CONTROL DATA
3600 COMPUTER

3600

SCOPE/GENERAL INFORMATION MANUAL

CONTROL DATA 3600 COMPUTER

SCOPE /GENERAL INFORMATION MANUAL

Control Data Corporation
8100 34th Avenue South
Minneapolis 20, Minnesota

Address any comments concerning this manual to:

Control Data Corporation
Technical Publications Department
501 Park Avenue
Minneapolis 15, Minnesota

October, 1963
Pub. No. 532

CONTENTS

DESCRIPTION 1

SCOPE is a monitor system which provides supervisory control of program execution for the CONTROL DATA® 3600 Computer. It facilitates programming and minimizes operating responsibilities, while offering maximum utility of computer time. It is designed so that maximum memory area is available for job execution.

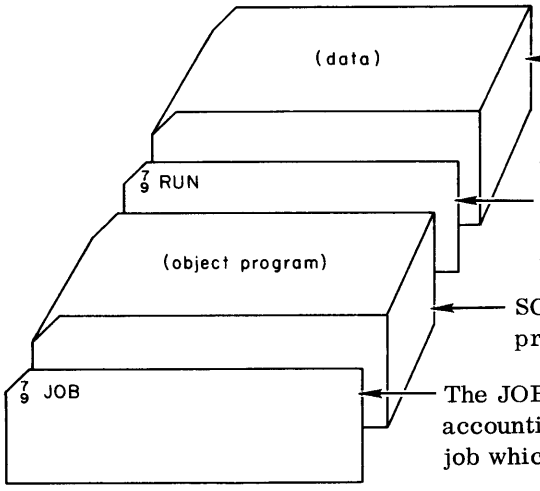
SCOPE program aids

- Job processing control
- Input/Output control
- Interrupt control
- Debugging aids
- Library preparation and maintenance

Minimum equipment configuration required for efficient processing

- 3604 computer module
- 3601 console with typewriter
- 3603 storage module (32,768 words)
- 3602 communication module
- 3606 bidirectional data channel
- 3620 magnetic tape controller with one R/W control
- Six magnetic tape units

The example demonstrates the simplicity with which a program may be executed under control of the SCOPE operating system.



Data, required by the object program, may follow it on the standard input unit.

The RUN control statement provides execution time limits and indicates the object program is to be executed. SCOPE transfers control to the object program.

SCOPE loads the relocatable binary object program into memory.

The JOB control statement provides identification, accounting information, and a time limit for the job which follows.

MAJOR FEATURES 2

- Job processing

  maintains accounting information
  initiates compilations and assemblies
  assigns equipment
  loads and links subprograms
  allocates memory
  assigns banks
  provides OVERLAY processing

- Debugging aids

  SNAP
  TRACE
  error dumps (recovery dumps)
  octal corrections
  memory map
  diagnostics

- Program requests

  input/output control
  external interrupt control
  tape handling, including labeling and continuation reels
  internal interrupt control
  sampling of time, date, equipment status, and available memory

- Library preparation and maintenance

  listing the table of contents
  directory
  preparing and editing library tape

• JOB PROCESSING

A monitor sequence consists of a group of jobs stacked on the standard input unit for processing under SCOPE. Jobs are processed in the order in which they appear, but the operator may alter this sequence by control statements. A job, delineated by JOB control statements, consists of one or more independent programs under the same account number: for example, named library programs (COMPASS, FORTRAN, ALGOL, COBOL), object programs, or library subroutines (SORT).

At the beginning of each job the previous job is closed out, saved tapes are unloaded, scratch and programmer units are released and equipment assignments and declarations from the previous job are cleared. The new job is logged in and the specified time limit is set.

Accounting information includes job identification, date, start, stop, and elapsed time. Job time limits and print line limits may be specified as well as the type of dump to be taken should the job terminate abnormally.

EQUIPMENT ASSIGNMENT

SCOPE locates and assigns physical equipment at run time according to specifications made in the control statements and requests. All references to input/output units are by logical unit numbers arbitrarily chosen by the programmer, and independent of any physical unit designation. Logical units are programmer units, scratch units, and system units.

Programmer units, which may be saved, are retained throughout the job for reference by the program and released at the end of the job.

Scratch units, for temporary use during the operation of a program, are released after each program execution.

System units, used by SCOPE and the programmer, are not released until the end of the monitor sequence. The Load-and-Go unit is an exception; it is available for scratch purposes during job execution. System units include:

Standard Input

Standard Output

Input Comment

Output Comment

Accounting

Library

System Scratch

Load-and-Go

SCOPE assumes that all programmer and scratch unit numbers refer to unique, high-density, labeled magnetic tapes, read or written in binary mode. If another hardware type, mode, or density is required, it is specified by either a control statement or a request. When a logical unit is requested, SCOPE searches the table of input/output equipment available to the computer, and makes the assignment.

