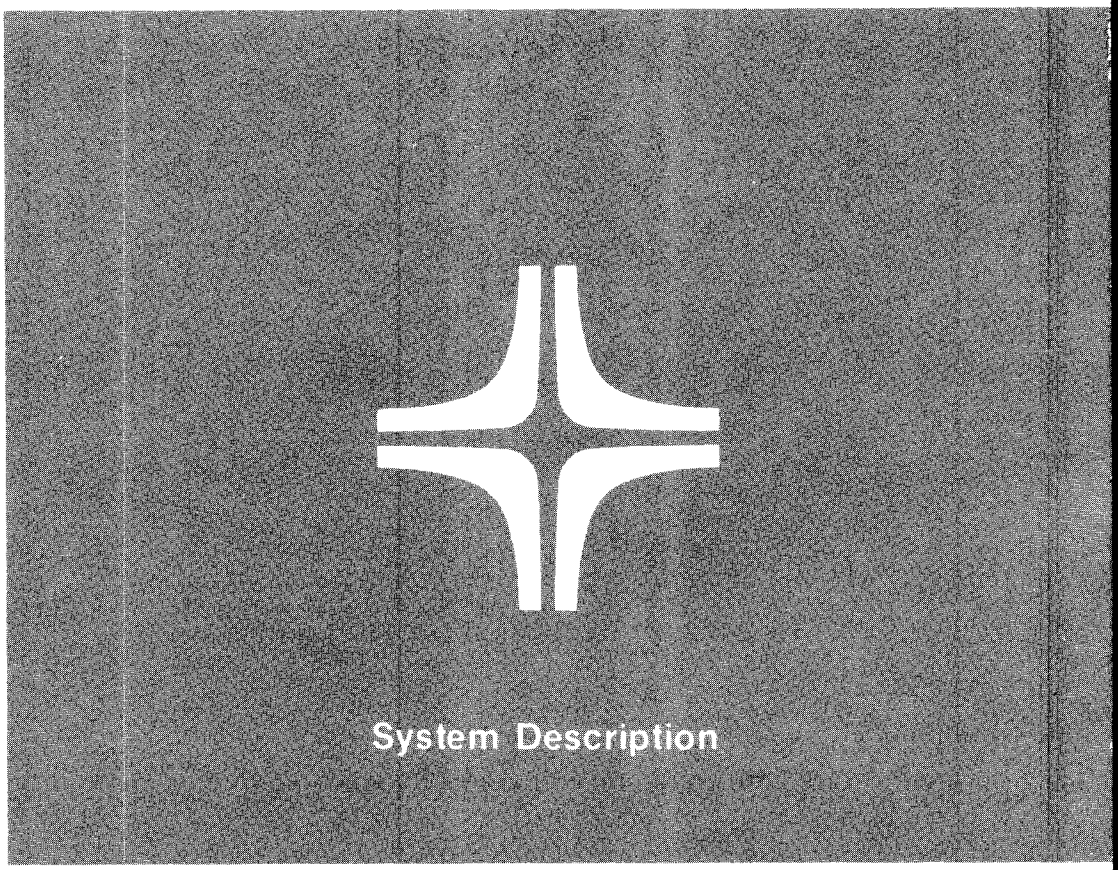
System 80 Model 8



SPERRY UNIVAC

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

Contents

FIGURES

TABLES

1. Introduction

Today, computer performance is being challenged by a variety of applications that have increased in scope and complexity. As a result, a significant number of established data processing systems cannot meet the requirements of these new applications because they lack the inherent processing capability. To meet this need, Sperry Univac has developed System 80 Model 8, an interactive data processing system for the established users who want to upgrade their data processing capabilities. It is designed for businesses that require a computer resource offering high performance at low cost and extended capabilities that encompass an unmatched range of computational requirements.

System 80 Model 8 features high-quality, compact, integrated hardware, complemented by an advanced, yet easy-to-use interactive software package: the SPERRY UNIVAC Operating System/3 (OS/3). Together, they make the computer a resource more widely useful to any business. Figure 1–1 depicts an installation of System 80 Model 8, showing the central processing complex with the system console and a variety of devices including manual and autoload diskettes, a high-speed printer, high-capacity, high-performance nonremovable disk units, and streaming magnetic tape drives.

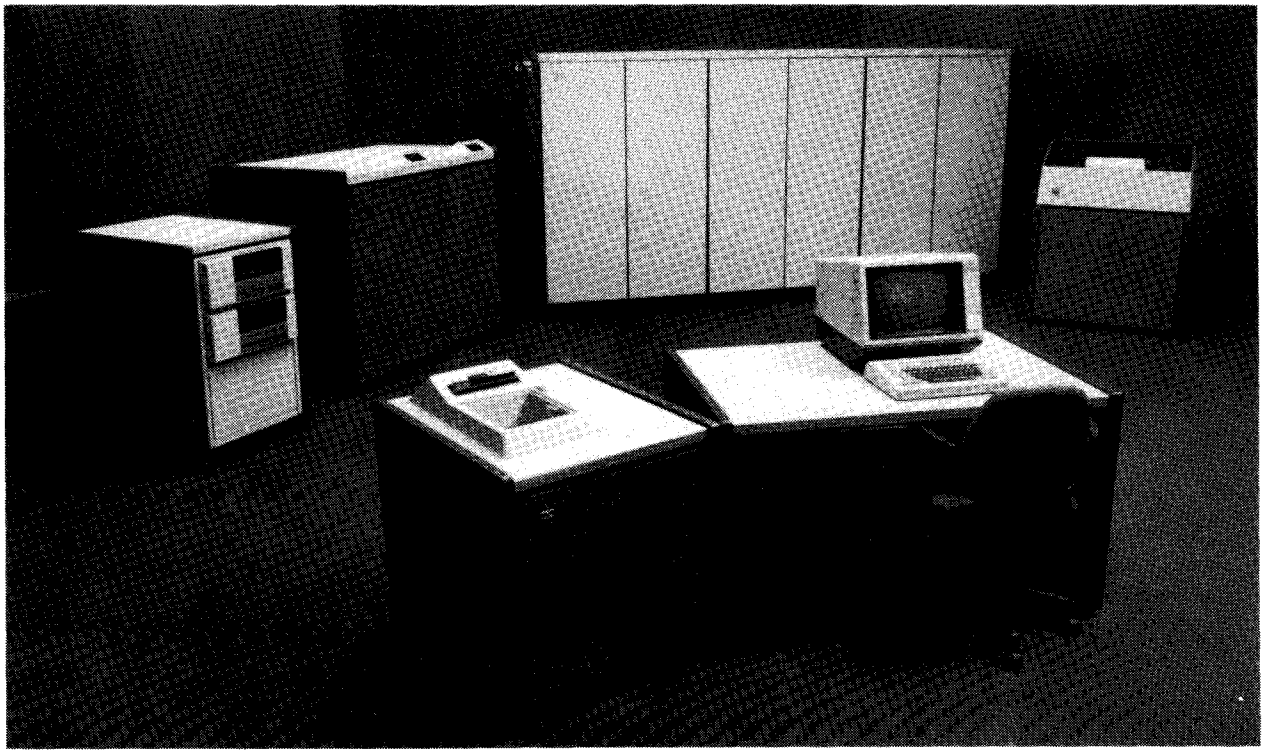


Figure 1–1. System 80 Model 8 Installation

Here are some of the advanced capabilities of System 80 Model 8 that increase its productivity:

- **Interactivity**

  System 80 Model 8 offers a number of interactive programming features, all initiated and controlled through the workstation. These features make the system easy to use, make it accessible to the novice or nonprogramming individual, and provide a number of programming tools that simplify the programming effort. A number of users can access the system simultaneously to perform interactive functions. These functions range from interactively generating source programs and job control streams to performing such functions as creating data files and transferring files between storage devices. Facilities are included to allow users to interactively enter variable data into jobs running in the system. In addition, the user can create screen format displays and interactive dialogs to simplify the inputting of data through the workstation and the displaying of output information.

  The interactive services allow for decentralization of system operation, job preparation and initiation through the workstation, interactive program development and execution, and the initiation of system utilities. They assist the novice user in using the system, and they increase the efficiency of experienced programmers.

- **Transaction Processing**

  Sperry Univac places information at the fingertips of the user through the SPERRY UNIVAC Information Management System (IMS). Employing a network of interactive workstations and data communications devices, personnel can instantly access and update user data files with no more knowledge than is needed to operate a keyboard. IMS offers a reliable, easy-to-use communications/data management package that adds a new dimension to computer applications through an inquiry/response capability in a real-time environment. Despite ease of use in the IMS system, security is paramount: all files are protected from unauthorized access, and extensive integrated recovery facilities are available.

  User support for IMS is in the form of applications programs called UNIQUE, which are supplied by Sperry Univac and require no programming effort on the part of the user. In addition, customized applications programs can be developed by using COBOL, RPG II, or the basic assembly language (BAL).

- **Multijobbing**

  Multijobbing increases throughput: the actual productivity of a data processing system as measured by the amount of useful computing work done per minute or hour. Multijobbing increases system productivity by interleaving the execution of job steps from more than one job. When any job step is waiting for an external event (I/O request) to occur before processing can continue, another job is given control. Priorities and available resources determine which job step is given control by the data processing system at any point in time. Up to 48 jobs can be processed concurrently by System 80 Model 8.

- **Advanced Hardware Technology**

  System 80 Model 8 incorporates a number of advanced hardware features that increase system efficiency and enhance user productivity. Among these features are:

  - Large-scale integrated circuits that increase system speed and reliability and thus increase the processing power of the system.

  - One megabyte direct access diskettes that are well suited as storage media for library and data files.

- High-density, high-capacity removable and nonremovable disk devices offer the user reliable and greatly expandable mass storage facilities. A fixed-head option is available for some nonremovable disk devices, offering a significant increase in access speeds and additional storage capacity.

- Autoload diskette drives, allowing the system to cycle up to 20 diskettes without user intervention.

- High-speed printers offering flexibility in character sets and line length selections, coupled with ease of operation, provide a maximum system printing capacity equal to any need.

- High-density, extremely reliable tape drives, including streaming tape drives, make SPERRY UNIVAC tape systems viable mass storage devices for all users.

■ Communications

Data communications use commercially available communications facilities to link a central processing site with remote sites to accomplish a variety of data processing applications. In meeting today's rising demand for this type of service, the System 80 Model 8 data processing system includes a sophisticated, yet easy-to-implement communications package – the integrated communications access method (ICAM) terminal support facility. All users, whether they are investigating communications for the first time or are upgrading their present capabilities, will find ICAM an indispensable tool. It is an integrated system capable of supporting several levels of communications processing. ICAM can be tailored to fit the user's needs, the type of service desired, and the installation configuration.

■ Distributed Data Processing

System 80 Model 8 can be easily incorporated into a distributed data processing network. Software is supplied that provides the interface required to match the systems to one another so that each system can be used to initiate a job at another location. Additionally, data files can be transferred between systems. Through distributed processing, programs running on one system can access files residing on another (remote system) and can even access other programs as files. Further, the information management system (IMS) we spoke of earlier can operate between systems, enabling the user to distribute transaction processing between OS/3 systems, or throughout an entire network. The establishment of a distributed data network allows all systems included in the network to share the processing load of the entire organization.

■ Data Base Management

The SPERRY UNIVAC data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs provide for the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of data bases are derived from the CODASYL data base specifications. Batch application programs and communications application programs may access a data base. The information management system may also access data bases created through DMS.

■ Conversion

In the purchase of any new system, the conversion of existing programs and data files is always uppermost in the customer's mind. Sperry Univac provides utility routines to convert data files generated for the SPERRY UNIVAC 90/25, 30, 40 systems and 9200/9300 and 9400/9480 systems; the IBM System/3 and System 32/34; and Honeywell Series 100, Series 200/2000, and Series 60, Level 62 and 64 into data files that are suitable for use on the System 80 data processing system. Also, information is provided on how to convert your present programs to System 80 formats.

- **Peripheral Compatibility**

  Users of SPERRY UNIVAC 90/25, 90/30, and 90/40 data processing systems who upgrade to System 80 Model 8 can connect many of their present peripheral devices to Model 8. The peripherals that may be connected include disk devices, tape drives, card readers, and printers. For further information on which peripherals may be connected to System 80 Model 8, refer to 4.3.

The System 80 interactive data processing system is an advanced and powerful computing system – a system that is both easy to use and specifically designed to meet the needs of today's business world. The system combines the latest in hardware technology with the proven OS/3 operating system, now enhanced by the inclusion of integrated, interactive software. The system offers a variety of programming languages, communications and data base facilities, interactive communications with the system, and interactive programming aids. The interactive features, coupled with the simplified operating system, make this system easy to use, simplifying information entry and retrieval to such an extent that even inexperienced and noncomputing professionals can access the system and achieve useful results.

System 80 Model 8 can grow with an expanding business. System 80 Model 8 has been designed to make expansion as easy as possible, both in the addition of peripheral devices and additional main storage. By growing easily with today's businesses, System 80 Model 8 can continue to meet new data processing challenges for years to come.

System 80 Model 8 sets a new standard in ease-of-use computing. It provides a powerful, versatile computing environment to meet even the most complex programming needs. The system incorporates the latest developments in both hardware and software technology and offers a number of advanced design features that make it unique among today's data processing systems.

2. Advanced Capabilities of System 80 Model 8

2.1. SIGNIFICANT FEATURES

This section highlights the features of System 80 Model 8 that make it unique as a total business data processing system: a powerful and flexible system that meets the needs of all users. Among these features are:

- The full range of interactive services
- Enhanced programming languages
- Communications and data management facilities, including distributed data processing
- Application packages for users with specific software requirements
- Conversion aids for those migrating from other systems

2.2. INTERACTIVE SERVICES

To meet the growing need to obtain immediate information and results from a data processing system, System 80 Model 8 offers interactive services. These services:

- Provide greater system usage and productivity through faster response times and enhanced data entry and retrieval methods
- Allow every member of an organization to access the system through easy-to-use facilities while maintaining a high degree of security through protective passwords and user identification features
- Simplify the entire programming effort by providing interactive communications to the system control software and the use of dialog and screen formats for control information entry

The interactive services include:

- Workstations
- General and language editors
- Screen format services
- Dialog processing services
- System dialogs

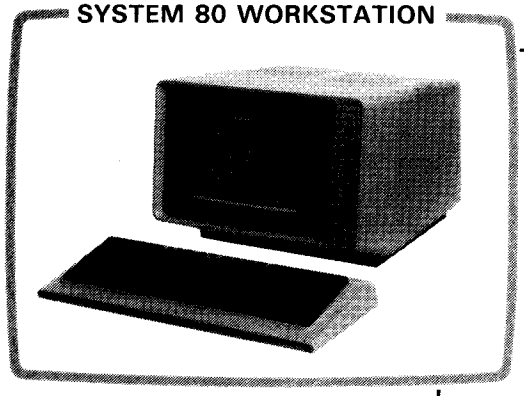
2.2.1. Workstation

In the interactive OS/3 programming environment, the workstation is the user's primary means of communicating with the system. A set of interactive commands can be issued directly from the workstation to:

- direct the operation of the system or a system component;
- query the system for specific information;
- direct the execution of a job; and
- interactively create and modify source programs, data files, job control streams, etc.

The workstation can also be used as an input/output device dedicated to a job or system component.

The workstation consists of a keyboard used for inputting entries and a video screen used to display the input entries and system responses.



VIDEO SCREEN

Displays:

- Application data
- System responses

KEYBOARD

Inputs entries through:

- Standard alphabetic and numeric keys
- Cursor control
- Function keys

The cursor control pad is used to control the positioning of the screen cursor. The function keys, when pressed, cause the system to perform a specific function. Some keys perform the same function at all times, some functions vary according to the system component that the workstation is in communication with, and others can be programmed by the user to perform a particular function.

To initiate workstation activities, the user must enter a LOGON message with accompanying user identifier and password. Once logged on, the user can begin any activity he chooses. The type of activities performed fall into two categories: control activities and data input/output activities. The user has commands available to control the initiation and execution of jobs and to initiate the various interactive services, such as the job control dialog or the editor.

The workstation is also used for input and output to system and user programs. It can be assigned to an IMS network to initiate and respond to IMS transactions. Or it can be assigned to a user job, in which case it is used as an input/output device for the job.

The workstation can also be used as a program development tool using the various interactive program development facilities, such as Editor, RPG II Editor, COBOL Editor, ESCORT, and BASIC.

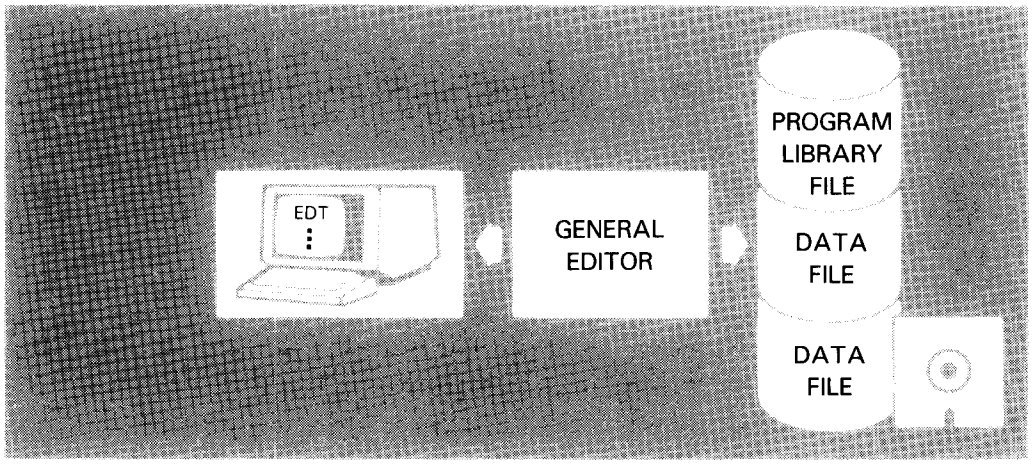
2.2.2. Editors

The OS/3 editors are easy-to-use, interactive, integrated components of OS/3 that help the user create and edit job control streams, data, and program files. The general editor features a comprehensive command set for creating and maintaining computer-based files on disks or diskettes. The language editors are specialized subsystems of the general editor and are tailored to handle the unique problems of creating and updating source language programs. The language editors can be called at any time during a general editing session. There are language editors available for the COBOL and RPG II programming languages.

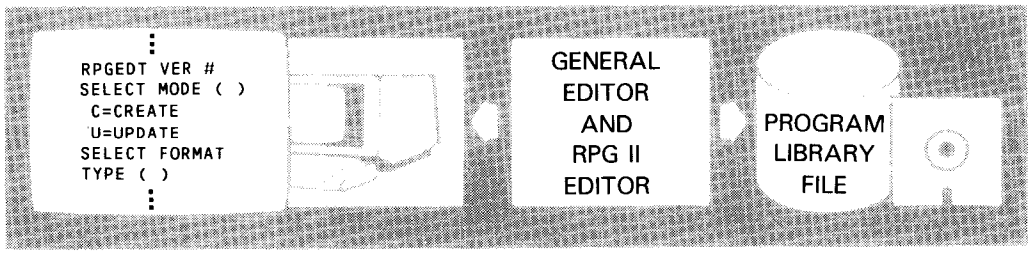
The OS/3 general editor has another specialized subsystem to aid in program development, the Error File Processor (EFP). EFP reads the error file produced when a program is compiled and displays the listing of errors, together with the appropriate lines of source code, on the workstation screen. Armed with a display of errors and the source lines containing the errors, the programmer can then quickly and easily correct those errors by using the general editor. Thus, a programmer can write, compile, and debug a program without having to wait for printouts, or even having to leave the workstation.

The general editor and the language editors are activated with a simple command from the workstation.

GENERAL EDITOR
IS USED TO CREATE AND EDIT DATA AND PROGRAM FILES



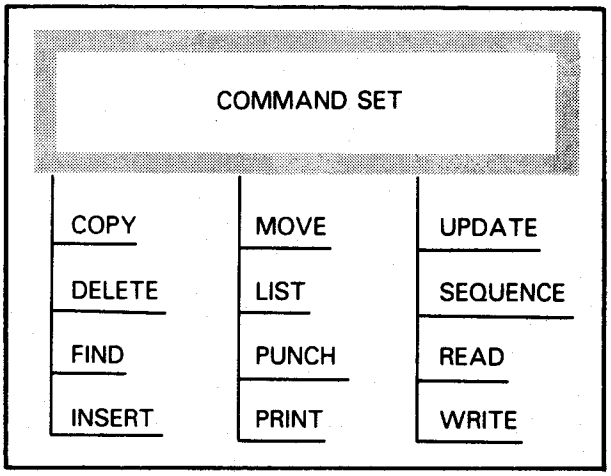
LANGUAGE EDITORS
ARE SPECIALIZED SUBSYSTEMS OF THE GENERAL EDITOR DESIGNED TO HELP CREATE AND EDIT SOURCE LANGUAGE PROGRAMS



The files created with the general editor and the language editors are displayed at the workstation screen as they are being created – making it easy to spot and correct errors as they occur. The same is true during an editing session – the files being edited are displayed at the workstation screen. When an editing command is keyed in, the user can see its effect on the file immediately.

The versatility of the general editor can best be described by examining the features it offers: commands, command modifiers, and procedures.

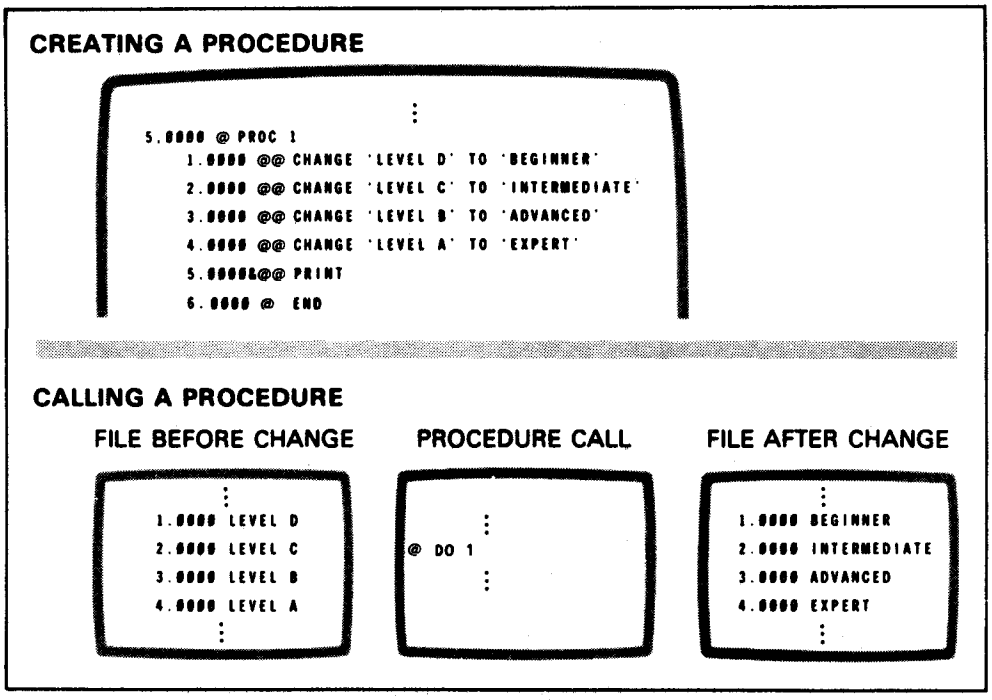
The general editor provides an easy-to-learn interactive command language. The command set calls these functions:



This is only a partial list of the functions available through simple, one-word commands.

The power of the general editor commands is enhanced even further by command modifiers that can be appended to each command. The general editor can be told to change all ABC character strings in a file to XYZ character strings, for example. There are many command modifiers available, but they all serve the same function – they tailor the general editor commands to the user's specific needs.

The general editor can also be used to create and call procedures. A procedure is simply a series of commands that perform a specific function. Once created, a procedure can be called with a simple command at the workstation keyboard. The procedures created with the general editor are stored in the editor work-space file.



The general editor operates in two modes, line mode and screen mode. In line mode, the user enters source code or data one line at a time. The examples up to this point have shown the editor operating in this mode. In screen mode, however, the editor allows the entry of up to 14 lines of source code or data at one time. This can save time when entering large blocks of data.

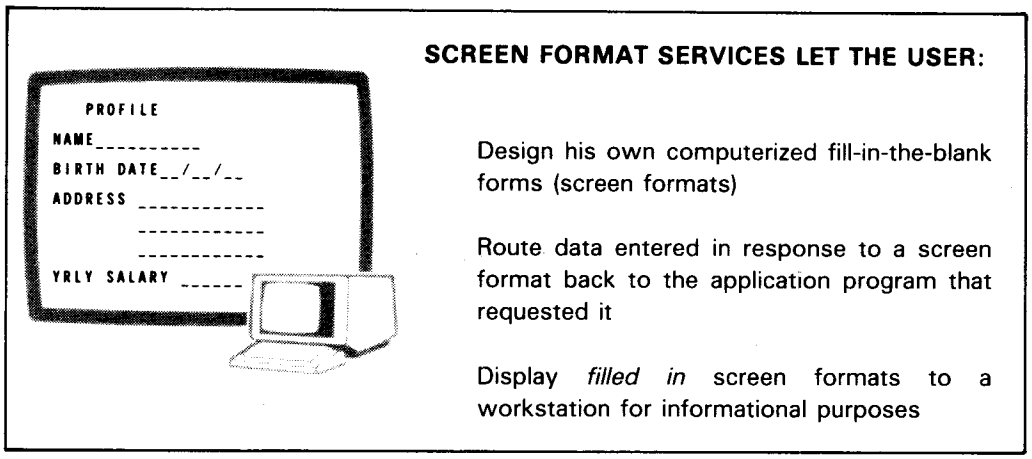
Screen mode is also useful for entering source code, in that the editor provides formatted screens that enable the programmer to enter source code faster. Screen formats are supplied for COBOL, FORTRAN, and RPG II. In addition, the editor provides a free-form screen that can be designed to accept uniform data. Screen mode can be advantageous for programmers who wish to enter source code quickly and do not need the syntax checking provided by the language editors.

The general editor also provides file protection. When editing a program or data file, a copy of that file is moved to the general editor work space file. The original file is left unchanged as a backup file and is not deleted until requested to do so.

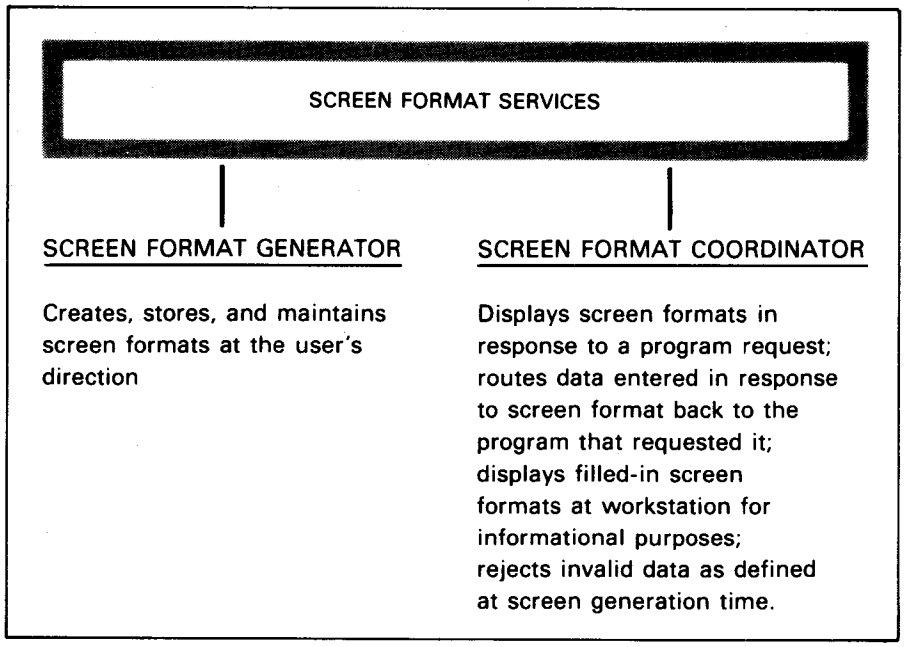
To summarize, the general editor and language editors of OS/3 are powerful interactive tools that help the user create and maintain all his computer-based files.

2.2.3. Screen Format Services

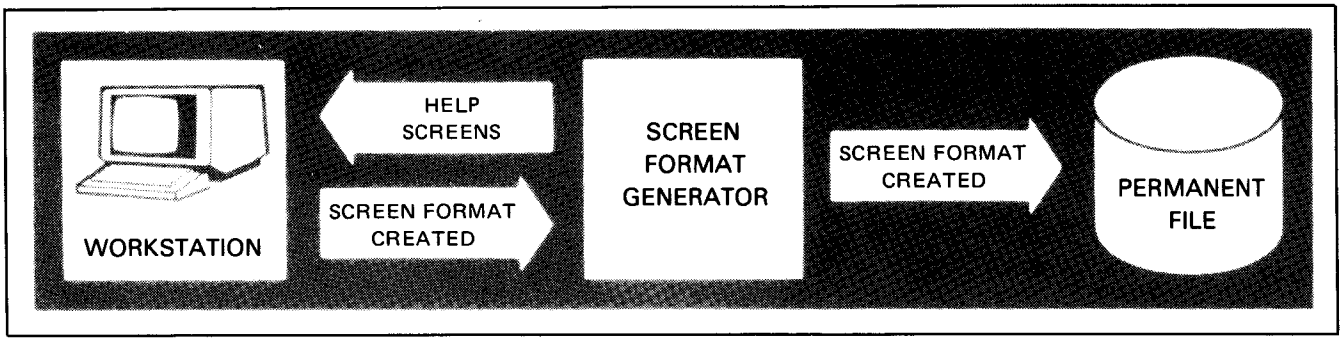
OS/3 screen format services simplify and standardize the process of entering variable data into a computer system.



Screen format services are comprised of two software components – the screen format generator, which helps the user create screen formats, and the screen format coordinator, which manages screen formats once they are created.



The screen format generator is activated by a simple command at the workstation. If the user needs assistance to create a screen format, it prompts him with HELP screens that explain the creation process. In addition, the screen format being created is displayed at the same time, making it easy to lay out a screen format exactly as desired. The screen format generator solicits information from the user about the variable data fields of the screen format, including how they should be displayed, when they should be displayed, and whether they are to be used for input, output, or both. Once the screen format is created, the screen format generator automatically stores it in a permanent library file.



The screen format coordinator is activated in response to a program request. It fetches the appropriate screen formats from the permanent file and displays them – either with blank fields for input or with the variable data filled in. The variable data is stored with the program. If the screen format called by a program was created as both an input and output screen format, the screen format coordinator handles the display of the screen format and variable data, accepts new variable data from the workstation user, and routes that data back to the application program that called the screen format coordinator.

SCREEN FORMAT COORDINATOR
DISPLAYS INPUT SCREEN FORMATS
AND THE WORKSTATION USER SUPPLIES THE VARIABLE DATA

```
INVNTRY
ITEM ____
PRICE ____
QTY ____
REORDER _____
```

Variable data fields are underscored.

```
INVNTRY
ITEM 1234
PRICE 0035
QTY 0010
REORDER 10/15
```

Workstation user supplies variable data that is routed by the screen format coordinator to an application program.

