will provide the Security Officer, OCS with the following information:

a. The normal hours of operation of the terminal. Attempts to use the terminal at other than the specified normal operating hours will be rejected unless prior arrangements for use of the terminal have been made with the Chief, Operations Division, OCS.

b. The names of personnel authorized to use the terminal, and any additional data that may be required to identify users for authentication purposes.

c. Any changes in the list of personnel authorized to use the terminal.

Every individual terminal user will be responsible for immediately reporting to the Chief, Operations Division, OCS any accidental display of information or other unusual occurrence at a terminal possibly connotating a failure in system security. In the event of repeated failures at one terminal or failures at several terminals, the Chief, Operations Division will render particular terminals or the entire system inoperative, until the problem has been diagnosed and remedial action has been taken.

5. Terminal Maintenance. No terminal maintenance will be performed without the approval of the Chief, Operations Division. Maintenance will be performed only by cleared personnel assigned to OCS or, in emergency situations, by uncleared persons escorted by employees of OCS or employees of the Responsible Office who are authorized access to the terminal.

No terminal will be replaced or modified without the approval of the Chief, Operations Division.

No terminal will be relocated without the approval of the Security Officer, OCS, who will be responsible for coordinating any proposed terminal relocation with the Office of Communications and the Office of Security.

No electrical equipment, including telephones, will be installed or relocated in the room in which a terminal is located without the approval of the Security Officer, OCS.

6. Terminal Log. Each Responsible Office will be provided with a regular copy of a Terminal Log which will contain information concerning terminal use, users' names, files accessed, unsuccessful log-on attempts and password and keyword failures. This log should be used for auditing terminal use.

7. Exceptions. Exceptions to the above procedures will be permitted only with the approval of the Director of Computer Services.

## VI. OCS RESPONSIBILITIES

### A. Available Manpower

OCS has nine staff people working on the interactive system. In addition, IBM has committed three people, 80 percent of their time, to aid in the design and implementation of the interactive system. Each of the staff members has special and unique skills and a loss of any one of these persons could seriously affect the effort. Cross-training and qualifications back-up is somewhat limited.

Each of the three applications divisions in OCS is responsible for determining whether applications should be implemented in the batch or interactive mode. The tradeoffs in many cases are subtle and managerial control must be firm. Since the interactive service is new and relatively few examples of interactive applications can be studies, creativity must play a dominant role in the systems analysis and design process.

### B. Development Schedule

The Development Schedule is described in terms of objective levels. No dates have been attached to any of these objective levels since each objective level involves many variables.

1. Objective Level One. The objective was to have the present OCS time sharing system, TSMON, running on the Mod 67 in March with the machine running as a Mod 65. This has been achieved and facilities are provided which are identical to that of the Mod 50.

2. Objective Level Two. There are some obvious deficiencies in the present system--in the monitor program system, in system services, and in the user programs. These deficiencies are being corrected; changes are being designed; and implementations are being integrated into the system. In one case, the package of TRUMP and TSAR is being replaced by a much more sophisticated package which will be called TORQUE II. Objective level two is to implement this improved version of the present system and its services on the Mod 67 which will be used as a Mod 65. Thus this level will see an improved version of the present system operating on a more powerful machine than the Mod 50. In objective levels one and two, none of the special power of the Mod 67 will be utilized.

3. Objective Level Three. This objective level is to have CP, the Control Program for the 67, operating with TSMON subordinate to it. CP must be changed to support TSMON. For example, code for the data cell must be inserted into the CP package. TSMON supports a 2701 control unit

for the COINS network; coding for this control unit must be inserted in CP. Coding for these devices in CP is being done concurrently with the redesign and reimplementation of TSMON modules. This objective has been met; CP is currently running with the present version of TSMON.