LOADER

Subprograms and library subroutines, written in different source languages, may be independently assembled into relocatable binary subprograms and subroutines. SCOPE loads the subprograms and subroutines, links them by relating external symbols to entry points, and allocates common areas. After the subprograms are loaded, the loader searches the SCOPE library directory for subroutines corresponding to the names of all undefined external symbols, then loads the subroutines, linking them to the subprograms.

Errors detected while loading are written on the standard output unit as loader diagnostics. Subprograms and labeled common—common areas which may be preset—may be patched at run time by octal correction cards. The loader establishes a program extension area for instructions which do not fit in the patched area.

MEMORY ALLOCATION

SCOPE maintains a record of available memory, which consists of those locations which have not been assigned to a subprogram, common block, or monitor routine. The limits of available memory may be obtained or changed by a SCOPE request. A memory map, written on the standard output unit before execution, includes a listing of names and core locations of all subprograms, program extension areas, labeled and numbered common areas, and entry points.

BANK

A storage bank may be specified at load time for each subprogram and common block. The programmer may specify that certain subprograms and common blocks be loaded into the same bank without identifying the bank. If banks are not specified, SCOPE loads the program into the bank into which the program most tightly fits. Bank zero is assigned only when the other banks cannot provide space.

OVERLAYS

Programs which exceed memory capacity may be divided into logically independent parts, stored on a tape, and called into memory when needed. The program is divided into a main program and any number of overlays, which may be further divided into segments. The main program, which resides in memory, transfers control to each overlay as it is called in. Overlays in turn transfer control to each segment as it is called. The locations assigned to the main program, overlays, and segments are contained in the memory map.

- **DEBUGGING AIDS**

SCOPE contains facilities to debug programs, make corrections, dump areas of memory at abnormal job termination, and obtain a map of absolute addresses assigned the program by the loader.

SNAP

Snap dumps are periodic dumps of specified areas of memory taken during execution of a running program.

TRACE

Trace dumps occur when jump instructions are executed within a specified range of the program.

At load time the dump area, format, and frequency are specified. A printout of console registers may also be requested. The dump information is written on the standard output unit.

RECOVERY

A recovery dump is taken if a program terminates abnormally. The dump, written on the standard output unit, is in octal with mnemonics. It may consist of a console dump, program dump, labeled or numbered common dump, or a dump of all memory except SCOPE.

CORRECTIONS

Corrections to an object subprogram may be submitted at load time. Instructions and labeled common data may be altered in the subprogram; numbered common may not. The loader assigns a program extension area within the limits of current memory to accommodate additional instructions and data. The corrections are relocatable and submitted in octal notation.

MAP

After the subprogram is loaded, a memory map is written on the standard output unit. It lists the absolute locations of subprograms, program extension areas, labeled and numbered common, and entry points.

DIAGNOSTICS

When the loader detects an error in the loading operation, a diagnostic is written on the standard output unit, and loading continues as though the error had not been encountered. No library subroutines are loaded, however, SCOPE terminates the job after loading the last subprogram and moves to the next job on the standard input unit.

• PROGRAM REQUESTS

Program requests specify input/output control, tape handling, internal interrupt, and special requests.

INPUT/OUTPUT

SCOPE handles all input/output requests made by a running program. SCOPE, using the equipment tables which it maintains, determines whether or not a physical unit has been assigned to the request. If a unit is not assigned, SCOPE locates and assigns a named or numbered tape for an input request or assigns an available unit of high density magnetic tape if the request is for output.

If the unit and a data channel are available, the unit is connected and the request is initiated. If the unit is available but no channel is available, the request is stacked temporarily. Control returns to the main program, and the request is processed when the channel becomes available. If the unit is not available, the request is rejected.

TAPE LABELS

When the end of a tape reel is reached, a continuation reel is assigned with a label identical to the current tape except the reel number is updated. On an initial request for a tape reel, a label is written or checked before initiating the operation. A logical tape unit is treated as a single physical tape for rewind, backspace, and other tape handling functions. Tapes are disconnected and rewound at the end of each job and if specified, are unloaded and saved.

EXTERNAL INTERRUPT

If an interrupt address is specified, control is transferred to an interrupt subroutine at the end of an input/output operation. Console registers are stored and control is transferred to the interrupt subroutine with the status of the unit

supplied. At the end of the interrupt subroutine, the programmer returns control to SCOPE, which restores the registers and processes any other interrupts before returning control to the main program.

INTERNAL INTERRUPT

Certain internal conditions may be specified to interrupt program operation when they occur. These include arithmetic faults, bounds (memory limits) faults, clock interrupt, and manual intervention interrupts. When the condition occurs, SCOPE saves the registers, clears the interrupt, and transfers control to an interrupt subroutine supplied by the programmer. After processing in the interrupt subroutine is completed, control is transferred to SCOPE which processes any other interrupts, restores the registers, and returns control to the interrupted program.