SCREEN FORMAT COORDINATOR
DISPLAYS OUTPUT SCREEN FORMATS FOR INFORMATION PURPOSES

```
INVNTRY
ITEM 1234
PRICE 0035
QTY 0010
REORDER 10/15
```

The screen format coordinator fetches the screen format from the permanent file and adds the variable data from the application program.

SCREEN FORMAT COORDINATOR
DISPLAYS INPUT/OUTPUT SCREEN FORMATS

```
INVNTRY
ITEM 1234
PRICE 0035
QTY 0010
REORDER 10/15
```

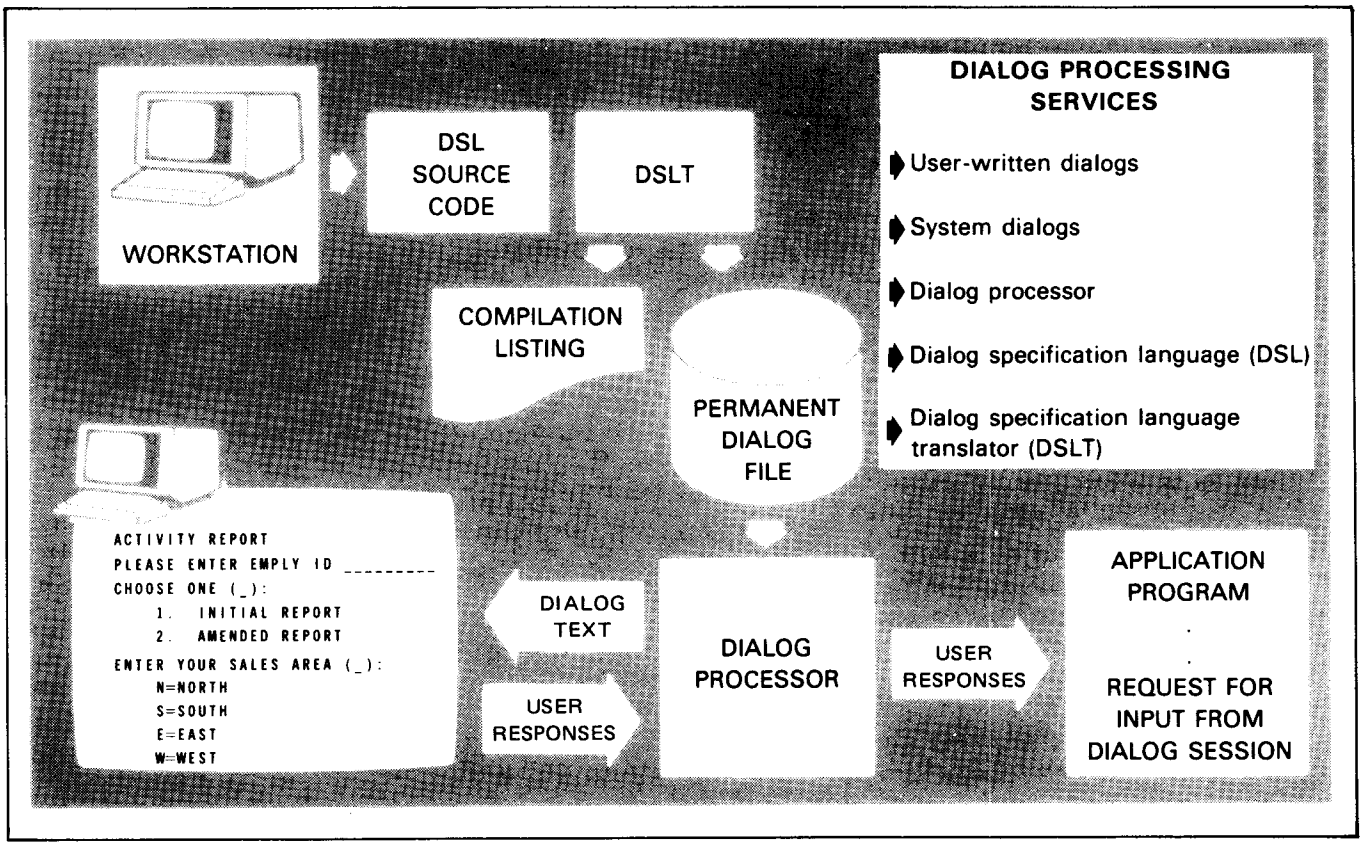
The screen format coordinator displays an output screen format and...

```
INVNTRY
ITEM 1234
PRICE 0035
QTY 1000
REORDER 12/15
```

...the workstation user supplies new variable data.

2.2.4. Dialog Processing Services

OS/3 dialog processing services make it easier to communicate with the programs and procedures in operation by providing an interactive, conversational link to the computer system.



System dialogs are interactive dialogs available from Sperry Univac that guide users through the processes of system generation, building job control streams, and initializing data utilities routines.

Dialog specification language (DSL) is a programming language designed by Sperry Univac specifically for the creation of interactive dialogs. The user can use DSL to write interactive dialogs. DSL source code is submitted to the translator (DSLT), which compiles it to produce the desired dialog and stores the dialog in a permanent file. The DSLT also produces a compilation listing.

Both user-written and system dialogs are managed by the dialog processor, which displays dialog text at a workstation screen, accepts user responses to the dialog, and routes that input to the application program that the dialog is designed to complement. In addition, the dialog processor produces a printed summary of each dialog session and (if requested) an audit file that contains a complete record of dialog responses. Besides providing a record of the dialog sessions, the printed summary and audit file can be used as guides to changing dialog responses in a subsequent session.

The application programs that call for input from a dialog session can be written in any language.

2.2.5. System Dialogs

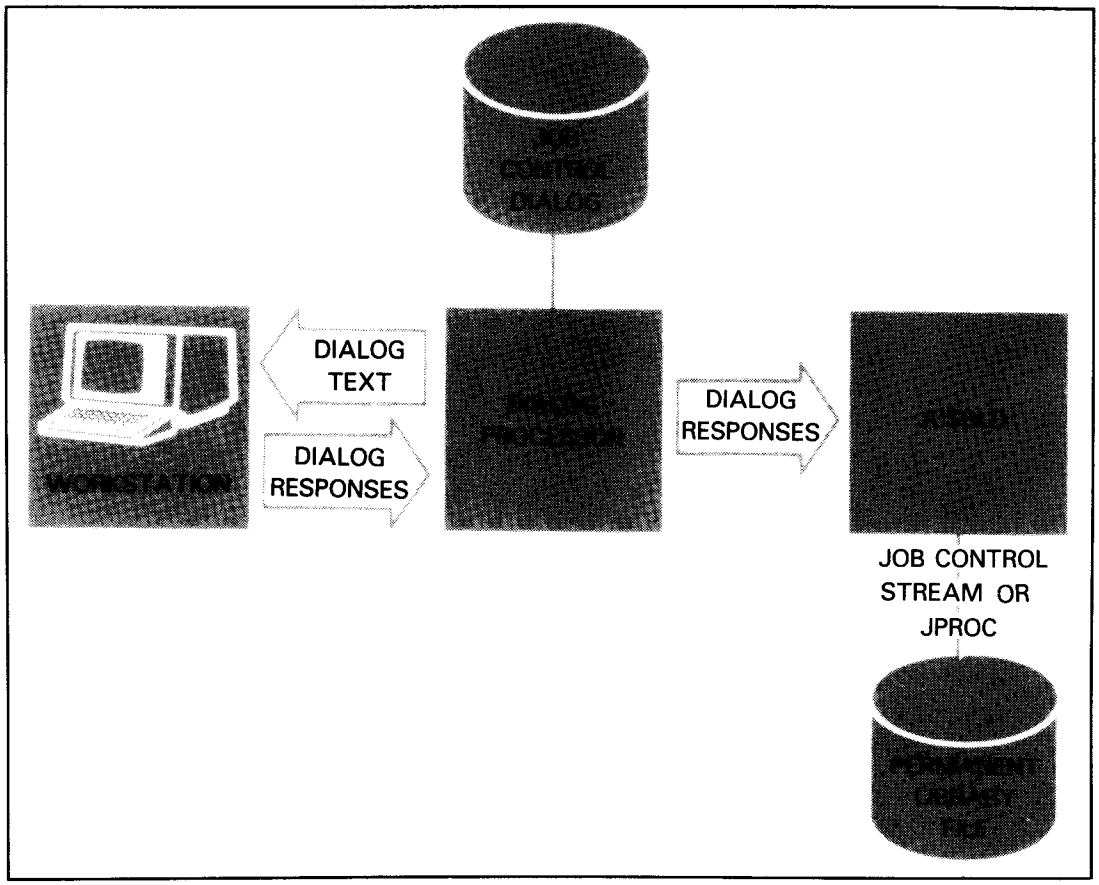
System dialogs, as previously noted, provide an interactive, conversational interface to the processes of system generation, building job control streams and job control procedures (jprocs), and initializing data utilities routines. In addition, they are designed to provide the user with tutorial assistance, when needed, in the form of HELP screens. HELP screens describe system concepts and specific dialog choices that need explaining in order to make a valid choice. System dialogs that use the dialog processor also offer an audit facility that allows the changing of dialog responses in a subsequent session.

2.2.5.1. Job Control Dialog

The job control dialog guides the user step by step through the process of building a valid job control stream or jproc. The statements and system jprocs that make up a job control stream are presented in the form of menu items, and the user is asked to choose those that are desired.

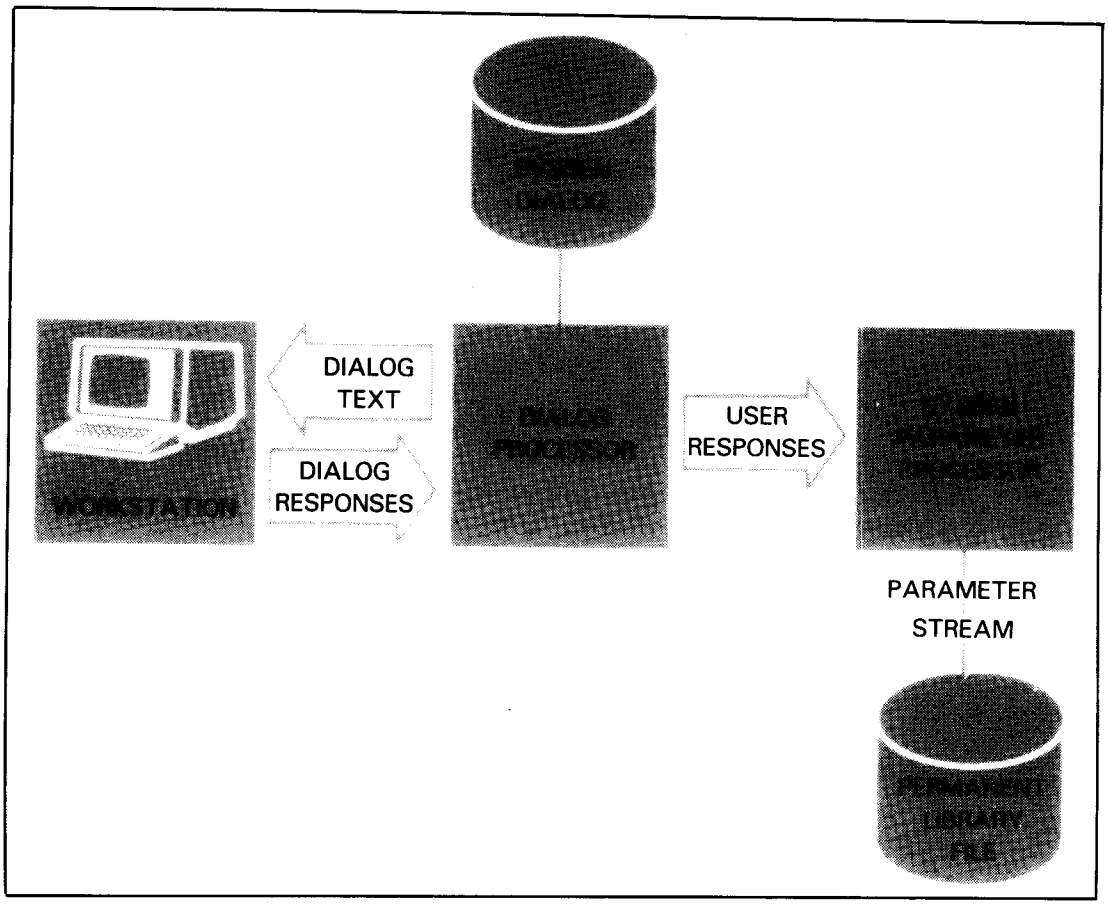
HELP screens can be displayed upon request whenever a dialog choice or a particular job control statement or system jproc explanation is needed. A more experienced job control user can use the dialog to build a control stream without using HELP screens – thus building the control stream quickly while still being constrained by the dialog to build a syntactically correct control stream. Novice users, on the other hand, can learn about job control as they are actually building a valid job control stream.

The job control dialog is initiated with a simple workstation command that activates the system program JC$BLD. JC$BLD activates the dialog processor, which then manages the dialog session with the workstation user. The workstation user's responses to the job control dialog are routed back to JC$BLD, which uses them to create a control stream or jproc and then stores the control stream or jproc in a permanent library file for future use. The user control stream can then be initiated with a run command from the workstation. Jprocs, of course, are initiated when the control stream that references them is run.



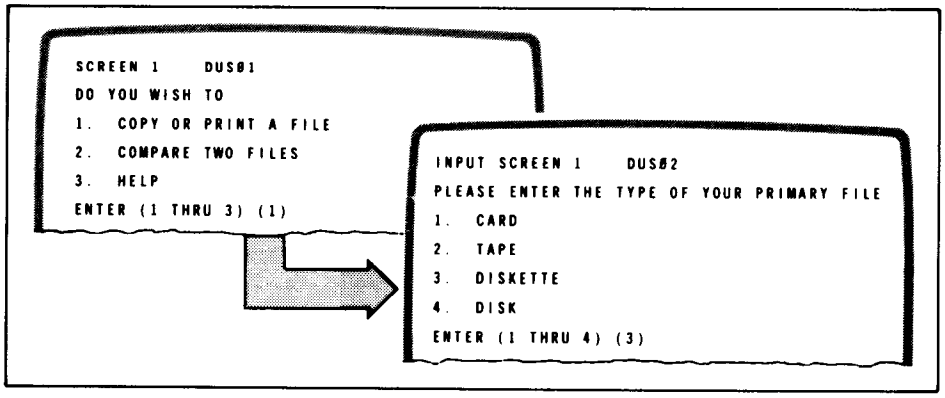
2.2.5.2. System Generation Dialog

The system generation (SYSGEN) dialog operates in the same manner as the job control dialog. It solicits SYSGEN parametric values that configure the operating system best suited for the user's processing needs. The SYSGEN dialog, like the job control dialog, is initiated with a workstation command. Responses to the dialog are routed to the SYSGEN parameter processor and then stored in a permanent library file.



2.2.5.3. Data Utilities Dialog

The data utilities dialog operates in a manner similar to the job control and SYSGEN dialogs, except that it does not use the dialog processor. It offers the same advantages, however. A simple workstation command activates the data utilities dialog, which then solicits information about the data utilities functions needed for use. HELP screens are available that explain the dialog choices. When the dialog is completed, the functions requested are automatically initialized.



2.3. ENHANCED PROGRAMMING LANGUAGES

System 80 Model 8 features a full range of versatile problem-oriented programming languages. These languages meet established standards and offer extended features unique to System 80. The languages provide the capability for programs to access workstations for dynamic data entry and display during execution. Language editors and the Error File Processor are available to support the interactive development of programs at the workstation. The implementation of a consolidated data management system offering a uniform file type specification for all devices offers a significant degree of device independence to user programs. These are but a few of the enhanced features available to the programmer using System 80 Model 8.

Two of the programming languages supported, BASIC and ESCORT, are fully interactive. Source statements for these languages can be entered directly through the workstation, with error messages provided for the isolation of syntax errors. Users can quickly and easily build programs with these languages.

ENHANCED PROGRAMMING LANGUAGES

ESCORT

ESCORT is an interactive language especially suited for developing file processing programs. This language is among the easiest languages to learn and use, and even the most inexperienced programmers can quickly and easily begin generating useful and productive programs. ESCORT offers extensive prompting at the workstation; users can get quick assistance by asking the system for help.

COBOL

The Common Business Oriented Language (COBOL) is a general purpose, yet powerful language designed with the business community in mind. COBOL offers a full range of programming services to meet common commercial needs, such as payroll, accounting, inventory, and personnel management. The COBOL supplied by Sperry Univac also offers a powerful sorting facility allowing users to sort data files on multiple keys. In addition, COBOL users can readily interface with the IMS and the DMS. Programmers can develop COBOL programs through the workstation by using the OS/3 COBOL Editor. The COBOL editor allows the programmer to enter source program statements through the workstation and provides syntax checking to aid in creating error-free programs.

FORTRAN IV

FORmula TRANslation (FORTRAN IV) is a powerful computing language that meets the problem-solving needs of the scientific community. Businesses that require a great deal of mathematical computing power should find FORTRAN IV an invaluable programming tool.

RPG II

The Report Program Generator II (RPG II) is a high-level language suitable for producing reports and maintaining files in a business environment. The basic logical flow of a program is provided to the user by the RPG II cycle. The user specifies the details of processing by using the input format, output format, calculations, and other specifications. Entering RPG II programs through the workstation has been simplified by the RPG II Editor; source program statements may be entered on the workstation by using formatted displays. RPG II on System 80 Model 8 also offers Auto Report, which accepts simplified RPG II source statements and creates a complete RPG II program.

BASIC

The Beginner's All-purpose Symbolic Instruction Code (BASIC) is a simple, straightforward programming language designed for general business applications. The salient feature of BASIC is its ease of use. BASIC is an interactive language that uses simple, English-language statements.

BASIC ASSEMBLY LANGUAGE

The Basic Assembly Language (BAL) is a flexible, machine-level language that offers a variety of features that allow the user to specify the most complex algorithms as a series of mnemonic symbols and command directives. Each mnemonic represents a single machine action. BAL is a complicated programming language that can be cumbersome to the inexperienced user, but can be of great value to the user with particular programming requirements.

2.4. INFORMATION MANAGEMENT SYSTEM

With the increased use of computers to store day-to-day business information came the need for users to be able to access and update that information quickly and easily. Sperry Univac meets that need with the information management system (IMS). IMS is a transaction processing system; all activities are initiated by an inquiry, and each inquiry results in a system response. IMS allows nonprogramming personnel to interactively access and manipulate large-scale data files through a network of workstations and terminal devices. Through IMS, any authorized member of an organization can use a terminal in the IMS network to instantly obtain information in the data files and to make changes to the data file. IMS interfaces with OS/3 distributed data processing software to enable users to use IMS in a network of remotely-located data processing systems. Users can access and update information on all the data processing systems within such a network. IMS is easy to install and specifically designed for use by computer personnel at all levels. It is modular for simplicity of use and maintenance, and configurable to fit an installation's requirements.

IMS FEATURES

INQUIRY/UPDATE LANGUAGE

IMS supplies an easy-to-use inquiry/update language called UNIQUE for general purpose file processing. Each UNIQUE command initiates a particular file processing activity. The range of commands includes record deleting, modification, and listing the contents of a file. Users with more specific file processing demands can generate and implement their own file processing programs. These programs, referred to as action programs, interpret terminal operator requests, perform the requested function, and issue the appropriate response to the initiating terminal. User action programs can be written in COBOL, RPG II, or the basic assembly language (BAL). IMS handles the scheduling of, and resource allocation for, action programs initiated by terminal operator requests.

DATA FILES

Any type of user disk data file can be accessed by IMS, including files generated by the data base management system (DMS). In addition, IMS provides the facilities for creating and accessing defined files. By writing a data definition, a user can logically redefine a file without actually altering the physical file structure. Selected items from records in several source files can be combined into a single defined file to be used by IMS action programs. DMS data bases can be interfaced by generating COBOL action programs that include DMS data manipulation language commands or by writing a data definition to redefine the data base.

RELIABILITY

IMS is an extremely reliable system providing extensive features to prevent accidental alteration or destruction of data files. A record lock facility prevents a user from accessing a file that is currently being updated by another user. If the updating transaction fails, the file is rolled back to the state it was in before the transaction. Files that are accidentally or incorrectly modified can be recovered by an offline recovery program either as they existed before being altered by transactions (backward recovery) or after modification (forward recovery). A quick-recovery facility provides backward recovery of all files adversely affected by a general system failure.

INTERACTIVE COMMANDS

In addition, to the terminal transaction processing provided by UNIQUE and action programs, a powerful set of interactive commands is available to assist with the administrative, operational, and educational aspects of IMS. Included is a set of general commands that can be used by individuals to control the processing of their own terminal and a master terminal command set issued from a designated master terminal to control the processing activity of the entire network. A terminal-to-terminal message command permits communication between terminals.

IMS offers a fast, reliable, and interactive method of accessing data files. It is an easy-to-use system capable of making any individual within a user organization into a potential IMS user, yet providing the security and file protection so important when dealing with vital data files.

2.5. DATA BASE MANAGEMENT SYSTEM

The data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs handle the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of DMS data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

DMS FEATURES

PHYSICAL DATA BASE CHARACTERISTICS

The device media control language (DMCL) defines physical characteristics of a user data base and the data dictionary.

LOGICAL DATA BASE CHARACTERISTICS

The description of data in a DMS data base is entirely separate from the manipulation of that data in application programs. This results in a higher level of data independence for application programs. All descriptions are done in a high-level language comparable to the data declaration language of COBOL.

STORAGE STRUCTURES

Data can be organized into three storage structures: sequential, tree, and network. The same record can belong to several different structures simultaneously and is stored directly by data base key, sequentially within area, by calculated key value, or according to its set relationship to other records in the storage structure. This reduces data redundancy and promotes processing efficiency.

MANIPULATION OF DATA

The data base is accessed by using a combination of data manipulation language (DML) statements and conventional COBOL statements in the procedure division of a COBOL application program.

SYSTEM SUPPORT FUNCTIONS

DMS supplies processors and utilities, collectively known as system support functions, that create, establish, and maintain a data base.

IMS/DMS INTERFACE

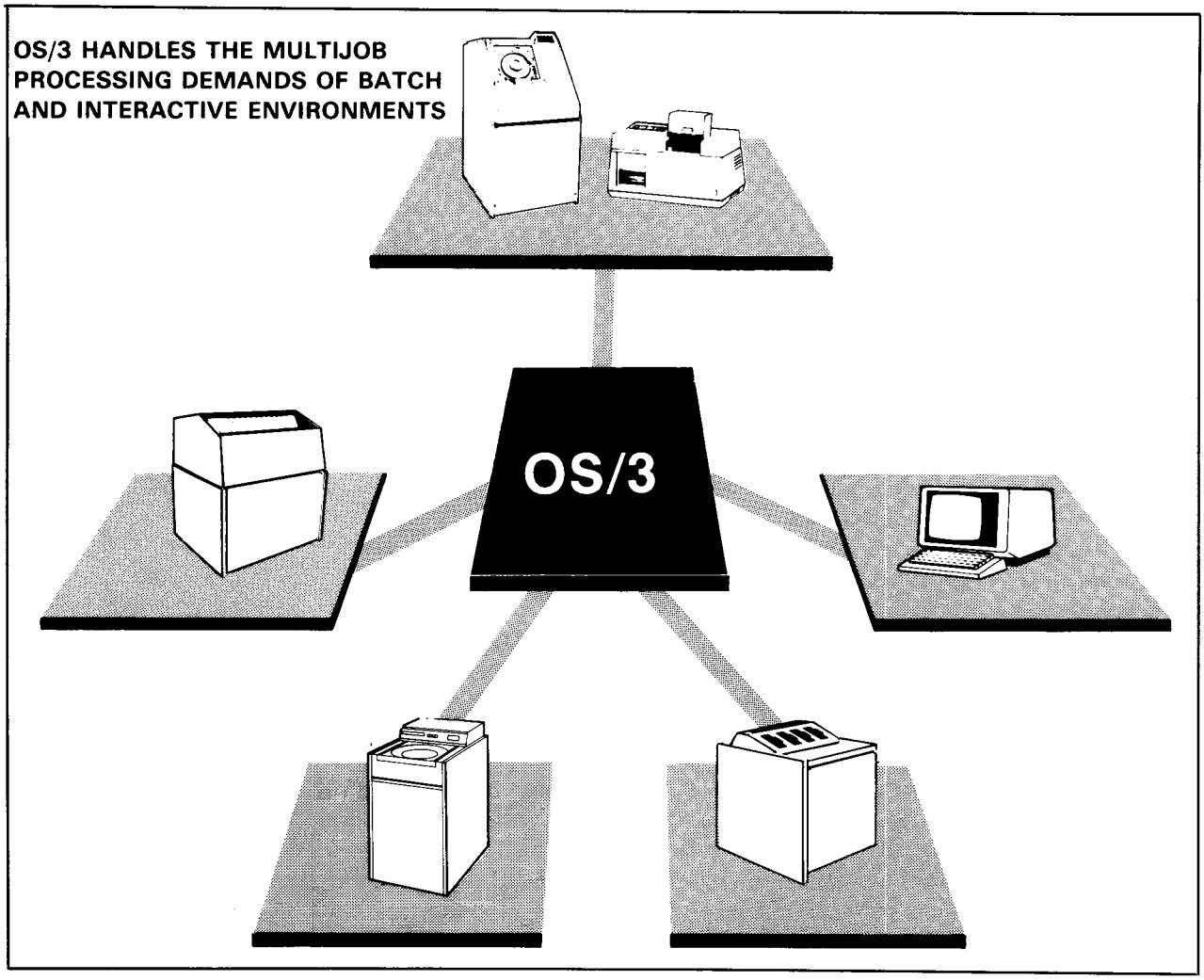
IMS user-written action programs or UNIQUE action programs can access DMS data bases.

RECOVERY

Offline recovery utilities use a journal file generated by the data base management system for forward and backward data base recovery. Online recovery automatically restores a data base after system crash or abnormal program termination.

2.6. JOB PROCESSING

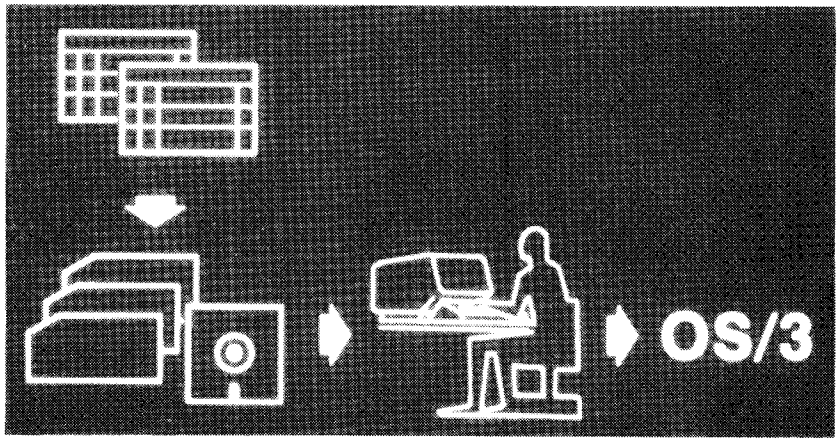
OS/3 is structured to handle multijob processing demands in both batch and interactive processing environments. Up to 48 jobs can be active concurrently in OS/3. A primary user interface to the system is job control, which is the OS/3 component that tells the system (at the user's direction) what hardware and software resources are needed to process a job as directed. The user must prepare a job control stream that specifies these requirements for every job in OS/3. The method of preparing the control stream varies, depending on whether the job is running in a batch or interactive environment. After the job's processing requirements are defined, the method of storing and running jobs also varies, depending on the environment.



2.6.1. Batch Job Processing

In a typical batch environment, the user builds a control stream for a job by coding and keypunching the stream on cards or a diskette and submitting the job to the system operator to be run. The user can communicate with the system operator through certain job control statements in the control stream that, when processed, are displayed at the system console. The primary responsibility for job processing rests with the system operator once the job is submitted. The user has considerable control over the way the job is processed, however, through the job control specifications. The user can also initiate the running of a job from the control stream of another job through a special job control statement that simulates many system operator functions. Job control streams and programs are stored through system console commands. In a batch environment, the results of the job are seen after it has been run, sometimes hours or days later. This contrasts with an interactive environment, where the user can see the job results displayed at the workstation screen almost immediately.

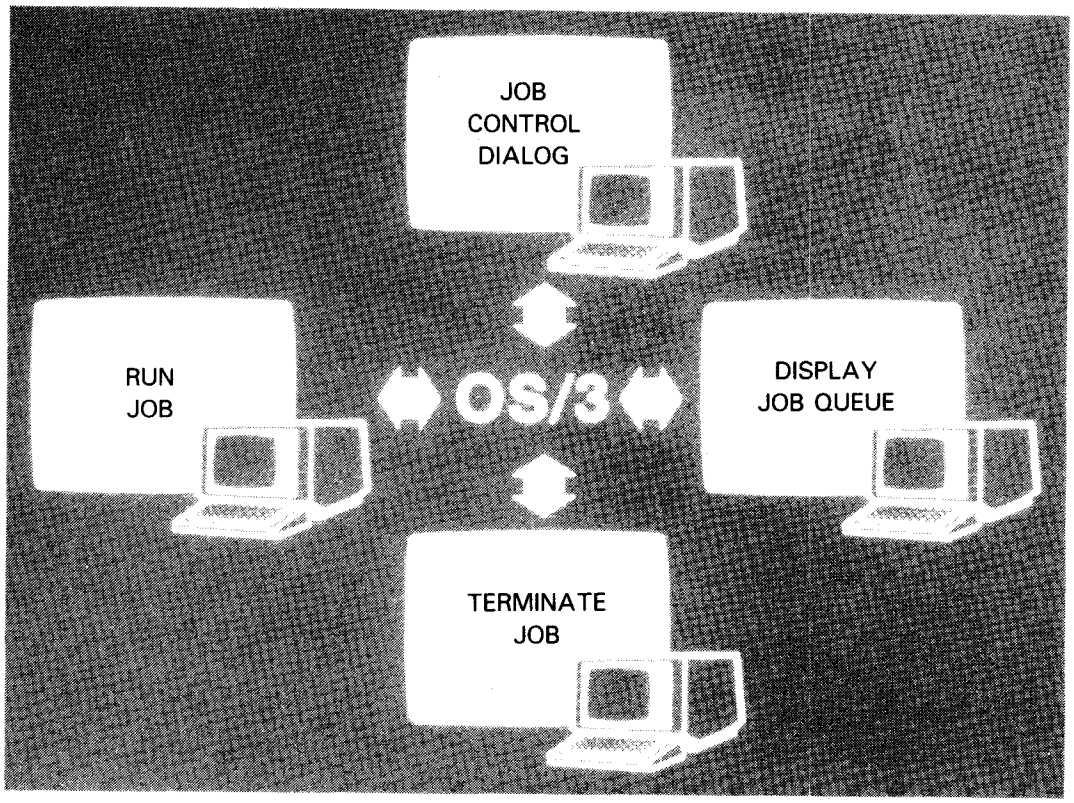
IN A BATCH ENVIRONMENT,
JOBS ARE RUN BY THE SYSTEM OPERATOR



2.6.2. Interactive Job Processing

In a typical interactive processing environment, the user communicates directly with the system from a workstation. The user can create program source code, build a control stream for a job, store job streams and programs, and run jobs from a workstation. Certain facilities of interactive job control allow variable values to be submitted for a job at run time, from the workstation, and to actually change the way a job is executed by performing dynamic skip functions from the workstation. The programs and control streams can be stored in permanent files from the workstation by using the general editor and the job control dialog. Other job processing functions, like connecting a workstation to a job, terminating a job, and requesting the status of a job, can be performed through simple workstation commands. In an interactive environment, in short, the user can control job processing from the time program source code is created to the time a job is run and terminated.