4. Objective Level Four. At this level it was desired to have CP controlling not only the TSMON of the previous level but also the operating system CMS which was especially designed for CP. This CMS system will offer the additional services of PL/1, FORTRAN, ALC, and a special edit and execute control function. Since CMS was specially designed to complement CP, it is anticipated that this will be a powerful addition to the system. There should be no major problems in implementing CMS under CP. A possible problem exists in that the present scheduling algorithm in CP would schedule the same total amount of time to all TSMON users as it schedules to each of the CMS users. Design is underway to change the scheduling algorithm in CP to compensate for this time allocation deficiency to TSMON. This system is tested and has been running at night.

5. Objective Level Five. In this level OCS plans to implement the standard operating system OS as another subordinate system to CP. Thus, the full facilities of the present system in OS would be available. Each user of OS would use an independent copy of PCP. OS has been successfully used and tested under CP by OCS.

6. Objective Level Six. In this level, OCS plans to install and test various other systems such as APL/360, TDMS (Time Sharing Data Management System), and ATS (Administrative Terminal System). Assuming good results and the ability to support them, these systems would be available with the various capabilities (and restrictions) to users.

As a development effort with a tentative December 1969 target date, the possibility of placing multiple copies of a single user version of TSMON subordinate to the CP system will be tested.

C. Training

1. Systems Personnel. Systems Personnel and two of the IBM personnel have attended a two week course on the CP and CMS software at Cambridge, Massachusetts, during November 1968. Other APS personnel have been trained on the TSMON system and its modules. A series of tutorials on CP, CMS, and TSMON has been completed for all members of the staff.

2. Applications Personnel. One member of the APS staff has been assigned as a customer and applications programmer interface. He is designing special tutorial sections which began in January 1969 for applications personnel who must understand the system so that they can adapt its power to their customer's problems. These tutorials will be internal OCS courses.

3. Computer Operators. One senior computer operator has been assigned as a permanent chief of the present interactive operations. He has the responsibility of training other personnel. This senior operator, the systems administrator, spent one week in November at Cambridge for detailed operator training. The operator function for the system is complicated since the systems administrator must operate many different systems concurrently. The systems administrator will be learning on the job; he will be present as each of these new systems (or new versions) are integrated into the system.

4. Terminal Users. Terminal usage briefings are available upon request for potential users. A person is available for group or individual briefings. The APS customer representative has already given several briefings explaining the interactive services that will be available to all users.

A six-hour course and demonstration on the use of terminals is being planned for potential customers. The course "Use of Interactive Terminals" and the dates will be published in the June 69 issue of the OTR Bulletin.

# APPENDIX A.
# ALTERNATIVE INTERACTIVE SYSTEMS

In April 1968, OCS began an evaluation of interactive systems with the intention of selecting a hardware system to replace the IBM 360/50 in January 1969.

Many systems were considered and evaluated by OCS. The systems studied in depth were:

a. TSMON - 50, the then current OCS time sharing system.

b. TSMON-RUSH - 50, the OCS time sharing system with LCS (large core storage).

c. ADEPT - 50, the SDC (System Development Corporation) system which is being tested by IPRD.

d. TSMON - 65, the OCS system on a Mod 65.

e. TSMON-RUSH - 65, the OCS system with LCS on Mod 65.

f. ADEPT - 65, the SDC system on a Mod 65.

g. TSS-67, IBM's time sharing system for the Mod 67.

h. CP-67, a virtual machine system written by Lincoln Lab and IBM for Mod 67.

i. TS-70/46, a system by RCA, written for the Spectra 70/46.

j. Duplexed small machines such as twin Mod 40's or twin S-70/46's.

The Executive Director-Comptroller, acting on the recommendation of the DD/S&T, approved selection of the IBM 360/67. It was ordered for January 1969 delivery, to coincide with the planned release date of the IBM 360/50.

There were many considerations in the selection of the Mod 67, not all of which could be identified as clear pluses or minuses. Selection was based primarily on the availability of the powerful control program (CP-67) which performed favorably in comparison performance tests.

Secondly, the system was in operation--not developmental--and did meet all objectives for the new system as previously determined by OCS. Negative factors were the high cost (although still within the planned budgetary limits), and concern about putting all interactive file material on one physical system (from a security standpoint).