SPECIAL REQUESTS

A running program may request SCOPE to position the system library tape at a record or directory on the tape. A running program may call upon the loader and if it is not in memory, SCOPE reads the loader from the system library. It is possible for the programmer to obtain or change the limits of memory by a special request. A request is also available for the running program to return control to SCOPE, causing normal termination of the program.

• LIBRARY

The SCOPE library contains SCOPE routines, assemblers, compilers, and other information required by the installation. It may include absolute binary and relocatable binary programs, binary and BCD data, subroutine directories and user defined information. A SCOPE library tape must begin with a binary record containing a bootstrap routine, a table of contents describing each record on the tape, and a label. Both the first and second record contain the SCOPE monitor system. The remainder of the tape contains routines added by the installation.

TABLE OF CONTENTS

The table of contents contains a list of the locations of the subroutine directories, end-of-file marks, subroutine records, tables, and data on the library tape. The table of contents may be listed on the standard output unit by using a SCOPE control statement.

DIRECTORY

A subroutine directory is a binary record that describes the subroutine records which follow it on the library tape. There may be any number of directories. The directory contains all the information required by the loader to define, load and link subroutines.

PREPARING AND EDITING

A new library tape may be prepared from source library tapes, subroutines or data on the standard input unit. The arrangement of records on the new library tape is determined by the order of control statements. Subroutine directories may be prepared and inserted on the tape as well as end-of-file marks. Records may be deleted or replaced and new records inserted to edit a library tape.

SCOPE PROGRAMMING 3

SCOPE processes jobs under directions supplied by the programmer and operator through control statements and program requests. Program requests are statements which may be included in assembly language (COMPASS) programs to perform operations such as input/output control, tape handling, internal interrupt, and equipment status checks.

CONTROL STATEMENTS

Control statements direct SCOPE in job processing.

JOB PROCESSING

The control statements identify the job and initiate compilation, assembly or execution of the object program. Control statements may assign a subprogram to a bank, establish logical equipment assignments, and describe overlays.

JOB,charge number, programmer identification, job time limit

A JOB statement signals job beginning, provides accounting information, identifies the programmer, and sets a processing time limit. It precedes all jobs.

BANK,(b_1), . . . , $name_i$, . . . , (b_2), . . . , $name_k$, . . .

b is a bank number, subprogram or subroutine entry point, or common block name

name is an entry point or common block name

This statement names a bank for each subprogram and common block or specifies that certain subprograms and common blocks be assigned the same bank.

EXECUTION

Library programs may be called to assemble or compile subprograms or programs. An object program on the standard input unit may be executed, an object subprogram may be loaded into memory from a logical unit and executed, or records may be transferred to a logical tape unit from the standard input unit.

program name(s_1, s_2) , $p_1, p_2, \ldots, p_n$

program name — is the name of the library program to be called

s — is a SCOPE parameter to correct or debug the library program

p — is a parameter interpreted by the library program

A program statement loads and executes library programs, such as FORTRAN, COMPASS, and COBOL.

FILE, logical unit

FILE END

A FILE statement transfers binary records following it on the standard input unit to another logical unit until a FILE END statement is encountered.

LOAD, logical unit

A LOAD statement loads relocatable binary subprograms into memory from a logical unit other than the standard input unit. (A binary object program on the standard input unit is loaded into memory unless a preceding control statement specifies otherwise. After the program is loaded, control returns to the next SCOPE control statement.)

RUN, time limit, print requests, recovery indicator, memory map indicator

A RUN statement transfers control to the object program in memory.

EQUIP , logical unit = declaration$_1$, declaration$_2$, . . . , declaration$_n$

declaration may concern hardware type, usage, logical unit equivalence, and label declarations

DEBUGGING AIDS

Corrections may be made to an object program at load time by octal correction cards. Existing instructions and data may be altered, additional instructions and data may be loaded into a program extension area.

SNAP, address, area to be dumped, dump format, frequency, identification

A SNAP is a periodic dump of specified areas in a running program.

TRACE, area containing jump instructions, area to be dumped, format of dump, frequency, identification

The TRACE statement produces a dump when jump instructions within a specified range of the program are executed.

OVERLAYS

A program may be divided into a main program and any number of overlays and segments.

MAIN, logical unit

MAIN defines the main portion of the program and the logical unit on which it is stored.

OVERLAY, logical unit, numerical identifier

OVERLAY causes SCOPE to create an overlay section of a program and to write it on a logical unit.

SEGMENT, logical unit, numerical identifier

SEGMENT directs SCOPE to create a segment section for an overlay and write it on a logical unit.