INTERACTIVE JOB PROCESSING PERMITS CONTROL OVER A JOB FROM BEGINNING TO END



2.7. MULTIJOBBING

System 80 Model 8 can concurrently process from 1 to 48 jobs, with each job consisting of one or more job steps, or tasks, executed serially. Each task can be a program compilation or the execution of a user- or system-supplied program. This capability allows the user greater flexibility in attaining maximum use of the system resources and in scheduling tasks.

OS/3 multijobbing consists of scheduling multiple jobs for concurrent execution. The allocation of processor time is based on a system switch list that contains information regarding program priorities, task synchronization, and input/output utilization. While one task is awaiting the completion of an external event (such as completion of an input/output request), OS/3 activates another task that is ready to run to ensure optimum utilization of the processor's capabilities. Because the majority of programs require support other than processing instructions, OS/3 multijobbing provides an effective method for the user to reduce processor idle time and increase system productivity (throughput).

OS/3 was designed around the multijobbing concept. All the software that Sperry Univac provides in the OS/3 package is designed to take full advantage of this multijobbing environment. The controlling software automatically, concurrently processes any user jobs submitted to the system. The interactive facilities are handled by the system in such a way that each individual workstation operator receives optimum response times to entered requests.

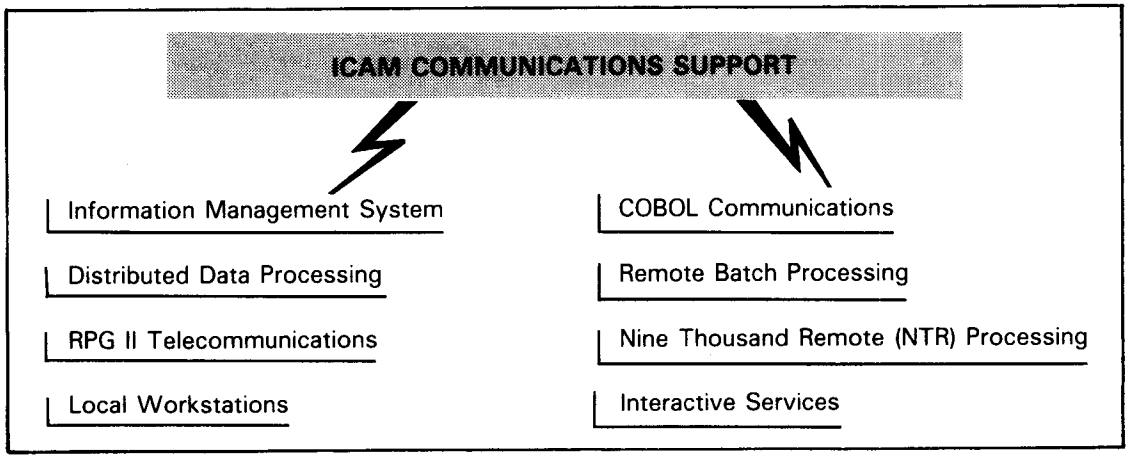
2.8. DYNAMIC RESOURCE MANAGEMENT

The OS/3 software has the capability of dynamically managing the resources available for use in the system. This dynamic management is provided for both interactive programming and batch activities. The system can dynamically manage such things as the main storage assigned to a job, unassigned main storage, workstations, and the allocation of peripheral devices, such as disks and diskettes. The dynamic management of system resources frees the interactive user from constraints of the physical devices and software components planned for use. In addition, the batch user with a program interfacing with a workstation user cannot always predict the extent of the resources the job requires. Thus, dynamic resource management frees all users from such dependencies and increases the efficiency and throughput of the entire operating system.

2.9. DATA COMMUNICATIONS

OS/3 provides an integrated, modular communications package designed for ease of operation while offering the communications user a system that provides a broad range of capabilities. This package, the integrated communications access method (ICAM) terminal support facility, can be configured to meet the needs of the individual system. ICAM offers several levels of support, each of which balances the services provided with the amount of system resources required. The small user can configure a simple, yet effective system without causing a severe strain on resources. The communications-oriented user can build a complex communications system that provides a full range of message handling functions.

ICAM provides the communications capability required to support a number of other Sperry Univac components. Among these are.



These functions are performed independently from the operation of ICAM because the comunications system has been designed to offer total device independence to both the user programs that interface with the system and to the system programs that interface with ICAM.

ICAM allows the user to configure a variety of terminal device types and communications lines into a single network usable by a number of user and system programs concurrently. ICAM has the responsibility for preventing conflicting resource assignments and for releasing facilities when jobs terminate.

ICAM PROVIDES

- The queueing of messages in main storage or disk (a user option) while waiting for servicing
- Specification of multiple destinations for a single message
- Scheduling of activities and support of user-defined priority levels
- Reconstruction of messages and queues from user-specified points in the event of system failures
- Accumulation of relevant statistical information

An ICAM configuration includes: the software modules included in the system during the system generation procedure; the communications lines; the terminal devices; the system utilities that the ICAM network requires; plus the programs that are to interface with the network.

OS/3 communications supports two basic types of communication: narrowband and wideband transmissions. Narrowband transmission permits voice-grade communications (referred to as dial-up), switched DDD (direct distance dialing), or privately leased lines. Wideband transmission permits data transmission over high-speed privately leased lines. Sperry Univac provides full- and half-duplex interfaces for its own terminals, as well as commercially available data sets.

2.10. DISTRIBUTED DATA PROCESSING

The SPERRY UNIVAC distributed data processing (DDP) system allows users having a number of separate processing systems to tie those systems together in a network so that all systems can share the data files and processing load of the entire organization. This distribution can bring many advantages to the user. The distribution of data files throughout an organization's data processing network can mean reduced need for mass storage at each processing location tied to the network. There is added security in that a disaster at any one location would not destroy all the data used within the processing network. The capability to distribute the processing work among systems in a DDP network can provide improved business operations and management control. Jobs can be decentralized and given to the location responsible for gathering and using the data. Distributed processing can also provide greater control over work priority, improved response times, and more efficient use of all the data processing systems within the network.

Distributed processing functionality in OS/3 is divided into three software packages, each offering the user different elements of distributed processing. Thus, the user may tailor his distributed processing software to include only the particular features his installation requires. The three software packages are the DDP transfer facility, DDP file access, and the IMS-DDP transaction facility.

The DDP transfer facility provides these capabilities:

- Site-to-site data file and program library transfers
- Remote job initiation and control
- Operator console control over remote site and routing of messages to remote operator console

The DDP transfer facility can be used to copy data files and program libraries from one system to another or to add data to an existing data file in a remote system. Data files can be deleted from a remote system. Data files and program libraries can be transferred between OS/3-based systems.

Users can utilize the DDP commands available through the DDP transfer facility to submit and initiate a job to a remote system through the local system. Output from the job can remain at the processing site or can be directed back to the initiating site. The transfer facility software can take advantage of spooling services if they are included in the system. Jobs running at another site can also be cancelled from the initiating system.

Messages can be sent to the system console of a remote system and the answer routed back to the initiating device. A device on a system can, in a limited way, operate as the system console for a remote system.

The DDP file access package provides these capabilities:

- A program can access, through job control statements, a disk file residing on a remote system and use it as if it resided on the local system.
- A program can access another program, running on a remote or local system, as if it were a disk file.

- Workstations may be connected to jobs running on a remote system.
- Control of programs can be transferred between workstations on either remote or local systems.

DDP file access also permits the routing of print and punch spooled output files to remote auxiliary printers.

The IMS-DDP transaction facility allows users to perform transaction processing between systems in a DDP network. Thus, a user on one system may perform inquiries and updates on files residing at a remote system, as well as on his local system. Both simple and dialog transactions may be performed through IMS-DDP.

The software required to support a DDP nework includes a configured communications network to physically link the included sites with distributed data processing processors at each site. The DDP processors interpret all DDP related commands and perform the requested functions. The DDP processors are designed to operate in an interactive environment and respond to commands issued from a workstation.

2.11. APPLICATIONS PROGRAMS

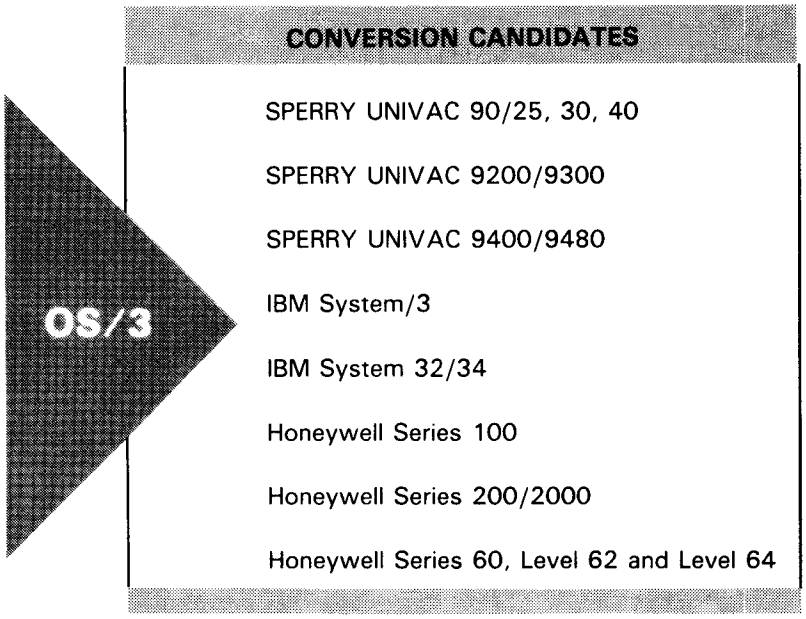
OS/3 readily lends itself to the support of diverse needs of the user organization. In addition to the major stand-alone applications systems, such as the UNIVAC Industrial System (UNIS), the OS/3 applications programs library contains over 700 programs and subroutines for interactive and batch use in the solution of a wide variety of business and technical problems. The interactive nature of OS/3 makes it especially suitable for the support of these operations without any unusual scheduling or data entry impact on the primary system users.

The list of available applications programs constantly fluctuates as Sperry Univac adds new programs, and modifies existing software, to meet the ever-changing needs of today's business community. Users wishing to know if Sperry Univac has an applications program that suits their needs can do so by contacting their local Sperry Univac representative.

2.12. CONVERSION AIDS

To convert an existing system to an OS/3 system, data files must be transferred to the peripherals in the new system, and those program elements not accepted by the new system must be changed. If the user is satisfied with the structure of the existing data base, file conversion becomes a relatively simple question of dumping existing files to a transportable medium and restoring them on the new system. If the instruction repertoire of the new system is substantially identical to that in the existing system, the conversion task is reduced to modifying the macros, file definitions, and incompatible instructions in the program.

To meet the needs of a converting user, Sperry Univac supplies a full range of conversion tools for a number of systems.



2.13. AVAILABILITY, RELIABILITY, AND MAINTAINABILITY

A prime consideration when purchasing a new computer system is the amount of productive computing time the user can expect from the system. The best computing system is of little value if it is unavailable for use because of a system failure. Sperry Univac recognized this and designed the hardware and software for System 80 Model 8 to be as available as possible by using reliable components and including extensive automatic recovery and maintenance features.

The hardware represents the state of the art not only in technology, but also in reliability. System 80 is designed to perform effectively under various conditions, and extensive testing ensures that any installed hardware is in perfect operating condition and was not damaged in transit.

The OS/3 software has proven itself through years of reliable service. It has enjoyed the long-term success associated with an effective and reliable product. However, even the most reliable systems available can suffer an occasional failure. To ignore the possibility would be unwise; therefore, Sperry Univac included the features needed to ease the impact of a system breakdown and to get the system up and running as quickly as possible. Sperry Univac supplies its customers with an impressive array of diagnostic, backup, and recovery facilities. These features cover all possible software and hardware failures. The system even monitors its own operation to help the user predict the possibility of a failure. Sperry Univac also provides direct assistance to System 80 users through the Remote Support Center. Through the remote maintenance interface and acoustic coupler provided with System 80, specialists at the remote support center can analyze hardware problems, recommend action to the system operator, and inform Sperry Univac customer engineering personnel of the nature of the problem. This can reduce substantially the number and duration of service calls.

To make System 80 Model 8 more reliable, Sperry Univac regularly issues software maintenance packages. These packages contain changes that can enhance system stability or forestall potential difficulties. Additionally, Sperry Univac responds to maintenance requirements by providing individually tailored software maintenance changes.

3. System Software

3.1. SOFTWARE DESIGN CONCEPTS AND CAPABILITIES

Through centralized control of all activities of System 80, the combined hardware and software capabilities are fully established and maintained to satisfy the requirements of all applications. The responsibility for efficient and flexible centralized control is borne by OS/3, which allows the programmer to use the system with relative ease, while relieving him of concern for the internal interaction between his program and other coexistent programs.

OS/3 is a comprehensive library of programs consisting of an executive, a collection of language processors, utility routines, and application programs. The programs that make up the executive are the OS/3 Supervisor and Job Control. Through its versatile job control language, OS/3 organizes and directs operations and system activities to achieve maximum use of computer facilities. The OS/3 supervisor provides the central control, interface, coordination, and allocation of the hardware to achieve optimum system utilization. The supervisor also controls the initiation, loading, execution, and termination of user jobs.

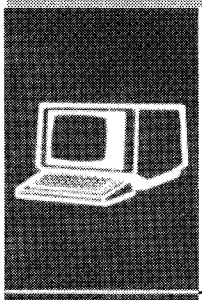
OS/3 has been designed and implemented to offer an efficient multijobbing environment that is needed to utilize the full capabilities of System 80.

The speed and hardware capabilities of System 80 are used to maximum advantage, and a given hardware configuration is used most effectively in the complex internal operating environment created by OS/3. This environment allows for the concurrent operation of programs; for immediate reaction to the inquiries, requests, and needs of many different users at remote and local workstations and terminals; for storage, filing, retrieval, and protection of large blocks of data; and for optimum use of all available hardware facilities while minimizing job turnaround time.

Ease of use is emphasized in the system. Interactive facilities simplify the user-to-system interface, allowing nonprogramming personnel to use the system. Work to be performed by the system is described through the OS/3 job control language, which was designed to minimize job turnaround time and operator intervention. The user may construct any logical combination of programs for a particular job by inserting the proper control statements in his job stream.

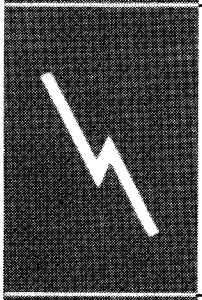
Jobs can be collected and entered into the system from many sources – central or remote. The executive controls the loading, allocation, and execution of the programs once they have been entered. Job steps that cannot be completed because of program error are automatically deleted from the system with appropriate diagnostic information. The console operator is, in effect, responsible only for participation in the data processing activities as directed by the executive.

FUNCTIONAL CAPABILITIES OF OS/3



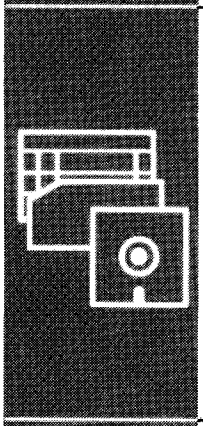
Interactive Processing

The executive provides the services required to meet the unique needs of the interactive user. It interfaces with the user through the workstation and with the interactive services to ensure that all activities proceed efficiently and that all interactive workstation users can perform their operations free of interference from other users. It coordinates the interactive activities with all other activities to provide a balanced processing environment so that the needs of one type of user do not interfere with those of another.



Communications Processing

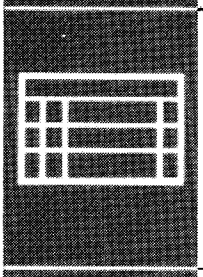
The executive efficiently responds to the demands of communications processing and gives preference to the operational needs of a communications system, these being the most critical requirements of the system. The executive functions dedicated to this type of processing receive the highest priority for scheduling within the system. The contingencies of message control are supported by procedures to roll out conflicting user programs, restart the system, and perform other necessary functions.



Batch Processing

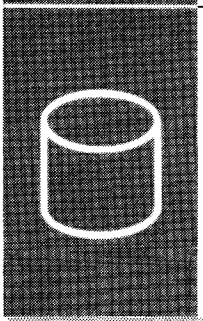
Facility of job preparation and submission, with minimization of job turnaround time, is a design feature of the executive. User-assigned priority by job can provide preferential service when batch jobs are submitted from workstations or remote terminals or where turnaround time is critical.

The executive provides an operational environment for a high volume of jobs. The user may specify preferred priorities for certain jobs within a group of jobs; this fulfills the user's responsibility related to multijobbing scheduling to achieve machine optimization. Automatic job-to-job transition, communication within jobs, and associated services are automatically provided by the system.



User Program Development

The executive interfaces with a specific set of source code language processors and editors that enable the user to write programs in COBOL, ESCORT, BASIC, FORTRAN IV, basic assembly language (BAL), and report program generator (RPG II). The output from any one of these language processors (object module), when processed by the linkage editor, will produce a program (load module) that can be executed on System 80.



Disk-Oriented Processing

The executive uses disk storage as buffers for accommodating any job backlog and for storing output data resulting from executed jobs. The buffering allows the system to operate independently of the essentially low-speed peripheral devices. All executable programs are obtained from disk storage through one of the system libraries maintained by job control. Temporary files required during program execution are generally assigned to disk storage rather than magnetic tape to facilitate disk-oriented processing. Operator participation is explicitly defined and is minimized as much as possible.

The programs of the OS/3 Executive - the supervisor and job control - provide control for coordinating and executing system-supplied and user programs and for furnishing a flexible processing environment. In concert with a collection of programming languages, utility routines, and application programs, Sperry Univac provides the user with a viable program library to take full advantage of the extended capabilities of System 80.

3.2. SYSTEM CONTROL SOFTWARE

3.2.1. Supervisor

The supervisor is a collection of sophisticated routines that provide the central control needed for the system's hardware and software, user programs, and interactive facilities to work together efficiently. Without the services provided by the supervisor, such features as multiple program processing, simultaneous interactive access, error control and recovery, and automatic resource management would not be possible. The supervisor manages and coordinates all system activities, handles randomly occurring events, initiates and coordinates the execution of batch and interactive programs, and provides advanced programming facilities. It is one of the most complex components in the system, yet the supervisor is what makes the System 80 an easy-to-use and efficient data processing system.

The supervisor is built around executable modules, or routines, each of which has a specialized function. Those routines commonly used by the supervisor always reside in main storage. Other less often used routines, called transients, are stored on the system resident volume and are loaded in main storage only when the supervisor needs them. This arrangement promotes supervisor efficiency: it minimizes the amount of main storage the supervisor uses by overlaying unneeded transients with newly loaded transients, and it eliminates the input/output time needed to load the most commonly used routines by keeping them resident.

SUPERVISOR FUNCTIONS AND SERVICES

- Supervisor interrupt requests - service the various interrupts generated by the system.
- Task switcher - determines the order in which the various tasks are given CPU time.
- Physical input/output control - controls the dispatching, queueing, and interrupt processing for all I/O devices directly connected to the system.
- Transient management - schedules, locates, and loads transient (nonresident) supervisor routines.
- Timer and day clock services - provide a system clock for timed, batch, or interactive activities.
- System console and workstation management - controls information to and from the system console and various workstations configured.
- Error recovery and logging - handles any program or machine error that causes an interrupt.
- Main storage management - allocates and deallocates main storage as required by other system control software.
- Diagnostic services - support a variety of diagnostic services to isolate and correct diverse error conditions.

The supervisor is configured during the system generation procedure (SYSGEN). During SYSGEN, users enter supervisor-related parameters to indicate those supervisor features they want to include as resident and as transient (nonresident). Additional features increase the main storage requirements of the supervisor. (This may be a consideration for smaller systems.) A number of supervisor generation parameters provide default values the supervisor used to handle certain conditions. Users can generate a number of separate supervisors during system generation, but only one supervisor at a time can be operating.

3.2.1.1. Supervisor Interrupt Requests

A supervisor interrupt is a request made to the supervisor to perform a function. Requests are generated by the various hardware and software components. They are called interrupts because they interrupt normal processor flow and must be handled in some way before processing can continue. OS/3 recognizes eight types of interrupts:

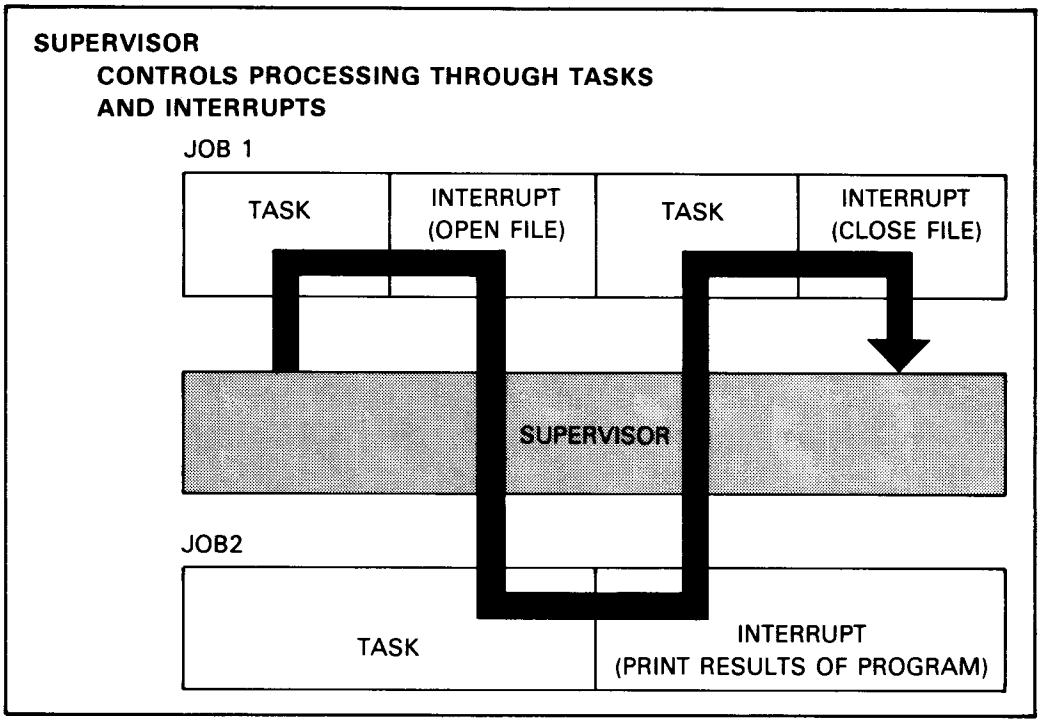
INTERRUPTS

- Supervisor call - occurs in response to the SUPERVISOR CALL (SVC) machine instruction. Although it is handled as an interrupt, programs routinely use the supervisor call to request supervisor services.
- Exigent machine check - indicates a malfunction in or around the processor from which the supervisor cannot recover.
- Repressible machine check - indicates a malfunction in or around the processor from which recovery is possible.
- Program check - occurs when the processor attempts to execute a nonexistent instruction or to execute an existing instruction in an illegal manner.
- Program event recording (PER) - provides dynamic monitoring of executing programs by storing information about the current instruction when a specified event occurs.
- Input/output - occurs in response to signals from I/O channels.

Some interrupts, like supervisor call or input/output, are routinely encountered; others, like program or machine checks, represent errors that the supervisor must handle with minimal system interruption.

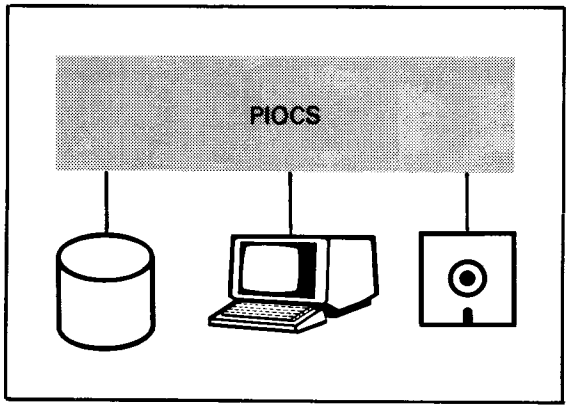
3.2.1.2. Task Switcher

Any activity that uses the central processing unit (CPU) is called a task. The system and users can initiate tasks. User-initiated tasks are interactive activities or functions requested through job control. The system initiates tasks to support user requests or as a part of normal system operation. The mechanism that coordinates the processing of all current tasks is the task switcher. The task switcher decides, based on established priority, which one of the tasks awaiting execution should be processed next.



3.2.1.3. Physical Input/Output Control System

OS/3 performs all input/output (I/O) operations with peripheral devices through the physical input/output control system (PIOCS). PIOCS handles the queueing and initiation of all I/O commands and the processing of I/O interrupts. OS/3 PIOCS is composed of general purpose software routines designed to provide maximum throughput on all peripheral devices and to allow for ease of expansion to support new devices.



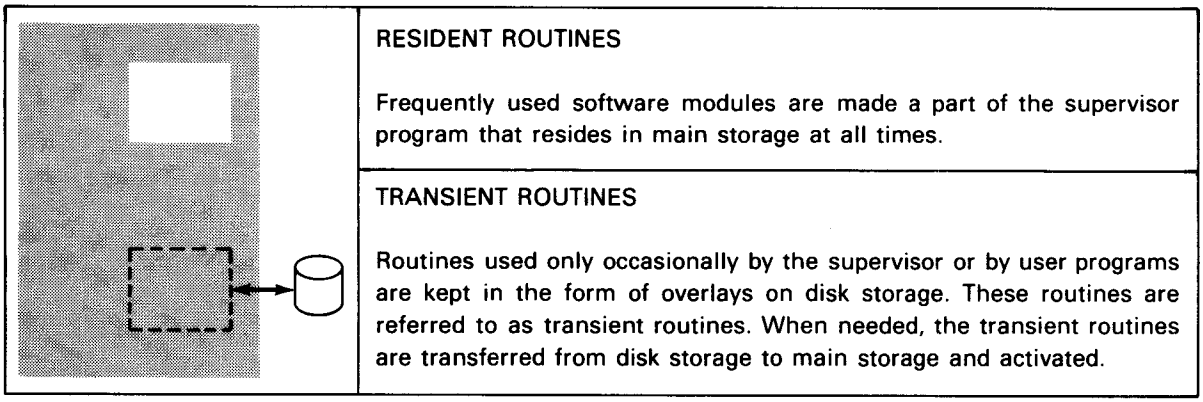
PIOCS receives control when data management issues a request for an I/O operation. Control is not returned to data management until the I/O request has been completed. However, other tasks in the system may be activated if their status indicates a ready-to-run condition. Requests for I/O operations are initially queued, by priority, in device and channel queues. Dispatching follows the queueing of an I/O order if the channel and device are free or, if not, upon completion of a previous order on the same channel. The I/O dispatch routines perform service functions, as required, such as disk address verification and parameter checking. Interrupts from I/O channels are serviced as a high priority function of the supervisor to free the channel for dispatching other I/O orders that may have been queued. Upon completion of an I/O order, a general I/O status analysis is performed to determine whether any abnormal conditions occurred. For normal I/O terminations, control returns to the I/O dispatcher. If no additional requests remain in the queue, control passes to the task switcher for return of control to another task.

If an error condition occurs, the appropriate device is flagged as unavailable for all tasks. For an error on the system resident device, the resident error recovery routine receives control.

Other error conditions require more detailed analysis. A device error recovery overlay routine is called in to complete processing of the error condition.

3.2.1.4. Transient Management

The supervisor takes full advantage of auxiliary storage to provide maximum services while keeping main storage requirements at a minimum. This is accomplished by having two types of supervisor routines:



The number of areas in main storage set aside to contain transient routines is specified by each user. An important function of the supervisor is managing these transient areas by monitoring their use, controlling the loading of a requested transient routine into a selected area, and transferring control to the transient. Only one disk access is required to load a transient. Open files, close files, and terminate jobs are examples of transient routines. Some frequently used transients can be made resident at the option of the user.

3.2.1.5. Timer Service Management

The central processor complex contains a high-resolution timer that can provide an interrupt after any time period greater than 1 millisecond. The calling task may specify a wait interval in milliseconds or seconds, or may specify a time of day at which an interrupt is to occur.

A simulated day clock provides the time of day to tasks upon request and is used by the supervisor for time stamping messages and job accounting entries.

3.2.1.6. System Console Management

System console management provides for displaying messages on the system console, with responses and commands coming from the operator. The screen images are rolled upward, with new display lines or operator input appearing on the bottom of the screen. Console management routines selectively delete messages from the top of the screen that do not require responses.

3.2.1.7. Workstation Management

Workstation management provides physical level support for the interactive services and data transfers for workstations dedicated to a job. Workstation management coordinates all the activities of the workstation with those of the requested functions to ensure efficient operation.

3.2.1.8. Error Recovery

Any error that causes a program interrupt is examined to determine the type of error, such as program check or protection violation, and the type of job involved.

The processing of error information by means of user-supplied subroutines is optional. Standard error control actions are initiated in the absence of user code. If an error is detected, the system is brought to an orderly halt, and a restart from this point is attempted. If recovery fails, information is collected for an orderly abnormal termination.

Invalid or inconsistent requests for supervisor functions are reported to the requesting program.

Machine check interrupts are examined by the system to determine whether the error is recoverable. If the error is not recoverable or recovery fails, the system is brought to an orderly abnormal termination. The pertinent information is collected and logged.

3.2.1.9. Error Logging

The error logging facility provides the capability to record hardware and software errors in the system error log file. The records placed in this file can be subsequently retrieved and statistical reports prepared.

The error logging facility is included in the system at system generation time and it is then that the type of errors to be recorded and the devices to be monitored are decided. Error logs can be collected for every device configured into the system. The type of error log records that can be collected are:

- Peripheral device errors
- Communications errors
- Machine check errors
- User specified errors

During the initial program load (IPL) of a new session, the operator is given the option of retaining the error log file from the previous session or resetting the file with either the same set of collection parameters or a new set. During the operation of the system, the operator can alter the collection of record types.

3.2.1.10. Multitasking

Multitasking is a programming technique that can significantly reduce the time required to process a job. Each job entered into the system consists of one or more functions, or tasks, for the system to perform. Normally, each job step has only one task actively vying for CPU usage. Through multitasking, the user can establish a hierarchy of independent tasks such that several tasks from a single job step can be active at the same time. The system takes control of the CPU away from one task when it is awaiting the completion of an external event and passes it to another task, so if that second task is from the same job, then that job gets processed faster. The separate tasks defined for each job step are called subtasks and vie for CPU usage as independent tasks along with the tasks from other jobs.