The Mod 67 was most attractive for several reasons:

a. It is a good time-sharing system with fast response and is fully compatible with other OCS batch processing systems. It can operate as a Mod 65, or when working in the interactive mode, its background processing power is at least 50% of a Mod 65.

b. The system comes very close to meeting a general capability for both multiprogramming and time-sharing.

c. CP-67 can provide not only a production time-sharing environment but also an experimental one by making each new set of facilities an independent virtual system under CP-67. Thus experiments can run concurrently with production to test subordinate time-sharing systems.

d. Conversion is fairly easy. TSMON can run under CP as soon as several new peripheral support modules are finished.

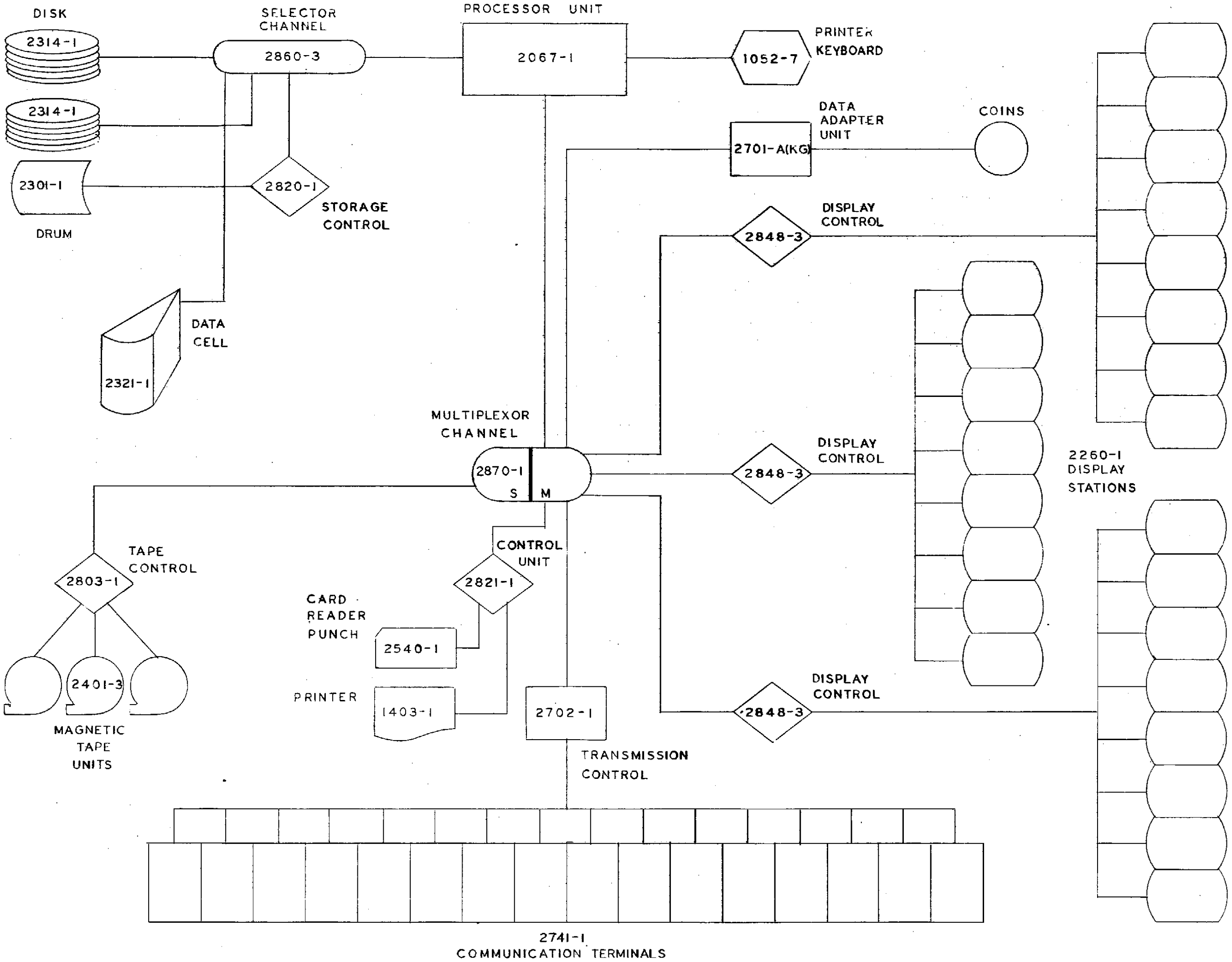
e. CP-67 has the power to expand greatly. At another installation it is giving excellent response, both computational and query type, to 20-30 terminals in a routine production mode. Another user occasionally has 60-80 terminals active. Response to peak load conditions is an important measuring factor of a time-sharing service, and the Mod 67 does have a powerful processing unit which should absorb much interactive processing.

