AD 680248

This document has been approved for public release and sale; its distribution is unlimited.

TM-3628/002/00

ADVANCED DEVELOPMENT PROTOTYPE

Distribution Of This Document Is Unlimited

Semiannual Technical Summary Report to the Director, Advanced Research Projects Agency for the period 30 January 1968 to 30 July 1968

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

DDC
RECEIVED
JAN 9 1969
B

Sponsored by Advanced Research Projects Agency

ARPA Order No. 939

57

BEST
AVAILABLE COPY

TM-3628/002/00

TECHNICAL MEMORANDUM

(TM Series)

ADVANCED DEVELOPMENT PROTOTYPE

Semiannual Technical Summary Report
to the Director,
Advanced Research Projects Agency

for the period
30 January 1968 to 30 July 1968

J. I. Schwartz, Project Director

SYSTEM
DEVELOPMENT
CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

The work reported herein was supported by the Advanced Research Projects Agency of the Department of Defense, and was monitored by the Information Processing Techniques Office under Contract DAHC15-67-C-0149.

Distribution of this document is unlimited.

ABSTRACT

This report describes work done under the Advanced Development Prototype contract from 30 January 1968 to 30 July 1968. The result of this work is ADEPT--a comprehensive information-processing system being implemented at SDC for operation on IBM 360 computers. The report includes an overview of the current status of system development, and a detailed description of the three major components of ADEPT: a time-sharing executive, a data management component (consisting mainly of the Time-Shared Data Management System), and a programmer's package, which includes a JOVIAL compiler, editing, debugging, and utility programs, a teletype interpreter (TINT), and an Interactive Programming Support System. Also included in this document are the names of staff members assigned to each of the three major project areas, as well as a listing of the documents produced in each area during this reporting period. Upon request, referenced documents will be made available to appropriate organizations.

CONTENTS

ABSTRACT 1

1. OVERVIEW 5

1.1 Introduction 5
1.2 System Components 5
1.2.1 Time-Sharing Executive 6
1.2.2 Data Management System 6
1.2.3 Programmer's Package 6
1.2.4 Hardware Configuration 6
1.3 Project Status 7
1.4 Organization of Report 8

2. TIME-SHARING EXECUTIVE 9

2.1 Introduction 9
2.1.1 Basic Executive (BASEX) 11
2.1.2 Extended Executive (EXEX) 14
2.2 Progress 14
2.2.1 Capabilities 14
2.2.2 Reliability 18
2.2.3 Performance 22
2.3 Plans 24
2.3.1 Capabilities 24
2.3.2 Reliability 26
2.3.3 Performance 26
2.4 Staff 27
2.5 Documentation 27

3. DATA MANAGEMENT SYSTEM 33

3.1 Introduction 33
3.2 Progress 33
3.2.1 Define 33
3.2.2 Load 34
3.2.3 Query 34
3.2.4 Compose 34
3.2.5 Update 35
3.2.6 Maintain 35
3.2.7 Reformat 36
3.2.8 Data Base Oriented Programming Language 36
3.2.9 META5 Adaptation 36
3.3 Plans 37
3.4 Staff 37
3.5 Documentation 38

4. PROGRAMMER'S PACKAGE 43

4.1 Introduction 43
4.1.1 JOVIAL Compiler. 43
4.1.2 Debugging Aid. 43
4.1.3 Editing Aid. 43
4.1.4 Utility Program. 44
4.1.5 TINT (Teletype Interpreter). 44
4.1.6 Interactive Programming Support System (IPSS). 44
4.2 Progress 44
4.2.1 JOVIAL Compiler. 44
4.2.2 Debugging Aid. 45
4.2.3 Editing Aid. 45
4.2.4 Utility Program. 45
4.2.5 TINT (Teletype Interpreter). 46
4.2.6 Interactive Programming Support System (IPSS). 47
4.3 Plans. 50
4.4 Staff. 51
4.5 Documentation. 51

APPENDIX: ADVANCED DEVELOPMENT PROTOTYPE SYSTEM BLOCK DIAGRAM 53

1. OVERVIEW

1.1 INTRODUCTION

Work on the Advanced Development Prototype contract was begun in January 1967 for the purpose of demonstrating--in an operational environment--the potential of automatic information-handling made possible by recent advances in computer technology, particularly advances in time-sharing executives and general-purpose data management techniques. The result of this work is a large-scale, multi-purpose system (known as ADEPT), which operates on IBM System 360 computers. The historical background and early development of this system can be traced in the two preceding reports in this series.*

Most of the basic features of the ADEPT system were operational during this reporting period. Accordingly, the Phase I version of the system was installed at the National Military Command System Support Center (NMCSSC) in June, and is now being subjected to extensive testing and evaluation. As improved versions of system components become available, they are installed at NMCSSC under the supervision of SDC personnel. Training sessions for Support Center personnel are also being conducted.

During the next reporting period, the ADEPT system will be installed at the Air Force Command Post, where it will be further tested. In addition, NMCSSC personnel are expected to begin to use the system experimentally and to provide valuable feedback on the operation of the system to its designers.

Information on the ADEPT system was disseminated to the military and governmental communities through a series of ADEPT-50 Symposia. The first symposium, which was held at SDC Santa Monica on 24 and 25 April, was attended by over 200 people. Sponsored by ARPA and SDC, it consisted of a set of briefings on the operation and capabilities of the system, and live demonstrations of major system components. Two more ADEPT-50 Symposia were held at Andrews Air Force Base, Maryland on 10 and 11 July. These two sessions, which were sponsored by SDC, DCA, and OASD Comptroller, were attended by more than 500 people. Briefings and demonstrations on the system were again presented.

1.2 SYSTEM COMPONENTS

The ADEPT system consists of three major components: a time-sharing executive; a data management system adapted from SDC's Time-Shared Data Management System (TDMS); and a programmer's package. Each of these components is discussed in turn below.

---

*TM-3628/000/00 and TM-3628/001/00.

1.2.1 Time-Sharing Executive

The time-sharing executive of ADEPT consists of two major pieces, one called the Basic Executive (BASEX), the other the Extended Executive (EXEX). BASEX contains elementary functions such as an input/output processor, a scheduler that will eventually permit the dynamic adjustment of priorities, an interrupt processor, and a basic sequencer. EXEX contains routines to interpret user commands, file-inventory routines, and various other aids for both programming and nonprogramming users. The Extended Executive is an "open-ended" module that permits expansion when necessary. ADEPT provides for multiple access to the computer through a variety of input/output devices, including typewriter keyboards, small tabular displays, and cathode-ray-tube graphic terminals.

1.2.2 Data Management System

The data management portion of the ADEPT system consists mainly of a set of integrated programs designed to handle the most frequently performed data management tasks. Included in this set are programs that allow the user to describe the entries in a data base, load them into the machine, ask questions about them, perform calculations on them, have them presented for his analysis, obtain hard-copy reports, and update and maintain the data base. Several related capabilities are also being developed as part of the data management portion of ADEPT; these include a procedure for integrating the data management features of TDMS with the computational capabilities of JOVIAL, and a means for reformatting existing data bases to TDMS format using the META5 language.

1.2.3 Programmer's Package

The third major component of the ADEPT system is designed to provide several powerful tools for programmers with varying skill levels. For the professional programmer, a JOVIAL compiler and a number of service programs (including editing and debugging routines) are provided. For the novice programmer who may occasionally wish to use a computer to solve short, "one-shot" problems, a user-oriented interpreter (TINT) is provided. In addition, an integrated system that provides extensive support to programmers at all levels is being developed as part of this portion of ADEPT. This system--based on SDC's Interactive Programming Support System--assists machine users in all of the programming processes, ranging from program composition, editing, execution, and testing, through program documentation.

1.2.4 Hardware Configuration

The equipment included in the ADEPT system currently consists of an IBM System 360/50 computer with 262,000 bytes of core memory; three selector channels for transfer of data between the CPU and drum, disc, and tape storage; and a multiplexor channel for transfer of data to and from interactive terminals and other input/output equipment. A block diagram of the current hardware configuration, showing equipment model numbers, speeds, capacities, and device addresses (in hexadecimal), is included as an appendix to this document.

1.3 PROJECT STATUS

During the six-month period ending 30 July 1968, the ADEPT system progressed from a partially checked out set of separate components to a relatively stable operational system. Improved capabilities and increased reliability were reflected in four new releases of the time-sharing executive. The initial releases of TDMS Load, Maintain, Update, and Compose operations were made during this period, as well as improved versions of Query, Define, and Reformat. All major components of the programmer's package are working well, including improved versions of the JOVIAL (J5) compiler and the on-line programming and debugging tools. A brief overview of the status of the three major ADEPT components is presented in the following paragraphs.

Work on the time-sharing executive portion of ADEPT during this reporting period concentrated on increasing capabilities while at the same time improving reliable performance. The ADEPT executive now provides the following major capabilities: a complete file-cataloging subsystem, pervasive security controls, an integrated Batch Monitor subsystem, dynamic memory allocation, interactive symbolic debugging, support for a variety of interactive devices, and an expanded command library of over 40 interactive commands. System throughput and reliability were also improved through changes in a number of areas, including control over system fabrication and installation/configuration management, error diagnostics and recovery procedures, memory management, scheduling, and input/output procedures.

All of the program segments of TDMS now provide some operational capability. Many of the TDMS components, however, still have various limitations that are expected to be removed in the coming six-month period. A particularly difficult problem--that of building completely error-free data bases with Load--seemed to be resolved at the end of this reporting period. Some efforts were also expended during this period on accommodating the needs of initial TDMS users, including publication of a set of interim user's guides, testing and analysis of various TDMS data bases, and consultation with users on error correction and recovery. (Satisfactory progress was also made on the two additional tasks in the data management portion of ADEPT--the JOVIAL/TDMS interface and the DBL data base reformatter.)

Work on the programmer's package component of ADEPT is progressing well. A major task during this period was that of modifying the input/output sections of various programs to allow them to operate under new releases of the ADEPT executive, which provided significant new capabilities--particularly in the Cataloger and SPAM. Improved versions of the JOVIAL compiler and the editing and debugging aids were released and have been heavily used. Several modifications were also made to TINT, including the addition of a program-save-and-reload feature. A new utility program was added to the programmer's package, designed to satisfy the initial tape-related print, punch and card read requirements of ADEPT users. A machine-language assembler was adapted from IBM's F-level Assembler for operation under the current ADEPT executive; this program is being maintained at its current level. Considerable attention was also focused on providing a capability

for composing and editing programs on-line, using a small CRT display (operating within IPSS).

1.4 ORGANIZATION OF REPORT

The organization of this report generally parallels the logical structure of the ADEPT system; each of the three major system components is described in a separate section. Included in each section is a general description of the component, the progress made in implementing it, and plans for developing the component in the coming six months. Also included are the names of staff members assigned to the three major project areas, and a list of the documents produced in each area during the preceding six-month period.

2. TIME-SHARING EXECUTIVE

2.1 INTRODUCTION

The ADEPT executive is a general-purpose time-sharing system. The initial system operates on a 360 Model 50 with approximately 260,000 bytes of core memory, 4 million bytes of drum memory, and over 250 million bytes of disc memory, shown graphically in Figure 1. With this machine configuration, ADEPT is designed to provide responsive on-line, interactive service, as well as background service to approximately 10 concurrent user jobs. It handles a wide variety of different, independent application programs, and supports the use of large random-access data files. The design--basically a swapping system--provides for flexibility and expansion of system functions, and growth to more powerful models in the 360 family.

ADEPT functions both as a batch processor (whereby problems are accumulated and fed into the computer for answering one by one) and as an interactive, or on-line, system (in which the user submits his requests directly to the machine in real time simply by typing them on a console. The user first identifies himself through the ADEPT console commands, and specifies his programs and data files).

Viewed as a batch system, ADEPT allows jobs to be submitted to console operators in some standard manner, or submitted from consoles via remote batch commands. In either case, jobs are "stacked" for execution by ADEPT in a first-in/first-out order. The stack is serviced by ADEPT as a background task, subject to the priorities of the installation and the demands of "foreground" interactive users. Viewed as an interactive system, ADEPT allows the user to work with a reactive typewriter, allowing computer-user dialogue in real time. Via ADEPT console commands, the user identifies himself, his programs, and his data files, and selectively controls the sequence and extent of operation of his job in an ad lib manner. A prime advantage of the interactive use of ADEPT is that the system provides for a library of service programs, later extendable, that permit the user to edit data files, compile or assemble programs, debug and eliminate program errors, and generally manage large data bases in a responsive on-line manner.

The ADEPT executive consists of two major components, referred to as the Basic Executive (BASEX) and the Extended Executive (EXEX). BASEX handles the major problems of allocating and scheduling hardware resources, and drives the input/output gear. It also has first-phase control of all CPU interrupts.

EXEX provides the interface between the user community and BASEX. It contains all of the processes that, as the name implies, are extensions of the basic processes and are user-oriented rather than hardware-oriented. Many functions that the system performs are non-urgent and need not be immediately accessible; they are, therefore, schedulable. These functions are also included in EXEX. Thus BASEX is always resident and EXEX is swappable.