To control the tasks and subtasks vieing for the CPU concurrently, the system provides a task priority mechanism that permits the user to establish up to 60 separate task priority levels. Tasks and subtasks can be executed at separate priority levels to ensure that in a multitasking environment the proper tasks are executed in the proper order. The number of task priority levels the system supports is determined at system generation. If a task is executed without a specified priority, it is automatically assigned the lowest priority configured into the system.

Multitasking can significantly improve the processing time of an individual job, and improve the throughput of the system on a whole, by reducing the amount of idle CPU time. The system software supplied by Sperry Univac takes advantage of the multitasking facility to increase the system's efficiency.

3.2.1.11. Diagnostic Debugging Aids

Diagnostic debugging aids provided by the supervisor include:

DIAGNOSTIC DEBUGGING AIDS
Monitor Routine
A hardware monitor interrupt enables the monitor routine to trace the execution of a program so that errors can be located and corrected. The routine interrupts each instruction before it is executed and tests for the conditions specified in the monitor input. When a specified condition is satisfied (a specified storage location, instruction, or instruction sequence is reached), the monitor can print out current program information, program status word (PSW), register contents, next instruction to execute, etc. and can suspend program execution or continue with or without monitor intervention. Trace conditions and information to be printed can be specified via the job control stream or entered at the system console. An entire program or a portion of a program can be monitored.

DIAGNOSTIC DEBUGGING AIDS

Snapshot Display of Main Storage

A partial main storage printout can be obtained to aid in the resolution of a problem. The area of main storage to be displayed and the time the display is to occur are identified by parameters at run time, or specified in the body of the program.

Main Storage Dumps

A main storage dump may be provided for diagnostic purposes under the following conditions:

- Abnormal Termination Dump for User Program

  This provides a main storage dump of a program region in hexadecimal, alphanumeric, or both, plus a formatted display of error codes, job-oriented tables, and supervisor information to assist the user in debugging.

- Program or Operator Request Dump

  This enables the operator or any program to request a main storage dump in the same format as the abnormal termination dump.

- System Failure Dump

  This routine is intended to be used when an abnormal condition occurs and other dump programs cannot be used.

Standard System Error Message Interface

An error message service routine provides complete and specific error messages. This routine locates the message in a disk file and transfers control to the system console handler for message display.

Error Response to User Jobs

Error codes returned by the supervisor to the calling program are standardized to provide a uniform interface for all system services.

If a user requires control returned after the detection of hardware failure or software exception, a user-supplied subroutine must be provided; otherwise, an orderly abnormal termination, which optionally may include a main storage dump, is called for the user job.

Program Checkpoint Restart

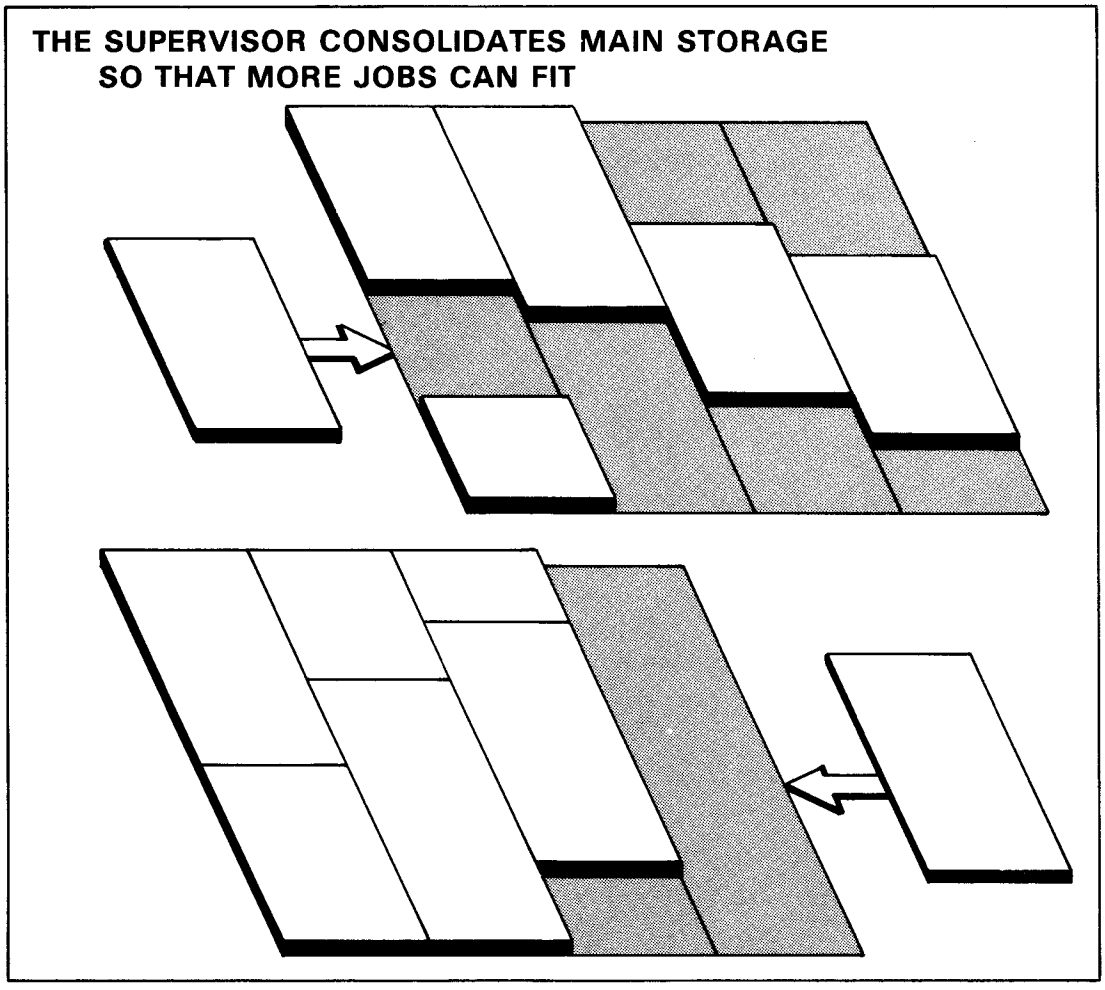
A checkpoint request facility provides for restarting a job with synchronization of all disk and tape sequential files. If unit record files are buffered to disk or tape, the position of these files may also be restored on restart.

3.2.1.12. Main Storage Management

The supervisor acts to ensure the efficient operation and proper use of the system's main storage. Primarily, the supervisor reserves the main storage space required by jobs, programs, and system routines and loads them into their reserved areas. The supervisor makes certain that they are loaded into the proper areas and at the proper time. In addition, the supervisor can:

- dynamically expand the main storage regions of certain programs, jobs, and routines when necessary;
- rearrange main storage to provide as much contiguous space as possible; and
- temporarily suspend and remove jobs from main storage to make room for preemptive priority jobs.

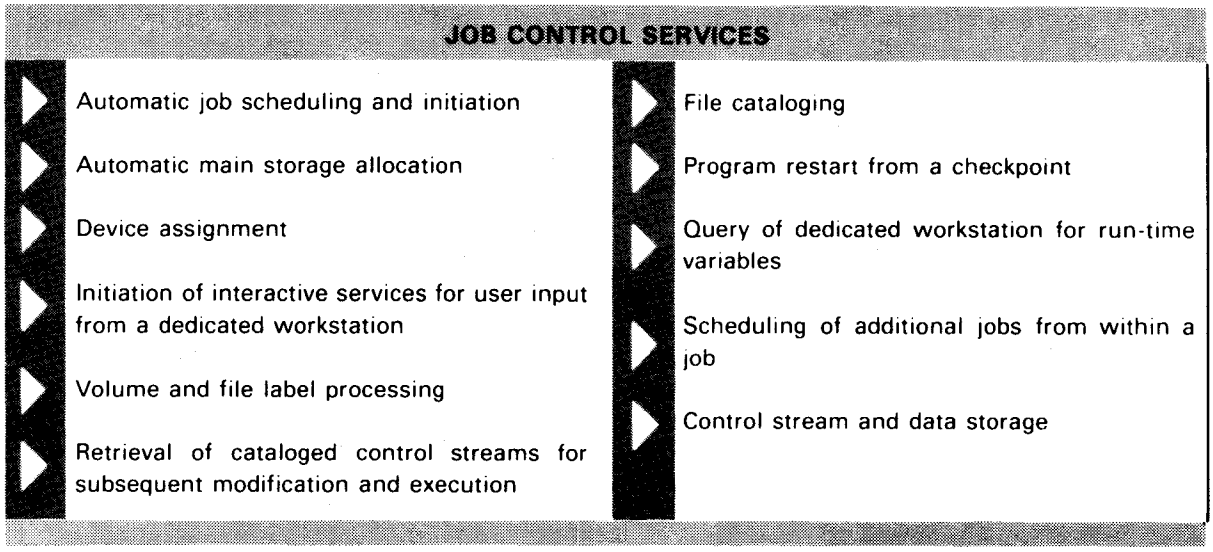
The main storage management functions the supervisor performs are designed to promote system efficiency and ease of use by streamlining main storage operations and providing as much automatic storage management as possible.



3.2.2. Job Control

Job control manages the system resources and prepares the jobs submitted for execution. A job represents a unit of work to be performed by OS/3. Each job consists of one or more job steps, each requesting the execution of a system or user program. Job control services are performed prior to the execution of the initial job step, during the transition between job steps, and at the conclusion of the job.

The services of job control are directed by the user through statements provided by the job control language (JCL). These control statements define the system resources required for proper execution of a job and facilitate the efficient management of these resources. OS/3 JCL is a flexible language that enables the user to specify the requirements for a variety of essential resources and affords a high degree of independence from limitations imposed by system configurations. Through the use of cataloged procedures, OS/3 effectively reduces the usual effort required when running frequently executed jobs.



A control stream is a group of sequenced statements, written in the OS/3 job control language, that defines a job and directs its execution. These statements are divided into these functional groups:

- Job Coordination
- Job Step Operation
- Device Assignment
- Regulation of Job Environment

3.2.2.1. Job Coordination

This group of control statements provides an interface that coordinates overall job execution.

STATEMENTS FOR JOB COORDINATION SPECIFY INFORMATION ON:

SCHEDULING PRIORITY

Indicates one of the following priorities is to be used when scheduling a job:

- Normal priority

  Used for regular scheduling considerations within the system

- High priority

  Used for rush scheduling

- Preemptive priority

  Used for urgent jobs that require immediate scheduling and execution

Within priorities, all jobs are scheduled for execution on a first-in, first-fit basis, with main storage allocated for preemptive jobs via the rollout/rollin capability.

MAIN STORAGE REQUIREMENTS

The main storage requirements for a job can be calculated automatically by OS/3 if all programs to be executed by the job either currently reside in a load library or are specified by the user through the job control language. The user has the option of specifying a minimum and maximum value.

The minimum value is the basic storage requirement needed to properly execute all programs within the job. The maximum value specifies an additional storage requirement that, if available, could be dynamically utilized by the programs within the job to improve and speed up job execution. To exploit this additional main storage, the programs within the job must be specifically designed to take advantage of the additional main storage allocation if it becomes available to the job.

JOB IDENTIFICATION AND DELIMITATION

Uniquely identifies the job and indicates its starting and ending points.

SCHEDULING A JOB FROM WITHIN A JOB

OS/3 JCL enables the user to serially execute jobs by allowing a currently active job to request another job.

RESTART OF AN INTERRUPTED JOB

Job restart from a specified program checkpoint is specified through JCL by giving the required checkpoint information and resource requirements for the recurring portions of the job. Checkpoint data must be established by the user prior to requesting job control to restart a job.

3.2.2.2. Device Assignment

This group of control statements is used to declare the devices required for the proper execution of a job.

STATEMENTS FOR DEVICE ASSIGNMENT SPECIFY INFORMATION ON:

DEVICE ASSIGNMENT SETS

OS/3 JCL offers a set of control statements that provide the information required to assign devices and establish the relationship between files or volumes and devices, such as the number of files per volume and the number of volumes that contain a file. The peripheral devices that may be assigned to satisfy the requirements of a job include card readers, printers, punches, disks, diskettes, and tapes.

DELETION OF FILES

This function deletes specified files from the system prior to the execution of a job step.

LOGICAL DEVICE ASSIGNMENT

This function provides the ability to temporarily change the logical unit number associated with a given device type from that specified at system generation time to one that will be used for a specific job, thus permitting the user to run existing control streams among different System 80 installations.

RELEASE OF PERIPHERAL DEVICES

This function is used to release peripheral devices that are presently assigned to a job and that are not required in subsequent job steps. Upon release, they are no longer available to the original job, and they are made available for subsequent assignment to other jobs being considered for initiation.

3.2.2.3. Job Step Operation

This group of control statements is used to specify job step operation. A job always consists of one or more job steps. One job step could compile a source program; another job step would link the object program, followed by a job step to execute it.

STATEMENTS FOR JOB STEP OPERATION SPECIFY INFORMATION ON:

PROGRAM EXECUTION

OS/3 JCL provides the facility to specify that a system or user program be loaded for execution from a system library or an alternate user library. A job step priority can also be indicated, with the lowest value indicating highest priority.

ALTERING PROGRAMS

This feature of job control allows programs to be altered at execution time. After being loaded into main storage, the module is altered according to the specified changes. However, the copy of the named module, in the load library on disk, remains intact.

DATA AND PARAMETER SPECIFICATION

OS/3 JCL allows the user to include information in the control stream as embedded data or program parameters. This information is stored in the temporary control stream library for subsequent retrieval by a system or user program.

3.2.2.4. Regulation of Job Environment

This group of control statements is used to regulate the environment of the job by (1) making designated job steps conditional on the outcome of a previous job step or on any errors that might previously have occurred during execution, or (2) modifying the environment of the job.

STATEMENTS FOR REGULATION OF JOB ENVIRONMENT SPECIFY INFORMATION ON:

JOB STEP OPTIONS

OS/3 JCL allows the user to specify certain optional software functions to be performed in a job step. The specified functions are effective only in the job step in which they are included. The user can request any of the following options:

- Alter

  Indicates that a loaded program is to be altered prior to being given control for execution.

- Binary Overflow

  Indicates that the user program to be loaded is given control for execution with binary overflow enabled.

- Decimal Overflow

  Indicates that the user program to be loaded is to be given control for execution with decimal overflow enabled.

- Load Module Construction

  Indicates that the source program being compiled is to be linked and executed using default linker parameters.

- No Abnormal Dump

  Indicates that a dump is not desired in the event of abnormal job step termination.

- System Dump

  Indicates that a system dump is desired in place of a job dump in the event of abnormal job termination.

- No Volume Label Check

  Indicates that header labels are not to be read and verified on disk, diskette, or tape volumes.

SKIP CONTROL STATEMENTS

This feature allows the user to bypass any number of control statements, including all statements in an executing control stream. The user can indicate a forward skip from any control statement to any other control statement in the stream.

JOB-TO-WORKSTATION COMMUNICATIONS

During execution, job control can communicate with the workstation by including statements that instruct the job to display queries on the workstation screen during the running of the job. The user has the option of altering the processing of the job at the point where the message was displayed.

JOB-TO-OPERATOR COMMUNICATIONS

This feature enables the user to communicate with the operator. The user can place a message of up to 60 characters anywhere in the control stream, and the message will be displayed on the screen of the system console during control stream processing.

DATE CHANGE

This allows the user to alter or modify the calendar date for a specific job by submitting a 6-character date consisting of the month, day, and year in any order, or a 5-character date consisting of two characters for the year and three characters for the day.

STATEMENTS FOR REGULATION OF JOB ENVIRONMENT SPECIFY INFORMATION ON:	
TRACE MODE	MAGNETIC TAPE POSITIONING
OS/3 JCL provides a trace option for use in program debugging. When the trace mode is requested, every instruction in the job step is examined before execution without interfering with the normal operation of the program, and the pertinent operational information for every traced instruction is printed by the system.	This feature allows the positioning of tape volumes prior to the execution of a job step. It can be used to position a data file or pre-position a multifile tape volume. It will space the tape volume forward or backward a specified number of tape marks or blocks, rewind the tape volume, or write a tape mark.

3.2.2.5. Dialog for Job Control Stream and Jproc Preparation

The job control dialog, supplied by Sperry Univac, is a product that leads the user step by step through the process of building a job control stream or a jproc from a workstation. It allows the user to choose the statements and system jprocs required from lists of menu items and requests parametric values where necessary. It also prompts with HELP screens, which when requested by the user explain the choices available. The dialog guides even inexperienced personnel in building valid job control streams. The job control dialog automatically stores the control streams and jprocs in either the system job control library file (YJCS) or an alternate library file identified by the user.

3.2.2.6. File Cataloging

The file cataloging facility provides a method to control file usage and to restrict the use of selected files to only authorized individuals. File cataloging also provides the capability to build and maintain generation files. In addition, file cataloging can be used as a convenience facility because it reduces the job control required to access a file.

To use the cataloging facility, the device assignment sets of the files to be cataloged are placed in the system catalog file. A device assignment set is the job control statements that define a file and its location. To place a device assignment set into the catalog, a cataloging statement is included in the device assignment set.

If the use of a cataloged file is to be restricted, protective passwords can be assigned to the file. A read password can be assigned to the file, and only those knowing the password can access the file. A write password can also be assigned, and only those knowing the password can make changes to the file.

Often, it is desirable to alter a file and at the same time retain a copy of the file as it was before alteration. The file cataloging facility provides this capability; the files so created are called generation files. For example, three generations of a payroll file could be kept: present payroll, week-old payroll, 2-week-old payroll.

A catalog manipulation utility is provided for use by the system administrator to perform certain maintenance functions on the catalog. This utility can be used to:

- assign a protective password to the entire catalog file to prevent its unauthorized access;
- make a backup copy of the file and restore the copy as necessary; and
- obtain a listing of the contents of the file.

3.2.3. Interactive Processors

Three interactive processors support the workstation-to-job command interface:

- Workstation command processors
- Dialog processor
- Screen format coordinator

3.2.3.1. Workstation Command Processors

The workstation command processors provide the interface between the workstation and the various functional components of the OS/3 software. The command processors handle:

- commands requesting control system functions, such as logging on, job execution, and initiation of the interactive facilities;
- commands issued to control the processing of jobs; and
- commands issued to the various interactive facilities.

3.2.3.2. Dialog Processor

The dialog processor supports the dialog processing services provided by System 80. Dialogs are used to enter variable data directly to a processing program through the workstation. The dialog processor coordinates the displaying of prefiled dialogs, extracts the data entered in response to the dialog, and routes the data to the appropriate user program for processing. The dialog processor functions in response to commands issued from within the processing program that is to make use of the information. When a program issues a request to the dialog processing services, the dialog processor locates and displays the dialog specified by the program. Several system functions use dialogs to make the system easier to use. The job control and system generation dialogs are two examples of such system functions.

The dialog processor maintains an audit file to save entries made to a dialog. The next time that dialog is encountered, the dialog processor can be instructed to read the entries from the audit file. The audit file can be used to make entries to all, or selected portions, of the dialog. The user supplies the information in the normal fashion for those parts of the dialog not filled in by the audit file. The audit file can be amended by the user, and a printed listing of the file, called an audit file summary, is presented each time it is created or altered.

3.2.3.3. Screen Format Coordinator

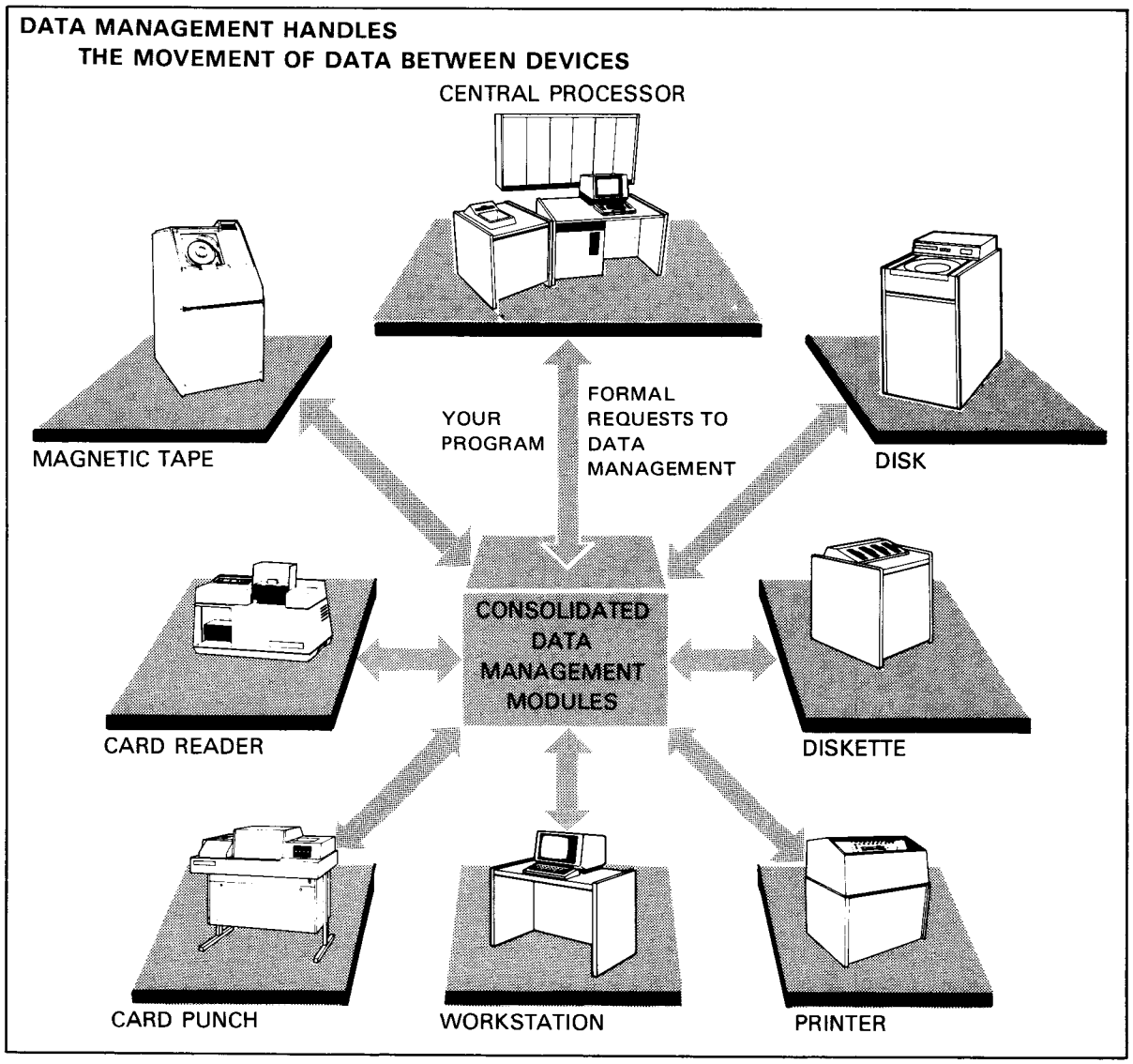
The screen format coordinator is responsible for interfacing the screen formats with the program and workstation users that use them. Screen formats are simply forms displayed on the workstation screen. Screen formats can be used by a program to output data to a workstation user or to input data from a workstation user. When a screen request is encountered, the screen format coordinator locates the requested format and then either displays it with the appropriate output data or displays it and waits for data to be input by the user. These functions are performed automatically in response to the screen format request commands embedded in the programs using the formats.

3.2.4. Consolidated Data Management

The consolidated data management system acts as a convenient intermediary between user programs and the input/output facilities of the supervisor and assists in the task of accessing data files on various peripheral devices. Data management relieves the user of the necessity of coding the routines for block/deblocking, buffering, and communication with the physical input/output control system. Consolidated data management provides an object code level interface for all higher level languages and a macroinstruction level interface for BAL users. The higher level language interface is transparent to the user.

The consolidated data management system offers the following features:

- A single access method for disk files
- Shared data management modules
- Device independence
- Support of the interactive features and workstation data transfers



3.2.4.1. Logical Input/Output Control System

The logical input/output control system (IOCS) modules that control each access method are sharable subroutines and are capable of supporting a single or multiple program environment. When referenced within a user program, these subroutines are dynamically loaded into main storage and can then be shared by many user programs.

3.2.4.2. Disk Access Method

The OS/3 consolidated data management access method for disk files is the multiple indexed random access method (MIRAM). MIRAM simplifies the data management requirements for disk files by offering a single access method that provides several ways of accessing a disk file. The records of a MIRAM file can be accessed:

- sequentially in order of placement;
- sequentially by ascending key;
- randomly by multiple keys; or
- randomly by relative record number.

The use of MIRAM conserves main storage space by providing a single data management module for disk access that can be shared by every user program, rather than having several data management modules, one for each access method.

3.2.4.3. Diskette Access Method

Consolidated data management provides for accessing the records of a diskette file:

- sequentially in order of placement;
- randomly by relative record number; or
- by data set labels.

3.2.4.4. Workstation Access Method

The workstation access method provides the logical level interface between the workstation and the system and user software. The workstation access method provides:

- Automatic support for the workstation as an interactive programming device, including the support of the interactive features that cause screen displays and the management of the screen displays
- Transferring application data and application messages between the workstation and user job
- Console-like capability between the user program, system console operator, and the workstation operators
- Support of workstation function key capabilities
- Device independence

These capabilities are provided automatically or may be controlled through a set of declarative and imperative macroinstructions that provide for accessing workstations at a logical level through the common data interface.

3.2.4.5. Magnetic Tape and Unit Record Device Access Methods

Magnetic tape subsystems, card readers, card punches, and printers are devices on which I/O operations are performed in sequential order; that is, records are handled from the first to last according to physical placement.

For these four device types, OS/3 consolidated data management provides sequential access method (SAM) modules to handle the access requests of user programs.

3.2.4.6. Device Independence

The consolidated data management system provides a significant level of device independence by providing a single system control table format to be used to control and define all file types (disk, diskette, printer, etc) instead of a unique format for each. This allows the files in a program to be changed from one type to another (e.g., disk to diskette) without requiring the user to alter the source code of the program that accesses that file. The only changes required would be to the job control statements that associate the file with a device. Of course, device-unique commands occurring in source programs, such as tape position commands, would need to be removed or altered accordingly. Thus, the OS/3 consolidated data management provides device independence by allowing logical data transfers among devices having dissimilar physical characteristics.

3.2.5. System Service Programs

3.2.5.1. System Librarians

System and user program libraries can be generated as either system access technique (SAT) files or as multiple indexed random access (MIRAM) files. To provide maintenance capabilities to the user for these program libraries, OS/3 includes two program librarians: the SAT librarian, and the MIRAM librarian.

The OS/3 SAT librarian is a set of integrated subroutines existing as a separate system entity and functioning as a housekeeping and maintenance tool for the system and user libraries generated as system access technique files. Programs and elements such as language processor source modules, language processor output (relocatable object modules), or system-executable load modules reside within a library (which is either a system or private file in the OS/3 environment). The storage, collection, creation, modification, correction, deletion, addition, duplication, and transposition of library modules are the primary tasks of the librarian. These housekeeping functions may be performed on entire program libraries, groups of program elements within a specified library, or individual program elements.

The librarian also transfers program library modules from one medium to another. While library modules reside primarily within disk files, the librarian is capable of transposing program libraries to or from a card, diskette, or tape medium.

Control statements supplied to the librarian via the control stream are the means by which the user directs the performance of the operations desired. Through the use of control statements, the librarian initiates such functions as copy, delete, add, compare, merge, compress, correct, group, rename, sequence, reproduce, list, and punch.

The librarian may modify existing libraries, create new libraries via copy or merge operations, or duplicate libraries in their entirety. The librarian also ensures that a given library does not contain any modules with identical names and types.

The listing and punching capabilities of the librarian allow a user to examine the contents of an individual program module or groups of program modules. A module may be listed by the printer as well as punched into cards. The user also may obtain a table-of-contents-type listing of all the program modules in a file. A correction facility allows modification and updating within program modules existing in a library. Source modules may be corrected or updated, as well as sequenced. When any program module is processed by the librarian, its format is verified.

At the option of the user, the librarian provides a map of the functions requested and the results of the action taken, along with diagnostic error indications and warnings, as appropriate. The extent and amount of information provided on the librarian control statements coded are dependent upon the options specified by the user. At the discretion of the user, the control statements themselves may be a part of the library map.

The MIRAM librarian is used to perform maintenance functions on program libraries generated as MIRAM files. System libraries generated as MIRAM files include the system screen format library and the saved, expanded run library. The MIRAM librarian can be used to:

- Copy all or selected modules from one library to another
- Delete selected modules from a library
- Print all or selected modules in a library
- Print a library directory consisting of all active header records
- Insert comments into a module header record
- Change the name or type of a module

The MIRAM librarian responds to a set of user control statements inserted into the control stream that executed the librarian. The librarian will produce a printed listing of all operations performed during a given librarian session. This listing is called a librarian map.

3.2.5.2. Linkage Editor

All OS/3 language processors produce object modules as the output of their compilation processes. These elements may subsequently be structured into a user-tailored executable program by the OS/3 linkage editor. Object modules from various language processor compilations can be structured by the linkage editor into a single loadable program embodying the segment and overlay characteristics described by linkage editor control statements. Additionally, the object modules thus collected may contain cross-references to each other for specific purposes of program execution and communication. Such cross-references between separately compiled modules are resolved by the linkage editor when these elements are collected and a loadable program is constructed.

The loadable program the linkage editor builds can consist of program segments fashioned into multiple regions. Each of these regions is structured as a hierarchical tree with specific boundaries inherently defined as branches of a tree trunk, or root segment. This structure permits a user to produce a program that is larger than the main storage area assigned for its execution. The various segments of a multiphase load module are loaded and executed as required by the logic of the program.

Cross-references made by the user (within compiled object modules) and resolved by the linkage editor in forming the loadable program may exist within or across segments. Segments are composed of one or more object modules, and the loadable program is composed of one or more segments, the initial segment always being designated as the root phase segment. The user's tools for establishing cross-references between processor object modules are:

- Language processor external reference declarations (EXTRN) used to create indirect references in requesting object modules.

- External definition declarations (ENTRY) used to create indirect definitions in satisfying object modules.