f. Manpower Resources. Several time-sharing systems have been written for IBM 360; each can be tested as virtual systems under CP-67, and the developed packages can be extracted or run unchanged. No other system will give us as much software flexibility or expansion capability as CP-67, thus it allows more human resources to be placed on solving customer problems. Because of CP's size and power, new devices and new functions can and will be added. Thus, OCS must be prepared to staff this effort with an adequate number of system programmers.

g. It has the best internal compartmentation hardware of any one machine, and has possibilities of being engineered to absolute compartmentation, thus significantly, if not totally, reducing the new technical security problem introduced by time-sharing multi-programmed systems.

## APPENDIX B.
## CONFIGURATION OF THE IBM 360/67

| Qty | Type | Description |
|---|---|---|
| 1 | 2067-1 | Processor Unit |
| 1 | 1052-7 | Printer Keyboard |
| 1 | 2301-1 | Drum Storage |
| 2 | 2365-2 | Processor Storage |
| 3 | 2401-1 | Magnetic Tape Units |
| 16 | 2741-1 | Communication Terminal |
| 1 | 2803-1 | Tape Control |
| 1 | 2820-1 | Storage Control |
| 1 | 2860-3 | Selector Channel |
| 1 | 2870-1 | Multiplexor Channel |
| 1 | 1403-N1 | Printer |
| 1 | 1416-1 | Train Cartridge |
| 24 | 2260-1 | Display Station |
| 2 | 2314-1 | Direct Access Storage Facility |
| 1 | 2321-1 | Data Cell Drive |
| 1 | 2540-1 | Card Read Punch |
| 1 | 2701-A(KG) | Data Adapter Unit |
| 1 | 2702-1 | Transmission Control |
| 1 | 2821-1 | Control Unit |
| 1 | 2841-1 | Storage Control |
| 3 | 2848-3 | Display Control |
| 6 |  | Feature #4635 Line Adapter on 2702-1, SN 20798 |



CONFIGURATION OF THE IBM 360/67

# APPENDIX C
# GLOSSARY OF TERMS

Address - A label, name, or number identifying a register, memory location, or I/O unit where information is stored.

Algorithm - A fixed step-by-step procedure for accomplishing a given result; usually a simplified procedure for solving a complex problem, also a full statement of a finite number of steps.

Batching - The stacking of a number of programs or jobs in a queue for processing. Adj. Describing the type of operating system which inputs a linear queue of workflow.

Byte - A sequence of adjacent bits (binary digits) treated as a unit. Bytes in most recent hardware contain 8 bits. All alpha/numeric characters for a given machine are bytes.

CP (Control Program) - A super control program which performs resource allocation for subordinate operating systems (OS, CMS, TSMON, DOS etc.). It can run only on the IBM 360/67.

Central Processing Unit (CPU) - The unit of the computer which interprets and implements the software instructions, performs the arithmetic and logic functions, and generally controls the logic flow of data to and from memory and I/O components.

Code - v. t. To write the man/machine language instructions for a machine. n. The product of above writing.

Compression - Applying a formula to the bit pattern of a word to convert it to abbreviated form in order to reduce its storage requirements.