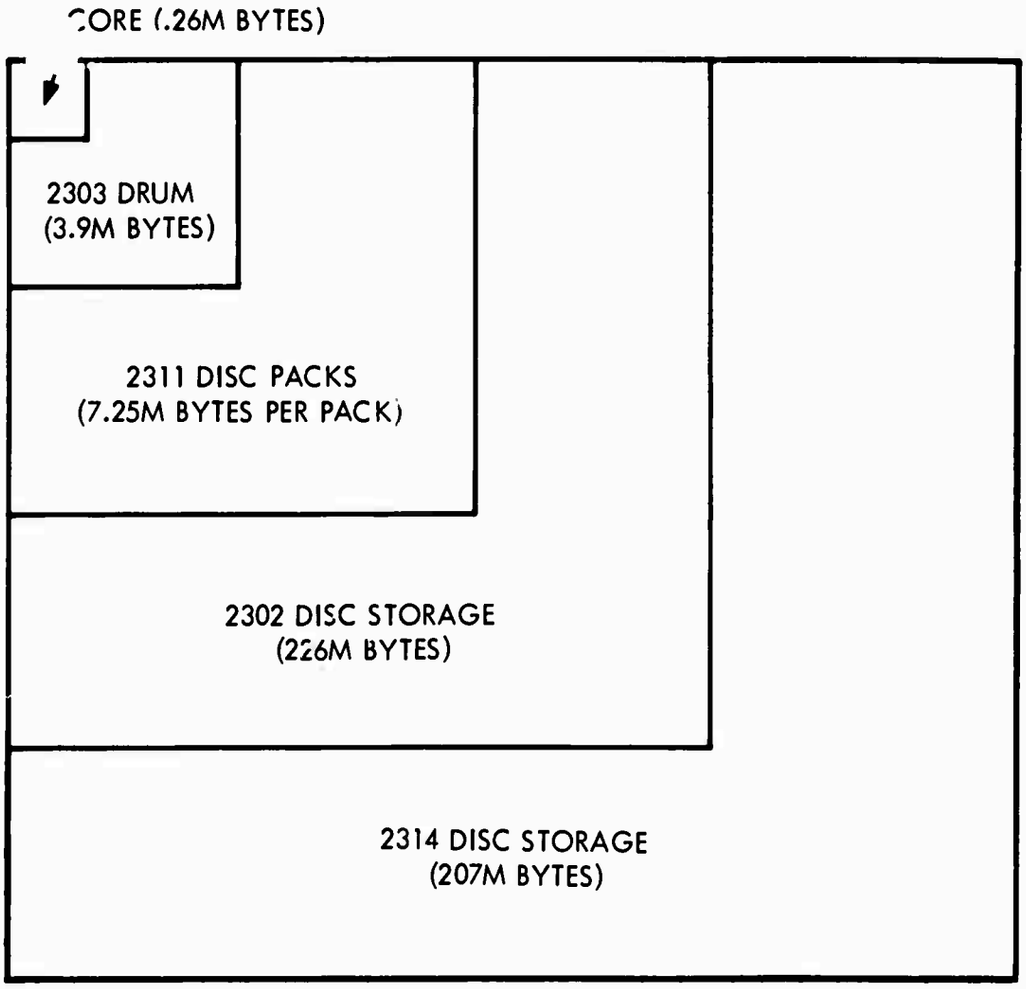


Figure 1. Relative Capacity of Various ADEPT Direct-Access Storage Media Available in Less Than 0.2 Seconds

The initial system that operates at SDC utilizes core, 2303 drum, 2311 disc packs, and 2302 disc storage. The NMCSSC system successfully utilizes 2314 disc storage in lieu of 2311 or 2302 discs. The architecture of the ADEPT executive is such that it permits any combination of the above types of disc storage in any amount.

The ADEPT executive is designed to operate in the lower quarter of memory, thereby providing three quarters of memory for user programs. With the current hardware configuration, ADEPT preempts the first 65,000 bytes of core memory, the bulk of which is dedicated to BASEX. EXEX then operates in user memory in a fashion similar to user programs. ADEPT is designed to operate itself and user programs as a collection of 4096-byte pages. BASEX is identified as certain pages that are fixed in main storage and that cannot be overlayed or swapped. EXEX and other programs are identified as sets of pages that move dynamically between main storage and swap storage (i.e., drum). It is necessary to maintain considerably more descriptive information about these swappable programs than about BASEX. This descriptive information is carried in a set of system tables that, at any point in time, describe the current state of the system and each program.

ADEPT views the user as a job consisting of some number of programs (up to three for the 360/50 configuration) that were loaded at the user's request. Implicitly EXEX is considered to be one of these programs. Only one program in the user's set may be active (eligible to run) at a time. When ADEPT scheduling determines that a job may be serviced, the current job in core is saved on swap storage, and the active program of the next job is brought into core from swap storage and executed for a maximum period of time, called a quantum. The process then repeats for other jobs. Figures 2 and 3 schematically depict these relationships.

2.1.1 Basic Executive (BASEX)

The Basic Executive is an integrated package of software components responsible for hardware resource management, including interrupt control, memory allocation and control, and scheduling of system and user program CPU access. Its principal components are:

- Basic Sequencer
- Scheduler
- Allocator
- Swapper
- Interrupt Handlers
- Input/Output Supervisor (IOS)
- SPAM
- Interactive Input/Output Handler (TWRI)
- Service Package
- Call Functions
- Central Tables

These components reside permanently in low core memory. They are never swapped as are the Extended Executive (EXEX) components and user programs.

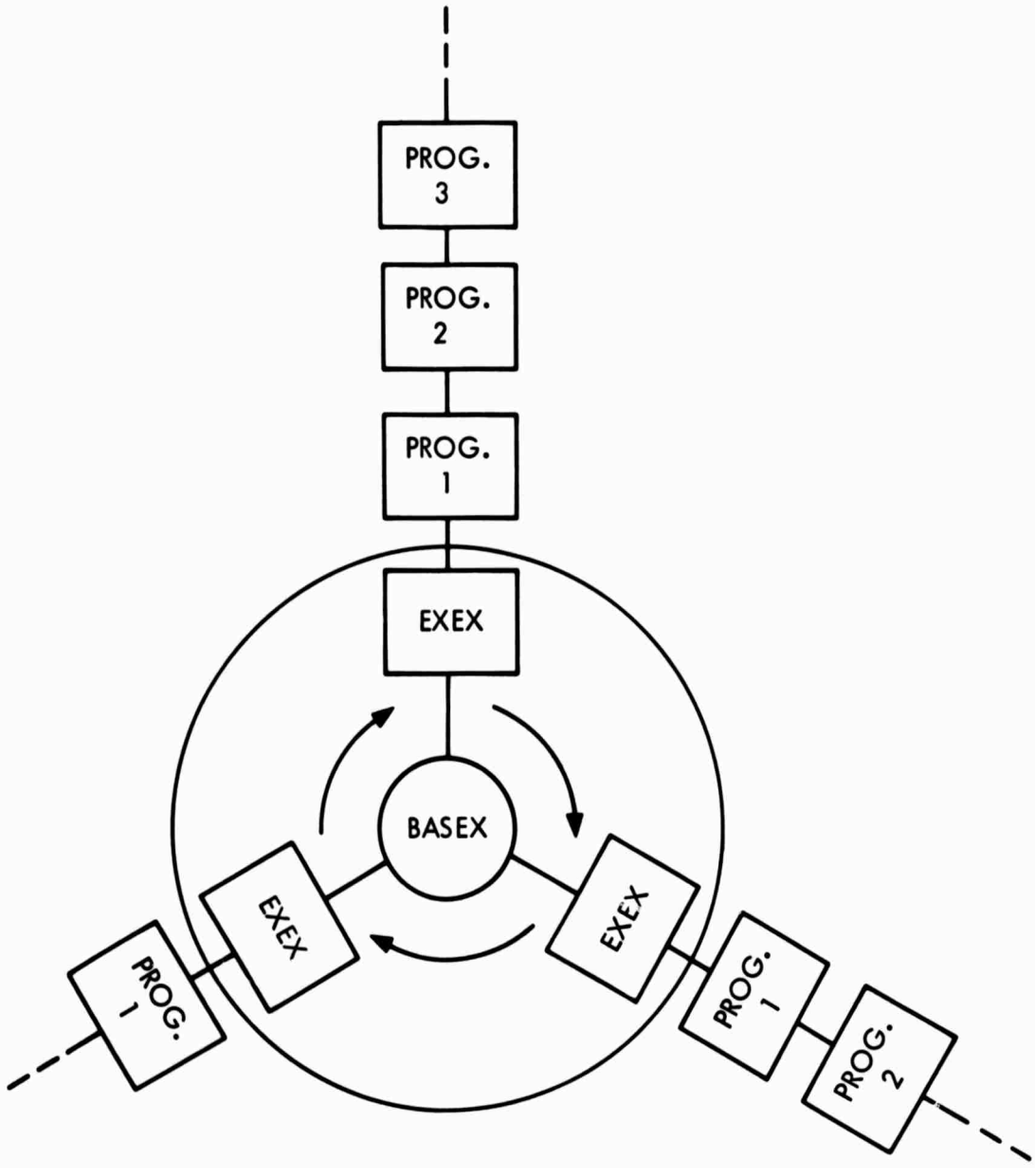


Figure 2. Simple Commutation of Users' Programs

This figure illustrates the relationship between users' programs, EXEX, and BASEX. Each spoke represents a user's programs, with his EXEX providing the interface with BASEX and the hardware resources. The number of users may, of course, be more than three.

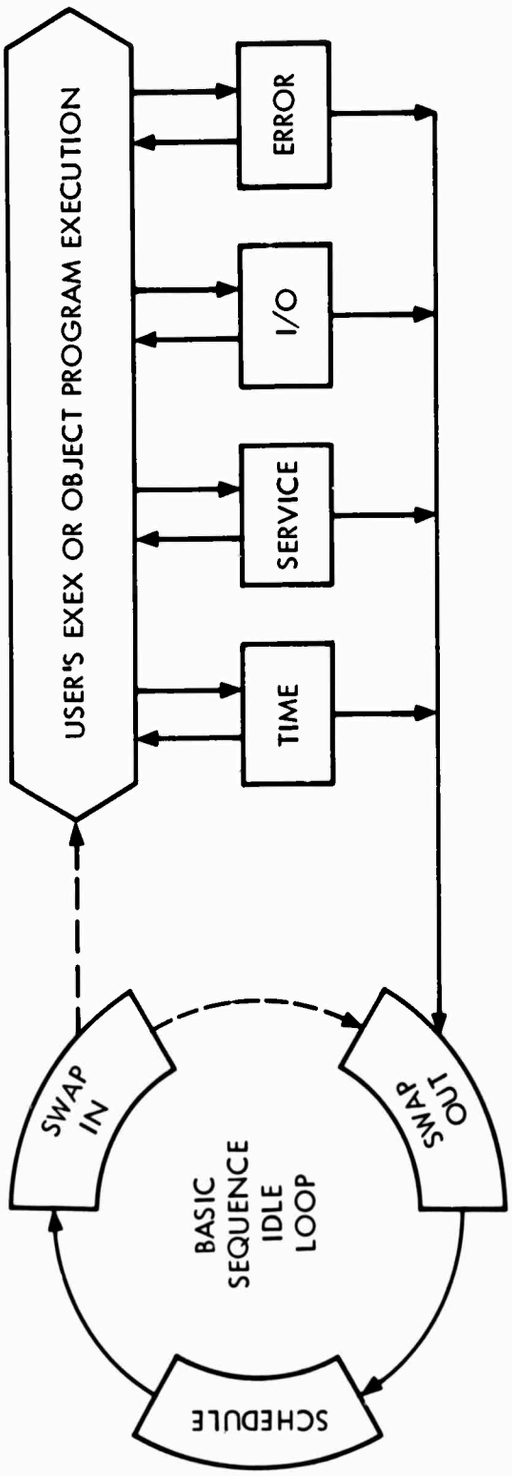


Figure 3. Basic ADEPT Sequence of Operation

This figure shows the basic operating system cycle: idle loop is interrupted by an external interrupt (an activity request); a program is scheduled, swapped into core from the drum, and executed; escape from the execution phase occurs when a quantum termination condition (e.g., time expiration, service or I/O call, error condition) is met; the program is then swapped out and control is returned to the idle loop (if no other programs are eligible to be scheduled).

2.1.2 Extended Executive (EXEX)

Unlike the tight, closed package of integrated BASEX components, EXEX is a loose, open-ended collection of semi-autonomous programs. EXEX is treated by BASEX as a user program, with certain privileges, and each user is given his own "copy" of the EXEX. It is transparent to the user that EXEX is reentrant and is being shared with other users, except for its data space (the job environment page is unique for each user). This structure permits flexible modification and tailoring of user-system interfaces in a localized, modular fashion.

EXEX consists of the following components:

- SVC Dispatcher
- Command Filter
- Command Library
- Cataloger
- SERVIS (formerly the Loader)
- Interface Routines
- Debugger
- Batch Monitor
- Networking Routines

The first two are resident parts of the ADEPT executive in low core, and may be viewed as additional BASEX components; the Command Library, however, is the bulk of EXEX--parts of which are selectively swapped into user core when needed. Irrespective of whether the EXEX components are resident or swapped, EXEX is always scheduled, like other user programs.

2.2 PROGRESS

Whereas during the previous reporting period our efforts were focused on system design and fabrication, this reporting period was characterized by considerable progress toward the three goals of executive capability, reliability and performance.

2.2.1 Capabilities

The capabilities of the ADEPT executive were increased in an evolutionary manner through a series of operational releases and experimental pre-releases as listed below:

Release	Date
1.0	30 October 1967
2.0	4 December 1967
2.5	15 December 1967
3X	15 January 1968
3.0	12 February 1968
4X	15 April 1968
4.0	20 May 1968
4.3	10 June 1968
5.0	15 July 1968
6.0	29 July 1968

The current release now possesses the following principal capabilities:

1. A complete file cataloging subsystem. The Cataloger manages 7- and 9-track tape drives, and 2302, 2311, and 2314 discs. Via the Cataloger, a user may maintain his files on private demountable storage volumes (i.e., tapes and disc packs) or on public on-line (POL) storage (i.e., disc). These files may be specified by the owner as Private (for his use only), or Public (for general use). Furthermore, access to Public files can be limited by the owner to read-only, write-only, or both read and write. All files are security-controlled by three concurrently operating locks: Classification, Need-to-Know, and Special Category access. These locks are set by the file's owner at the time of file creation; the keys to opening the locks are derived dynamically by the file subsystem when a file is accessed by the user community.