LIBRARY

The library preparation routine may be used to prepare a new library, edit an existing library, or list the table of contents of a library.

PREPARE, new tape name (source $name_1$, source $name_2$, . . . , source $name_n$)

PREPARE statement designates the name of the new library tape and specifies the names of source library tapes from which information will be taken.

EDIT (name of source tape)

Editing statements are used to delete, replace, or insert records on a library tape.

LIST (library name)

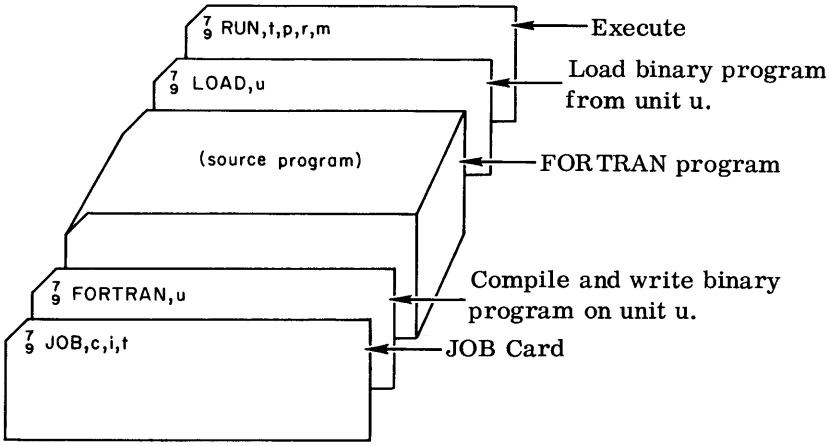
This statement lists the table of contents of a library on the standard output unit.

SAMPLE PROGRAMS

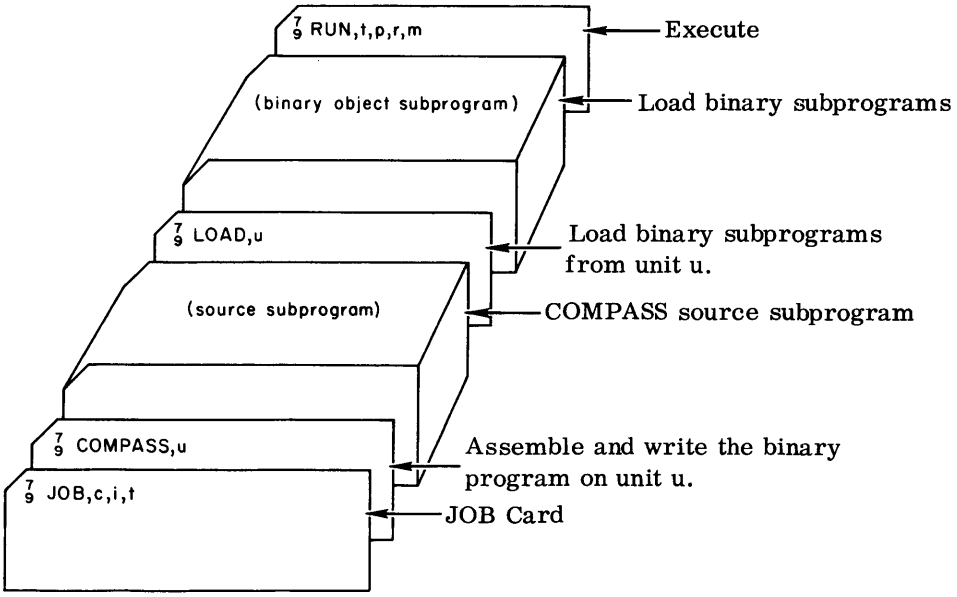
Sample programs demonstrate the arrangement of control statements and subprograms on the standard input unit.