Conversational Capability - The attribute of certain software which augments interaction with the user in a prompting or pedogogic manner. Several conversational man/machine languages (e.g., SOLVE, BRUIN, etc.) and operating systems have been written. Conversational versions of several common man/machine languages, such as FORTRAN and PL-1, have been implemented on a few computer systems. CMS, although lacking some typical conversational features, is an example of a conversational operating system.

Core Allocation - The assignment of specified blocks of storage to receive blocks of instructions and/or data.

DOS - A disc oriented operating system for the IBM 360 with fewer capabilities than O. S.

Data Set - A collection of related records treated as a unit. In the past, commonly referred to as a file.

Debugging - The process of isolating and correcting all malfunctions and/or mistakes in a piece of equipment or a computer program.

Diagnostics - Pertains to the detection, discovery, and further isolation of a malfunction or a mistake.

Dynamic Relocation - The relocating of computer programs and data in memory while the program is executing. In some systems, particularly in time sharing systems, this relocation may be frequent such as between sequential instructions. Generally, special hardware is necessary to perform this relocating efficiently.

Hardware - The mechanical, magnetic, electrical and electronic devices or components of a computer.

Hashing - A method of converting a word into a bit pattern which can be used for various purposes, such as error checking or as a simplified and efficient address to the location of the word or a record containing the word.

I/O - The abbreviation for Input/Output.

IPL - The abbreviation for Initial Program Load which refers to loading the operating system. This is the first function to be performed in placing a software system on hardware for processing.

Indexing - A method of quickly locating information in a file; look-up consists of a search of the index in which is stored the address of the location for the information.

Interactive - Describes a computer system or method in which the user interacts directly with the machine, entering commands and receiving responses. Generally, it is assumed that the delay between command and response is short enough to keep the user's mind occupied with the problem at hand (i. e., not diverted elsewhere while the machine is busy).

Interface - A common boundary; e.g., a boundary or unit between two systems, two software modules, or two devices. In computer systems, interfaces among software modules are designed very carefully, since the volume and speed of data transfers at these points is high and prone to excessive error rates.

Language - A defined vocabulary and grammar used by humans to give commands to a computer. Such languages range from primitive cryptic languages in symbolic mnemonics to sophisticated expressive languages in scientific notation or in natural language vocabulary and syntax.

Library - The filed collection of available and use-proved computer programs, either printed, kept on tapes, or in random-access file for quick checking, reference, or use with or without modification.

Multiprocessing - Describes a system in which two or more central processing units of the same type are connected to share processing managed by a single operating system. Programs which are processed may use any of the central processors dynamically assigned by an availability-demand algorithm. Generally, multiple connections of systems in which the scheduling algorithm does not permit program sharing of the CPU's are not considered full or true multi-processing systems (e.g., the Attached Support Processing System for the IBM 360). A true multiprocessing system such as the CDC-6500, the GE-600 series, and the IBM SIESTA (2 IBM 360/65's connected) is designed to operate either (a) in a master/slave mode in which one CPU contains the single operating system or (b) in an equivalence mode in which the operating system uses all CPU's.

Multiprogramming - A technique in an operating system for handling numerous routines or programs simultaneously by overlapping or interleaving their execution: i.e., permitting more than one program to share machine resources. Typically, programs in such a system are processed by a priority scheduling algorithm which allots the highest priority program all the central processor time which it can profitably use.

Operating System - An organized collection of software techniques and procedures which creates a capability to process programs efficiently. Operating systems generally are designed to utilize the hardware efficiently (batch systems) or to service customers efficiently (interactive systems). The two principal functions of a system are to distribute resources such as memory, I/O components, and arithmetic/logic power to programs, and to manage the data sets being processed and stored on-line.

On-line - Descriptive of a system and peripheral equipment or devices in which the operation of such equipment is under the control of the central processing unit.

O.S. (Operating System) - The official name for the most sophisticated of the various IBM produced systems for the series of larger standard IBM 360's. (Models 40, 50, 65, 75, 85.)

Overhead - The amount of time the Central Processing Unit (CPU) is devoted to performing System's functions, i.e., the allocation of resources and data management.