   In concert with the Cataloger, on-line SERVIS commands LISTF and DELETE permit users to (1) selectively display on their terminals an inventory of their files, all Public files, or all files on a given volume; or (2) purge their files from the file inventory.

   As the largest single component of the ADEPT executive (57,000 bytes), the Cataloger was written in a new, experimental programming language called MOL-360,* which satisfied the conflicting demands for higher-level source language and flexible machine code (i.e., code that provides access to non-standard hardware features). The Cataloger design and checkout was considerably enhanced by the use of MOL-360, while simultaneously showing the validity of MOL-compilers for difficult machine-dependent programming

*MOL-360 (Machine-Oriented Language for the 360) is a "higher-level machine language." It was developed under an ARPA-sponsored SDC research project on metacompilers. (See TM-687/010/00, "Information Processing Techniques Semi-annual Technical Summary Report to the Director, ARPA.")

2. Pervasive security controls. Integrated throughout the ADEPT executive are software controls for safeguarding security-sensitive information. The conceptual framework is based upon four "security objects": user, terminal, file, and job. Each of these security objects is formally identified: user by an up to 12-character name; terminal by its hardware address; file by its 8-character name, form (e.g., binary program, card images, etc.), owner-name, and volume number; and job by its internal job-table entry location. Each object is also described by a three-tuplet security property: Classification (e.g., TOP SECRET, SECRET, etc.), Need-to-Know, and Special Category (e.g., SIOP, CRYPTO, etc.). At system initialization time, user and terminal security properties are pre-stored by security officers via the system component SYSLOG. SYSLOG also permits the association of up to 64 passwords with each user. At LOGIN time, a user identifies himself (his unique, up to 12-character name) and enters his private password to validate his identity. The LOGIN component of ADEPT validates the user and dynamically derives the security three-tuplet for the user's job as a complex function of the user and terminal three-tuplets. The job's three-tuplet is subsequently used as "keys" when access is made to ADEPT files. The file's three-tuplet acts as the locks under control of the file subsystem.

   Security controls are also involved in the control of classified memory residue. Hardware and software memory protection is extensively used. Software memory protection is achieved by interpretive, legality checking of memory bounds for I/O buffer transfers, legality checking of device addresses for unauthorized hardware access, and other possible user program attempts to seduce the operating system into violating security controls.

3. An integrated Batch subsystem. The RUN command permits on-line users to enqueue jobs into the single ADEPT Batch job stream. Subject to various possible scheduling disciplines, the Batch Monitor component of ADEPT services these jobs in priority order in the ADEPT background. (Interactive terminal-controlled jobs are serviced in the foreground.) Remote job entry is thus permitted. Currently, the RUN command is limited to the "short form," where only specific production programs (compiler and assembler) can be called. The RUN command will be expanded in two directions: (1) permitting other production programs, and (2) servicing the "long form," where a file of commands will be accepted by the Batch Monitor. These areas are further explained below under "Plans" (see Section 2.3.1).

4. Dynamic memory allocation. As a resource-sharing system, ADEPT provides a wide variety of dynamic memory management capabilities for user and system use. Memory management includes both core and drum memory, allocated in pages of 4096 contiguous bytes. The 2303 drum has an 800-page capacity, whereas the "H" size core memory is viewed as 64 pages. The Allocator component of ADEPT manages a page map for each program. The map reflects the correspondence between drum and core pages, established initially by the ADEPT SERVIS component at LOAD time. User programs may manage their own page maps via a number of Allocator calls. To acquire more drum and core pages than initially loaded, the GETPAGE call may be invoked; to release pages, FREEPAGE call; to permit page sharing, SHARE call; to modify (activate or de-activate) the swappable set of program pages, ACTDEACT call; and to copy the page map, the ESTACOP call. The user program may also load additional pages from disc or tape via the SERVIS calls APPEND and OVERLAY. Via these calls, skilled users can achieve efficient use of time and memory. Most EXEX components use these calls for just such purposes.

   The single most important performance aspect of ADEPT is in the management of the drum memory by the Swapper component. By marking only those pages changed (i.e., written onto) during a program's time-slice, considerable reduction in swap time has been attained. This technique for efficiently managing memory is further described below (see Section 2.2.3).

5. Interactive symbolic debugging. ADEPT JOVIAL programmers can now debug their programs on-line using symbolic item names and statement labels as address parameters for the Display, Set, Breakpoint, and Goto DBUG commands. This has been achieved by allowing a compiler-produced dictionary to be loaded with the program as the result of the SERVIS component LOADD command. DBUG has been modified to use this dictionary for all symbolic address references.

6. Interactive devices. The prototype system at SDC has been expanded to permit the use of a variety of interactive terminals. It is now possible to support Model 33/35 Teletypes, both local and remote (hardwired or dialup), IBM 2741 and 1052 typewriters, IBM 2260 and CCI cathode-ray tube devices. With all these devices, users may correct typographical errors by line- or character-level cancellation. In addition to these devices, ADEPT supports the large IBM 2250 graphic display and the IBM 1053 hardcopy printer for the 2260 CRT.

7. Expanded command library. Over 40 interactive commands are now available to users. Although the four commands of LOGIN, LOAD, GO, and LOGOUT are all that is necessary to use ADEPT, the expanded vocabulary provides more knowledgeable users greater power and flexibility in their use of the system. In addition to these console commands, a variety of service calls (SVC's) are available to programs. Table I summarizes both console and program calls.

2.2.2 Reliability

After achieving executive capabilities beyond those minimums needed for a meaningful user environment (i.e., Release 3), emphasis was shifted toward attaining higher system reliability. This goal was translated into the following activities:

1. System fabrication. Better control was gained over the production of new releases, thus minimizing errors of omission and commission. The Load and Initialization Package (LIP) continued to be the principal tool for system fabrication. Work on LIP during this period focused on cleaning up old features to make them work better or simpler, and adding new capabilities. In the former area, the structure of the master system deck has been simplified by the elimination of extra, unnecessary control cards. In the latter area, LIP now clears core (a confusing set of operator actions was previously required); it also clears the protection keys prior to system loading. LIP permits the dumping of disc and core memory, and the listing of all system components.

   Additional tools have been generated as shakedown testers for various system components, including the Cataloger, SPAM, IOS, and the interactive terminal package (TWRI).

2. Installation and configuration control. Some work has gone into greater parametric control of system decks for producing systems for different hardware configurations (e.g., NMCSSC has no 2302, or 2741 terminals). More work is needed in this area. The system will accept 2302, 2311, and 2314 discs, 7- and 9-track tapes, and a wide variety of terminals. The quantity and addresses of all devices are now controlled via run-time patches to configuration tables.

   Considerable effort was expended to ease the operator's task in initiating the system from a cold start. It now takes about 3 minutes. Communication with the operator via the 1052 typewriter has been simplified. He has a number of options, which include varying devices on/off-line, initiating hardware status checks, clearing and testing the drums, initiating the security SYSLOG procedures, and setting the date and time of day.

Table I. Summary of ADEPT Executive Functions

Function	Type of Component	Called	SVC Number* Dec.	Hex.	Description
ACTDEACT	B	I	17	11	Activate or deactivate program page(s).
APPEND	X,C	E,I	35	23	Load a new program into the next available core page immediately following the initially loaded program.
B (break)	X,C	E	39/40	27/28	(A debugging command.) Insert or remove a breakpoint in user's program.
CANCEL	X,C	E	48	30	Cancel a batch job that has been previously requested by the RUN command.
CATALOG	X	I	37/38/ 42	25/26/ 2A	Call the EXEX Cataloger to open, close, change, or delete tape and disc files.
CLOSE	X	I	38	26	(One of the CATALOG functions.) Close a file.
CYLS	X,C	E	--	--	List the number of available cylinders.
D (dump)	X,C	E,I	39	27	(A debugging command.) Display contents of specified portions of user's program.
DBUG	X,C	E	39	27	Request use of the EXEX debugging commands.
DELETE	X,C	E	--	--	Delete a file. (Same function as below, except this one is externally called.)
DELETE	X	I	38	26	(One of the CATALOG functions.) Delete a file. (Internally called via an SVC call.)
DEVICE		I	20	14	Provide the current time, date, computer model, size of core, software release version, whether or not device is enabled, and the on-off status of the various devices attached to the system.
DIAL	X,C	E	--	--	Send a message from one console to another.
DIALOFF	X,C	E,I	55	37	Do not allow messages to be received from other consoles. Dial messages are normally allowed.
DIALON	X,C	E,I	56	38	Allow messages to be received from other consoles. Clear the "not accepting message" indicator which is set by the DIALOFF command.
DISC	X	E	--	--	Provide information on available disc space.
DISMISS	B	I	01	01	End the quantum. Immediately reschedule the program for the next quantum.
DRIVES	X,C	E	--	--	Provide information on available tape and disc drives.
DRUMS	X,C	E	--	--	Provide information on available drum space.
ESTATCOP (copy)	B	I	18	12	(One of the Allocator functions.) Copy the ESTAT table.
EXCP	B	I	10	0A	Execute a channel program for non-cataloger controlled devices, e.g., 2250, card equipment.

B = BASEX, X = EXEX, C = Command Library, E = externally called via console command, I = internally called via SVC.

*Functions in the Command Library that are permanently resident BASEX or EXEX components are not assigned SVC numbers.

Table I. Summary of ADEPT Executive Functions (Cont'd)

Function	Type of Component	Called	SVC Number* Dec.	Hex.	Description
FORGET	X	E	--	--	Cancel a request to the Cataloger for a tape or disc volume. This command causes a no-wait read on the terminal.
FREEPAGE	B	I	15	0F	(One of the Allocator functions.) Release page(s).
GETPAGE	B	I	14	0E	(One of the Allocator functions.) Assign new page(s).
G (go to)	X,C	E,I	39	27	(A debugging command.) Specify a point in user's program at which to begin or resume execution.
GO	X,C	E,I	45	2D	Initiate program execution. Stop executing (this) calling program and go to execute the named program.
LISTF	X,C	E	--	--	Obtain a list of the specified files.
LOAD	X,C	E,I	34	22	Load a program at location 10,000 (hexadecimal) and following.
LOADD	X,C	E,I	34	22	Load a program and a dictionary from the same unit.
LOGIN	X,C	E,I	47	2F	Log into the time-sharing system. Internally executable by the Batch Monitor and item EXEX components.
LOGOUT	X,C	E,I	53	35	Exit from the system. Omit all active programs in job. Total quit.
OPEN	X	I	38	26	(One of the CATALOG functions.) Open a file.
OVERLAY	X,C	E,I	36	24	Load a new program at the lowest core page assigned to an initially loaded program.
PRINT	B	I	08	08	Print a line of text in the user's IBM 2741, Model 33 or 35 Teletype, or IBM 2260.
QUIT	X,C	E,I	49	31	Remove a program from the job (and system).
READ	B	I	09	09	Read a line of text from the user's IBM 2741, Model 33 or 35 Teletype, or IBM 2260.
RESTORE	X,C	E,I	44	2C	Restore a program that wss previously saved in non-relocatable binary form.
RESTORED	X,C	E,I	44	2C	Restore a program that was previously saved and its associated dictionary, if any.
RETURN	X	I	46	2E	Return to the original calling program.
RUN	X,C	E	48	39	Initiate remote-batch operations specified in the file name.
S (set)	X,C	E	39	27	(A debugging command.) Alter the contents of specified portions of user's program.
SAVE	X,C	E,I	43	2B	Save an initially loaded program on a specified device in non-relocatable binary form.
SHARE	B	I	16	10	(One of the Allocator functions.) Share pages of (this) calling progrsm by assigning some or all of its pages to other programs.

B = BASEX, X = EXEX, C = Command Library, E = externally called via console command, I = internally called via SVC.

*Functions in the Command Library that are permanently resident BASEX or EXEX components are not assigned SVC numbers.

Table I. Summary of ADEPT Executive Functions (Cont'd)

Function	Type of Component	Called	SVC Number* Dec.	Hex.	Description
SPAM	B	I	11	0B	Call SPAM to read, write, modify, delete, position, or search for records on a tape or disc file.
STATUS	X,C	E	--	--	Provide information on the status of named program.
STOP	B,C	E	--	--	If there is no *program:name* parameter, stop current active program; otherwise, stop named program.
STOP	B	I	00	00	Stop current active program.
TIME	B,C	E,I	19	13	Provide the current time and date and the accumulated total CPU time used.
USERS	X,C	E	--	--	Provide information on the current number of ADEPT users.
WAIT	B	I	02	02	Wait for time or I/O completion.
The following functions can be issued from the operator's terminal only:					
BQUIT	X,C	E	--	--	Quit the present batch job, but continue processing.
BGO	X,C	E	--	--	Start the Batch Monitor.
BSTOP	X,C	E	--	--	Stop the Batch Monitor according to the conditions specified by the command *program:parameters*.
RESTART	X,C	E	--	--	Cause a "break key" operation for a specified terminal.
SJREAD	B	I	29	1D	Read system subjob.

B = BASEX, X = EXEX, C = Command Library, E = externally called via console command, I = internally called via SVC.

*Functions in the Command Library that are permanently resident BASEX or EXEX components are not assigned SVC numbers.