These declarations are used by the linkage editor to resolve cross-references between two or more object modules that are linked together at link-edit time. Executable program generation is, therefore, essentially a two-fold process:

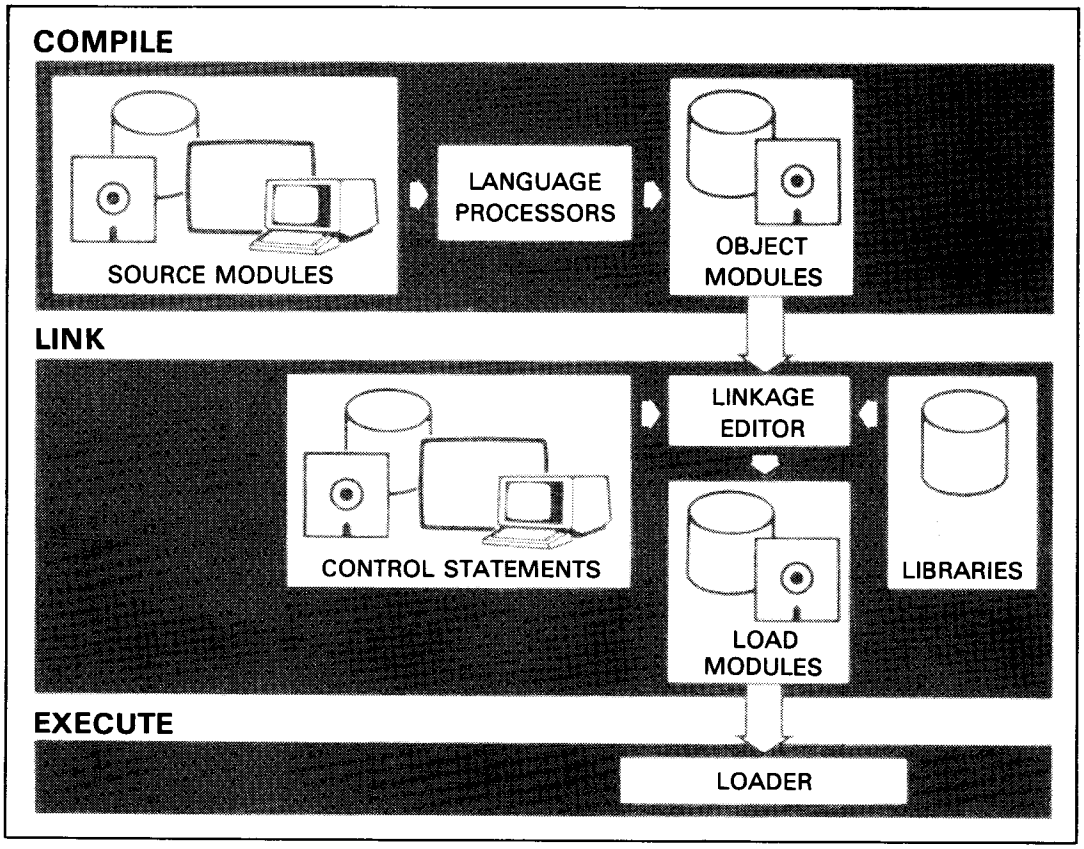
- The source program code must be compiled by the various language processors to produce one or more object modules.

- The various object modules must be combined by the linkage editor to create a single, executable load module.

At link-edit time, the linkage editor assigns, to the object modules included in the load module, a new relative address based upon their new relative position within the load module. Any cross-references that exist between object modules are satisfied by replacing them with the relative address of their respective definitions or references, as the case may be.

The output of the linkage editor is, therefore, a loadable executable program acceptable to the OS/3 loader.

The role of the linkage editor in program preparation is as follows:



The ability of the linkage editor to construct a single executable load module from several object modules has the following major advantages:

- If a change is required in one of the included object modules, only that object module must be compiled or assembled again.
- The various source modules may be written in the appropriate language and combined into a single executable program.
- Routines common to two or more object modules need be assembled or compiled only once and the resulting object code linked as needed; the result is reduction in the total time required to generate an executable program.

In addition to the basic linking function, the linkage editor performs the following:

- Searches an appropriate library and, either on request or automatically, incorporates object elements other than those in its primary input
- Performs program modification by deleting and rearranging control sections of an object module as directed
- Produces an optional overlay structure to be used by the supervisor during loading
- Reserves space automatically for common storage requests generated by the language processor

3.2.5.3. Disk, Diskette, and Tape Initialization Routines

The various tape, disk, and diskette initialization, or prep, routines check the condition of the magnetic storage media and prepare them for use by the system. The disk and diskette prep routines respond to a set of keyword parameters inserted into the job control stream while the tape prep routine is executed with no special parameters required.

3.2.5.4. Dump Routines

The system has a variety of dump routines to aid in the diagnosis of subtle hardware and software errors. The routines include a system dump routine and user job and program dump routines.

DUMP ROUTINES

SYSTEM DUMP ROUTINE

The system dump routine provides a printout of all, or selected portions, of the contents of main storage. The routine can be initiated unconditionally by the user or can be preset to automatically begin if a system failure occurs. The dump obtained is the same in either case.

The system dump listing is divided into several parts, each part corresponding to a system component. The sections are clearly labeled with the appropriate heading for ease of use. A typical system dump listing contains:

- The contents of low order storage
- The physical unit, system information, and channel control blocks
- The system switch list
- Translated job region
- The supervisor
- Hexadecimal job region
- Free region

The dump listing can be in hexadecimal, decimal, or EBCDIC format.

USER DUMP ROUTINE

It is possible for a job to terminate abnormally, or crash, with no apparent reason. Often the only way to determine what caused the job to crash is to list all the information in the system relating to the job. Such a listing is called a job dump, and two routines are available for obtaining a job dump. They are: the job dump routine, and the user end-of-job (EOJ) dump. Both routines produce a listing containing the same type of information, but the formats are different.

3.2.5.5. Software Maintenance Packages and Changes

Software maintenance packages and corrections provide maintenance changes to System 80 software that enhance system stability or forestall potential difficulties. Software maintenance packages are issued on a regular basis, while software maintenance changes are provided to meet individual requirements.

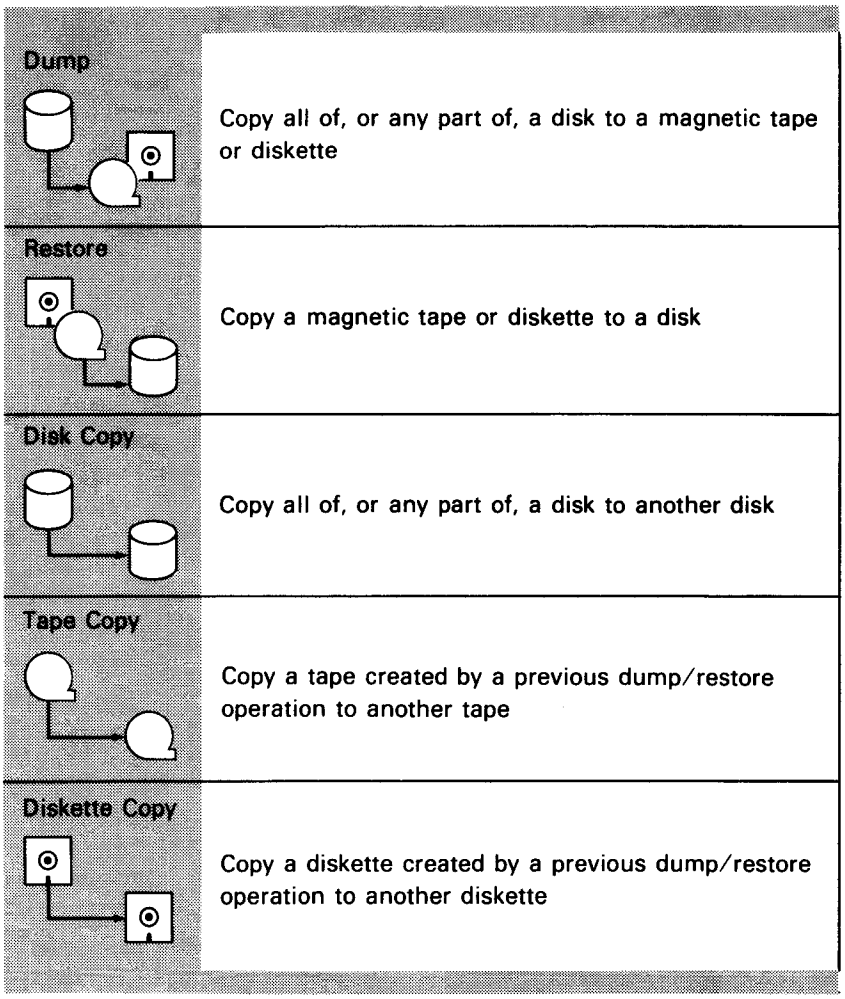
Software maintenance packages and changes are issued on diskettes. They are installed interactively and perform all functions necessary to apply and test the changes. Instructions on application and a list of contents of each package or change are included on the diskette. The user must execute the interactive software maintenance change list routine to obtain a copy of application instructions and contents.

3.2.5.6. Catalog Manipulation Utility

The catalog manipulation utility is used to maintain the system catalog file, which holds the device assignment sets for often-used user files. A file whose device assignment set has been placed in the system catalog file can be accessed thereafter through one job control statement. The catalog manipulation utility enables the system administrator to assign a protective password to the catalog file, make backup copies of the catalog file, and restore these copies to disk when required. The utility operates in response to a set of command statements embedded in the job control stream that executes the utility.

3.2.5.7. Disk Dump/Restore

The disk dump/restore utility allows the user to make backup copies of the contents of a disk volume. Using the dump/restore utility, the user can perform these operations:



The disk dump/restore utility can be executed as an interactive program or run as a batch job. If executed as an interactive program, the user initiates the disk dump/restore utility through a simple keyin command. Control specifications for the program are entered in response to a dialog displayed on the workstation screen. Once the user has traversed the dialog, providing all appropriate information, the program is executed and performs the specified operation.

If executing the disk dump/restore utility as a batch job, through job control, the user must prepare a job control stream. The appropriate program control statements are included in the job control stream.

3.2.6. System Installation Facilities

System installation is the essential process of installing the SPERRY UNIVAC System 80 hardware, integrating into it the Operating System/3 (OS/3) software, and generating this software so that it fits the user's particular needs. Once hardware installation is complete, the user can install and generate the OS/3 software.

3.2.6.1. Software Installation Facilities

Sperry Univac delivers OS/3 software on a set of release diskettes, on tapes, or on disk. Software installation involves the transfer of this delivered software to the nonremovable disk pack integrated in the System 80 processor complex. Once the delivered software has been copied onto the integrated disk, the disk is referred to as the system resident volume, or SYSRES. The SYSRES disk holds all system software and must remain on line when the system is in operation.

To accomplish software installation, Sperry Univac provides installation routines as part of the standard OS/3 release. These routines provide the user with the capability to install:

- The initial release of OS/3 software
- Any new software received between major releases
- Updated software as Sperry Univac releases major improvements to OS/3
- Software maintenance packages and changes

3.2.6.2. System Generation Facilities

System generation, or SYSGEN, is the process whereby the user defines the system's hardware configuration to OS/3 and generates, or creates, the control elements required to meet particular processing requirements. This responsibility is placed in the user's hands because it is the user who is most familiar with his or her own requirements.

To help the user accomplish system generation and to simplify the procedure, Sperry Univac provides these SYSGEN facilities:

- Dialog
- Parameter processor
- Job control streams

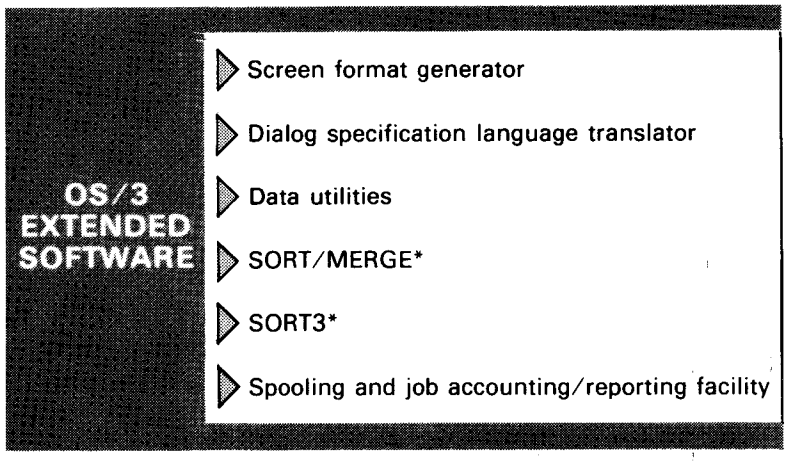
The SYSGEN dialog is an easy-to-use facility that helps the user prepare and process required SYSGEN parameters directly through the workstation. The user makes SYSGEN parameter selections in response to queries displayed on the workstation screen. For omitted or incorrectly specified parameters, OS/3 provides default values that ensure that a usable system is generated. The dialog accepts user choices and passes them to the SYSGEN parameter processor. The parameter processor, in turn, validates the submitted parameter selections and generates a series of job control streams based upon the selections and lists these streams for the user. The user then executes the actual system generation by interactively running the generated control streams through simple workstation keyins.

3.2.6.3. Installation Verification Programs

Sperry Univac supplies a set of installation verification programs that can be run after SYSGEN is complete to test the functional capability and operation of the various components included on the resident disk unit.

3.3. EXTENDED SYSTEM SOFTWARE

The extended system software provides a number of optional features and improves the utilization and operation of the system. The extended system software is available as a package, including all these components:



**Available as separate items*

3.3.1. Screen Format Generator

The screen format generator provides the user with the capability to interactively create template-like formats for display on workstation screens. These formats can be used to simplify entering variable data and parameters and to present output data on the workstation screen in a formatted manner.

Users can generate screen formats to meet particular requirements and applications. The screen format generator permits the user to exercise great control over the appearance of the screen format. The user can include as many fields within the screen as he chooses and specify the characteristics of each field. Users can specify the length of the field; editing attributes; whether the field is input, output, or both; and the disposition of the screen after use.

The screen format generator includes extensive prompting features to assist the user in the generation or modification of formats. Formats generated are stored in a screen format file where they may be accessed by the programs that need to use them or by workstation users for further modification. Formats are accessed through user-defined format names. The screen format generator also provides error detection capabilities during the creation and modification of screen formats.

Screen formats often prove to be the most efficient and accurate way of entering variable data into the system. If properly constructed, they can provide a way for inexperienced users to access the system.

3.3.2. Dialog Specification Language Translator

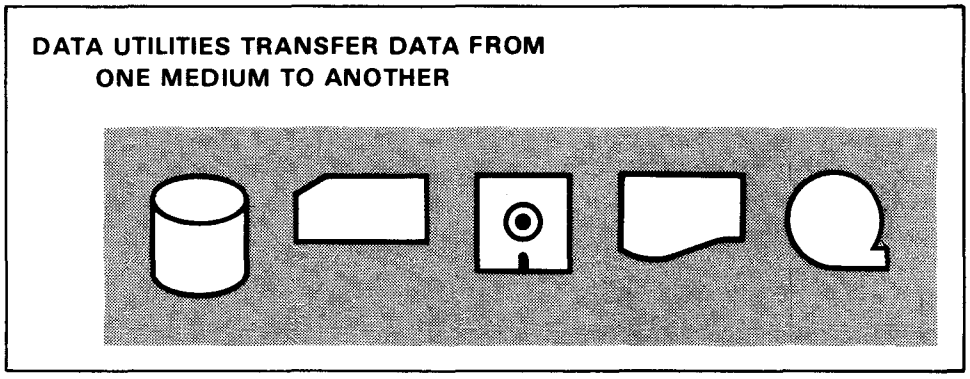
The dialog specification language translator allows the user to create his own job-oriented dialogs. It accepts dialog specification language source code as input and uses that input to generate a dialog display on a workstation screen. Dialogs consist of a series of queries to which the user responds with the appropriate information. Users can use the dialog specification language to produce dialogs to meet their requirements. The user assigns a name to each dialog generated through the dialog specification language translator, and this name is used to access the dialog. Each language has commands to access a dialog. Once accessed, the dialog is displayed on a specified workstation screen, and the information entered in response to the dialog is passed to the requesting program for processing.

3.3.3. Data Utilities

The data utilities (DATA) program performs a number of essential data file maintenance and manipulation functions. Through the data utilities program, the user can:

- Make a copy of a data file on the same type of storage device or a different type
- Reformat the records of a data file so that it can be used in another program
- Select or delete specific areas of a file for testing or report preparation
- Produce a printed listing of a file in a variety of formats
- Compare duplicated files to their originals to check for discrepancies

The DATA program can process files stored on any of the supported storage media: disk, diskette, tape, and cards. Files can be transferred from one medium to any other medium, and printed listings can be made from files on any medium. Users can also perform such device-related functions as label checking, tape rewinds, sequence checking, and write protected files.



The DATA program can be executed as an interactive program or run as a batch job. If executed as an interactive program, the user initiates the DATA program through a simple keyin command. Control specifications for the program are entered in response to a dialog displayed on the workstation screen. Once the user has traversed the dialog, providing all appropriate information, the DATA program is executed and performs the specified function.

If executing the DATA program through job control, the user must prepare a job control stream. The appropriate DATA program control statements are included in the control stream. The DATA program has the same capabilities regardless of how it is executed.

3.3.4. SORT/MERGE

The OS/3 SORT/MERGE can operate in the following manner.

SORT/MERGE

INDEPENDENT	SUBROUTINE
As an independent sort/merge program defined and initiated by job control, SORT/MERGE is constructed as a processor that consists of interrelated modules operating within the framework of a system-driver program. Each module is designed to perform a specific function. As SORT/MERGE progresses through the execution phases, the modules required are called by the driver program, loaded into main storage, and executed.	As a modular subroutine integrated into a user program, SORT/MERGE can be integrated into a user program to allow the user flexibility and freedom, with respect to the external format and source of input records, and to the external format and disposition of the output records. SORT/MERGE can be called from a language processor program that includes the verb specifying this operation. The modular structure gives the user efficient operation despite variations in hardware configurations and data requirements. The modular structure allows a module to be changed, replaced, deleted, or added.

CAPABILITIES

- An interface that permits disk or magnetic tape to be used as work areas
- Input and output that may be associated independently with disk, diskette, or magnetic tape
- Sorting of blocked or unblocked records
- Sorting of fixed-length or variable-length records
- Handling of seven types of key field formats:
  - Character
  - Binary (signed or unsigned)
  - EBCDIC data in ASCII collating sequence
  - Decimal (signed zoned and unsigned zoned)
  - Leading and trailing sign numeric
  - Overpunched leading and trailing sign numeric
  - Floating point (single and double precision)
- Specification of up to 255 key fields
- Sorting of noncontiguous key fields in ascending or descending sequence
- User specification of alternate collation sequence
- Execution of input and output own code
- Sorting of two or more different characters with the same collating value (multiple character sort)
- Shared input and reserved output devices
- A merge-only capability
- Performing of data validity and data integrity checks during sorting
- Convenient restart procedure

3.3.5. IBM Compatible Sort

Sperry Univac also supplies an IBM compatible sort (SORT3). SORT3 is compatible with sort programs provided by IBM for its System/3, System/32, and System/34 data processing systems. SORT3 is an OS/3 program initiated by means of job control. This sort can process disk files, diskette files, tape files, and card input and:

- Rearrange the records in a file
- Reformat the records in a file
- Select specific records from a file
- Summarize fields in the records

SORT3 is capable of performing three different types of sorts: full record sort, tag sort, and summary sort.

The output from the full record sort is 10-byte (binary) relative record numbers of the records in the input file. The tag sort output is a file of sorted records that can contain control fields and data, control fields only, or data only. The output of the summary sort can be any of the following:

- Control fields, data fields, and summary data
- Data fields and summary data
- Control fields only
- Summary data fields only
- Data fields only
- Control fields and summary data fields

3.3.6. Spooling and Job Accounting/Reporting

Spooling (Simultaneous Peripheral Operations On Line) is a technique that increases the throughput of System 80. Data from low-speed peripheral devices is transferred to disk storage independently of the program that will use the data. When a user program is logically retrieving data from a low-speed peripheral, it is physically retrieving the data from a higher speed device. On output, the user program logically specifies a low-speed device, but the images are physically recorded on disk storage and later, under system control, transferred to a low-speed device. Spooling of output allows concurrent use of a specified device by multiple programs. OS/3 spooling consists of these routines:

- Input Readers

  Input readers accept data files from local or remote subsystems. When these data files are used, no distinction is made as to the method of submission.

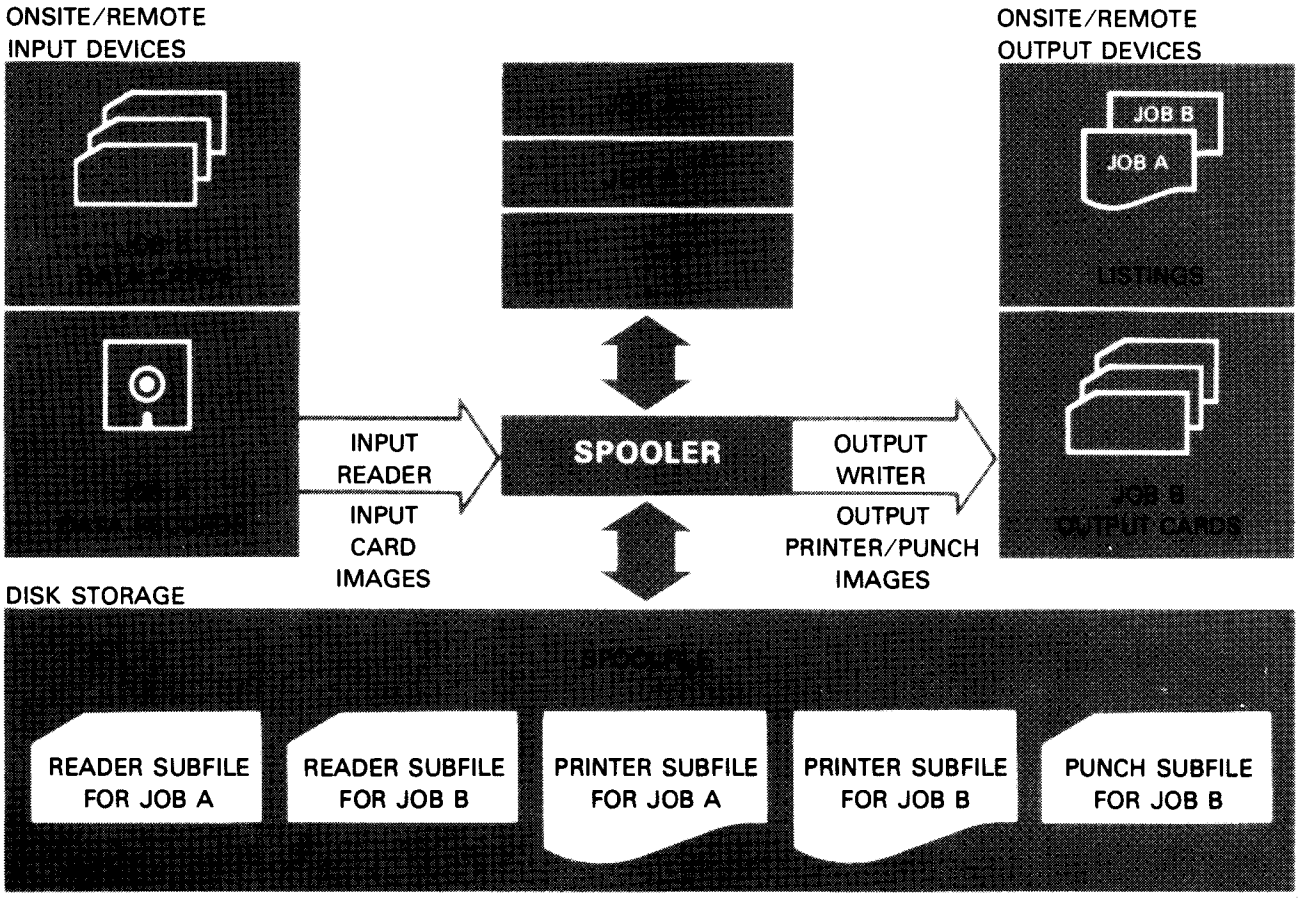
- Spooler Cooperative

  The spooler routine stores card data in the spool file until it is required by a user program. Data output from a user program is stored by the spooler in the spool file until a printer or punch is available.

- Output Writers

  Output writers provide local or remote users with output from user programs that receive identical service regardless of final destination. In addition, the operator of the printer can control forms recovery at page increments, control the number of copies, and control starting and stopping.

Here is how several active jobs use the concurrent spooling capability of OS/3:



As a normal part of spooling operation, job accounting information is generated for each job run on the system. This information is included in the normal message output log, but the information is not saved. However, the user can specify that all the accounting data generated for each job be saved in a spooling LOG file.

The system console and job log accounting utilities gather selected information from the system spool LOG file, sort the information according to user specifications, and produce a printed report. The system console and job log accounting utilities are:

- System log accumulation utility
- Joblog report program

The system log accumulation utility transfers all of, or selected portions of, the spooling LOG file onto a disk or tape file for use as input to user accounting routines. The system log accumulation routine is initiated by running a canned job control stream.

Once created, the file can be used by the joblog report program to create a printed report containing selected job accounting information. This program gathers the information and sorts it according to user specifications. The joblog report program is initiated by running a canned job control stream. Parameters allow the user to select the type of accounting records to be included in the report and the manner in which they are to be sorted.

3.4. MENU SERVICES

The Menu Service software allows the System 80 user to make use of menus in his data processing activities. Menus are screen displays that list, by number, various items including user and system programs, interactive commands, and other system activities. The user chooses which item he wants to perform and enters its corresponding number in a space provided on the menu screen. The activity is then performed, without the user's having to make any further keyins.

Figure 3–1 shows a menu, supplied by Sperry Univac, of system functions:

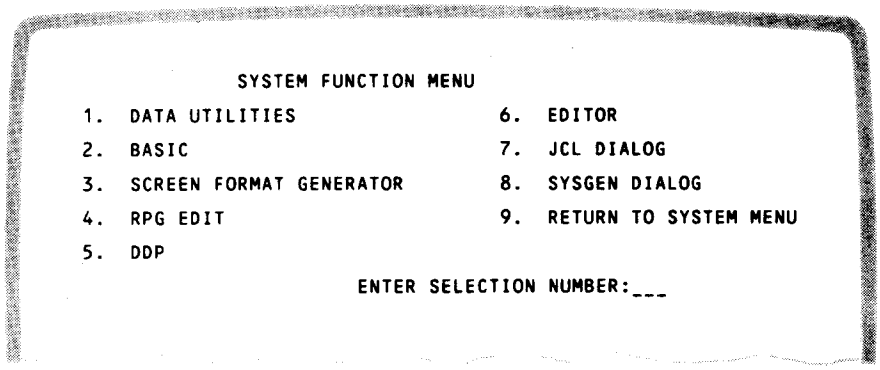


Figure 3–1. OS/3 Menu Screen

Menus also provide assistance for users unfamiliar with the system or a particular item on a menu. Menus can provide HELP screens, screen displays that explain menu choices to the user. Figure 3–2 shows a HELP screen explaining the programming language BASIC, one of the choices on the menu shown in Figure 3–1:

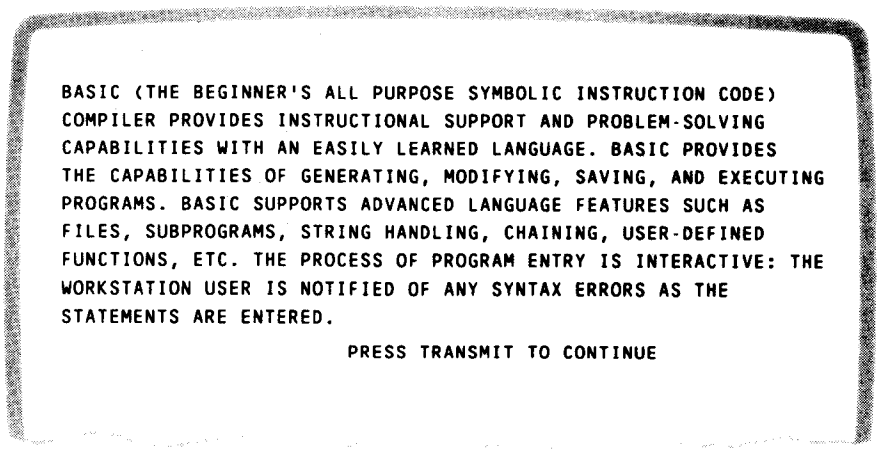


Figure 3–2. HELP Screen

OS/3 Menu Services consists of two components, the menu processor and the menu generator. Each of these is discussed further in the following subsections.

Sperry Univac supplies system menus with OS/3. The menu in Figure 3–1 is such a menu. These menus give users lists of system programs and interactive services commands that they may easily choose and execute.

Sperry Univac also supplies a utility that allows users to create their own menus for use in their own application programs. That component is the menu generator.

3.4.1. Menu Processor

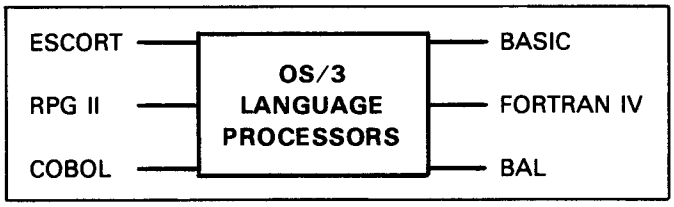
The menu processor retrieves and displays menus on workstation or terminal screens, handles user input from the menu, and arranges for the execution of whatever program or command the user chooses. The menu processor also keeps track of multiple menu sequences, keeping the menus displayed in the order specified, and responding to user commands to display other menus.

3.4.2. Menu Generator

The menu generator enables the user to create, modify, and display menus. The user is engaged in a dialog to create the menu. He is able to create not only the menu, but HELP screens for all or some of the items offered by the menu. The menus the user creates can cause programs and interactive services commands to be executed, can call other menus, and can return constant data to a program.

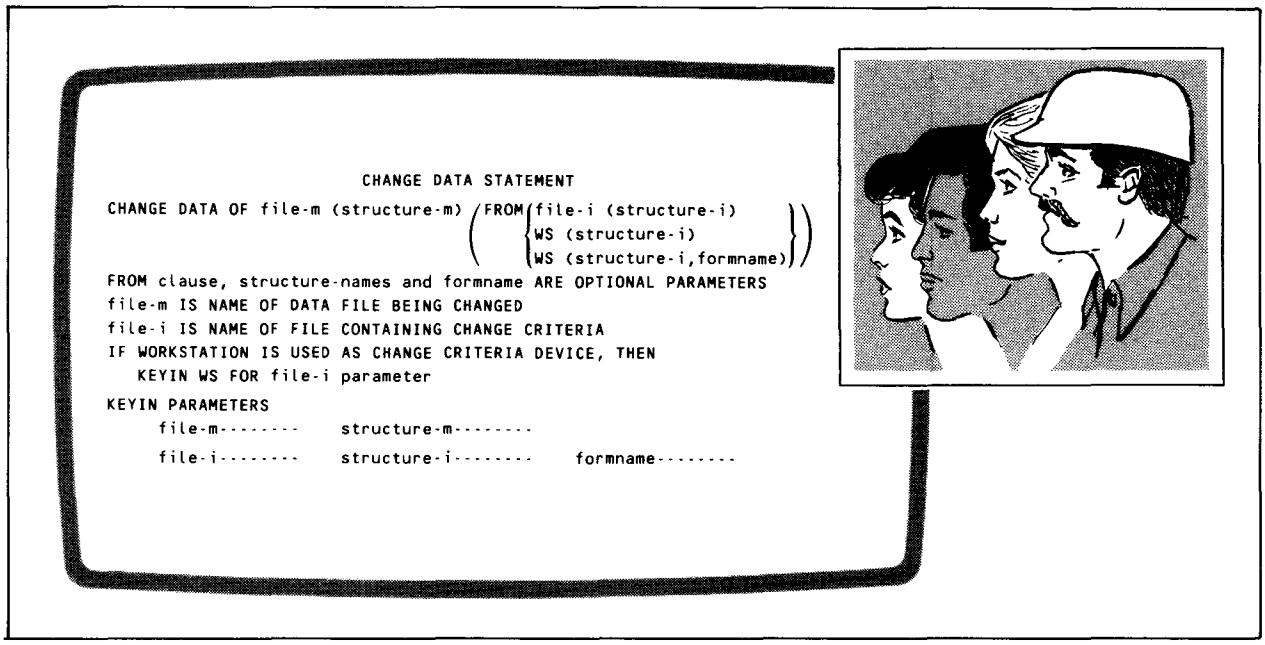
3.5. LANGUAGE PROCESSORS

Six language processors allow flexibility in preparing programs for use with OS/3:



3.5.1. ESCORT

ESCORT is a high-level, interactive programming language used to generate file processing and data handling programs. It is easy to use, and requires only a minimum of training, yet has the power of more complex data handling languages. The ESCORT language employs English verbs and clauses to express programming operations. Thus, even a programming novice can quickly begin producing useful programs. Here is a typical ESCORT screen display:



ESCORT can be used to deveop programs for report generation, data entry and retrieval, and transaction processing. An ESCORT program can create, sort, or merge files; add, delete, sort, or replace records within a file; issue prompting messages to a workstation for input data and validate the input data. Computational operations available include addition, subtraction, multiplication, division, and modulo, a division operation in which the result is the remainder only.