COMPILE AND EXECUTE

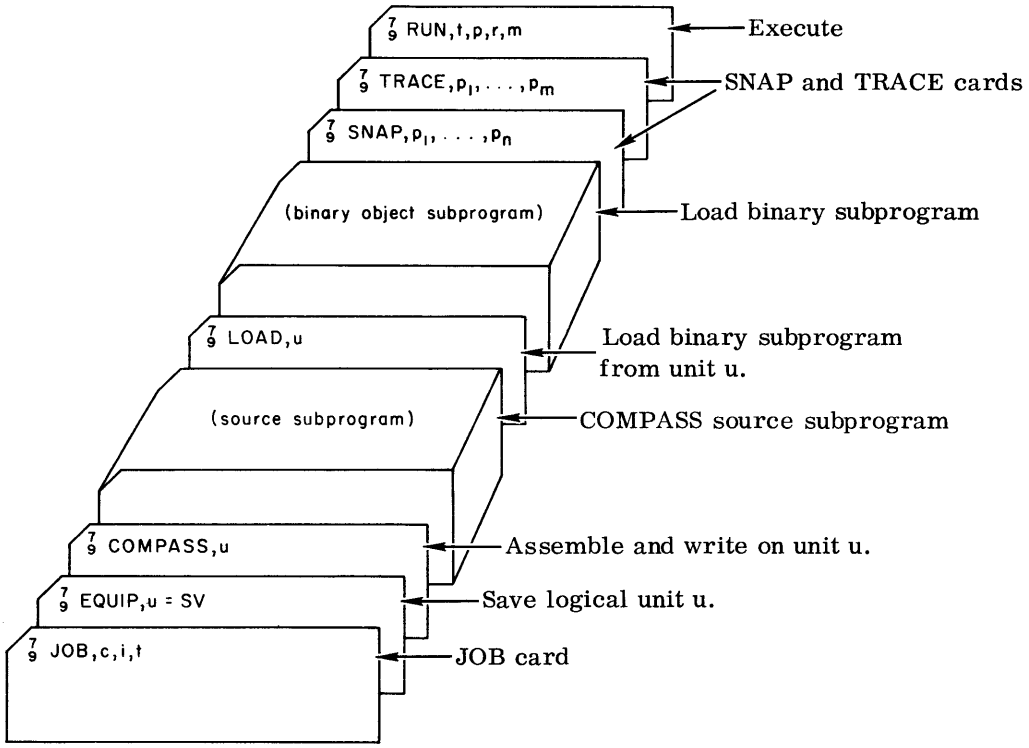
Assemble a FORTRAN program on unit u. Load and execute the program.



Assemble a COMPASS subprogram on unit u. Load with a binary subprogram and execute.

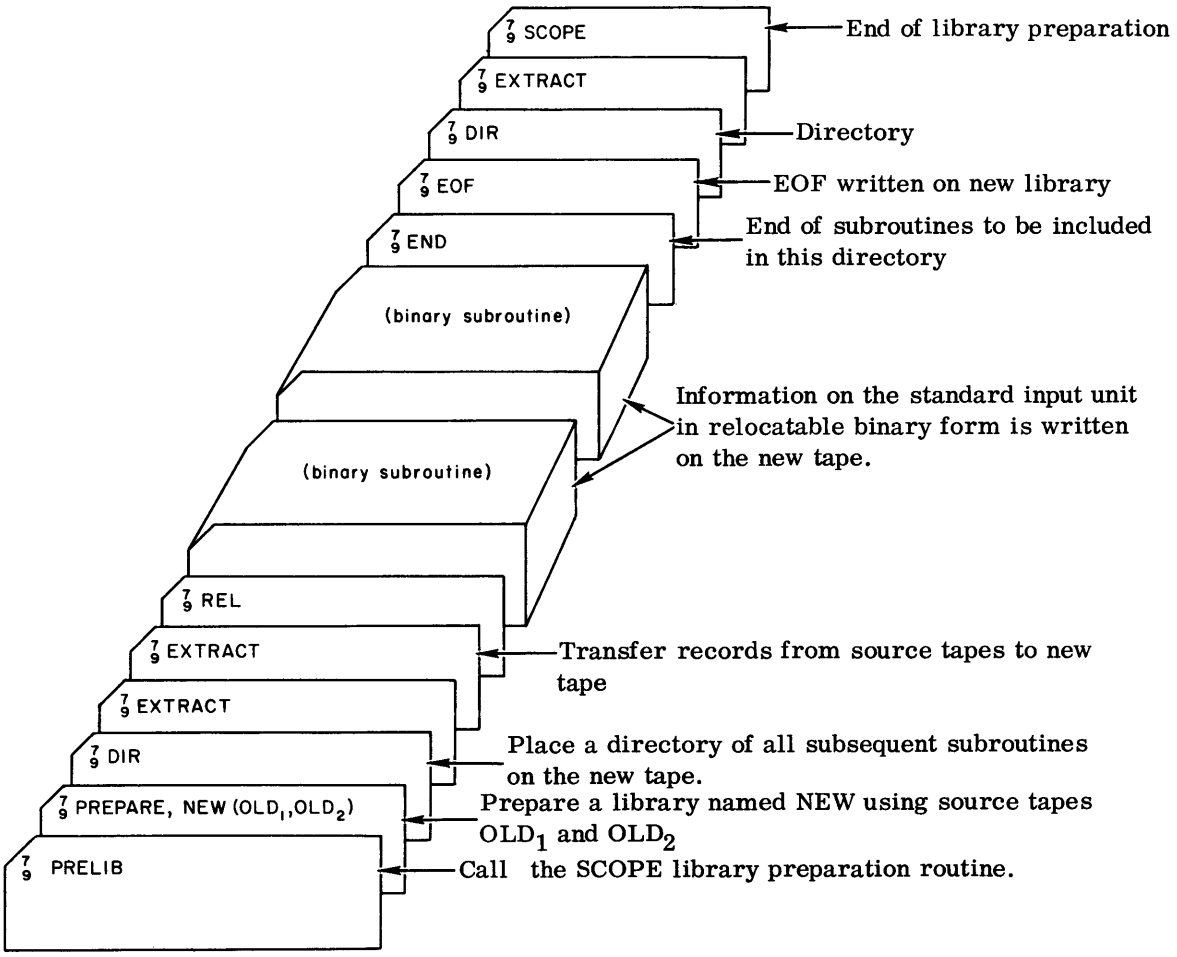


Include debugging aids in previous example and save unit u.