3. Error diagnostics and recovery. The decision to incorporate a primitive debugging capability (BXBUG) into the resident (BASEX) executive has paid handsome dividends. New releases can be intimately probed on-line for cause and effect, whenever system errors arise; often the trouble can be fixed immediately, without requiring a system restart. BXBUG is our first line of defense for diagnosis and repair of software errors. Our second line involves improved error messages.

   Hardware errors are now less catastrophic than in the last reporting period. The principal reason has been the incorporation of retry capability into the input/output components of the ADEPT executive. Additional software to better sense and handle recoverable errors (including both hardware and user abuses) also contributes to the improvement. Now, rather than causing a system crash, only the user-job affected is aborted, while other users continue unperturbed and often unaware of the difficulties. Finally, better pre-conditioning of the system at initialization time often avoids trouble later on with marginal components. This is particularly true for marginal drum tracks which are communicated to the Allocator so it can avoid their allocation.

   With these techniques, the stated minimum of two hours mean-time-to-failure (MTF) of software has been achieved with the initial NMCSSC ADEPT executive (Release 4.3). This first installation of the ADEPT system (at the Support Center) took place the end of May, with formal operation the first week of June as scheduled. Current Support Center capabilities are those of Release 6.0, and this site will continue to stay abreast of new, reliable executive releases. The second ADEPT installation, at the Air Force Command Post, is currently in progress, with formal operation expected by early August.

   Other "tests by fire" for ADEPT were achieved at three successful ADEPT-50 Symposia, at which live system demonstrations were given to hundreds of interested military and civilian personnel. The symposia were held at SDC, Santa Monica on April 24 and 25, and at Andrews Air Force Base, Maryland, on July 10 and 11.

2.2.3 Performance

Improving the system's performance is now our principal objective. Efforts are being directed toward balancing the equation of fast response time and high throughput via improved scheduling. However, improving performance also requires that we pay attention to details of user facilities--smoothness of man-machine dialog. Finally, performance improvement implies continual reflection on internal system design in an ongoing effort to lower system overhead. Measurement and recording are the latest tools employed in these efforts.

In contrast to Release 2, Release 6.0 provides an approximate five-fold improvement in throughput at a nominal cost in response time. The specific system changes responsible for this improved performance include:

1. Dynamic page marking. Whenever a user program is swapped into core, its pages are set in a read-only condition. As the program executes, it periodically attempts to store data (write) in its write-protected pages. The resulting interrupt is fielded by the system. After satisfying itself that the store is legal for the program, the executive marks the target page as "written," turns off the write protect for that page, and resumes the program's execution. The situation repeats for each additional page written. At the completion of the program's time slice, the Swapper has a complete map of all the program's pages that were changed. Only the changed pages are swapped out of core. Preliminary measurement of this scheme shows that about 20 percent of the pages are changed, and hence for every five pages swapped in, only one need be swapped out, for a total swap of six pages, rather than the full swap of ten pages (five in, five out). The scheme thus makes the drum appear to be 40 percent faster.

2. Improved scheduling. The current time slice is now a full two seconds. This raises response time to about five seconds, but it improves system throughput by lowering the swap frequency (and hence swap overhead). In addition, programs are now rescheduled after an I/O request, if there is sufficient time remaining in the time-slice. This technique has improved 2260 service dramatically. The scheduling algorithm is still round-robin for all jobs (including the Batch Monitor), though this will soon be changed to a multi-queue discipline.

3. Buffered output. Interactive terminal output is now buffered by the system, thereby permitting some overlapped I/O and program execution.

4. Priority SVC's. Priority processing of selected Allocator calls has yielded a four-fold improvement in the speed of the LOAD command, which makes heavy use of these calls. It now takes about 30 seconds to load a large program (30 pages) with heavy system usage.

5. SPAM improvements. Recent improvements have lowered the overhead for "position-type" I/O calls. With these improvements, SPAM (the ADEPT Sequential Partitioned Access Method) can calculate and position disc heads directly to the proper file record. Previously, a sequential search through all previous records on a track was required.

2.3 PLANS

With the installation of ADEPT at a number of military facilities, we expect the next review period to be concerned with the general task of modifying the system in response to live user feedback. These modifications should continue to increase the capabilities, reliability and performance of the ADEPT executive.

2.3.1 Capabilities

An impressive list of new features--those suggested by our current user population, and those scheduled previously for later work--are awaiting their turn for implementation. These new features, noted below, will be handled in the same evolutionary manner as before in new system releases. It is anticipated, however, that the frequency of new releases will decrease.

1. The cataloging subsystem. The major features to be added in this area involve the handling of semi-private files, the CHANGE call, and the handling of multiple volume files. Semi-private files are those with a formal Need-to-Know list of user names who may access a file. The CHANGE command will allow the owner of a file to change all the inventory control information stored with the file, such as its name, form, owner, security three-tuplet, etc. Currently, files cannot exceed the capacity of the volume upon which they are stored. This limitation will be relaxed with multiple-volume files. This new capability also affects the SPAM component of ADEPT, which also must be modified.

2. Tighter security control. Semi-private files will expand the Need-to-Know dimension of the file security object. Attention must be given to the control aspects of this change. Also, new features will be required to permit user fabrication of Need-to-Know lists. During the next reporting period, various security audit trails will be incorporated into the system. As a minimum, these audit trails will include recording of all user LOGIN/LOGOUT, file OPEN/CLOSE, program LOAD/QUIT, and CHANGE call usage. Further, experimental work will be started soon into the development of an internal SPY program and a Security Operations Station (SOS). The SPY program will act as a simulated security guard--trying all the "locks and doors"--as it makes its periodic check of the system security integrity. The SOS will be an on-line, real-time, human-controlled security surveillance station. From the SOS, a security officer will be able to monitor and interact with the security controls of the system.

3. Expanded batch operations. The RUN command will be expanded to permit any production program to be executed in the batch mode via the short form. Such production programs will be defined as those that have adopted the conventions of the Batch Monitor

defaults for input/output file types and formats. The long-form RUN command will accept a disc file name as its basic parameter. This file will contain a prescripted list of normal system commands, e.g., LOAD, GO, etc. The Batch Monitor will be modified to read and execute each command in the command file. In this manner, the long form of the RUN command will act like a system macro. Considerably greater batch-processing capabilities should result from this feature.

4. Reentrant housefiles. Public files that are always guaranteed to be on public storage are called "housefiles." In a machine with no dynamic relocation hardware (such as the 360/50), reentrant programs--which share common read-only code--are useful principally to save drum (swap) storage, and then only if they are frequently used programs, e.g., housefiles. In ADEPT, most application programs will eventually be reentrant housefiles. The SERVIS component of the ADEPT executive is being modified to service reentrant housefiles.

5. User operating subsystems. If it were possible for a user program to arm and field its own interrupts (and possibly those of other programs in its job), and if the program were also able to capture the terminal input or output of a program in its job, then it would be possible for users to develop their own operating systems that run under ADEPT. Such "user operating subsystems" could be used to provide higher-level control programs for language processors or application programs. They could be used to achieve software emulation and hence transferability of application software between different operating systems. Networking control programs is but one other example.

   In the coming reporting period, interrupt arming and terminal capture control features will be implemented as prerequisites for a "user operating subsystem" capability. The Batch Monitor will then be redesigned internally (invisible to the user community) to act as an operating subsystem. Another operating subsystem will be a network protocol package to allow the coupling of ADEPT systems into an ADEPT network.

6. Interactive devices. It is expected that new terminal types will be tied into the ADEPT system. Already on order is an Inktronics printer/keyboard and a Model 37 Teletype. Also, discussions are continuing on other devices such as the Bunker-Ramo BR-90 CRT display. Of considerable impact is the delivery of a peripheral processor to replace the IBM 2702 controller. This processor must be programmed to service all terminal devices while concurrently managing graphic terminals and the RAND Tablet. It is anticipated that this effort will not begin until late in

the next reporting period, due to the long lead time for processor delivery. Selection of the particular processor has not been finalized due to the wide variety of new equipment on the market that needs to be examined.

7. More commands. The open-ended set of commands will continue to be expanded as the needs arise and requests can be fulfilled. The following commands are expected shortly: SEARCH (locate a particular file), CREATE (create a Need-to-Know list), VARY (switch a peripheral device on/off-line), CPUTIME (print on-line CPU time used), LISTU (print on-line the names of all users logged in), EXPLAIN (give a brief explanation of system commands), WRAPUP (log out all jobs, record accounting/audit trail information).

2.3.2 Reliability

With system installation in the field, greater numbers of error reports are anticipated. Thus, the next reporting period will concentrate on rapid error recovery and good system maintenance. Further efforts on parameterization of system fabrication will be made, in particular, by greater use of assembly macros and table-driven code. We know we can further improve error recovery. For example, by switching to alternate drum tracks on drum write errors, we can significantly increase MTF.

2.3.3 Performance

A variety of changes aimed at improving system performance are now being designed. These improvements will be checked by full-scale system recording and measurement.

1. Improved Scheduler. A multi-queue scheduling algorithm is currently under going checkout. The Scheduler sorts programs into three classes: interactive, compute-bound interactive, and batch. Batch programs get all the time not usurped by the other classes. Interactive programs get top priority, in round-robin fashion. Compute-bound interactive programs in reality follow their own multi-queue scheduling discipline: the more computation time requested, the larger the time-slice but at longer intervals. It is expected that the new algorithm will further improve both response time (for interactive class users) and throughput, by lowering swap overhead.

2. Overlapped swapping and execution. By initiating execution before a program is completely swapped into core, we expect to further lower swap overhead. The scheme appears sound, though only preliminary design has started.

3. System tuning. With selected recording and measurement of component execute times, attention will be directed at improving and tuning existing system routines to lower system overhead. Particular attention is being focused upon I/O transfers.

It is expected that these changes will maintain or even improve current performance levels as system overhead increases to accommodate new security and operational reliability or control features.

2.4 STAFF

C. Weissman, Staff Head
C. E. Fox, Technical Assistant
S. M. Aranda, Project Leader
R. R. Linde, Project Leader
R. E. Martin, Project Leader (on-site)
P. S. Baker
Martha Bleier
Frances Farrand
B. D. Gold
Patricia Kribs
S. G. Swerdlow
A. Tschekaloff

2.5 DOCUMENTATION

Many of the documents describing the ADEPT executive were released in SDC's "Note" series. These documents are internal working papers only, and have not been cleared for open publication.

Kennedy, P. R. Preliminary design specifications for the ADEPT-50 time-sharing executive and ADEPT-50 utility programs: Introduction to the series and table of contents. SDC document N-23741/000/05. 25 July 1968. 3 pp.

Introduces a document series that specifies preliminary design specifications for ADEPT; includes volume numbers, titles, and current issue information for preliminary design documents.

Aranda, S. M. Preliminary design specifications for the advanced development prototype (ADP) time-sharing system: Proposal for extended executive component SERVIS. SDC document N-23741/023/00. 12 January 1968. 7 pp.

Describes an initial set of on-line commands to be processed by a new EXEX component called SERVIS. These commands will allow users to determine (on-line) what files they have on the 2302 disc and the 2311 or 2314 disc packs, and to delete the files when necessary.

Kennedy, P. R. Preliminary design specifications for the ADEPT central tables. SDC document N-23741/031/00. 1 February 1968. 40 pp.

Describes the ADEPT central tables, including the core-resident, job environment page, data page, and the EXEX communication tables.

Weissman, C. Initial ADEPT security controls. SDC document N-(L)-23741/125/00. 20 May 1968. 25 pp.

Describes ADEPT-50 time-sharing executive security controls. Overall techniques are noted; however, those features available with the initial system release are emphasized.

Kennedy, P. R. User's guide for the ADEPT-50 time-sharing system, the programmer's package, and miscellaneous utility programs--Introduction to the series and table of contents. SDC document N-23759/000/06. 24 June 1968. 3 pp.

Introduces a document series that provides information on the use of ADEPT system components; includes volume numbers, titles, and current issue information.

Martin, R. E. ADEPT interim interactive device package user's guide. SDC document N-(L)-23759/001/01. 25 June 1968. 10 pp.

Describes the ADEPT-50 time-sharing executive, the programmer's package, and other programs that run under the executive. This guide describes how to use the interactive device package, a basic executive component. The package supports remote and local IBM 2741 terminals, remote and local Model 33 and 35 teletypes, and local IBM 2260 display scopes.

Aranda, S. M. and Baker, P. S. ADEPT cataloger user's guide. SDC document N-23759/002/02. 11 April 1968. 51 pp.

Describes procedures to be followed in using the ADEPT Cataloger; includes a discussion of Cataloger call procedures, request table formats, and special terms used in connection with the Cataloger.

Tschekaloff, A. ADEPT SPAM user's guide. SDC document N-23759/004/02. 15 May 1968. 27 pp.

Provides instruction for ADEPT users on how to use SPAM for reading, writing, altering, positioning, and searching for records in disc- and tape-based file structures. Includes a discussion of procedures for making SPAM calls, and request table formats.

Bleier, M. B. ADEPT loader user's guide. SDC document N-23759/021/03. 29 May 1968. 13 pp.

Provides instruction for ADEPT users on how to use the Loader for loading programs into the system. Includes a discussion of Loader calls, registers that must be set, input and output parameters, and commands recognized by Loader.

Brewer, R. A. ADEPT debugger user's guide. SDC document N-23759/022/00. 25 April 1968. 14 pp.

Explains how to take interactive debugging actions at a user console while operating a program in the ADEPT system. Presents instructions for debugging programs written in a language other than JOVIAL, and special instructions for debugging programs written in JOVIAL.