ESCORT programming sessions are conducted directly at the workstation, and the ESCORT compiler employs prompting messges for data entry. The amount of prompting used depends on the programming expertise of the user.

The novice user can conduct a session in interactive tutorial mode. In tutorial mode, ESCORT uses a question-and-answer technique to lead the user through the various steps involved in creating an ESCORT program. When the user is unsure of how to proceed, he may request HELP displays, which explain the various choices offered by the question-and-answer display screens.

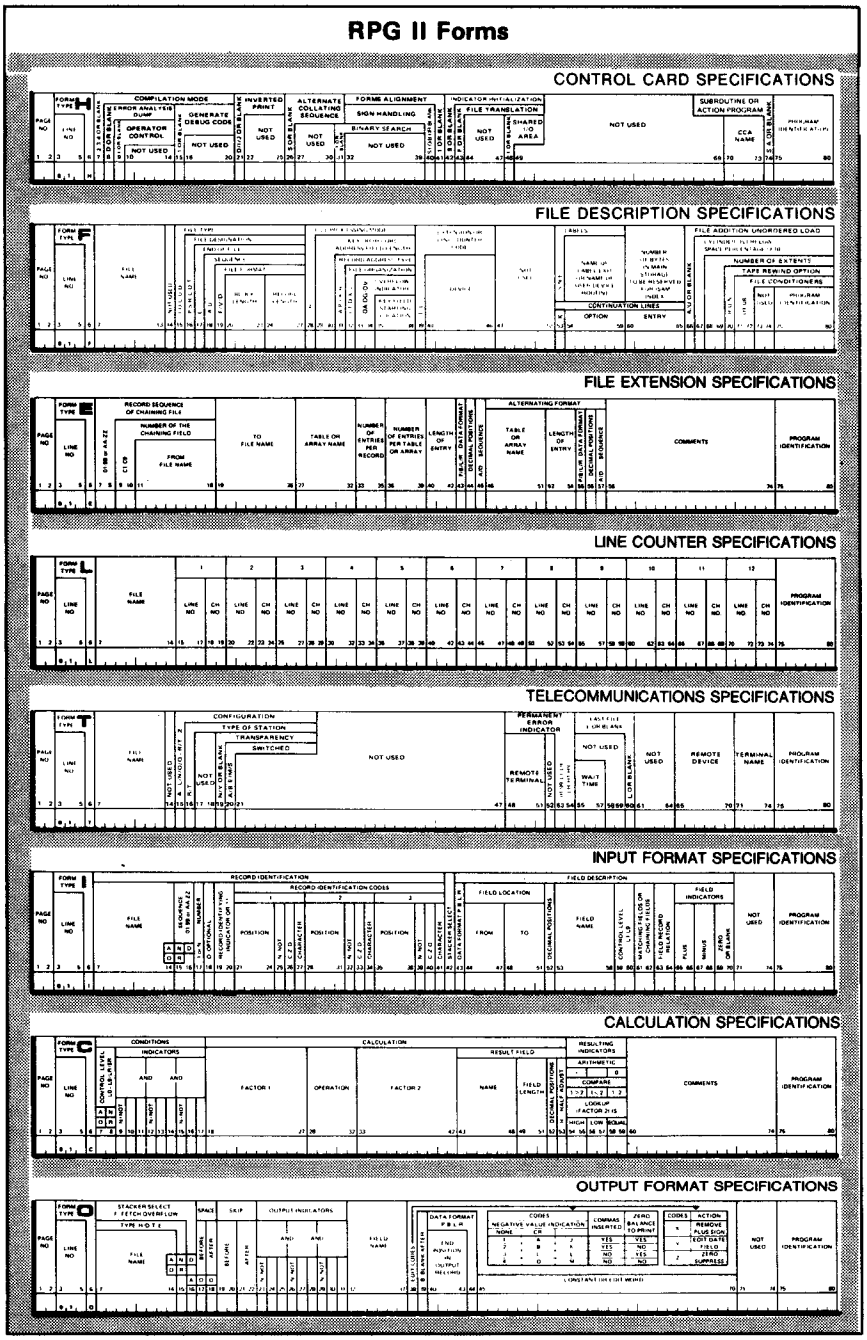
The experienced user can conduct a session in interactive program mode, allowing faster creation of ESCORT programs, with less prompting. The experienced programmer can use program mode in two ways. He can create programs by filling in the blanks on menu screens displayed by ESCORT. Or, he can enter ESCORT statements directly onto a blank screen. This is the quickest way to create an ESCORT program. The compiler offers immediate syntax checking of ESCORT statements and sends messages when more extensive errors are detected. These errors can be corrected immediately through the workstation.

ESCORT employs the common data interface for data transfers, and all data files are MIRAM files. In addition, field names and attributes for the records within a file are contained in control data for the file. Thus, the programmer is freed from knowing detailed data descriptions for existing files and is able to concentrate more fully on how to express the actions to be taken to solve his problem.

3.5.2. RPG II

The OS/3 report program generator (RPG II) is a processing language that generates an object program from a series of interrelated specifications. RPG II provides the basic logic of the program with the RPG II cycle. Through the various RPG II specifications forms, the user can define records and indicate the operations to be performed on these records. Each record is then processed by the same RPG II program cycle, and the user-specified operations are performed, provided the conditions specified for the operation are satisfied.

To define a report program for generation, the requirements are listed by the programmer on specifications forms.



The information coded on these forms becomes the actual input to the system. The compiler generates an object module for input into the linkage editor; the subsequent output is a loadable program module. The system can be instructed to display the RPG II formats on the workstation screen and the user can file the proper entries. Users may also enter unformatted RPG II source statements. Enhancements have been included in RPG II that enable the user to develop more sophisticated report programs within the existing framework of the language. These enhancements are:

- Telecommunications interfaces are supplied via an RPG II specifications form offering extensive communications capabilities to the RPG II user.
- Eight control-stream user switches (indicators) that may be used to condition calculations, input files, output files, or specific output records.
- IMS action programs can be written in RPG II.
- Workstations are accessible through programmed operations.

Sperry Univac offers two facilities to simplify RPG II source program entry:

- RPG II editor
- Auto report facility

The RPG II editor is an extension of the general editor and is used specifically to create and maintain RPG II source programs. The RPG II editor is initiated at the workstation and provides a number of ways for the user to enter and update the source programs. The experienced RPG II user can indicate at the start of the session that he will be entering the source statements in a free-form format. The less experienced programmer may find it more advantageous to use the positional format. In this format, the columns that must be filled in are displayed. The novice user can take advantage of the specification format display. If this display is used, the standard RPG II specifications format is displayed along with detailed explanations of their uses and all possible entries. The source programs generated can be stored in the program library that the user chooses and can be redisplayed in either of the formats for updating.

The RPG II auto report facility accepts simplified specifications and standard RPG II source statements and generates a complete RPG II source program. By simplifying programming, auto report increases programmer productivity.

3.5.3. COBOL

Common Business Oriented Language (COBOL) is a programming language oriented toward problems in business applications. The language is similar to the English language, rather than a notation that considers the technical aspects of a particular data processing system. The source programs are easily transferable among systems that accept *American National Standard COBOL X3.23-1974.* Each of these systems provides a COBOL compiler to translate the COBOL source program into a machine-oriented object program. The ability to advance from one generation of equipment to another in a logical, orderly, and rapid manner is assured through this limited machine dependence. Source programs written in COBOL consist of four major divisions:

COBOL DIVISIONS

IDENTIFICATION

This division contains information identifying the source program and the output of a compilation; the author, installation, and so forth also may be identified.

ENVIRONMENT

This division specifies a standard method of expressing those aspects of a data processing problem that are dependent on the physical characteristics of a specific system; also, it permits specification of the compiling system hardware characteristics, input/output control techniques, and so forth.

DATA

This division describes the data that the object program is to:

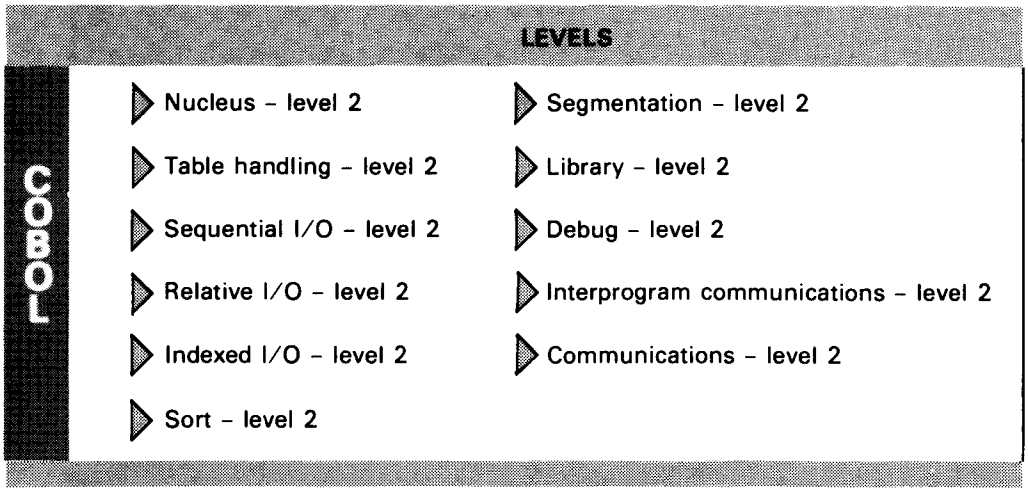
- accept as input;
- manipulate;
- create; or
- produce as output.

The division is further divided into sections to facilitate the description of data contained in input or output files or developed during the program, or present as constant information to be used in the object program.

PROCEDURE

This division describes the logical steps that must be taken in the solution of the data processing problem.

The levels of *American National Standard COBOL X3.23—1974,* modules are:



To simplify COBOL source entry, Sperry Univac supplies the COBOL editor. It, like the RPG II editor, is an extension of the general editor and allows the user to create and maintain COBOL programs from a workstation.

Both novice and experienced programmers can use the COBOL editor. The novice can create programs by using specification format displays, which act as fill-in-the-blanks screens. The more experienced user can create programs by using either the positional or free-form formats. These offer very little prompting, but enable the user to enter program statements very quickly. The COBOL editor allows the user to store the programs he creates and to display already created programs for updating.

3.5.4. BASIC

The Beginner's All-purpose Symbolic Instruction Code (BASIC) language is an interactive programming language designed to be easy to use yet meet the requirements of both business and scientific programming. The BASIC language available on the OS/3 operating system complies with the *American National Standard Minimal BASIC, X3.60-1978* and includes Dartmouth features and compatibility. It provides a powerful, but simple set of commands that allow the novice to learn the language quickly, and yet gives the experienced programmer an extensive list of features for various applications.

BASIC is an interactive language, and all source statements can be entered directly at the workstation with the results and error messages displayed on the screen. All source statements are checked for syntax errors as they are entered, and a message appears on the screen if a line is in error. The BASIC source program can be compiled directly at the workstation, and compilation errors can be corrected immediately. During an interactive BASIC session, users can input, modify, execute, and save programs.

The OS/3 BASIC compiler has facilities for arithmetic operations, data file processing, matrix generation and processing, and logical operations. Subroutines and string operations may be used in a BASIC program.

3.5.5. FORTRAN IV

FORTRAN IV is a programming language designed for performing mathematical computations required for solving engineering and scientific problems. FORTRAN IV is also useful for many nonscientific data processing applications.

FORTRAN IV is designed so that the user can express an algorithm in a way natural to the problem. The user requires minimal considerations of the particular characteristics of the system in which the program is executed. Procedures defined outside the FORTRAN IV program, and possibly written in a language other than FORTRAN IV, can be referenced by name and thereby be made an implicit part of the program.

FORTRAN IV is a proper superset of the *American National Standard FORTRAN (X3.10–1966)*. It is also a compatible superset of the IBM/DOS 360 FORTRAN IV. This system features code optimization, high-performance I/O, and extended functional capability.

3.5.6. Basic Assembly Language (BAL)

OS/3 assembly is a versatile and detailed symbolic language. The combination of a macro facility and the ability to handle procedural directives gives the assembly language unique capabilities. Each instruction within the language is assigned a mnemonic to denote the particular hardware function performed.

The symbolic format for writing an assembly language instruction, macroinstruction, or procedural call consists of three basic symbolic fields; use of the fields requires conformity to simple rules so that efficient translation of symbolic code to object code can be performed. The fields are:

- the label field, which may contain a symbolic name used to provide an entry point or a label for a block of data or a block of instructions;
- the operation field, which must contain a mnemonic instruction code or the name of a macroinstruction or procedural call; and
- the operand field, which provides for a variety of uses ranging from simple to complex specifications.

Combining names, parentheses, arithmetic operators, and self-defining terms into operand expressions makes possible the use of highly sophisticated coding sequences. Operand expressions gain power by being able to include location counter references and literals.

A wide range of data types is provided for constant generation and storage definition. Binary, hexadecimal, decimal, fixed-point, floating-point, and character formats can be used to specify absolute values in the source code.

Output from the assembler run consists of a complete listing of symbolic coding, generated object coding, diagnostic messages, and a cross-reference listing. A relocatable object module is produced that is suitable for linking to other modules prior to loading for subsequent execution.

The assembler recognizes a set of directives that can be used to direct its operation. These directives allow the user to control program sectioning, base register assignment, the format of the output listing, sequence checking, and other auxiliary functions.

The assembler includes a macro facility that can reduce the effort required to write patterns of coding, either repeated in one program or common among several programs. The flexibility of the macro facility allows a macro to be written so that the pattern of coding generated can vary widely, depending upon the parameters supplied with the call. Macro definitions may be specified in two formats: macro and proc. The elements of each format type may not be mixed within a definition; however, definitions of both types are permitted within a program. OS/3 provides the user with a comprehensive selection of system macros that interface with data management, the supervisor, and other elements of the software.

3.6. THE GENERAL EDITOR

The general editor provides the capability to interactively generate and edit source programs, data files, and job control streams. It can also be used to create and edit textual information. The editor responds to a set of commands designed specifically to aid in the generation and formatting of text. The general editor can also store the created material in a program library or a data file. The editor command repertoire provides the user with the ability to:

- Create, add to, delete from, and modify text
- Create, modify, and merge files
- Copy files and texts

The editor offers file protection facilities that prevent a file being modified by the editor from being inadvertently destroyed or incorrectly altered either by direct action of the user or through some system failure.

The general editor can operate in two modes: line mode and screen mode. Each mode offers the user full editor functionality.

In line mode, the user creates or edits material one line at a time. After each line is created or edited, the user transmits the line of information to the editor, which places that line in a work file created for that session of editor use. Lines of material, displayed to the user one at a time, roll up the screen as he either creates or edits more lines.

In screen mode, the user may enter from 1 to 14 lines at a time. He may use special formatted screen displays for entering source programs in the COBOL, FORTRAN, and RPG II programming languages. He may also use a free-form screen display. The free-form display provides tabs, which the user may set for uniform entry of data or to code assembly language programs. The free-form display also provides the user with the ability to create data files without having to run a user or system program, simplifying the process of bulk data entry.

The general editor offers another aid to users creating programs, the Error File Processor (EFP). The EFP lets the user see errors in his source program displayed on the workstation screen immediately after his program has been compiled, without having to wait for a printout of the compiler error listing. The user can then correct those errors in his source program by using the general editor.

The EFP works by reading an error file created at the user's specification during the compilation of his program. It also reads the file in which the user's source program is stored. It then displays each error message in the error file with the line or lines of source code in which the error was found. The user can thus see his mistakes and quickly correct them by using the general editor.

3.7. INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM)

As with all the OS/3 software, the design emphasis for the integrated communications access method is on ease of use for the casual as well as the major real-time communications user. A wide range of system options provides support for users committed to a real-time multijobbing environment without penalizing those whose needs are more modest. ICAM is available for workstation support as well as for terminals supported over a communication line through the ICAM Terminal Support Facility.

ICAM offers the communications user these features:

ICAM FEATURES

MESSAGE QUEUEING

Message queueing stacks complete messages in main or disk storage while they wait to be serviced by a remote device handler or a message processing program. A single message queue consists of one or more messages with their header segments linked together. The message text that overflows the header segments is contained in additional segments, which have secondary links out of the header segment. A network may contain one or more message queues associated with lines, terminals, process files, or program-related queues called LOCAP (LOCal APplication) files.

A user, when defining the network configuration, may select line or terminal message queueing. Terminal queueing results in a single queue being defined for each terminal on a communication line. Line queueing results in a single queue being defined for all terminals associated with a line.

MULTIPLE DESTINATION ROUTING

ICAM provides for the multiple routing of messages. A single message may be routed to up to 255 destinations. The message is released from its queue only when all deliveries have been made. It is up to the user to ensure that the message text sent to each device is appropriate to the particular device.

ACTIVITY SCHEDULING AND PRIORITY CONTROL

ICAM provides an activity scheduling routine which, at the user's option, can also provide priority suspension and scheduling services. Small systems that do not require suspension capability may have only a single level of scheduling. Large systems with critical timing requirements are provided with multiple-priority-level scheduling and suspension capability.

TIMER SERVICE

A centralized timing service for control of active data buffers and for scheduling an activity is provided for use by all ICAM software elements. The timer service also enables the user to specify the time interval for polling communication devices.

CHECKPOINT/RESTART

Restart procedures can reconstruct message queues to their status at the point of system hardware or software failure. The reconstruction involves only complete messages; incomplete messages must be reprocessed after the recovery operation. Status messages are transmitted to remote terminals to effect an orderly restart and provide assurance that messages are not lost in the system. Restart is supported only in conjunction with a checkpoint and journaling feature. Checkpoint results in the complete network definition being written on a tape or disk file.

JOURNAL CONTROL

Journal files are required when a user of ICAM requires a recovery/restart capability, or when a statistical accounting of the network operation is required. The journal control routine provides for the recording of message processing events and message data on a disk or magnetic tape file. To reduce the number of I/O accesses required, each journal entry is transferred into a larger buffer area for staging prior to being written on the output device.

STATISTICS ACCUMULATION

ICAM maintains statistics that reflect the operating status of the communication environment. These statistics are maintained by line and by terminal and are available to a user program upon request. The accumulation contains totals for types of information, such as:

- **Number of messages received**
- **Number of input retransmission requests**
- **Number of messages transmitted**
- **Number of output transmission requests**
- **Number of poll messages**
- **Number of no-traffic responses**

These statistics are maintained in the network definition and provide the user with information not generally available from journal files.

PUBLIC DATA NETWORK SUPPORT

ICAM supports the following public data networks:

SWITCHING	NETWORK NAME	LOCATION
Packet	DATEX-P	W. Germany
Packet	DATAPAC	Canada
Packet	TRANSPAC	France
Circuit	NORDIC	Sweden
Circuit	DATEX-L	W. Germany

ICAM FEATURES

DYNAMIC BUFFER EXPANSION

ICAM automatically acquires additional network buffers as its need for more buffers increases. The user is not required to respecify the amount of buffer space needed.

At system generation time, the individual user can choose the level of support needed to meet the site requirements. Levels of support are arranged by combining the appropriate ICAM library elements. The levels are named for the interface between the user's message processing program and the ICAM modules combined into a loadable software module. Each interface contains its own unique set of macroinstructions.

ICAM INTERFACES

STANDARD INTERFACE

The standard interface is a conventional GET/PUT interface for communications that automatically queues input and output messages in network buffers. In this interface, the processing program requests data from a process file or input queue and not directly from the line. Conversely, output is placed in a destination queue or output queue for transmission by ICAM. An extended set of macroinstructions known as message processing procedure specifications are optionally available to this interface for additional processing of messages.

TRANSACTION CONTROL INTERFACE

The transaction control interface provides the communications facilities needed by the information management system (IMS). It is available only to IMS. After defining the communications network, the operation of the transaction control interface is completely transparent to the IMS user.

ICAM is a modular software package capable of supporting either simple or complex communications environments. A single ICAM configuration can provide concurrent support for multiple user programs that handle a variety of terminals and line types.

ICAM prevents conflicting facility assignments and releases facilities when jobs terminate. User programs are provided with macroinstructions that perform the following services:

- control table generation;
- data transfers to and from user-specified buffer areas;
- communication facility initialization and control; and
- dynamic terminal and poll table entry alterations in the communications control areas.

The OS/3 components that make up ICAM are:

ICAM COMPONENTS

CHANNEL CONTROL ROUTINE (CCR)

The CCR provides the physical input/output interface to single line communications adapters (SLCA) and the specific type of communications subsystems.

REMOTE DEVICE HANDLERS (RDH)

The remote device handlers provide the software logic and control required to interface the unique characteristics of specific remote devices to the other ICAM components.

RDHs are available to accommodate a wide range of terminal equipment, including the following SPERRY UNIVAC equipment: local(for IMS) and remote System 80 workstations, UNISCOPE 100 and 200 Display Terminals; UTS 400 Universal Terminal System; UTS 4000 Universal Terminal System, 1004 Card Processor System, BC/7 Business Computer, UDS 2000 Remote Data Entry System, V77 minicomputer systems; and TELETYPE* teletypewriter models 33, 35, 37 and 38. Binary synchronous communications procedures are also supported.

Device handlers provide all device-dependent functions that are required to permit other ICAM components to function independently of the terminal device. These device-dependent functions include:

- Station and device polling where applicable
- Error detection and correction
- Code translation
- Accumulation of operation statistics
- Compression and decompression of data where necessary

COMMUNICATIONS NETWORK CONTROLLER (CNC)

The CNC coordinates message flow between the remote device handlers and either main storage or a disk-based message queue. The CNC is the ICAM component that places incoming messages on their appropriate processing queues or submits them to special system functions (the user program interfaces are described later) for disposal. It provides the interface to user-specified message processing procedures specification (MPPS) routines and user-generated routines. The CNC also detects the presence of a message on an outgoing (destination) queue and provides the orderly transmission of that message.

Additional functions performed, either directly or indirectly, by the CNC include:

- providing the control to dynamically modify a network in accordance with a changing operating environment;
- monitoring the orderly shutdown of ICAM during end-of-job processing (this service is extended to both ICAM and user programs); and
- scheduling the message user service transcriber (MUST) routines when new activity is detected and there is an outstanding request waiting.

Trademark of the Teletype Corporation.

DISTRIBUTED COMMUNICATIONS ARCHITECTURE TERMINATION SYSTEM (DCA TS)

The DCA TS serves as coordinator of the flow of messages between local and remote message queues. DCA TS places incoming messages in appropriate message queues or submits them to special system functions for processing. It also ensures the orderly transmission of outgoing messages by managing outgoing message queues. The DCA TS also protects messages from being lost when communications links break down and provides data throttling and recovery services.

COMMUNICATIONS CONTROL AREA (CCA)

A CCA contains all the tables required to define and control a specific network configuration. CCAs can be tailored to specific needs.

ICAM controls a network by setting indicators and flags in the line, terminal, and queue tables within the CCA. These flags control polling, indicate the operational status of communication hardware, and reflect the current disposition of message queues.

Each CCA contains a pool of network buffers that are under the control of ICAM. Incoming and outgoing messages are temporarily staged in these network buffers during their active transition through the system. Buffers from this pool provide the base when main storage queueing is specified. When disk storage queueing is specified, these network buffers provide intermediate storage during the active input or transmit phase of a message.

MESSAGE USER SERVICE TASK (MUST)

The MUST routine provides a message staging service that isolates a message processing program from the device dependence that is usual in data communications programs. MUST copies input message data into designated work areas in the message processing program from the network buffer pool. It is also responsible for copying output message data into the network buffers from designated work areas in the program.

Variations of a MUST routine provide support for specific user program interfaces, such as remote batch processor (RBP), RPG II, and IMS. The MUST routine isolates these functions from the CNC and device handlers, obtaining the maximum commonality of ICAM components.

DEFERRED USER SERVICE TASK (DUST)

The deferred user service task component is a series of ICAM overlays that performs functions that are not time-dependent or that are used infrequently.

DUST performs such functions as:

- ICAM initialization
- Line connect and autodialing
- Network initialization
- Program termination
- Subsystem parameter loading
- System console message capability

GLOBAL USER SERVICE TASK (GUST)

GUST allows the formation of a global, or nondedicated, communications network permitting several user programs to access the network concurrently. The GUST routine controls network and line requests and releases in a global environment.

In addition to providing a communications interface between a terminal network and System 80, ICAM can provide the following interfaces.

ADDITIONAL ICAM INTERFACES

RPG II TELECOMMUNICATIONS

RPG II users are provided with a simplified remote I/O capability through RPG II telecommunications and the ICAM direct data interface. Support for RPG II telecommunications is extended to all the devices already listed under remote device handlers, as well as the IBM System/3, the IBM System 360/20 (BSCA equipped), and the IBM System 360 (with OS or DOS BTAM binary synchronous communications support).

COBOL MESSAGE CONTROL SYSTEM (CMCS)

Users can write COBOL programs that access and use the resources of their ICAM system. COBOL programs interface with ICAM through the COBOL message control system (CMCS), which handles the flow of control information and data between COBOL programs and ICAM. CMCS translates the COBOL communications verbs into ICAM instructions and translates ICAM error and status conditions into COBOL format. Its access to ICAM is through the standard interface. Message processing procedure specification features can be included in COBOL communications programs.

REMOTE BATCH PROCESSING (RBP)

ICAM provides an RBP capability that permits jobs to be entered into the computer system from a variety of remote terminal devices. The software components that perform the RBP functions operate as system tasks and are referred to as symbionts.

These symbionts may be configured as input only, output only, or input and output. In small computer installations, the execution of the input symbiont, the user's job, and the output symbiont occurs serially. In larger systems, the entry and execution of a specific job are serial, but more than one job may be processed concurrently.

Remote symbionts may be loaded and initialized manually via the system console, or these remote jobs may be processed automatically in response to a call from a remote station. An instruction set is available to send job requests, activate workstations, defer processing, obtain status, and deactivate workstations.

IBM 3270 EMULATOR

The IBM 3270 Emulator provides a method for connecting the OS/3 System 80 to an IBM host system. It allows System 80 workstation users to access applications and IBM program products running on an IBM host. To do this, System 80 emulates, or pretends to the IBM host, to be a 3270 terminal system. The 3270 emulator:

- Acts as an IBM 3271 control unit, providing communication between System 80 and the IBM host system
- Supports both user programs and System 80 locally connected workstations

REMOTE TERMINAL PROCESSOR (RTP)

The Remote Terminal Processor permits a System 80 to act as a remote job entry terminal to one or more IBM host processors. RTP enables System 80 to send jobs to an IBM host, send and receive files on tape, diskette, or punched cards, and send and receive messages to the host processor.

RTP simulates an IBM multileaving workstation by using the Houston Automatic Spooling Program (HASP) protocol through binary synchronous communications (BSC) facilities. It interfaces with the following IBM software systems:

- Job Entry System 2 (JES2)
- Job Entry System 3 (JES3)
- Asymmetric Multiprocessing System (ASP)
- Houston Automatic Spooling Program (HASP)

3.7.1. NTR System Utility

The nine thousand remote (NTR) system utility gives a user the ability to have System 80 act as a remote terminal to a SPERRY UNIVAC 1100 Series system. The NTR is a combination system utility and job task that may run concurrently with other job tasks. It makes use of a modified version of the direct data interface. NTR provides data transmission control, operator-to-operator communications, and control features for status and command functions between both systems. The NTR utility controls local input and output tasks through use of a combination of data management macroinstructions and a set of special NTR macroinstructions for user own code control of device buffers.

3.8. INFORMATION MANAGEMENT SYSTEM

The information management system (IMS) is an interactive transaction processing system. IMS is ideal for users who have a need for rapid access to up-to-the-minute information. Information retrieval and updating can be performed by personnel with no formal data processing training.

IMS is transaction oriented. This means that processing is triggered by an input message from a remote terminal (either a display terminal or a teletypewriter). Applications programs, called action programs, process the input message, access data files, and display an output response at the terminal.

The user creates his own online IMS system in a configuration process in which he defines the communications network, user files, action programs, and optional IMS features. Any number of different IMS load modules can be configured. This allows the user to tailor each online IMS system to a specific application and conserves main storage. The configuration process is simplified by an IMS configuration job control procedure and by the use of default parameters.

IMS provides a collection of action programs called UNIQUE, which retrieves, updates, and displays data via a simple inquiry/update terminal language. To use UNIQUE, the user first creates *defined files* from elements of existing files using an IMS utility called the *data definition processor.* A password capability is provided with UNIQUE, which allows the user to limit access to data files or to certain elements within those files.

For specialized file processing or message formatting needs, IMS supports user-written action programs in COBOL, RPG II, and basic assembly language (BAL). Action programs operate under the control of IMS and access files via function calls to IMS. Programming is simplified because IMS handles all communications and file I/O functions. IMS software allocates resources, and schedules action programs, to process each transaction. IMS provides file protection via a record locking facility and online and offline recovery.

IMS can run in a multijobbing environment because IMS is executed as merely another program under control of OS/3.

IMS can operate in a distributed processing environment through the IMS-DDP transaction facility. Through it, the IMS user can define and perform transaction processing between interconnected OS/3 data processing systems supporting IMS.

3.9. DATA BASE MANAGEMENT SYSTEM

The data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs handle the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of DMS data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

DMS FEATURES

PHYSICAL DATA BASE CHARACTERISTICS

The Device Media Control Language (DMCL) uses a COBOL format to describe physical characteristics of a data base and data dictionary by supplying page and area sizes within the data base, the number of data base buffers, recovery options, and file identification.

LOGICAL DATA BASE CHARACTERISTICS

The Data Description Language (DDL) uses a COBOL format to describe the logical characteristics of a data base. The schema DDL describes the entire data base, and the subschema DDL describes the portion of the data base used by one or more application programs. Thus, many different subschemas may exist for the same data base depending on the specific collection of data required by different application programs using the data base.

STORAGE STRUCTURES

Data items are organized into record types, and record types are organized into logical relationships called sets. Records are grouped into three basic logical data structures: sequential, tree, and network, according to record set linkages. In addition, data is organized physically by record types within pages, and pages are organized within areas that are subdivisions of storage. Users can control physical placement of records in a data base and control access to the records by assigning a location mode and set order. This capability increases access efficiency.

MANIPULATION OF DATA

The data base can be accessed by writing data manipulation language (DML) statements in the procedure division of a COBOL application program. COBOL programs containing DML are processed by a DMS preprocessor to produce American National Standard COBOL source programs that can be subsequently compiled by the COBOL compiler.