CREATE A LIBRARY TAPE

Create a library tape called NEW from two source library tapes called OLD_1 and OLD_2 and subroutines on the standard input unit.



PROGRAM REQUESTS

SCOPE controls the execution of input/output operations and the handling of internal interrupts. Requests are written in the form of COMPASS language macro instructions, which the COMPASS assembler translates into requests to SCOPE. SCOPE may execute the request immediately, stack the request temporarily and return control to the next instruction, or return control to a reject address if the request cannot be executed. When an input/output request is completed, SCOPE gives control to the programmer-defined interrupt subroutine. In the interrupt subroutine, the programmer can determine if the request was completed successfully and initiate remedial procedures if it was not. Requests may be given to specify tape labels, establish operating modes or to save tapes at job termination.

Internal interrupt conditions which may be selected to produce program interrupts include errors resulting from shift and arithmetic instructions, memory guard for program operation and data storage, and time limits for program execution.

Following are the programmer requests as they would appear in the COMPASS program.

READ or WRITE (logical unit, control word address, reject address, interrupt address)

The following tasks are performed.

Assign logical unit numbers to physical units

Select an available channel

Stack a request if a channel is unavailable

Respond to external interrupts

Initiate input/output operations

Locate and label a continuation tape when one is needed to complete an output operation

Reject a request if the unit is unavailable

REOT or WEOT (logical unit, control word address, reject address, interrupt address)

Read end-of-tape and Write end-of-tape permit:

Data Transmission beyond an end-of-tape mark

Establishment of a logical end-of-tape condition prior to reaching the physical end of a tape reel

STATUS (logical unit)

STATUS of any logical unit may be requested at any time. Status is always executed by SCOPE prior to entering an external interrupt subroutine. Status information includes logical unit, availability, reel number, control word, next control word address, and equipment status.

MODE (logical unit, reject address, usage, format)

usage is read/write, bypass, or read only

format is even or odd tape parity; high, medium or low density tape

A MODE request restricts the usage of a tape unit or selects a density or recording mode for the unit. If the unit is unavailable, control is returned to the reject address.

TAPE CONTROL REQUESTS

control name (logical unit, reject address, interrupt address)

Control Name	Function
BSPR	Backspace one record
BSPF	Backspace one file
REWIND	Rewind to load point
UNLOAD	Rewind and unload
SKIP	Skip to end-of-file
ERASE	Erase 6 inches of tape
MARKEF	Mark end-of-file

LABEL (logical unit, location of name, edition, reel)

The LABEL request describes a tape label and defines a reel of tape to be referenced by the next control function which moves the tape forward. For an input file, the label is checked. For an output file, the label is written.

SAVE (logical unit)

A SAVE request indicates that a tape is to be saved at job completion. All reels of the saved logical unit are rewound and unloaded; a SAVE message is written to the operator.

INTERNAL INTERRUPTS

SELECT (interrupt designator, interrupt address)

SELECT identifies the interrupt condition and location of a routine to be entered should that condition occur.

REMOVE (interrupt designator)

REMOVE releases the interrupt designated.

BOUND (lower bound, upper bound, reject address, interrupt address)

BOUND sets the limits of memory to be used. Referencing a word out of bounds causes interrupt.

UNBOUND This statement releases the last set of bounds and reinstates the preceding set of bounds.

Interrupt designators:

SHIFT	Shift Fault
DIVIDE	Divide Fault
EXOV	Exponent Overflow Fault
EXUN	Exponent Underflow Fault

OVER	Fixed Point Overflow Fault
M1604	1604 Mode Alert
TRACE	Trace Mode Alert
MANUAL	Manual Interrupt Alert
ADDR	Non-existent Address Fault
INST	Illegal Instruction Fault
OPER	Operand Parity Fault

CLOCK INTERRUPTS

LIMIT (duration, reject address, interrupt address)

LIMIT imposes a time restriction at the end of which the program is interrupted.

FREE This statement releases the most recently imposed time limit.

TIME This statement requests the remaining time until the next LIMIT interrupt and the time of day.