Bleier, M. B. ADEPT LOGIN user's guide. SDC document N-23759/023/00. 2 May 1968. 6 pp.

Instructs the prospective ADEPT Time-Sharing System user in how to use the LOGIN command.

Bleier, M. B. User's guide for DIAL (an ADEPT console command). SDC document N-23759/050/00. 7 March 1968. 6 pp.

Instructs the prospective ADEPT user on how to use the DIAL console command.

Kennedy, P. R. ADEPT executive console commands--a user's guide. SDC document N-23759/060/00. 23 May 1968. 90 pp.

Describes the ADEPT time-sharing executive, the programmer's package, and other programs that run under the executive. It describes how to use the console commands to communicate with the ADEPT executive. Examples of user/system dialogue are also included.

Martin, H. G. ADEPT time-sharing system file forms. SDC document N-23759/062/00. 6 June 1968. 5 pp.

Describes the seven file forms that are presently defined or reserved.

Foote, E. B. ADEPT protection, dynamic page marking, and program interrupt code. SDC document N-23759/065/00. 20 June 1968. 3 pp.

Describes the protection used on the IBM 360/50 and the fringe benefit of marking changed pages, which decreases swap time by 38 percent.

Aranda, S. M. User's guide for DEBE (ADEPT card file handling utility program). SDC document N-23759/300/01. 23 April 1968. 4 pp.

Describes the use of DEBE, an ADEPT card file handling utility program. DEBE permits the user to copy card files from the card reader, tape, or disc onto the printer, card punch, tape, or disc.

Kameny, I. M. FILEDUMP user's guide. SDC document N-23759/310/00. 4 April 1968. 13 pp.

Describes methods of using FILEDUMP, a program which allows the user to list (on-line or off-line) in EBCDIC or HEX records within files stored on 7- or 9-track tape or disc.

Bernzott, P. E. Operator utility function user's guide. SDC document N-23759/400/00. 4 March 1968. 16 pp.

Describes the initial release of the ADEPT operator utility function. The function is a self-contained program which may only be loaded by the system operator. It is designed to satisfy the initial print, punch, and card read requirements of ADEPT users.

Aranda, S. M. List of available housefiles on the SDC Santa Monica ADEPT-50 TSS. SDC document N-23759/601/00. 14 May 1968. 4 pp.

Lists the housefiles which are available to all ADEPT-50 users at SDC, Santa Monica.

Gold, B. D. and Kribs, P. LIP (loading-and-initiating) procedures. SDC document N-23759/701/00. 4 June 1968. 14 pp.

Describes how to load and initiate the ADEPT time-sharing system. Addressed to computer operators and ADEPT executive system programmers.

Bleier, M. B. ADEPT SYSLOG user's guide. SDC document N-(L)-23759/702/00. 7 May 1968. 11 pp.

Instructs the prospective ADEPT time-sharing system computer operator or security officer in how to use SYSLOG, a routine that sets up security classification and category data for each terminal that is to operate in the system, and the tables that are used by the LOGIN routine. The input deck and related tables are described.

Kennedy, P. R. ADEPT-50 time-sharing system bulletin: Introduction to the series and table of contents. SDC document N-23853/000/01. 24 June 1968. 1 pp.

Base volume of the N-23853 series. The bulletins in this series announce pertinent information concerning the ADEPT-50 time-sharing executive. The ADEPT-50 system is being implemented on the IBM 360, Model 50 computer at SDC in Santa Monica. This volume contains a table of contents for the entire series.

Martin, H. G. ADEPT-50 time-sharing system bulletin: Number 1--file copy and other utility programs. SDC document N-23853/001/00. 24 May 1968. 2 pp.

This bulletin contains a description of a file copy program that summarizes the utility programs available.

Bleier, M. B. ADEPT-50 time-sharing system bulletin: Number 2--new system commands. SDC document N-23853/002/00. 17 June 1968. 2 pp.

This bulletin contains a description of four new system commands available in Release 4.0.

Aranda, S. M. ADEPT-50 time-sharing system bulletin: Number 3--standardized file identification. SDC document N-23853/003/00. 20 June 1968. 7 pp.

Describes procedures for standardized file identification to be used by all programmers who write programs to operate under ADEPT.

Kennedy, P. R. ADEPT-50 time-sharing bulletin: Number 4--SERVIS (formerly the loader). SDC document N-23853/004/00. 26 June 1968. 1 pp.

Presents the following information about the use of SERVIS: a warning about saving a program in an existing file, setting the program name in general registers 5 and 6, and program control.

Kennedy, P. R. ADEPT-50 time-sharing system bulletin: Number 5--temporary restriction in format of the SAVE command. SDC document N-23853/005/00. 9 July 1968. 2 pp.

This bulletin describes a temporary restriction in the format of the SAVE command.

Kennedy, P. R. User's guide for the ADEPT executive time-sharing system. SDC document TM-3881/000/01. 11 June 1968. 350 pp.

Provides the prospective ADEPT-50 programmer-user with a general description of the ADEPT-50 executive time-sharing system. Includes an overview of the system, a complete description of external and internal user/system interfaces, terminal operating instructions, user/system dialog and programming examples, miscellaneous reference tables, a glossary of system terminology, an index, and an annotated bibliography.

Kennedy, P. R. The ADEPT-50 system: a general description of the time-sharing executive and the programmer's package. SDC document TM-3899/100/01. 7 June 1968. 60 pp.

Presents a general description of two major elements of the ADEPT-50 system--the time-sharing executive and the programmer's package. It also includes a discussion of the user/executive interface, a descriptive summary of executive functions, and an annotated bibliography.

3. DATA MANAGEMENT SYSTEM

3.1 INTRODUCTION

The data management component of the ADEPT system consists principally of an adaptation of the Time-Shared Data Management System (TDMS), combined with some additional tools in the areas of a data base oriented programming language, and the specialization of META5 to symbolic file conversions for TDMS.

During this reporting period, TDMS developed from a small set of partially checked out programs into a substantially operational system. The first two versions of TDMS were installed at the National Military Command System Support Center (NMCSSC) in the Pentagon; the system is now undergoing extensive testing and evaluation by a military user community. In spite of several limitations, indications are that the system is being exercised successfully.

TDMS consists of an integrated set of operations, each one of which is designed to do some part of the total data management task facing a user. The operations being delivered with the ADEPT system are Define, Load, Query, Compose, Update, Maintain, and Reformat. In addition, some utility programs useful in testing and analysis of TDMS data bases have been created. Each of the TDMS operations has been coded and subjected to extensive checkout on the IBM 360/50 at Santa Monica to the point where each of them has reached some operational capability. During the late spring and early summer, a set of interim user's guides were published, covering all major TDMS operations. On the 1st of June, our first delivery was made to the Pentagon for installation in the NMCSSC. This was followed in July with a second delivery, correcting errors and adding additional capabilities.

The system was first successfully demonstrated in March from RADC at Rome; it was subsequently demonstrated at two ADEPT-50 Symposia in Santa Monica and Andrews Air Force Base.

3.2 PROGRESS

The sections that follow discuss the major TDMS operations, the progress made in implementing them, current capabilities and limitations of the system, and progress made in implementing the other (non-TDMS) data management tools.

3.2.1 Define

Define is the TDMS operation through which the user names and describes the structural relationships of his data. It is the first operation the user must perform when he actually starts to use TDMS.

Define has been operational since early spring. One of the capabilities provided by Define is the specification of legalities for input formats and values. During the Load operation, the system uses these specifications to error-check incoming data. At the end of this reporting period, these optional legality specifications were the only capabilities in Define with known errors; they are expected to be corrected soon.

3.2.2 Load

Load is designed to accept all types of data, either in batch or interactive mode, perform the necessary validity checks, and allow on-line error correction, if desired. The output of the Load operation is a TDMS data base, a self-defined file with self-defined entries and a completely inverted cross-index.

The current version of Load performs only a limited number of its planned functions. It will build a satisfactory TDMS data base from batched data only. It does not yet perform many of the necessary validity checks nor output satisfactory error messages. Missing features also include interactive input, error correction, add-on capability, and HALT and RESTART operations. All these features will eventually be included in the Load operation. We had expected to have most of these features working by this time, but the problem of getting the Load program to construct completely error-free data bases has been more difficult than anticipated, and has diverted our attention. Due to the nature of cross-indexing, there is no such thing as a partially correct TDMS data base. Any errors in the linkage internal to the data base can be considered catastrophic errors. From the more recent successes we have had in loading data bases without errors, we feel confident that our energies can now be returned to incorporating within Load the many features still scheduled.

3.2.3 Query

Query allows TDMS users to retrieve data from a TDMS data base in an ad hoc fashion. The major commands--SHOW, PRINT, and DESCRIBE--have been working well for many months, allowing users to formulate retrieval requests employing a great variety of conditional expressions ("WHERE clauses"). Query has been the most heavily used of the TDMS components. There are occasional errors reported, but by and large it seems to be working with all features that were originally planned for the initial delivery.

During this reporting period, the testing of Query revealed that the planned output formatting for both the SHOW and PRINT commands was not as useful as desired. Accordingly, new formats were developed and included in the program. These all seem to be working well.

3.2.4 Compose

Compose is a very large report generator. It is a two-phase operation: the first phase is that of designing a report in which Compose acts in a highly

interactive fashion with the user to design a report format according to user instructions. The language used in doing this job is extremely powerful and yet has proven to be relatively easy for people to learn. The second phase is the actual production of the desired report; it is analogous to a compile-and-go operation.

Both phases of Compose are currently operating fairly well. There are, however, quite a few features that were not implemented by the end of the reporting period; work is continuing on all of these. The list of limitations includes no off-line output, no report modification capability, limited or no use of many of the format control commands like LINK, MASK, etc., no tables or derived variables, and only one element in the SORT list.

3.2.5 Update

Update permits a user to dynamically change, add, or delete single values or entire entries in a data base as it exists on the disc. This operation is an extremely complex job, requiring linkage changes throughout the data base every time an Update command is exercised.

We anticipated that Update would not be available as early as many of the other operations. However, the program checkout phase has been going exceptionally well. The SET command in Update is now working on all test cases that have been tried. The ADD and REMOVE commands are all coded and well into checkout. In addition, the basic Query commands, PRINT, SHOW, and DESCRIBE, have been incorporated within the Update operation. Once the ADD and REMOVE commands are in, only format and validity checking will remain to be done in Update.

3.2.6 Maintain

Maintain represents a significant research effort within TDMS. The intention of this operation is to provide the user with a generalized means for merging, subsetting, extracting, ordering, restructuring, and updating data bases. The Maintain programs are designed to accept one or two different data bases, an on-line description of the desired output data base, rules for the selection of data, and the transformations that are required. As with Compose, the capability to interactively describe a task on-line, save it for repetitive use, and modify it as desired has been designed.

Of the eight tasks originally isolated as a meaningful set of generalized maintenance tasks, only Batch Update and Copy and Cleanup have been scheduled for inclusion in initial versions of TDMS. The Copy and Cleanup task is required to reallocate spares in a data base that has been extensively modified by use of the Update operation. This task has been coded, but checkout has been delayed due to lack of personnel. The Batch Update task is aimed at handling large volumes of changes rather than the small volumes expected to be handled on-line by Update. In the Batch Update task, all the interactive part describing the desired actions has been coded and checked out. However, only the SET command

is working properly in the production end of the task operation. The delays in getting a satisfactory operating Batch Update task have been caused by the same kinds of problems that have delayed the Load operation. Progress is being made and work is proceeding as rapidly as possible on both the ADD and REMOVE commands, as well as on task description modification, tables and derived variables, full arithmetic capabilities, and a number of the other features.

3.2.7 Reformat

Reformat is intended to provide a straightforward method of converting symbolic data that already exists in machine-readable form into the numbered field data set format required by the TDMS Load operation. The basic capabilities originally designed for Reformat have been coded and tested for both standard fixed-field input types and indentured fixed-field types.

This latter input type is the one being used by the FCS as operating in the NMCSSC. We anticipate that Reformat will be used heavily by the initial users for converting existing data files, but as yet no results have been reported. Planned work on extending or revising the Reformat operation will be delayed until some use of existing capabilities (both Reformat and DBL) indicates the more beneficial approach to take.

3.2.8 Data Base Oriented Programming Language

This work is aimed at providing a bridge between the computational capabilities of JOVIAL and the data base manipulation ability of TDMS. A JOVIAL program is being developed which will utilize parts of the TDMS system and will retrieve data from a standard TDMS data base. This JOVIAL routine may then be compiled as part of any JOVIAL program, allowing the programmer to perform whatever processes he wishes. In addition to a capability to retrieve data from an existing TDMS formatted data base, the JOVIAL subroutine utilizes the Query language of TDMS so that a user need not learn new forms or expressions.

Work on this task was temporarily suspended during the previous reporting period, awaiting the resolution of the problem of segmenting the TDMS retrieval package. Now that this problem has been resolved, work has been resumed on this task. During the coming reporting period, coding of the program will continue, and it will be checked out using TDMS data bases.

3.2.9 META5 Adaptation

A special-purpose language (known as DBL--Data Base Language) has been adapted from META5 for the purpose of reformatting data bases into a form acceptable to TDMS. DBL is intended for use by nonprogrammers, and thus is relatively nonprocedural in its use. It allows the user to describe the format of a data base and those format conversions he wishes to effect. He need not be well versed in TDMS conventions, since the burden of actually performing the conversions is borne by the DBL compiler. The language is simple enough to be learned in a few hours.

Compilers for DBL and META6 (an extension of META5) are now operating on both the IBM 360/50 and Q-32 computers. Programs written for either compiler are completely compatible, that is, the same program will operate on either machine. Major emphasis during this reporting period was placed on adding features to the DBL language. These features included integer arithmetic, new data structures (such as conversion lists, literal- and value-class variables), and conditional statements (the ALGOL Boolean and relational operators). Other features were added to the language in order to shape it to user's needs.