The DML contains statements that call a specified subschema, establish contact with the DBMS, open and close areas of the data base, find data base records and get them into working storage, store new records, modify and delete existing records, insert and remove records from set occurrences, save current information, and test for set membership.

SYSTEM SUPPORT FUNCTIONS

System support functions include three types of utility processors: generation, report and statistics, and recovery utilities. Users control these utilities via the utility processor language (DUPL) commands. Some DMS utility activities include data base initialization, dump and restore, and data base recovery.

IMS/DMS INTERFACE

IMS action programs can access DMS data bases via data manipulation statements embedded in COBOL action programs. Also, DMS data bases can be used to build a defined file accessible via COBOL, RPG II, or BAL action programs, or UNIQUE.

RECOVERY

A journal file is generated by the DMS to support the operation of offline recovery utilities. The offline recovery utilities can perform forward recovery of a data base to its final state or backward recovery to any selected run-unit checkpoint. Online recovery automatically restores a data base after a system crash or abnormal program termination.

3.10. DISTRIBUTED DATA PROCESSING

OS/3 distributed data processing permits the user to distribute files between remote data processing systems, engage those systems to cooperatively process jobs, allow those jobs to access files and other programs on remote systems, and perform transaction processing between remote systems.

These functions are accomplished through three software packages: the DDP transfer facility, the DDP file access, and the IMS-DDP transaction facility.

3.10.1. DDP Transfer Facility

The DDP transfer facility allows the user to view each system in the DDP network as an available resource for scheduling and executing his work. Using simple commands, the user can initiate job distribution and file transfer within the system without concern for the requirements of the hardware and software of each system, or the communications protocols needed to initiate and monitor the distribution of a job. The DDP transfer facility consists of two pieces of software, the job transfer manager and the file transfer manager.

The job transfer manager allows the user to initiate execution of a job on any system within the DDP network and to monitor the execution from the initiating site. Any printed or punched output generated by the job is normally routed to the initiating system. However, the initiator can request routing of this output to any available system in the network.

The job transfer manager provides commands to:

- Submit jobs to the DDP network
- Monitor the execution of submitted jobs
- Cancel a submitted job
- Communicate with a remote operator's console
- Issue instructions to a remote operating system
- Respond to messages issued by a job executing on a remote system

The file transfer manager provides the capability to transfer sequential files from one system to another or to duplicate file structures between systems.

Files can be transferred between systems that contain the same file handling facilities, in which case no data transformation is required. Files with serially-accessible records may be transferred between systems of dissimilar architecture with or without character conversion. That is, a file can be transferred as a bit stream to be reformatted by the user, or it can be converted (character translation only) during the transfer process to the internal code recognized by the receiving system.

The file transfer manager does not support automatic reformatting of records within a file, or converting records to their equivalent form in the destination system, when transfer is between systems of dissimilar architectures. In this case, the contents of the file are treated as all character data or bit string data. Neither record sensitivity, record sequence, nor numeric field characteristics are recognized or adjusted during the transfer operation.

File transfer can be directed into the file area of the destination system or, if the destination file area is in use, to a temporary file area in the destination system. Transfer from the temporary file to the destination file is made when the destination file becomes available.

FILES TRANSMITTED BETWEEN SYSTEMS	
DATA FILES	PROGRAM LIBRARIES
Data formatted in the OS/3 MIRAM format. Files transmitted from OS/3 are MIRAM files; files transmitted to OS/3 will be created in MIRAM format.	Any directly-accessible module in a program library can be transferred between systems, or an entire library can be transferred.

The file transfer manager permits the user to:

- generate a file directory to catalog the characteristics of files;
- transfer copies of data files and program libraries;
- delete a file from the file directory; and
- obtain a listing of a file's characteristics from the file directory.

3.10.2. DDP File Access

The DDP file access permits a user program to, through OS/3 job control language, access files residing on remote OS/3 systems within a DDP network. DDP file access also permits user programs to exchange data and control information. The user can transfer program control between systems and reroute spooled output.

The DDP file access permits:

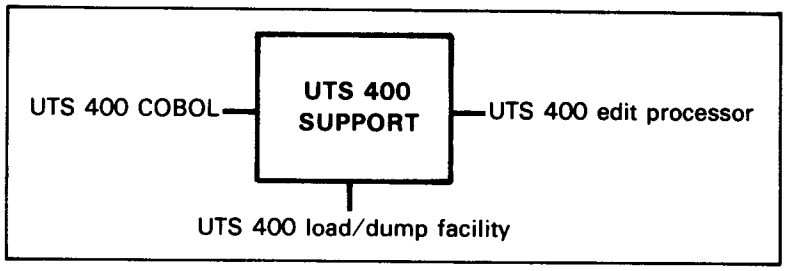
- user programs to access disk files residing on a remote OS/3 host through the // DVC job control statement;
- programs running on remote hosts to access each other as files;
- the connection of workstations to jobs running on either local or remote hosts;
- the sending of job control messages, such as PAUSE messages, between local and remote hosts;
- the use of the ORI job control option statement to transfer program control to another workstation on either a local or remote host; and
- the rerouting of print or punch output files to different host computer systems. Also print output files may be routed to remote auxiliary printers connected to remote workstations.

3.10.3. IMS-DDP Transaction Facility

The IMS-DDP transaction facility permits the user to perform transaction processing between remote systems. The transaction facility supports the transfer of transactions between systems operating OS/3 IMS. The transaction facility provides the user with an easy and flexible way to specify his particular DDP transaction needs through parameter information supplied when he configures IMS.

3.11. UTS 400/4000 SUPPORT

OS/3 offers the following software components to support the SPERRY UNIVAC UTS 400 Universal Terminal System (UTS 400) and the SPERRY UNIVAC UTS 4000 Universal Terminal System (UTS 4000):



3.11.1. UTS 400 COBOL

UTS 400 COBOL is a high-level, business-oriented international language with features to complement the capabilities of the UTS 400 and UTS 4000 terminals. The UTS 400 COBOL compiler meets the ANSI X3.23-1974 standards and the ISO recommendations on COBOL. The compiler also contains extensions to provide for interactive data entry, program control, and screen management. The UTS 400 COBOL compiler executes under control of OS/3 and produces a compiled program that can be downline loaded to the terminal or placed on a diskette.

3.11.2. UTS 400 Edit Processor

The UTS 400 edit processor allows the user to create and manipulate text data on a diskette file. It provides an easy and efficient way to create and update line-oriented files of data. Lines may be inserted, replaced, deleted, or changed in any order.

The edit processor permits functions such as string searches, line deletes, insert or delete lines, moves, and print. Error messages advise the user of mistakes made in entering data. Lines up to 118 characters in length may be inserted in an edit processor file.

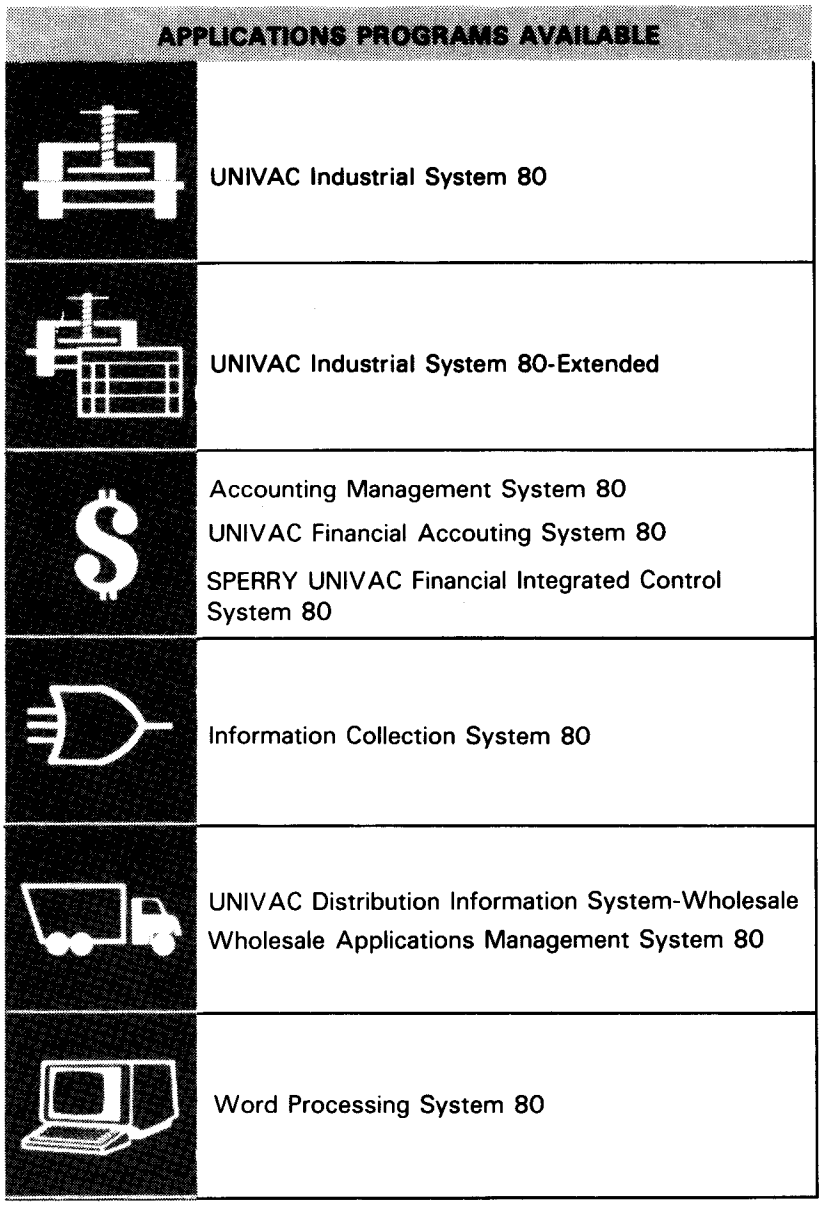
3.11.3. UTS 400 Load/Dump Facility

The UTS 400 load/dump facility provides the following facilities for the use of the UTS 400 COBOL and edit processor:

- a communication program to extract a load module from an OS/3 library and to transmit this module (downline load) to a remote UTS terminal; and
- a communication program to accept UTS 400 dump information and store it on an OS/3 dump file.

3.12. APPLICATIONS PROGRAMS

Sperry Univac offers a wide variety of applications programs with the System 80 data processing system. These programs are designed to meet the data processing requirements of particular applications. The list of available packages varies as Sperry Univac institutes changes to meet the needs of its customers.



3.12.1. UNIVAC Industrial System 80

The UNIVAC Industrial System 80 (UNIS 80) is a comprehensive manufacturing production and inventory planning and control system. UNIS 80 provides the facilities to meet the requirements of a total manufacturing control system, including features for production scheduling, inventory control, and material handling. It is an easy-to-use, interactive system that permits even the most inexperienced personnel to rapidly become proficient in system operation. Terminal screen displays, called menus, lead the user through the system, thus making it available to any member in the organization. UNIS 80 interfaces with an integrated data base and permits multiple interactive users to access the system simultaneously as well as providing noninteractive, or batch, support. UNIS 80 is modular and consists of the following application modules.

PRODUCTION ENGINEERING DATA CONTROL

This feature collects, maintains, and retrieves all basic engineering and production data, used as primary information and as a basis for other applications in manufacturing. It handles bills of material, parts data, manufacturing routings, and work center data. The user is provided with:

- Part engineering data
- Single-level bills of material
- Indented bills of material
- Summarized bills of material
- Manufacturing routings
- Tool data
- Work center data
- Single-level where-used
- Indented where-used
- Summarized where-used
- Highest-level where-used
- Work center where-used
- Tool where-used

PRODUCT COSTING

This feature takes information from the data base to accumulate standard costs for materials, labor, and overhead. Areas covered by this application include:

- Cost data entry
- Part cost
- Cost buildup (roll-up or fold-in)
- Costed bill of material

CUSTOMER ORDER PROCESSING

This feature provides capabilities for gathering and using customer, vendor, and part or end-item information to process customer orders from receipt to shipment. This application includes:

- Customer search
- Customer shipping, billing, and part information
- Entry and control of customer orders
- Inventory checking and availability
- Profitability control
- Order status
- Picking slips and confirmation
- Shipping documents
- Invoicing
- Returned goods processing

INVENTORY STATUS AND CONTROL

This function provides stock and replenishment order transactions and current inventory status. Typical applications include:

- Stock receipts, issues, transfers, adjustments
- Order status
- Stock location control
- Stock status reporting
- Cycle count

FORECASTING AND ANALYSIS

This feature provides the ability to statistically forecast requirements and to analyze usage. Available features include:

- ABC analysis
- Requirements and scrap factor forecasting – linear, trend, and seasonal models
- Model analysis

MASTER SCHEDULING

The manufacturing user is able to construct a master schedule based on data in the data base and make suitable changes as necessary. The system provides:

- Master schedule report
- Two-level master scheduling
- Lead-time picture
- Master schedule list
- Master schedule load analysis

MATERIAL REQUIREMENTS MANAGEMENT

This feature provides the capabilities for both statistical order control and material requirements planning (MRP) processing to be used selectively by part. Among the facilities available are:

- MRP (regenerative and net change)
- Stock and order status reporting
- Order point control (optional)
- Order action recommendations
- Lot sizing
- Requirements by part and order

UNIS 80

PRODUCTION PLANNING

This feature provides the ability to schedule work and determine capacity requirements, such as:

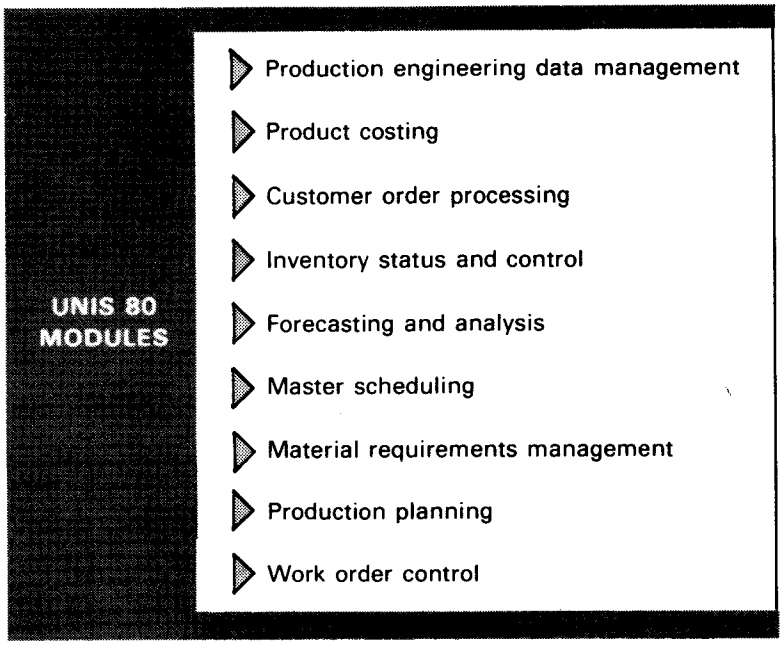
- Work order scheduling
- Capacity requirements planning (infinite)
- Capacity requirements planning (finite)
- Lead-time reduction factors

WORK ORDER CONTROL

- Work order release
- Material availability checking
- Shop floor reporting
- Scrap reporting
- Work order status
- Dispatch reporting

3.12.2. UNIVAC Industrial System 80-Extended

The UNIVAC Industrial System 80-Extended (UNIS 80-E) is a version of the UNIS 80 system offering all of the features of that system. In addition, UNIS 80-E allows the user to modify the system to meet particular needs. Included in the system are the following UNIS 80 modules:



3.12.3. Accounting Management System 80

The Accounting Management System (AMS) provides the user a series of packaged financial applications, written in RPG II, that provide all functions necessary to standard business accounting. AMS consists of four subsystems: Accounts Payable, Accounts Receivable, Payroll, and General Ledger.

AMS FUNCTIONS

ACCOUNTS PAYABLE

The AMS Accounts Payable subsystem includes all functions needed to build and maintain the accounts payable files, select the invoices to be paid, issue the appropriate checks, and print reports reflecting the system operations. These files contain all the needed vendor information, such as name, address, telephone number, year-to-date purchases and payments, and discount-lost data.

ACCOUNTS RECEIVABLE

The AMS Accounts Receivable subsystem includes all functions needed to build and maintain accounts receivable files. These files contain all required information on customers, such as name, address, telephone number, year-to-date sales figures, and credit limits. Daily accounts receivable transactions are summarized by the system to generate appropriate entries for the AMS General Ledger subsystem.

PAYROLL

The AMS Multi-State Payroll subsystem is designed after the manual methods familiar to accountants and bookkeepers. Personnel working with the payroll subsystem need no data processing training or experience.

The payroll subsystem is ready to use and provides all accounting records required by the Internal Revenue Service. It can accommodate multicompany and multidivisional payrolls. The AMS Multi-State Payroll subsystem can also accommodate paying an employee out of multiple taxing jurisdictions and multibank checking.

GENERAL LEDGER

The AMS General Ledger subsystem provides control of accounting records, including an audit trail of entries, and the balancing and validation of all bookkeeping entries. Balance sheets and income statements are produced.

The general ledger subsystem accepts input from the accounts payable, accounts receivable, and payroll subsystems.

3.12.4. UNIVAC Financial Accounting System 80

The UNIVAC Financial Accounting System 80 (UNIFACS 80) is a series of packaged applications, written in ANS'74 COBOL, that provides the user with all the standard features and functions of a business accounting package and gives the user added capabilities in the areas of personnel record-keeping and budgeting.

UNIFACS 80 FUNCTIONS

ACCOUNTS PAYABLE

The UNIFACS 80 Accounts Payable subsystem provides all functions needed to maintain accounts payable records, select invoices for payment, issue appropriate checks, and produce reports needed for control and future planning.

The accounts payable package performs such functions as invoice entry and validation, duplicate invoice monitoring, recurring payment processing, cash requirements forecasting, check printing, invoice aging, Federal Form 1099 reporting, and use tax reporting.

ACCOUNTS RECEIVABLE

The UNIFACS 80 Accounts Receivable subsystem provides all functions needed to maintain accounts receivable records. The subsystem is designed to allow maximum control and flexibility in processing cash payments and in controlling outstanding receivables.

General ledger transactions created by the accounts receivable subsystem are accepted directly by the UNIFACS 80 General Ledger Subsystem. The accounts receivable subsystem also creates the sales history necessary for sales analysis functions.

PAYROLL/PERSONNEL

The UNIFACS 80 Payroll/Personnel subsystem provides a comprehensive, easy-to use payroll processing system, with additional features for personnel record-keeping. Payroll/personnel supports a variety of pay categories and all federal and state tax calculations. It can process up to 20 regular deductions per employee, manually override rate, and process hand-written and voided checks.

The payroll/personnel subsystem also provides for labor distribution, payroll distribution, EEO reporting, and personnel history maintenance.

GENERAL LEDGER/BUDGETING

The UNIFACS 80 General Ledger/Budgeting subsystem maintains all necessary general ledger and budgeting information for many types of users, including those with multicompany requirements. The subsystem can process different companies, divisions, departments, etc, as well as any combination of accounting periods in the same run.

General ledger/budgeting is capable of handling current period transactions for financial systems ranging from 4 quarterly periods per fiscal year up to 13 four-week periods, including the standard 12 monthly periods. The general ledger/budgeting subsystem accepts automated input from the UNIFACS 80 accounts payable, accounts receivable, and payroll/personnel subsystems.

In addition to the general ledger functions, the subsystem provides the user with budgeting functions. Multiple budgets may be created and maintained. Customized budget reports and actual-versus-budget comparisons are available through the use of the Financial Reporter budgeting subsystem.

3.12.5. SPERRY UNIVAC Financial Integrated Control System 80

The SPERRY UNIVAC Financial Integrated Control System 80 (SUFICS 80) offers the user an advanced, interactive financial modeling and planning system. SUFICS 80 uses English-language-based statements, allowing the user to easily construct and change models of all or part of a company. The easy-to-use SUFICS 80 statements also allow users to produce a wide variety of reports to fill analytical and forecasting needs.

SUFICS 80 is a powerful analytical tool that enables the user to perform "what-if", statistical, and risk analyses. SUFICS 80 also offers hierarchical consolidation, providing 3-dimensional modeling and consolidation – all through the use of English-like SUFICS statements.

The analytical capabilities of SUFICS 80 can be applied to such diverse areas as:

- Budgeting
- Cash flow forecasting
- Capital investment analysis
- Long-term and strategic planning
- Market planning; product launching and planning
- Corporate modeling
- Corporate financing
- Inflation evaluation
- Merger and acquisition evaluation
- Cost estimating
- Product profitability analysis
- Consolidations
- Statistical analysis and forecasting

3.12.6. Information Collection System 80

The Information Collection System 80 (ICS 80) is a highly efficient online information collection system implemented as action programs to IMS. It offers a practical and economical solution to information and data collection problems. The system helps to ensure optimum utilization of computer resources, aids in the introduction of online information processing, and provides the following important capabilities:

ICS 80 CAPABILITIES

- Online collection of data and information. The same workstation used for the collection of data and information may also be used for file inquiry and updating.
- A full range of data validation and checking routines. These may be application-dependent checks specified by the user.
- Information and data collection handled simultaneously with other processing in a multiprogramming environment.
- A simple implementation language for specifying the formats to be used to enter data.

3.12.7. UNIVAC Distribution Information System – Wholesale

The UNIVAC Distribution Information System – Wholesale (UNIDIS – WHOLESALE) offers a complete distribution control system designed to optimize cash flow, increase profits, streamline operations, and improve customer service. UNIDIS – WHOLESALE ordering strategies maximize inventory while holding down cost, and offer the user positive control over all goods flowing into and out of the user organization. It is a real-time system allowing the user to respond immediately to customer orders and retrieve billing information quickly and easily.

UNIDIS – WHOLESALE functions can be broken down into order entry, stock control, and inventory management.

UNIDIS – WHOLESALE FUNCTIONS

ORDER ENTRY

UNIDIS – WHOLESALE offers real-time, online order entry with immediate response capability. In addition, order entry offers:

- Online availability determination/reservation
- Specialized delivery instructions and comments, including item substitution
- Standard ship-to and bill-to address
- Automatic discount and pricing capabilities
- Profitability control
- Customer credit limit control
- Invoice transactions
- Automatic pick list generation and route optimization
- Inventory and demand history updating
- Pre- and post-billing accounting procedures
- Blanket order, back order, and drop shipment processing
- Automatic assignment of customer order identifiers

STOCK CONTROL

Stock control provides for the control of goods from their receipt on the shipping dock, through count verification, inspection, repair, and rework until the goods reach stock or scrap. Stock control offers:

- Online processing
- Receipt verification against purchase orders
- Quantity tolerance verification
- System control of goods movement
- Inventory updating
- Generation of financial transactions
- Location control of all goods, stock, and nonstock

UNIDIS - WHOLESALE FUNCTIONS

INVENTORY MANAGEMENT

Inventory management provides a sophisticated set of statistical features to analyze demand patterns and suggest replenishment strategies. The system provides the following features:

- Demand models with automatic model analysis
- Service level specification
- EOQ calculation
- Tracking signal/demand filters
- Alarm reports
- Graphic representation of demand patterns and forecast model
- Product group or warehouse processing

3.12.8. Wholesale Applications Managment System 80

The Wholesale Applications Management System 80 (WAMS 80) is an online, interactive system that provides the basic wholesale distribution management functions required by today's business environment.

WAMS 80 consists of four subsystems: Inventory/Sales Analysis, Order Entry/Billing, Credit Return, and Expanded Sales Analysis.

WAMS 80

INVENTORY/SALES ANALYSIS

The WAMS80 Inventory/Sales Analysis subsystem includes the functions necessary to create and maintain the inventory file. The inventory file includes required information on a product, such as pricing levels, stock levels, reorder levels, sales data, vendor number, and substitute product designation.

ORDER ENTRY/BILLING

The WAMS Order Entry/Billing subsystem is an online interactive system. It achieves the classic ordering goals of having inventory and customer information instantly available for the user, while accurately performing all of the required order processing operations. This includes identifying products, determining stock availability and pricing, identifying the shipping customer, discounts applicable, commission allocations, applicable taxes, producing invoices and pick slips, producing billing records for accounts receivable, maintaining all sales records, and producing management reports.

CREDIT RETURN

The WAMS 80 Credit Return subsystem is an online interactive system. It achieves the goals of having inventory and customer information instantly available, while accurately performing all of the required returned merchandise operations. This includes identifying products, discerning pricing and customer discounts, identifying the shipping customer, determining commission allocations and tax liabilities, producing credit invoices, producing credit records for accounts receivable, maintaining credit records and information, and producing various management reports.

EXPANDED SALES ANALYSIS

The WAMS 80 Expanded Sales analysis subsystem provides expanded sales reports to the user. The reports produced by this subsystem are detailed and tailored to the needs of each user. This subsystem is used in conjuction with the WAMS 80 Order Entry/Billing and Credit Return subsystems to extract input data. The extracted data is then reformatted, stored, and utilized in the various reports. Files from the WAMS 80 Accounts Receivable subsystem and Inventory/Sales Analysis subsystem are accessed to obtain supporting data for these reports.

This subsystem provides monthly reports on product analysis, customer analysis, territory analysis, and monthly and year-to-date sales reports by customer/product class, as well as by salesman/customer/product class. Comparative analysis by customer/product class and salesman/customer/product class is also provided.

3.12.9. Word Processing System 80

The Word Processing System 80 (WPS 80) is an informational processing system that can significantly reduce the time and effort required to generate, correct, update, and manipulate text. It provides high-speed editing, formatting, and computer processing of documents, such as form letters, without repetitive typing.

WPS 80 allows personnel such as secretaries, typists, and clerks to use powerful computerized text editing with ease. Through WPS 80, standard letters or documents, or individual pages, paragraphs, or sentences can be quickly retrieved from storage; revisions or corrections can be quickly made; and revised, error-free copies can be printed at the rate of up to 540 words per minute.

WPS 80 offers the following features:

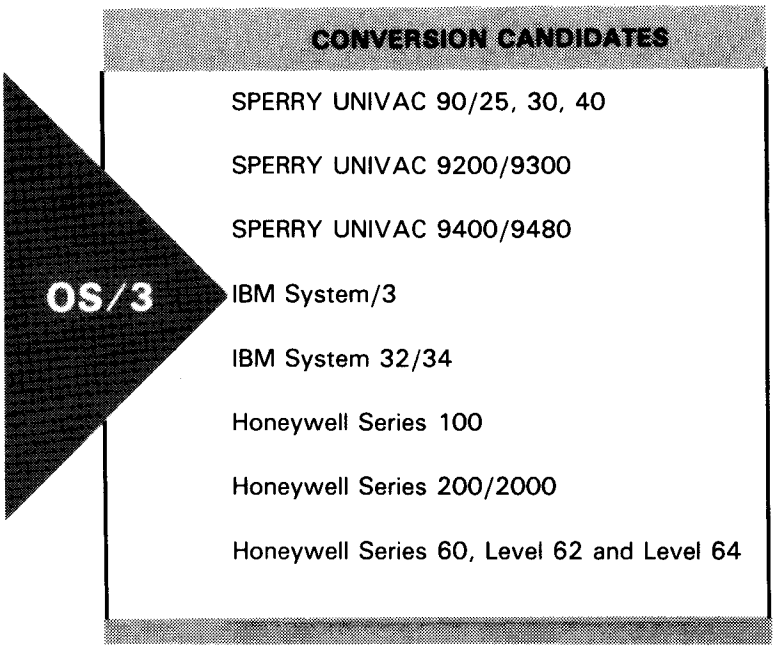
- Variable vertical spacing
- Prints envelopes using name and address of letter
- "Fill-in" feature permits entry of variable data on standard forms
- Automatic paging
- "Cut and paste" feature permits moving or copying of text within a document or between documents
- Access-controlled classifications provide security for documents

- Global searching of files for specified data
- Prints letters, envelopes, or labels in zip code or other sequence

WPS 80 operates using a series of menus to ensure that personnel with no data or word processing experience can quickly and easily learn its use. When a screen of text is completed, a single keystroke removes that text from the screen, stores it, and displays another screen.

3.13. CONVERSION AIDS

In the rapidly changing world of the computer industry, the need is constantly arising to upgrade the present computing system to one that is more powerful, economical, or that meets changing needs. Because of its advanced design – offering power and versatility with economical operation – System 80 is seen as being the growth path for a number of other systems.



As a result, Sperry Univac supplies a number of conversion aids to users who decide to convert from one of these systems to System 80. In addition, there are a number of areas of compatibility between System 80 and the other systems – making the conversion a fairly straightforward process.

3.13.1. IBM System/3

OS/3 software provides a significant amount of compatibility with the IBM System/3. Among the major areas of compatibility are:

- A System/3 compatible sort, SORT3, that accepts System/3 parameters.
- An access method, MIRAM, that is functionally compatible with the System/3 disk access method.
- A System/3 mode on the OS/3 RPG II compiler that permits direct compilation of System/3 RPG II source programs.

- Parameter specifications for $DELET, $COPY, $KCOPY, $DCOPY, and $MAINT can be used directly on System 80 to duplicate the utility functions.

- An OCL processor that accepts and processes System/3 OCL control streams

In addition, Sperry Univac supplies a number of conversion aids for those areas of incompatibility between OS/3 and System/3:

- Disk data file conversion procedure

- A Model 10, 12, and 15 source and proc transcriber

- OCL to JCL converter (JCLCON801)

3.13.1.1. Disk Data File Conversion

System/3 data files must be first dumped to magnetic tapes or diskettes using one of the System/3 utilities, for example $KCOPY. The tapes or diskettes are then submitted to the OS/3 data utilities to reload the files to disk storage devices.

3.13.1.2. Model 10, 12, and 15 Source and Proc Transcriber

Model 10, 12, and 15 source module and OCL procedure libraries must be copied to either diskette or tape transfer files. These files are in turn transcribed to OS/3 libraries by the source and proc transcriber, COPYS3. Diskette transfer files can be generated directly from the source and procedure libraries, using the IBM library maintenance program ($MAINT). If tape transfer files are to be generated, the libraries must be copied to disk first, using $MAINT. The disk files holding the libraries are then copied to tape, using the IBM copy/dump program ($COPY).

3.13.1.3. OCL to JCL Converter (JCLCON801)

IBM System/3 operation control language (OCL) streams can be quickly converted to OS/3 job control language (JCL) streams through the OCL to JCL converter, JCLCON801. JCLCON801 executes on the OS/3 system, accepts System/3 OCL streams, and generates as closely as possible OS/3 JCL streams. Where direct conversion is not possible, the converter attempts to provide as similar a function as possible. Where translation is not possible at all, the converter makes no changes to the code and issues a warning message indicating the areas not translated.