DATE This statement requests the calendar date. Time and date are returned in the A and Q registers.

OTHER REQUESTS

LIBRARY (library unit, reject address, location of record name, record number)

The LIBRARY request directs SCOPE to position the library at a named library record or at the directory which precedes the named library, or to indicate the number of records a tape has moved under programmer control.

LOADER This statement request calls the loader into memory and loads a program.

MEMORY The limits of available memory may be obtained or changed by means of the MEMORY request.

EXIT This request returns control to the monitor from a running program, causing normal termination of the program.

OPERATOR CONTROL STATEMENTS

The following control statements are used by the operator for job sequencing.

SEQUENCE, job sequence number

SEQUENCE assigns the job a sequence number. It is not required.

END SCOPE This statement signals SCOPE that the processing of the job stack is completed.

END REEL This statement terminates the current reel of the standard input unit and causes SCOPE to locate the next reel.

PAUSE, job sequence number

PAUSE signals SCOPE to halt processing of the standard input unit just prior to loading the specified job. Processing is halted until a complete message is received from the operator.

REPEAT This statement repositions the standard input unit at the beginning of the current job. A job in process is first completed.

NEXT, job sequence number + increment

NEXT positions the standard input unit at the beginning of the job specified. SCOPE completes the job it is processing when the command is given.

PHYSICAL EQUIPMENT DEFINITION

The AET statement obtains a listing of the contents of the Available Equipment Table, alters the status of a unit, and establishes or alters the entire table entry.

AET , entry ordinal in AET, { unit on which the entry is written | availability of the unit | value to replace the value in the entry }

OPERATING SCOPE 4

SCOPE minimizes operating responsibilities and simplifies operating procedures by maintaining accounting information, making memory dumps, and providing recovery information.

INITIALIZATION

The operator loads SCOPE from the library tape by autoload. Switches indicate the channel and equipment to be connected and the function to be executed. The first record loaded from the library contains a bootstrap routine which performs initialization and requests the time and date from the operator. The operator responds via the input comment unit. SCOPE locates the library tape and the standard input unit which contains the job stack. Any changes or requests concerning the physical equipment available to SCOPE may be entered via a control statement.

JOB SEQUENCE

Jobs are executed by SCOPE in the order in which they appear on the standard input unit. The operator may designate job sequence numbers and alter the order of execution by control statements.

SATELLITE

The Satellite processes job input and output for SCOPE and loads and lists tapes and punches cards. Since only SCOPE can make equipment assignments, the Satellite interrupts SCOPE processing to make requests. If a Satellite is assigned, its stacked requests are accepted before a new job is initiated.

COMMUNICATION

The operator interrupts SCOPE by pressing the manual interrupt button and enters a message via the input comment unit. The message may be either a control statement for SCOPE, or information for the program, in which case control is given to the programmer's manual interrupt subroutine.

RECOVERY

Before control is given to a loaded program, recovery information, including a record of SCOPE equipment tables, date, time, job sequence number and identification, and some of the loader tables, is saved on the system scratch unit. When a run terminates normally, the job is closed out and the next control statement is read.

If a program terminates abnormally, or if the SCOPE equipment tables do not tally, the entire job is terminated. An audit routine tests and reconstructs, if necessary, the critical equipment tables before the job is terminated. The recovery routine is called from the SCOPE library tape to make the core dump specified by the programmer. SCOPE is reloaded and proceeds with the next job.