Operational experience in the use of DBL was also gained during this reporting period in the reformatting of approximately 25 data bases. These were converted from formats acceptable to other machine programs to those employed by TDMS and TSS-LUCID.

3.3 PLANS

During the next reporting period, we expect that many of the current limitations of TDMS will be removed. We also expect that the use testing will uncover other unknown errors, limitations, and requirements. Within manpower limitations, changes will be made to correct the problems discovered. In addition, a continuing effort will be made on timing studies and changes for program improvement. Work on a few major projects should also begin during the next reporting period. These will include design study for the inclusion of text-processing in TDMS, and a design effort on the inclusion of management assistance tools (sophisticated statistical functions, linear programming, etc.).

In the DBL area, work will continue on embedding META6 fully in the DBL system. The DBL compiler will also be changed to produce assembly language programs, rather than the currently produced META6 programs.

Further efforts during the next six months will be directed at providing DBL with the capability to recognize free format data bases (in addition to the fixed format data bases it currently recognizes). Work will also begin on further generalizing DBL output so that the system will be able to translate data bases from TDMS/LUCID format to some other (as yet unspecified) format, such as FFS.

3.4 STAFF

A. H. Vorhaus, Staff Head
- Beverly Bourne, Technical Assistant
- R. H. Barsalou
- J. R. Shiban
- Capt. J. Krause (on loan from AFICCS)
- A. Ude (on loan from NMCSSC)
- Lt. J. Wassem (on loan from ESD)

TDMS Language Project

J. D. Reynolds, Project Leader
Nora Bosak
S. H. Brown
D. P. Crowson
D. D. Fess
Celeste Hand
H. W. Henley
Alice Lanes
J. F. Murray
Jean Saylor
S. Shapiro
R. P. Stewart

TDMS Preparation and Maintenance Project

C. A. Kribs, Project Leader
K. M. Brandon
P. A. DeSimone
J. Farell
Leah Fine
R. Gates
J. P. McGahey
M. Peltz
R. B. Reynolds
Margaret Rybak
W. M. Shasberger
W. H. Stone

Data Base Oriented Programming Language

D. D. Fess (part time)

META5 Adaptation

M. Schaefer

3.5 DOCUMENTATION

Some of the documents describing the data management portion of the ADEPT system were released in SDC's "Note" series. These documents are internal working papers only, and have not been cleared for open publication.

Shiban, J. R. The time-shared data management system interim user's guide. SDC document TM-3849/000/01. 17 April 1968. 3 pp.

Establishes a documentation series for TDMS. This series includes a set of interim user's guides for all TDMS operations.

Shiban, J. R. and Wassem, J. ADEPT/TDMS interface user's guide. SDC document TM-3849/001/00. 4 April 1968. 23 pp.

Assists the ADEPT Time-Sharing System user in operating TDMS. Also includes the system rules and conventions for operating in TDMS.

Krause, J. Define interim user's guide. SDC document TM-3849/002/01. 10 June 1968. 17 pp.

Provides instructions for prospective users of the Define operation during the period of initial familiarization.

Krause, J. Load interim user's guide. SDC document TM-3849/003/00. 22 April 1968. 28 pp.

Provides instructions for prospective users of the Load operation during the period of initial familiarization.

Ude, A. Query interim user's guide. SDC document TM-3849/004/00. 29 February 1968. 22 pp.

Provides instructions for prospective users of the Query operation during the period of initial familiarization.

Ude, A. Update interim user's guide. SDC document TM-3849/005/00. 17 April 1968. 16 pp.

Provides instructions for prospective users of the Update operation during the period of initial familiarization.

Ude, A. Compose interim user's guide. SDC document TM-3849/006/00. 14 June 1968. 38 pp.

Provides instructions for prospective users of the Compose operation during the period of initial familiarization.

Krause, J. Maintain interim user's guide. SDC document TM-3849/007/00. 3 June 1968. 48 pp.

Provides instructions for prospective users of the Maintain operation during the period of initial familiarization.

Wassem, J. Convert interim user's guide. SDC document TM-3849/009/00. 29 February 1968. 10 pp.

Describes how to use Convert, an interactive Q-32 program used to convert TSS-LUCID data bases into TDMS format.

Vorhaus, A. H. The time-shared data management system (TDMS) for the IBM 360/65. SDC document N-(L)-23550/000/03. 11 March 1968. 4 pp.

Establishes a documentation series for TDMS. This series includes documentation of table and file design, and program design specifications.

DeSimone, P. A. and Kribs, C. A. TDMS I/O package: communication and data specifications. SDC document N-(L)-23550/060/00. 19 February 1968. 30 pp.

Describes the environment and methods for calling procedure XIO, which is used to perform all TDMS input/output operations. Presents a general description of the tables and items used, as well as a more detailed description of each item. Also contains the JOVIAL definitions of the tables and items, as well as bit layouts of the tables.

Stone, W. H. Message loader program specifications. SDC document N-(L)-23550/127/00. 19 July 1968. 18 pp.

Describes Message Loader (ML), a program running under ADEPT which loads or lists messages in the TDMS message file. ML, which is not part of TDMS, is used when a greater number of messages are to be loaded or listed than it would be appropriate to do with an interactive device. Principal input to ML is a prestored tape of card images; the output of the list option is a DLO tape.

Peltz, M. SEGMOD: TDMS component segment modification program. SDC document N-(L)-23550/606/00. 26 March 1968. 3 pp.

Describes SEGMOD, a conversational program designed to facilitate on-line modifications to TDMS program segments. By using a predetermined name, a user can call into core any segment, modify it using DEBUG and restore it to the segment file.

Peltz, M. EMOVE: Multiple disc file to tape copy program. SDC document N-(L)-23550/607/00. 10 June 1968. 4 pp.

Describes EMOV, a program that copies onto a single tape reel up to 20 disc files under the ADEPT system. This program bypasses the single file per tape limitation currently in ADEPT by organizing the disc files as subfiles (groups of records) on the tape.

Peltz, M. ELOD: TDMS segments load program. SDC document N-(L)-23550/608/00. 17 June 1968. 3 pp.

Describes ELOD, a conversational program operating under ADEPT that modifies the existing TDMS program segment file by loading newly compiled segment binary decks.

Schaefer, M. Interim user's guide for DBL: Introduction to the series and table of contents. SDC document TM-3870/000/00. 12 March 1968. 1 pp.

Establishes a document series for DBL user's guides. Includes volume numbers, titles, reissue and modification data.

Schaefer, M. Interim user's guide for DBL: General description. SDC document TM-3870/001/00. 8 March 1968. 34 pp.

Describes the general features of DBL. Provides examples of its use, a sample DBL program and explanation, definition of DBL terms, and a complete overview of the DBL language.

Wassem, J. and Schaefer, M. Interim user's guide for DBL: Sample program. SDC document TM-3870/002/00. 8 March 1968. 14 pp.

Describes a sample DBL program used to reformat a typical fixed-field data base on personnel information. Contains a listing of the data base items, the DBL program used to convert the data to TDMS format, and some sample output data.

Schaefer, M. and Book, E. DBL interim user's guide: ADEPT usage instructions. SDC document TM-3870/004/00. 2 July 1968. 9 pp.

Presents instructions for the ADEPT Time-Sharing System user on how to convert data base formats using DBL and META6. Includes instructions on loading DBL, examples of user/program interaction, a discussion of META6 operation, and conventions used for input/output operations.

Vorhaus, A. H. The time-shared data management system (TDMS) language specifications. SDC document TM-3370/000/02. 11 March 1968. 3 pp.

Establishes a documentation series for TDMS. This series includes language specifications for all TDMS operations.

Bosak, N. The language specifications for the ma tain operation of TDMS. SDC document TM-3370/006/00. 31 May 1968. 40 pp.

Describes the Maintain operation of TDMS, which is used to create an output data base from either one or two input data bases. There will be eight separate functions within the Maintain operation, each one independent of the others. Two of the functions, Batch Update and Copy and Cleanup, are described in this document. The other six functions will be described in future documents.

Reynolds, J. D., Bosak, N. and Shiban, J. R. The language specifications for the load operation of TDMS. SDC document TM-3370/007/00. 8 March 1968. 25 pp.

Describes the Load operation, which creates a data base file using a data base description produced by the Define operation of TDMS and formatted input data. The input data format and the language for user interaction with the Load operation are described in this document.

4. PROGRAMMER'S PACKAGE

4.1 INTRODUCTION

The programmer's package being developed for use in the ADEPT system contains four elements for the professional programmer: a compiler, a debugging capability, an editing capability and a utility program. A teletype interpreter, known as TINT, is also provided for the user who is not a professional programmer. In addition, an Interactive Programming Support System--which combines these capabilities into a single system--is being developed.

4.1.1 JOVIAL Compiler

JOVIAL is a general-purpose programming language well suited for a variety of different applications, including scientific and engineering problems involving numeric computation, business problems involving large data files, and logically complex problems involving symbolic data. Because of the optional control it provides over the details of storage allocation, JOVIAL is especially suitable for problems requiring an optimum balance between data storage and program execution time.

The JOVIAL compiler being developed for use by the professional programmer in the ADEPT system has all of the capabilities of Basic JOVIAL, as well as several extra features requested by the initial users of the system. Some of these additional capabilities include provision for longer literals and the ability to compile program segments independently and then combine these segments at execution time. This compiler is compatible with a JOVIAL compiler delivered to the initial system users, which runs under their (OS/360) operating system.

4.1.2 Debugging Aid

The debugging program being developed as part of ADEPT provides an on-line capability for the professional programmer to look at and change his program and program data during execution, and to switch between "execution" and "look-and-change" modes.

4.1.3 Editing Aid

The editing program provided in ADEPT gives the professional programmer an on-line capability to maintain and modify his program in source language. It may also be used to generate original code. The editing commands available at present are: COMPOSE, INSERT, REPLACE, DELETE, MOVE, COPY, SEQUENCE, and CHANGE. Also available are the utility commands: DISPLAY, SAVE, and QUIT.

4.1.4 Utility Program

The utility program provides some basic program maintenance services needed by the programmer--namely, card-to-tape and tape-to-printer and punch conversions.

4.1.5 TINT (Teletype Interpreter)

TINT is an interactive programming system aimed at the casual programmer who has small-to-moderate sized programs. It uses a dialect of JOVIAL, and combines techniques of compilation and interpretation.

TINT is designed to bridge the language gap between the computer and the non-programmer user; it operates interpretively, on-line. The on-line nature of the system makes it convenient to use. With TINT, the user can create, check out, execute, modify and re-execute a program directly from a remote teletype--and can often carry out the entire programming process at a single sitting. TINT is particularly suited to compact programs such as short mathematical problems, subprogram checkout, and other "one-shot" operations.

4.1.6 Interactive Programming Support System (IPSS)

The goal of the Interactive Programming Support System (IPSS) is to permit all of the programming processes--composition, editing, execution, testing and documentation--to be carried out as parts of a single, coordinated activity centered around an interactive compiler. The system unifies techniques that are usually embodied in separate functional programs so that the programmer need not know which particular program is performing a specific task. The system, of course, is intended for a time-sharing environment, with user interaction via a CRT/keyboard console or teletypewriter.

4.2 PROGRESS

A major effort during this period has been spent on modifying the input/output sections of programs to allow them to operate with the latest releases of the ADEPT Executive. These releases contained significant new capabilities in the Cataloger and SPAM, and the previous simulations of these capabilities were replaced by actual code.

4.2.1 JOVIAL Compiler

The version of the JOVIAL compiler running under Release 3 received some shake-down from users early in this reporting period, showing few bugs. In parallel with this shakedown, the changes needed for Release 4 were implemented. Near the end of the period, the Release 4 version was made available to users. A number of substantial test cases were compiled, testing many capabilities including compool and compool assembly. It should be noted that the compiler itself has been completely compiled.

In addition to checkout of the compiler, some changes have been made to speed up the compiler. The primary changes have been in the use of input/output capabilities such as buffering and packing of records. Also, interactive operation of the compiler has been made smoother.

4.2.2 Debugging Aid

The debugging program--which was designed, coded, and mostly checked out in the previous reporting period--was put to use in Release 3 early in this period. Users found a number of minor errors which were easily corrected. The program has now been used extensively throughout and is working well. Only a small amount of work was needed to make the program operate with Release 4, since the debugger uses few files and much of its input/output is either via the interactive console or the system loader. Portions of the program were recoded, however, in order to meet the Basic Executive's space requirements.

4.2.3 Editing Aid

The editing program (which was released in January) was modified to operate with Release 4. In addition, the available options for input and output were increased to make the program applicable in a broader range of situations. Input is now accepted from either 7- or 9-track tape in addition to disc. The data may be in S1 format (header information, records located sequentially) or S2 format (no header information and sequential ordering--essentially, the form of prestored "foreign" tapes). Records may be packed. Output may also be to 7- or 9-track tape in addition to disc.

For users who have become familiar with the operation of EDIT, there is an option to select a terse conversational mode. In this mode of operation, EDIT saves typing time by communicating with abbreviated messages. For example, one of EDIT's normal queries is TAPE OUTPUT OPTIONS/7TRACK/9TRACK. The short form is OUTPUT NONE/7/9.

4.2.4 Utility Program

A utility program designed to satisfy the initial tape-related print, punch and card read requirements of ADEPT users was released. New tape files can be created from card inputs or existent tape files can be printed or punched. (Though later versions will include disc file capabilities, this version is restricted entirely to tape-based files.) The program makes the necessary Cataloger calls to open and close files, supplies needed SPAM and I/O buffers, generates the required channel programs, and monitors and drives the I/O devices by making appropriately timed IOS calls. The user specifies the operation he wishes performed on his file, and provides labeling and accounting information.