PL/1 (Programming Language #1) - This procedural language was designed by IBM as a general purpose language for 3rd generation computers. It has features which enhance the utilization of the most recent hardware advances. In addition to the usual compile time diagnostics, it has execute time diagnostics, and sophisticated on/off optional debugging features. The language has the capabilities for scientific, commercial, real-time, and command/ control use. In addition to its extensive use on IBM 360, it is being implemented on 3rd generation equipment on several other manufacturers.

Page - A specified number of bytes which can be instructions and/or data. The size is fixed per system and is usually a power of two such as 4096 ($2^{12}$) for the IBM 360/67.

Paging - The dynamic transferring or exchanging of pages between memory and external direct access I/O components. In general use, the term implies the dynamic exchanges of the executable pages of a program from a virtual memory to and from a real memory. In the TSMON system, paging as used in PAM, the paging access method, means the management of efficiently packed data sets.

Partitioned Core Storage - Separating core memory into partitions. Generally, this is done so that independent programs can be placed in separate partitions for concurrent processing. The technique of using fixed size partitions is generally easier to implement than that which uses dynamically variable sized partitions. The general use of the term implies fixed size partitions.

Peripheral - Not part of the main unit. Peripheral components of a computer include such devices as tape drives, card readers/ punches, printers, drums, and disks. Sometimes, small computers which support a larger computer are termed peripheral systems.

Queues - A number of items or persons waiting for service.

RJE (Remote Job Entry Terminal) - Small remote peripheral computer to handle input and output to and from a larger processor--typical input media is card reader and typical output media is a Printer and Punch. Normal use is to enter batch-type jobs remotely.

Register - A term to designate a specific computer unit for processing a group of bits or bytes. Registers have specific functions which generally can be exploited in efficient ways during programming.

Remote Terminal - A communication device remotely situated but electrically connected to a computer providing services of the system.

Routines - A sequence of machine instructions that carry out a defined function.

SYSGEN - Process of generating the operating system. In this process the specific capabilities which are desired for the system are extracted from the master system file, and matched to the specific hardware configuration.

Software - All of the products which are required to produce the algorithms, instructions, and procedures to make the hardware produce the desired results.

SPOOL'ing - Simultaneous Peripheral Operation On-Line which implies that the central computer will simultaneously perform the initial input operations (generally card reading) for one task, the final output operations for another task (generally printing and card punching), and the main processing for a 3rd task.

TSMON - Time Sharing Monitor System which has been developed by the Office of Computer Services for the IBM 360.

Third Generation - Term describes computers designed and produced after 1964; distinguishing advances include:

1. Solid logic technology.
2. More efficient packaging of miniaturized components.
3. Sophisticated interrupt logic.
4. Powerful I/O multiplexing.
5. Capability of processing both "byte" and "word" data.

Typical software advances include: advanced multiprogramming and time-sliced operating systems with conversational language capabilities.

Throughput - Total production of system.

Time-Sharing - The sharing of predefined central processor time among multiple concurrent users. Central Processor time, in addition to other resources such as memory and storage space, can be allocated to concurrently running programs in operating systems, whether these systems are batch or interactive service oriented. The term, time-sharing, as generally used implies use in an interactive system. In an interactive system the technique for sharing the time is to interrupt the time distribution to each user on either of two factors, a) unable to use central processor time prifitably, i.e., he is in the WAIT state, or 2) exhausting of his time allocation. In the interactive system, generally all users have equal priorities and equal time allocations.

Turnaround - The particular amount of time that is required for a computation task to get from the programmer to the computer, onto the machine for a test or production run, and back to the programmer in the form of the desired results.

Virtual - is synonomous with imaginary.

Virtual Machine - An imaginary machine (X bytes of core, Y tape drives, Z cylinders of direct core storage) which is requested by a computer user; a central computer in turn simulates this imaginary machine and provides the user with equivalent services.

Virtual Memory - The imaginary core memory which is simulated by a central machine for a user; this imaginary core memory actually is derived from some combination of core and direct access storage.