ORGANIZATION OF SCOPE 5

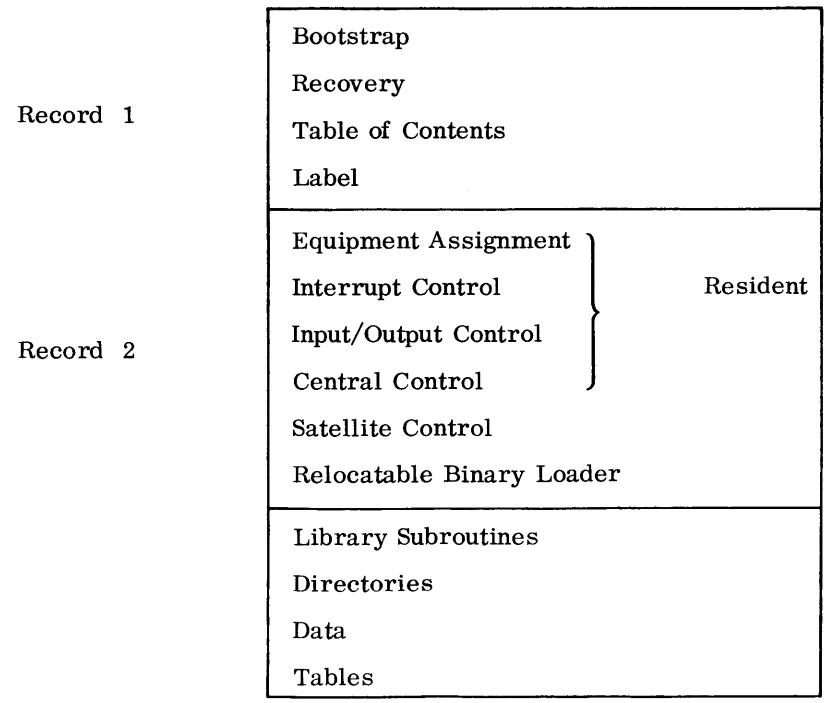
SCOPE LIBRARY

The SCOPE program contained on the SCOPE library tape is composed of routines and tables which monitor job processing. SCOPE is contained in the first two records of the SCOPE library tape as shown below. The remainder of the tape may contain subroutine directories, library subroutines, data, and tables. The first record contains the bootstrap which reads in SCOPE, recovery which terminates incorrect jobs and continues with the next job, the library table of contents, and label.

RESIDENT

Resident is that portion of SCOPE which remains in memory at run time. Located on the second record of the library tape, resident is composed of equipment assignment, interrupt control, input/output control, central control routine. Satellite control, and the relocatable binary loader are also in the second record. Resident and the bootstrap with recovery routines occupy the lowest region (00000_8) of bank zero. Other SCOPE programs are loaded next to resident.

Sample Library Tape



LOADER

Loads and links subprograms

Detects errors and provides diagnostics

Patches subprograms and labeled common

Assigns program extension areas

Selects banks

The central control routine of SCOPE transfers control to the loader for loading drivers and programs to be executed. The loader may also be called indirectly by a running program. When loading is completed, control is returned to the calling program. All errors detected are written out as diagnostics on the standard output unit. The loader provides for patching subprograms and labeled common and assigns a program extension area if necessary.

The loader loads into memory all subprograms, except those in the first two records of the library tape. It selects the bank, if there has been no other bank declaration. The bank selected, other than zero, is the bank having the amount of available memory into which the subprogram or block of common most tightly fits. Bank zero is assigned only when the other banks cannot provide space.

After loading and relocating all subprograms, the loader links them by entry points and external symbols. Linkage is made possible by elements, usually produced as output of COMPASS, FORTRAN and COBOL, read into loader tables. When the loading process is terminated, the loader searches the SCOPE library directory for subroutines corresponding to the names of all undefined symbols. It loads the library subroutines into memory, links them, and returns control to the calling program.

An object program is loaded into the highest locations of available memory, followed by labeled common and the program extension area. Numbered common in bank zero overlays the loader, beginning at the first loader location and extending upward in memory. In other banks, numbered common begins at the lowest location and extends upward in memory. The loader area is not cleared before control is transferred to a loaded program. If numbered common is not used, the loader will remain in memory.

EQUIPMENT TABLES

The SCOPE equipment assignment routine maintains tables concerning physical equipment and logical units. None of the tables is available to the programmer directly.

Available Equipment Table is a record of the equipment available to the system at a particular installation. It is part of SCOPE and may be altered temporarily in memory by an operator control statement.

Unit Status Table, established by SCOPE, reflects the current status of each physical unit.

Running Hardware Table contains information about each logical unit referenced in the program.

Tape Label Inventory contains cross references between tape labels, logical units, and physical equipment.

PROGRAMMING TRAINING CENTERS

3330 Hillview Ave.
Palo Alto, California

●

8100 - 34th Ave.
Minneapolis, Minnesota

●

11428 Rockville Pike
Rockville, Maryland

●

Room 223, Terminal Building
Newark Airport
Newark, New Jersey

●

5630 Arbor Vitae
Los Angeles 45, California

CONTROL DATA SALES OFFICES

ALBUQUERQUE • BEVERLY HILLS • BIRMINGHAM • BOSTON
CHICAGO • CLEVELAND • DALLAS • DAYTON
DENVER • DETROIT • HONOLULU • HOUSTON
HUNTSVILLE • ITHACA • KANSAS CITY, KAN. • LOS ALTOS • MINNEAPOLIS • NEWARK
NEW YORK CITY • OMAHA • ORLANDO • PALO ALTO • PHILADELPHIA • PITTSBURGH
SAN DIEGO • SAN FRANCISCO • SEATTLE • WASHINGTON, D.C.

INTERNATIONAL OFFICES

BAD HOMBURG, GERMANY • MELBOURNE, AUSTRALIA • LUCERNE, SWITZERLAND
STOCKHOLM, SWEDEN • ZURICH, SWITZERLAND • PARIS, FRANCE • OSLO, NORWAY

CONTROL DATA
CORPORATION

8100 34th AVENUE SOUTH, MINNEAPOLIS 20, MINNESOTA

Pub. No. 532