This version of the utility function supports three operations:

LIST	Specifies the printing of an existent tape file.
PUNCH	Specifies the punching of an existent tape file.
CARDIN	Specifies the creation of a system-compatible tape file from a deck of cards.

The program was designed primarily to handle tapes generated within the ADEPT system. However, it is also possible to perform the LIST and PUNCH operations with tapes generated outside the system. Specifically, the program is compatible with the formats of tapes prepared for the SC-5000 electronic printer and Q-32 peripheral equipment. The program will also accept and prepare tape files that are blocked.

The utility function is a public program created and controlled from an interactive console. It operates in the ADEPT time-sharing environment, and thus is allotted quanta and is swapped in and out of core as though it were a user program. However, large-scale I/O operations could possibly degrade the system if performed during periods of high load. Therefore, the scheduling and initiation of the utility function is handled by the system operator, who has sufficient information to decide when the load is appropriate and required machine configurations are available.

4.2.5 TINT (Teletype Interpreter)

The TINT program was modified to operate under Release 4 and was successfully checked out. There are only a few users at present, but they have encountered no problems. Since TINT operates well from remote, interactive consoles, it has been used frequently for demonstrations.

A few modifications have been made to accommodate variations in computer configuration. Versions have been compiled which use the 2302, 2314, and 2311 discs as permanent on-line storage.

A new capability has also been added to TINT: the ability to save and later reuse a TINT program. The command

? SAVE *filename*

causes TINT to store a copy of the user's program in permanent on-line storage. The command

? LOAD *filename*

brings the code back, ready for execution or further modification through TINT's editing functions. If the user does not wish to use the storage device normally

used, he may ask TINT--via the interactive console--to store his program on another device. When he later wishes to load such a program, however, he is responsible for telling TINT on what device it is stored.

4.2.6 Interactive Programming Support System (IPSS)

Major components of IPSS became operable during this reporting period. Work on the use of CRT displays for composing and editing programs on-line received considerable attention. Progress was made in the use of the IBM 2250 display and keyboard as the machine-programmer communication link.

The display is used to convey information both from IPSS to the user, and to IPSS from the user. IPSS either requests input and waits for the user to respond, or it outputs information in response to user actions and again waits for the next user response. User responses may be answers to specific questions from IPSS, program statements, or requests for IPSS action.

Use of the CRT greatly speeds up output from IPSS and enables it to minimize the amount of typing the user must do. In the case of error correction and editing, IPSS regenerates program statements so that the user can modify them as necessary, and does not have to retype entire statements. The speed also enables the user to make effective use of the display as a substitute for leafing through program listings. IPSS permits the user to change his program in several ways: he can delete lines, replace lines, insert new lines, and move lines. He can also change "signs" within statements in a particular area of his program. These tasks are all done by typing the appropriate edit commands. The user can save himself extra typing if the change consists of modifying statements. The command

```
#COPY M,N $
```

displays lines M through N on the CRT and the user can then treat the lines as if he had typed them there. For example, suppose that a statement has been incorrectly composed as

```
GOYO OVER $
```

IPSS will display the erroneous statement, and will position the CRT cursor under the "Y," as well as displaying an error message. The user need only type the letter "T" and "send" the entire line back to his program file. If he had been using a typewriter-like terminal, he would have had to retype the entire incorrect statement and everything after it, or type in positioning information.

Several "leafing through the program" capabilities have been provided. Suppose that in the course of debugging, the programmer does not remember every place the identifier XX is used.

The following sequence of commands shows how the programmer can make use of saved program information, and can use displays to find and correct errors in program logic.

The user first types

#DISPLAY SET USED XX

IPSS clears the face of the CRT and displays the information

XX S3 U5 B7

This indicates that the identifier XX is set in line 3, used in line 5, and both set and used in line 7. The user then types

#ADD DISPLAY 3

which adds statement 3 to the current display. If he then wishes to see another line, say statement 7, he types

#ADD DISPLAY 7

which adds statement 7 to the existing display (see Figure 4).

The face of the display has been organized into three logical "blocks" to facilitate these uses (see Figure 5). The number pair shown in Figure 5 represents block and field numbers; e.g., 3,0 represents block 3, field 0. Each line is composed of one field except for block 1, where the line is divided into two fields.

```
#ADD DISPLAY 7

        XX S3 U5 B7

0003.   INITIAL.  XX=10000$

0007.             XX=XX+1$
```

Figure 4. Set-Used Information

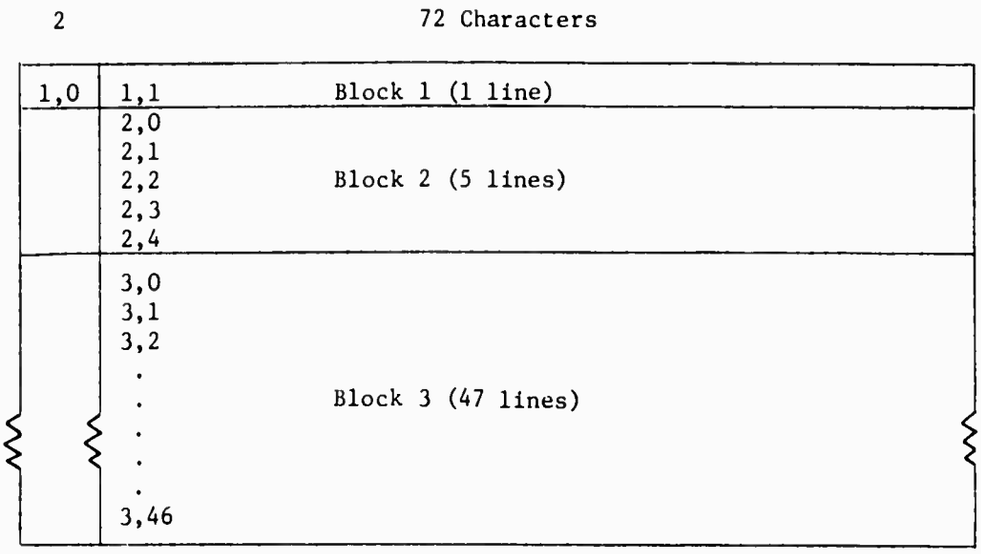


Figure 5. Display Organization

A "block" is roughly defined as a physical segment of the display surface, set aside for some specific function. Block 1 is the area where IPSS requests user actions. IPSS signals the user when it is ready for input by displaying two question marks (??) in position 1,0 (block 1, field 0) and by placing a cursor in position 2,0. A descriptive message is written in position 1,1. Block 2 is the work area where the user responds to IPSS input requests. Since the cursor has been placed in position 2,0, typing on the keyboard causes input to appear there. The user is restricted by IPSS from typing into block 3. Block 3 is the area where IPSS displays information requested by the user or automatically generated in response to some user action. Every line but the first contains one field of 72 characters. The first line is divided into two fields of 2 and 72 characters each. Block 1 consists of 1 line, block 2 consists of 5 lines, and block 3 consists of 47 lines. The size of the blocks can be varied for experimental purposes.

The length of the program is not limited to the length of block 3. When the program exceeds the length of block 3, lines at the top are removed and the new lines appear at the bottom. The line identifiers M,N in the command

#COPY M,N

can refer to any lines of the program, visible or not. The seemingly more straightforward technique of moving the cursor to the appropriate position in block 3 has not been included in IPSS. The technique of determining which lines have been changed in block 3 is difficult in the type of display equipment we

ultimately hope to use--namely, small tabular displays. We have avoided the use of some of the expensive, complex hardware available with the 2250 (such as light pens and function keys) in order to be able to move to smaller equipment. We are assuming just the ability to move a cursor and send an interrupt to the main computer.

A paper describing this work was written and accepted for presentation at the 1968 Fall Joint Computer Conference.

4.3 PLANS

All of the programs in the programmer's package have been checked out and are in the user shakedown phase. It is expected that a variety of new bugs will be uncovered as new users with different characteristics begin using ADEPT. We do not expect any of them to constitute serious problems.

In addition to receiving maintenance, the various programs will be modified and refined to increase their usefulness and smoothness of operation. EDIT and TINT will be modified so that a user can pass programs between these subsystems. We are considering the possibility of having a common format for files of "lines" (cards, teleterminal input, printer output) which is acceptable to all utility programs.

We are also studying the feasibility of having TINT output a user's program in a form acceptable to JOVIAL. Many of the differences between TINT and JOVIAL can be taken care of mechanically (e.g., JOVIAL requires <u>all</u> data to be defined, whereas TINT has numerous default conditions). A few TINT capabilities will require that the JOVIAL library have some new subroutines (e.g., READ and PRINT). We expect there will be only a few TINT capabilities that would require extensive effort to convert. In these cases, the TINT user will probably have to program around the deficiency.

Work on IPSS will concentrate on providing a single, integrated compiler. (At the present time, the compiler and the editing facilities are independent entities, requiring the user to switch between them himself. More importantly, modification of the source language program requires a recompilation.) In addition, a number of ideas are being developed on the operation of the CRT display. As the existing capability becomes more widely used, any human engineering faults should be exposed.

The debugging program will remain stable, relative to capabilities. As the executive is refined, corresponding changes will be made in the debugger. For example, the debugger may be moved from the Basic Executive to the Extended Executive to meet space requirements.

Work will continue on improving the performance of the JOVIAL compiler. The input/output speed can be effectively increased by increasing the packing density of the information being passed, or even eliminating some output in certain cases. The dictionary and the library are prime targets. There are a number of proposals being considered which will improve the code produced by the compiler; some of these will be implemented. (Such improvements are, of course, reflected in the compiler's operation, since the compiler itself is written in JOVIAL.)

As new users come on the system, new demands are generated for additional utility tools. A program to copy files between tape and disc (with some capability to modify format) has been written and is ready for shakedown. Additional capabilities in the utility area will be considered for implementation.

4.4 STAFF

E. H. Jacobs, Staff Head
- H. Bratman, Compiler Systems Project Leader
- G. M. Cady
- H. G. Martin
- R. N. Mathur
- Ellen C. Perstein
- N. A. Sandin

4.5 DOCUMENTATION

Many of the documents describing the programmer's package component of the ADEPT system were released in SDC's "Note" series. These documents are internal working papers only, and have not been cleared for open publication.

Brewer, R. A. ADEPT debugger user's guide. SDC document N-23759/022/00. 25 April 1968. 14 pp.

Presents instructions on the use of DBUG, the ADEPT on-line debugging program. Includes a description of the debugging commands and detailed instructions on how to debug both machine-oriented programs and JOVIAL programs.

Bratman, H. and Perstein, E. C. User's guide for IPSS--interactive programming support system. SDC document N-23759/200/02. 25 June 1968. 30 pp.

Provides instructions for ADEPT programmers on how to use IPSS to syntax check and edit JOVIAL programs. Includes explanations of the commands available, possible error situations, and a list of numbered error messages.

Kameny, I. M. User's guide for EDIT. SDC document N-23759/210/01. 10 June 1968. 27 pp.

Explains the use of EDIT, an on-line program that can be used to compose, change, or rearrange a file of symbolic text. Includes a description of files used by EDIT, the file descriptive information requested by EDIT, usage instructions, commands available, and possible error situations. (EDIT replaced a similar program: MODIFY.)

Sandin, N. A. JOVIAL (prJ5.3) compiler user's guide. SDC document N-23759/250/00. 9 April 1968. 17 pp.

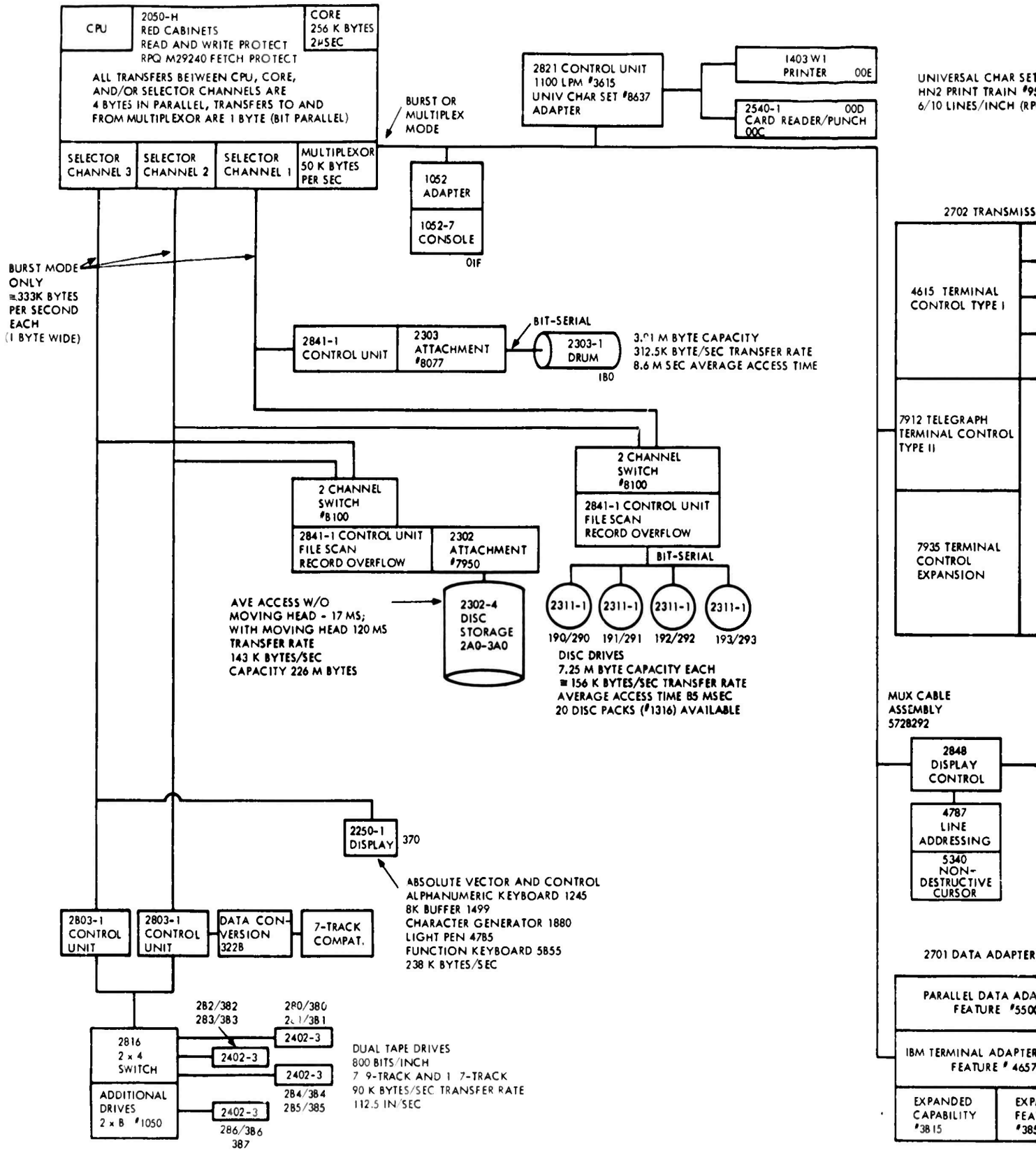
Describes the use of the JOVIAL compiler that operates under the ADEPT executive. Includes several detailed examples, showing user input, program responses, I/O formats, and files used by the compiler.

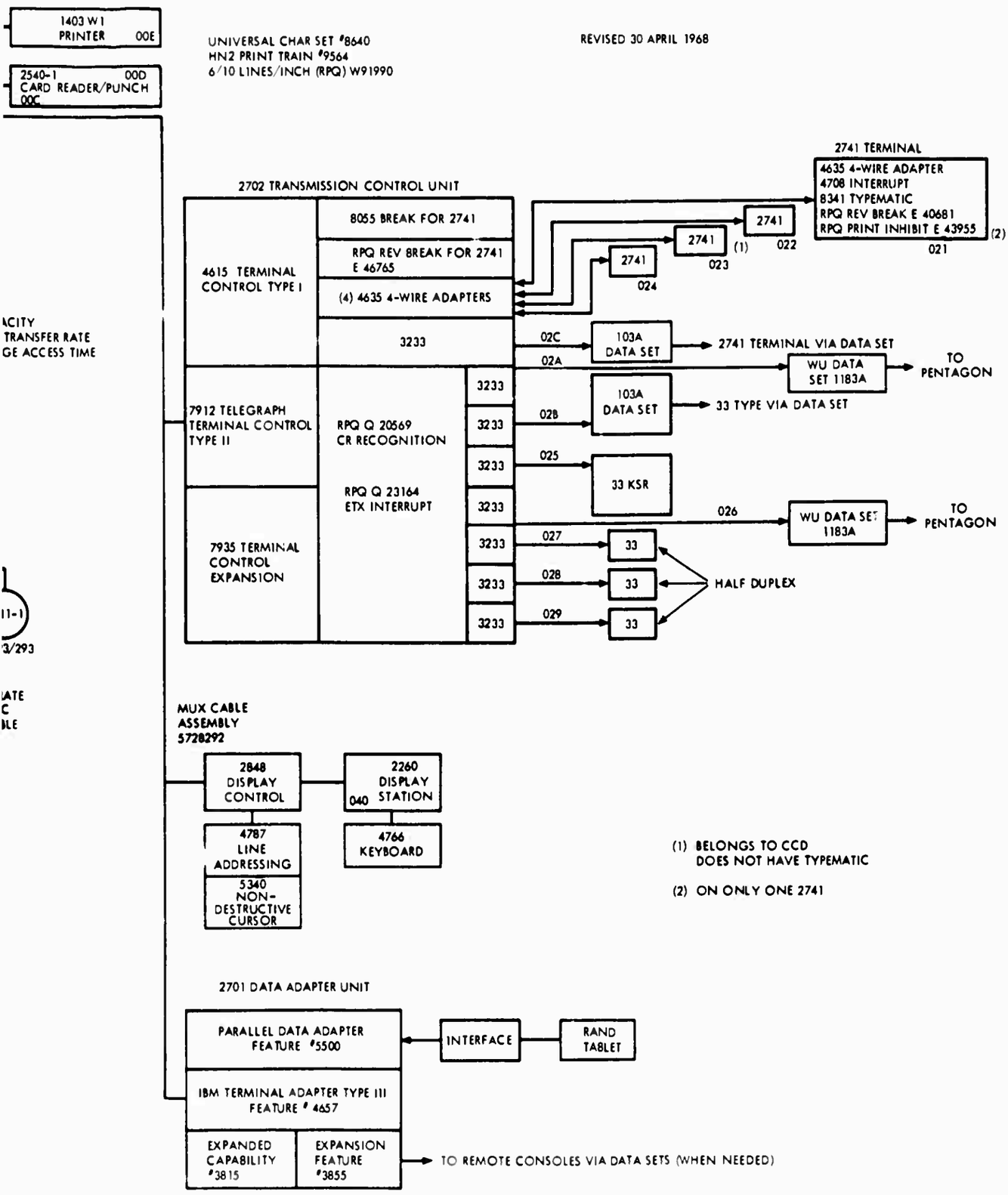
Bernzott, P. E. Operator utility function user's guide. SDC document N-23759/400/00. 4 March 1968. 16 pp.

Describes the initial release of the ADEPT operator utility function, a self-contained program that can only be loaded by the system operator. It is designed to satisfy the initial print, punch, and card read requirements of ADEPT users.

Mathur, R. N. TINT1.1 user's guide. SDC document N-23759/502/00. 1 July 1968. 6 pp.

Describes the use of TINT1.1, a teletype interpreter operating under the ADEPT executive. Includes instructions for initiating and terminating a job, and for saving a program and loading it from any disc.


CPU
2050-H
RED CABINETS
READ AND WRITE PROTECT
RPQ M29240 FETCH PROTECT
CORE
256 K BYTES
2μSEC
ALL TRANSFERS BETWEEN CPU, CORE, AND/OR SELECTOR CHANNELS ARE 4 BYTES IN PARALLEL, TRANSFERS TO AND FROM MULTIPLEXOR ARE 1 BYTE (BIT PARALLEL)
SELECTOR CHANNEL 3
SELECTOR CHANNEL 2
SELECTOR CHANNEL 1
MULTIPLEXOR 50 K BYTES PER SEC
BURST OR MULTIPLEX MODE
2821 CONTROL UNIT
1100 LPM #3615
UNIV CHAR SET #8637
ADAPTER
1403 W1 PRINTER 00E
2540-1 00D
CARD READER/PUNCH 00C
UNIVERSAL CHAR SET
HN2 PRINT TRAIN #95
6/10 LINES/INCH (RP
1052 ADAPTER
1052-7 CONSOLE
01F
2702 TRANSMISS
BURST MODE ONLY
≅333K BYTES PER SECOND EACH
(1 BYTE WIDE)
4615 TERMINAL CONTROL TYPE I
2841-1 CONTROL UNIT
2303 ATTACHMENT #8077
BIT-SERIAL
2303-1 DRUM
180
3.01 M BYTE CAPACITY
312.5K BYTE/SEC TRANSFER RATE
8.6 M SEC AVERAGE ACCESS TIME
7912 TELEGRAPH TERMINAL CONTROL TYPE II
2 CHANNEL SWITCH #8100
2841-1 CONTROL UNIT
FILE SCAN
RECORD OVERFLOW
2 CHANNEL SWITCH #8100
2841-1 CONTROL UNIT
FILE SCAN
RECORD OVERFLOW
2302 ATTACHMENT #7950
BIT-SERIAL
7935 TERMINAL CONTROL EXPANSION
AVE ACCESS W/O MOVING HEAD - 17 MS;
WITH MOVING HEAD 120 MS
TRANSFER RATE
143 K BYTES/SEC
CAPACITY 226 M BYTES
2302-4 DISC STORAGE 2A0-3A0
2311-1
2311-1
2311-1
2311-1
190/290
191/291
192/292
193/293
DISC DRIVES
7.25 M BYTE CAPACITY EACH
≅ 156 K BYTES/SEC TRANSFER RATE
AVERAGE ACCESS TIME 85 MSEC
20 DISC PACKS (#1316) AVAILABLE
MUX CABLE ASSEMBLY
5728292
2848 DISPLAY CONTROL
4787 LINE ADDRESSING
5340 NON-DESTRUCTIVE CURSOR
2250-1 DISPLAY
370
ABSOLUTE VECTOR AND CONTROL
ALPHANUMERIC KEYBOARD 1245
8K BUFFER 1499
CHARACTER GENERATOR 1880
LIGHT PEN 4785
FUNCTION KEYBOARD 5855
238 K BYTES/SEC
2803-1 CONTROL UNIT
2803-1 CONTROL UNIT
DATA CONVERSION 322B
7-TRACK COMPAT.
2701 DATA ADAPTER
PARALLEL DATA ADA
FEATURE #550
IBM TERMINAL ADAPTER
FEATURE # 4657
EXPANDED CAPABILITY #3815
EXP
FEA
#385
2816 2 x 4 SWITCH
ADDITIONAL DRIVES 2 x B #1050
282/382
283/383
280/380
281/381
2402-3
2402-3
2402-3
2402-3
284/384
285/385
286/386
387
DUAL TAPE DRIVES
800 BITS/INCH
7 9-TRACK AND 1 7-TRACK
90 K BYTES/SEC TRANSFER RATE
112.5 IN/SEC


1403 W1 PRINTER 00E
2540-1 00D CARD READER/PUNCH 00C
UNIVERSAL CHAR SET #8640
HN2 PRINT TRAIN #9564
6/10 LINES/INCH (RPQ) W91990
REVISED 30 APRIL 1968
2741 TERMINAL
4635 4-WIRE ADAPTER
4708 INTERRUPT
8341 TYPEMATIC
RPQ REV BREAK E 40681
RPQ PRINT INHIBIT E 43955
(2)
021
2741
022
2741
(1)
023
2741
024
2702 TRANSMISSION CONTROL UNIT
4615 TERMINAL CONTROL TYPE I
8055 BREAK FOR 2741
RPQ REV BREAK FOR 2741 E 46765
(4) 4635 4-WIRE ADAPTERS
3233
02C
103A DATA SET
2741 TERMINAL VIA DATA SET
02A
WU DATA SET 1183A
TO PENTAGON
7912 TELEGRAPH TERMINAL CONTROL TYPE II
RPQ Q 20569 CR RECOGNITION
RPQ Q 23164 ETX INTERRUPT
3233
02B
103A DATA SET
33 TYPE VIA DATA SET
025
33 KSR
026
WU DATA SET 1183A
TO PENTAGON
7935 TERMINAL CONTROL EXPANSION
027
33
028
33
029
33
HALF DUPLEX
ACITY
TRANSFER RATE
GE ACCESS TIME
11-1
3/293
ATE
C
LE
MUX CABLE ASSEMBLY 5728292
2848 DISPLAY CONTROL
2260 DISPLAY STATION
040
4787 LINE ADDRESSING
5340 NON-DESTRUCTIVE CURSOR
4766 KEYBOARD
(1) BELONGS TO CCD DOES NOT HAVE TYPEMATIC
(2) ON ONLY ONE 2741
2701 DATA ADAPTER UNIT
PARALLEL DATA ADAPTER FEATURE #5500
INTERFACE
RAND TABLET
IBM TERMINAL ADAPTER TYPE III FEATURE # 4657
EXPANDED CAPABILITY #3815
EXPANSION FEATURE #3855
TO REMOTE CONSOLES VIA DATA SETS (WHEN NEEDED)

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

System Development Corporation
Santa Monica, California

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Advanced Development Prototype Semiannual Technical Summary Report to the Director, Advanced Research Projects Agency for the period 30 January 1968 to 30 July 1968

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

J.I. Schwartz, Project Director

6. REPORT DATE

30 July 68

7a. TOTAL NO. OF PAGES

54

7b. NO. OF REFS

6

8a. CONTRACT OR GRANT NO

DAHC15-67-C-0149

b. PROJECT NO.

c

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

TM-3628/002/00

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report describes work done under the Advanced Development Prototype contract from 30 January 1968 to 30 July 1968. The result of this work is ADEPT -- a comprehensive information-processing system being implemented at SDC for operation on IBM 360 computers. The report includes an overview of the current status of system development, and a detailed description of the three major components of ADEPT: a time-sharing executive, a data management component (consisting mainly of the Time-Shared Data Management System), and a programmer's package, which includes a JOVIAL compiler, editing, debugging, and utility programs, a teletype interpreter (TINT), and an Interactive Programming Support System. Also included in this document are the names of staff members assigned to each of the three major project areas, as well as a listing of the documents produced in each area during this reporting period. Upon request, referenced documents will be made available to appropriate organizations.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Advanced Development Prototype (ADEPT)						
ADEPT						
Information processing						
Time-sharing						
Data management						
JOVIAL compiler						
Teletype interpreter (TINT)						
TINT						
IBM 360						
Interactive Programming Support System						
Time Data Management System						
TDMS						