3.13.2. IBM System 32/34

OS/3 software provides a significant amount of compatibility with IBM System/32 and System/34. Among the principal areas of compatibility are:

- A sort program, SORT3, that accepts System/32 and System/34 sort parameters

- Compatible RPG programming languages

- A mode of operation of the OS/3 RPG II compiler that permits direct compilation of many System/32 and System/34 source programs

- A highly compatible RPG II Auto Report facility

In addition, Sperry Univac supplies a number of conversion aids for those areas of incompatibility between OS/3 and System/32 and System/34:

- Disk data file conversion procedure
- Utility program for transcription of source and procedure program libraries (COPYS3)
- OCL to JCL converter (JCLCON802)
- Utility program for converting IBM screen format specifications (S & D) to OS/3 screen formats

3.13.2.1. Disk Data File Conversion

IBM System/32 and System/34 data files can be transcribed to OS/3 disk by first running the IBM TRANSFER procedure to produce a diskette. This diskette is then input to the OS/3 data utility. The data utility can then produce a user-tailored OS/3 disk file.

3.13.2.2. Source and Procedure Library Transcriber

System 32/34 source and proc files can be transcribed to OS/3 files by using the SPERRY UNIVAC COPYS3 source and proc transcriber (also a System/3 conversion aid). The IBM libraries are first copied to a diskette through the IBM $MAINT utility, and this diskette is used as input to COPYS3.

3.13.2.3. OCL to JCL Converter (JCLCON802)

IBM System/32 and System/34 operation control language (OCL) streams can be quickly converted to OS/3 job control language (JCL) streams through the OCL to JCL converter, JCLCON802. JCLCON802 executes on the OS/3 system, accepts System/32 or System/34 OCL streams, and generates an OS/3 JCL stream as close as possible to the OCL stream. Where direct conversion is not possible, the converter attempts to provide as similar a function as possible. Where translation is not possible, the converter makes no changes to the code and issues a warning message indicating the areas not translated.

3.13.2.4. S & D Converter

Sperry Univac provides a utility program that converts the S & D specifications used on System/32 and System/34 to produce formatted screen displays to OS/3 screen formats. The S & D conversion utility executes on the OS/3 system, accepts the S & D specifications, and generates OS/3 screen formats as similar as possible to the IBM formatted sceen displays. When a screen function cannot be converted, the converter makes no change and issues a warning message indicating the areas not converted.

3.13.3. SPERRY UNIVAC 9200/9300 System

The following areas of compatibility exist between the 9200/9300 system and System 80:

- The OS/3 RPG II compiler provides a 9200/9300 mode that permits direct compilation of 9200/9300 RPG programs on System 80 without source code translation.
- 9200/9300 sequential tape files developed on the UNISERVO VI-C or UNISERVO 12 tape devices can be mounted on System 80 UNISERVO 10 tape drives and processed directly by OS/3 programs.

In addition, Sperry Univac supplies a number of conversion aids for those areas of incompatibility between OS/3 and the 9200/9300 system. These aids are:

- 9200/9300 data file transcriber to convert 9200/9300 data files to OS/3 format (UNLOAD/DATA)
- Assembly language translator (TRASM3)
- COBOL and COPY translator (COBTRN305)
- Library transcriber (COPY93)

9200/9300 CONVERSION AIDS

9200/9300 DATA FILE TRANSCRIBER

The data file transcriber (UNLOAD) supplied by Sperry Univac is executed on the 9200/9300 system to copy 9200/9300 disk data files to a tape file. This tape file can be used as input to the OS/3 data utility, DATA, which in turn generates the appropriate disk file from the tape.

9200/9300 ASSEMBLY LANGUAGE TRANSLATOR

The 9200/9300 assembly language source statements can be translated into OS/3 basic assembly language statements through the 9200/9300 to OS/3 assembly language source translator (TRASM3).

COBOL AND COPY TRANSLATOR

The 9200/9300 COBOL source programs and COPY library elements can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN305 translator.

9200/9300 LIBRARY TRANSCRIBER

The 9200/9300 library files can be converted to OS/3 format through the OS/3 COPY93 library transcriber. COPY93 accepts a 9200/9300 formatted tape as input and produces an OS/3 formatted disk file.

3.13.4. SPERRY UNIVAC Operating System/4 (OS/4)

OS/3 offers a high degree of compatibility to the 9400 and 9480 users who have been operating under OS/4. OS/4 RPG and FORTRAN source programs can, for the most part, be recompiled by the OS/3 compilers. Any changes required will be minor. A conversion guide that details all the steps required to migrate from OS/4 to OS/3 is available. For those areas of incompatibility, Sperry Univac supplies the following conversion aids:

- Job control language translator (JCON1)
- Assembly language translator (ASMTRN)
- COBOL and COPY translator (COBTRN301)
- Data file converter (DCON4)
- Library transcriber (COPY94)

OS/4 CONVERSION AIDS

OS/4 JOB CONTROL CONVERTER

The OS/4 to OS/3 job control language converter (JCON1) supplied by Sperry Univac converts OS/4 job control source statements to OS/3 compatible job control statements. Input to the JCON1 utility can be a magnetic tape containing only control streams (no procs) produced by an OS/4 FILE command, cards, or an OS/3 disk file created by an OS/3 FILE command, the COPY94 utility, or the OS/3 librarian. JCON1 outputs to cards, the printer, or to a disk file.

OS/4 ASSEMBLY LANGUAGE TRANSLATOR

OS/4 basic assembly source statements can be translated into OS/3 assembly statements through the OS/4 to OS/3 assembly translator (ASMTRN).

COBOL AND COPY TRANSLATOR

OS/4 COBOL source programs and COPY library elements can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN301 translator.

DISK DATA FILE CONVERTER

Disk data files can be converted to OS/3 format by using the disk data file converter (DCON4) to dump the files onto tape and then inputting the tape to the OS/3 data utility that, in turn, builds the appropriate data file.

OS/4 LIBRARY TRANSCRIBER

OS/4 library files can be converted to OS/3 format through the OS/3 COPY94 library transcriber. COPY94 accepts an OS/4 formatted tape as input and produces an OS/3 formatted disk file. The input tape must be generated through the OS/4 tape and disk librarians.

3.13.5. Honeywell 100 Series

Users migrating from the Honeywell 100 Series data processing systems will find a high degree of compatibility with System 80. For those areas of incompatibility, Sperry Univac offers the following conversion aids:

- COBOL translator (COBTRN304)
- Data file translator (TAPCON)

3.13.5.1. COBOL Translator

Honeywell 100 Series COBOL source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN304 translator.

3.13.5.2. Data File Translator

Honeywell data files can be converted to OS/3 format by using the data file translator (TAPCON). The Honeywell data files must be copied to tape or card, then submitted to TAPCON for reformatting. The resultant tape file can be converted to the ultimate intended file media.

3.13.6. Honeywell 200/2000 Series

Sperry Univac provides the following conversion aids for those users converting from the Honeywell 200/2000 Series systems to System 80:

- COBOL translator (COBTRN302)
- EASYCODER converter (ETC3)
- Data file translator (TAPCON)

3.13.6.1. COBOL Translator

Honeywell 200/2000 Series COBOL source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN302 translator.

3.13.6.2. EASYCODER Converter

Honeywell EASYCODER source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the ETC3 translator.

3.13.6.3. Data File Translator

Honeywell data files can be converted to OS/3 format by using the data file translator (TAPCON). The Honeywell data files must be copied to tape or card, then submitted to TAPCON for reformatting. The resultant tape file can be converted to the ultimate intended file media.

3.13.7. Honeywell 60 Series, Level 62 and Level 64

The following conversion aids are available to those users converting to System 80 from the Honeywell 60 Series, Level 62 and Level 64 systems:

- Program library and data file translator
- COBOL translator

3.13.7.1. Program Library and Data File Translator

Honeywell data files and program libraries can be transcribed to OS/3 format through the program and data file translator (TAPCON). The Honeywell data files and program libraries must be copied to tape or card, then submitted to TAPCON running on OS/3 for reformatting. The resultant tape files can be converted to the ultimate intended file media.

3.13.7.2. COBOL Translator

Sperry Univac provides a translator to convert Honeywell 60 Series COBOL to OS/3 compatible ANSI 1974 COBOL.

3.13.8. SPERRY UNIVAC 90/25, 30, 40

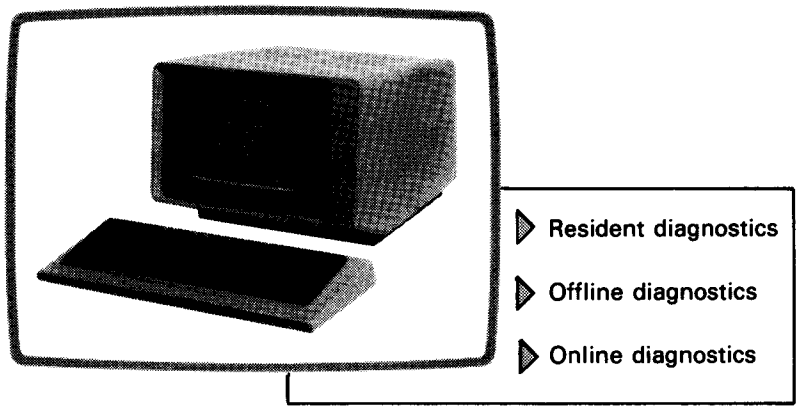
Users of SPERRY UNIVAC 90/25, 90/30, and 90/40 computer systems will find conversion from their present system to System 80 Model 8 a simple, straightforward, and easy task.

Series 90 users operating under OS/3 using consolidated data management (CDM) can simply move their files from their Series 90 system to System 80, and commence production immediately. The only changes a Series 90 user may have to make would be to job control streams, to adjust for peripherals not carried over to System 80 Model 8.

Series 90 users using basic data management can also just move their files from the Series 90 system to System 80. System 80 Model 8 can operate under basic data management, enabling the user to convert to CDM at leisure. Consolidated data management is necessary to use the interactive facilities available with OS/3. Sperry Univac makes available a program, DTFCDI301, to aid in the conversion of assembly language programs from basic to consolidated data management.

3.14. MAINTENANCE AND DIAGNOSTIC SOFTWARE

The System 80 hardware diagnostics are used by the user and Sperry Univac personnel to isolate and identify system hardware faults. The hardware diagnostic system consists of:



The hardware diagnostic routines are designed to operate in an interactive programming environment and make use of displayed dialogs and messages for initiation and entry of control information.

HARDWARE DIAGNOSTICS

RESIDENT DIAGNOSTICS

The resident diagnostic routines are an integral part of the system control software and perform basic error checking on the central processor complex. The central processor complex includes the central processor, control storage, and the resident disk control hardware. The resident diagnostics also check the paper peripheral, diskette, and workstation controllers along with the single line communications adapter. These checks are performed automatically whenever the system is powered on or reset.

OFFLINE DIAGNOSTICS

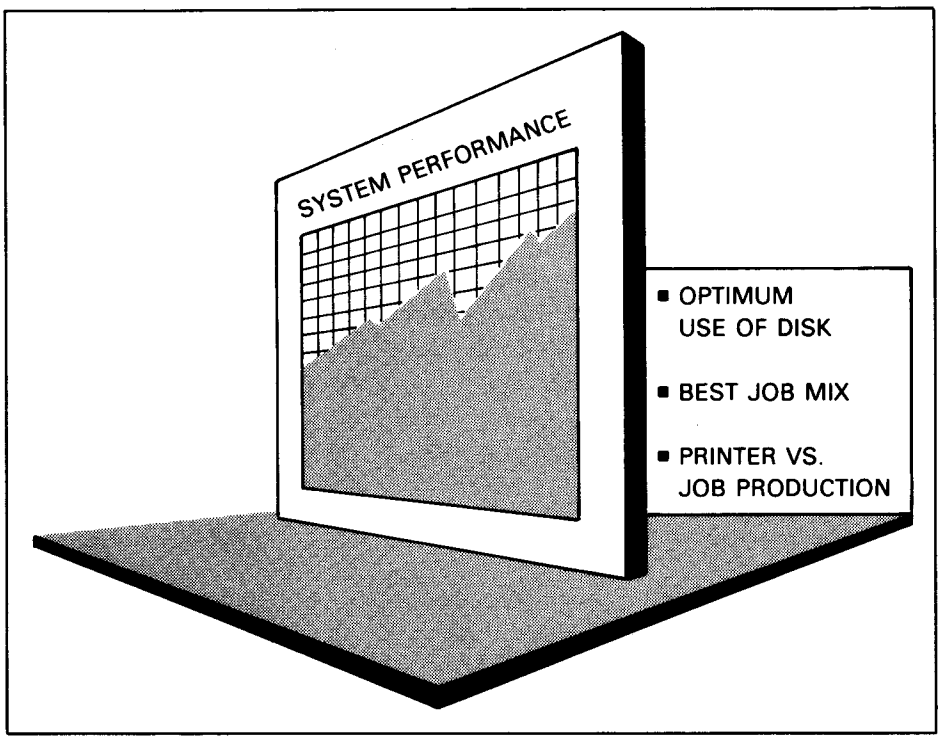
The offline diagnostics consist of microdiagnostic and macrodiagnostic routines. The microdiagnostic routines are used to isolate detected hardware faults to the particular component responsible. Initiated at the console, these routines diagnose problems in the central processor, control storage, and main storage. The macrodiagnostics check the operability of the complete repertoire of system instructions.

ONLINE DIAGNOSTICS

The online diagnostic routines check all peripheral and communications devices to ensure proper functioning and operation. These routines operate under control of the operating system and run concurrent with user jobs. During normal system operation, errors are logged to allow interrogation and analysis by these routines. Error log edit and analysis programs are available to display individual error events and summary information.

3.15. THE SYSTEM ACTIVITY MONITOR

The system activity monitor is used to measure system efficiency by monitoring and recording various system activities. It is an optional software feature configured during system generation. The monitor measures such functions as central processor and main storage usage, disk usage, input/output operations, supervisor interrupts, and, in a system configured for communications, various communications activities. Using the information gathered by the monitor, the user can detect possible production bottlenecks, develop an optimum job mix, change system variables to enhance system performance, and determine if the best use is being made of the system's peripheral devices.



The system activity monitor operates in two modes: monitor class mode, and event trace mode.

In monitor class mode, the system activity monitor groups the data it collects into classes of system activity. Monitored data can be output to the system console, or written to a disk data file. The system activity monitor measures such data as input/output activity, disk storage activity, and communications activity.

In event trace mode, data is collected about each occurrence of a specific event. The system activity monitor keeps track of two types of events: I/O requests by device, command code, and requesting job; and disk cylinder address reference patterns.

Data collected in either mode can be used to generate reports. A report program called SAMRPT can generate either tabular statistical reports or histogram reports.

4. Hardware

4.1. MODEL 8 COMPONENTS

System 80 Model 8 has processing power triple that of an average medium-scale system, along with a significantly larger data base. High-speed technology is used throughout the system, permitting a processor power up to 3.5 times greater than the SPERRY UNIVAC 90/30. The system features main storage capacity up to 8 MB. More powerful capabilities bring improved performance for extensive applications.

The Model 8 processor complex, in either its minimum configuration or with maximum expansion, comprises field-installable modules, including basic processing and input/output control components.

4.1.1. Central Processor

The central processing unit (CPU) is the controlling center that provides the data path and control logic for instruction execution, system control, and I/O support. The user has access to two sets of 16 general-purpose registers and four floating-point registers.

4.1.2. Main Storage Processor

The main storage unit (MSU) stores data in a 4-byte word format. Address and data are checked for correct parity. Double and single-bit error correction, as well as data alignment and error correction code (ECC) are provided by the MSU. Storage protection assures data and program integrity in the system. Main storage is available from 1 MB in the minimum configuration to a maximum of 8 MB. Main storage expansion from 1 to 4 MB is accomplished in 1-MB increments. Expansion from 4 to 8 MB is accomplished in 2-MB increments. Additional bits are also provided for error detection and correction.

The main storage byte sizes available are:

- 1,024,000
- 2,048,000
- 3,072,000
- 4,096,000
- 6,144,000
- 8,192,000

4.1.3. Control Storage

System 80 Model 8 contains 10K words of control storage. Each word is 80 bits long. Control storage is divided into 8K words of read-only memory (ROM) and 2K words of random-access memory (RAM).

4.1.4. Input/Output Microprocessors

Two input/output microprocessors (IOMP) can be used in Model 8. The IOMP is used to support low-speed peripheral subsystems, such as those attached to a diskette controller, workstation controller, paper peripheral controller, integrated magnetic tape controller, remote printer attachment, or intercomputer control unit. Up to eight controllers can be attached to an IOMP.

Each IOMP can also support up to 14 communications lines.

4.1.5. Shared Direct Memory Access

Microcode for shared direct memory access (SDMA) is used with the IOMP to support various low and medium data-rate peripherals. These may include workstations, diskette drives, card readers, card punches, magnetic tape drives, and intercomputer controllers. All of these peripherals interface the IOMP through their controllers that are attached to I/O buses.

An I/O bus using SDMA microcode supports concurrent operations with up to eight controllers and a UNISERVO 10 magnetic tape subsystem. Data transfer rate with SDMA is 200 KB per second.

4.1.6. Multiple Line Communications Multiplexer

Microcode for the multiple line communications multiplexer (MLCM) is used on the I/O bus to interface up to 14 single line communications adapters (SLCAs), with half-duplex lines operating at low, medium, or high speeds. The MLCM can also interface up to seven SLCAs with full-duplex lines operating at 56 KB per second.

A SLCA is attached to the IOMP via its I/O bus. Buffer pool management, data chaining, and command chaining are controlled by microcode for the MLCM as well as the IOMP hardware. The MLCM can support a wide range of communications protocols through a mix of SLCA and MLCM microcode. The MLCM I/O bus transfers data at the rate of 128 KB per second.

4.1.7. Single Line Communications Adapters

Various types of single line communications adapters (SLCAs) provide data communications capability for the system. An SLCA includes a data communications interface (DCE), communications controller, and I/O bus interface. SLCAs interface the MLCM I/O bus.

A loadable microprocessor in the communications controller supports data communications protocols of the following characteristics:

- Synchronous and asynchronous communications
- Line speeds up to 56K baud
- Numerous data communications subsystems
- Functionality of 90/30 system communications multiplexer (CMM) model to the extent that user code does not change
- 90/30 system interfaces to foreign device handlers

SLCAs can also accommodate different DCE interfaces operating at low, medium, or high speeds. Low speed operates at a line speed up to 2400 baud; medium speed operates at a line speed of up to 19.2K baud; and high speed operates at a line speed of up to 56K baud synchronous.

4.1.8. Integrated Disk Channel

System 80 Model 8 may be configured with up to six integrated disk control units (IDCUs) on five selector channels. Up to 24 disk drives may be connected to the system in this way. Additional Series 90 100/200-MB disk drives may also be configured on the selector channels.

4.1.9. Instruction Set

The instruction repertoire for Model 8 consists of 91 general instructions, 53 floating-point instructions, and 31 privileged instructions.

INSTRUCTION FORMATS

Each format consists of an operation code and two or more fields that specify addresses of the operands. Instructions can be two, four, or six bytes in length. All instructions must start on an even-address boundary.

REGISTER-TO-REGISTER (RR) INSTRUCTIONS

RR instructions process data contained in registers. The maximum length of the data is a double word in floating-point registers and a full word in general registers. Data can be a signed or unsigned binary number, a short- or long-format floating-point number, or a decimal number, depending on the specified operation. Operand 1 specifies either a register or mask. Operand 2 specifies a register. Some RR instructions use both operands as an immediate data operand.

REGISTER-TO-INDEXED-STORAGE (RX) INSTRUCTIONS

RX instructions process data between registers and indexed storage. A double word is the maximum length of data handled. Data can be a signed or unsigned binary number, short- or long-format floating-point number, or decimal number. Operand 1 specifies a register or mask. Operand 2 specifies a main storage location.

REGISTER-TO-STORAGE (RS) INSTRUCTIONS

RS instructions perform multiple register and storage operations, as well as data shifting. The first and third operands specify the numbers of two general registers or the boundaries for general multiregister usage. Operand 2 specifies a main storage location.

STORAGE (S) INSTRUCTIONS

S instructions perform operations on main storage locations, I/O channels, and other System 80 hardware locations. The operand, called operand 2, is specified by a 4-bit base register number and a 12-bit displacement. Although most S instructions form the operand 2 address by combining the base and displacement contents, some use only the base register.

▶▶▶

INSTRUCTION FORMATS

STORAGE-AND-IMMEDIATE-OPERAND (SI) INSTRUCTIONS

SI instructions perform operations on an 8-bit value, called immediate data, and an operand in main storage. Operand 2 specifies the immediate data or mask. Operand 1 specifies a 1-byte or half-word main storage location.

STORAGE-TO-STORAGE (SS) INSTRUCTIONS

SS instructions perform operations on two operands located in main storage. In logical operations, the operands are assumed to be equal in length and can vary from 1 to 256 bytes. In decimal operations, the operands can be of different lengths and can vary from 1 to 16 bytes.

4.1.9.1. Nonprivileged Instruction Set

Nonprivileged instructions process fixed-length binary numbers, floating-point numbers, packed and unpacked decimal numbers, and EBCDIC or ASCII characters. Data can be transferred between main storage and the user program set of general registers, as well as from one location in main storage to another main storage location. The operations of shifting, branching, and logical functions also are included.

4.1.9.2. Privileged Instruction Set

Privileged instructions are used by the operating system software when operating in the supervisory state. In this state, all installed instructions are valid and can be executed. This set of instructions includes facilities to load and store the contents of low-order main storage and to load the writable section of the microprogram control storage. Privileged instructions cannot be included in a user program.

4.2. HARDWARE CONFIGURATIONS

The extensive capabilities of System 80 Model 8 permit you an almost unlimited variety of system configurations, from which you can tailor one to your particular needs. The system is configured from the basic processor complex to maximum expansion with two expansion cabinets. Figure 4–1 illustrates the Model 8 configuration.

4.2.1. Basic System

The basic, or minimum, Model 8 includes the central processor unit and a minimum set of I/O subsystems. The central processor complex and subsystems are as follows:

- Central processor unit
- 10K 80-bit words control storage

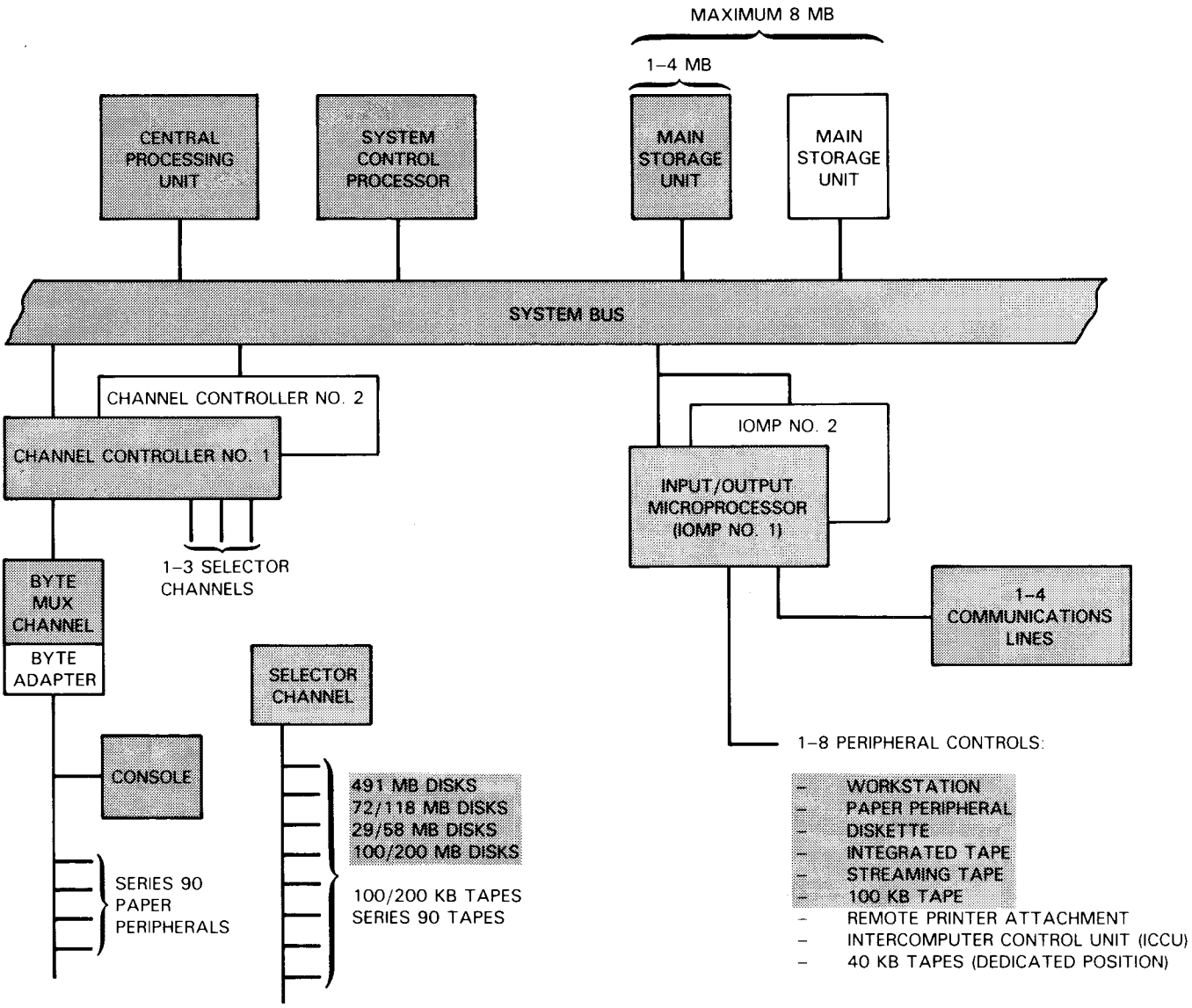
- Main storage processor and main storage unit with 1 MB of main storage
- Either one nonremovable-media disk drive, or two removable-media disk drives are required in a basic system.
- An input/output microprocessor (IOMP) channel with a diskette controller and workstation controller
- One diskette drive
- System console
- A paper peripheral controller and line printer, or a line printer attached to the byte adapter

The console can be located up to 30 feet (9 meters) from the processor cabinet. The diskette subsystem can be located up to 50 feet (15 meters) from the processor or I/O cabinet.

4.2.2. System Growth

System 80 Model 8 can expand greatly from its minimum configuration to meet the increasing data processing demands of a growing business. Following are some capsule descriptions of the expansion capabilities of System 80 Model 8:

- Main storage expansion from 1 to 8 MB
- Up to 24 nonremovable-media disk drives; up to 11 billion bytes of online disk storage
- Removable-media disk storage using 100/200-MB capacity disks interfaced through the integrated selector channel
- Up to 120 locally connected workstations
- High-speed magnetic tape subsystems
- Magnetic tape subsystems for backing up nonremovable-media disk storage
- Additional paper peripherals, including printers, card readers, and punches
- Intercomputer control units, capable of interfacing with System 80 Models 4, 6, and 8 and Series 90 systems running under OS/3
- Up to three additional diskette drives, for a system total of four drives
- Up to 28 communications lines



NOTE:

System Maximums
- 8 MB Main storage
- 5 Selector channels
- 120 Directly attached workstations
- 28 Communications lines

Not all system maximums can coexist due to cabinet space considerations.

Figure 4–1. System 80 Model 8 System Configuration

4.3. PERIPHERAL SUBSYSTEMS

System 80 Model 8 can operate with a wide variety of peripheral subsystems. They are illustrated and briefly described in 4.3.1 through 4.3.8.5.

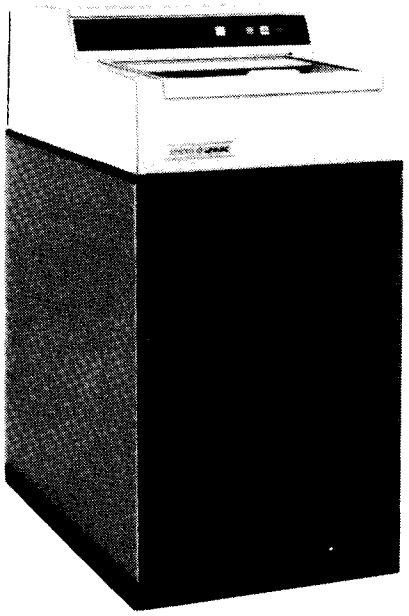
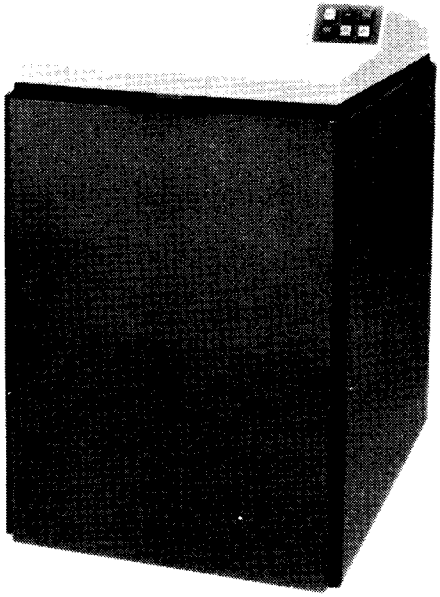
4.3.1. Disk Subsystems

Up to 24 disk drives in various configurations can operate in the Model 8 system. A minimum system requires at least one nonremovable-media disk drive or two removable-media disk drives.

4.3.1.1. 8416/8418 Disk Subsystems

An integrated controller located inside the Model 8 processor cabinet controls up to eight 8416 or 8418 disk drives. The drives operate through the integrated disk channel.

DISK SUBSYSTEM

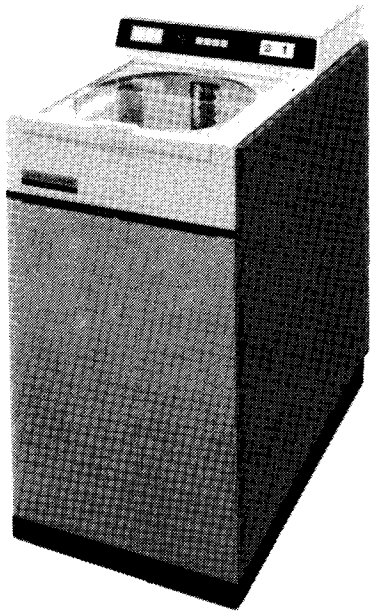
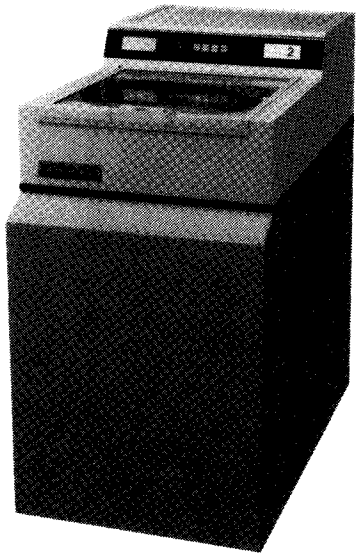
8416	8418
	

	8416	8418
Record format	256 bytes	256 bytes
Tracks per cylinder	7	7
Number of cylinders	404 + 5 spare	404 + 5 spare (low density) 808 + 7 spare (high density)
Data rate	625 KB/s	625 KB/s
Average access time	27 milliseconds	33 milliseconds
Data capacity	29 MB	29 or 58 MB
Disk media	Removable	Removable
Rotational speed	2800 rpm	2800 rpm

4.3.1.2. 8430/8433 Disk Subsystems

The 8430 and 8433 disk subsystems use their own type 5039 control unit in a freestanding cabinet. The disk subsystem is connected to the system through the selector channel. A single control unit supports up to 16 disk drives in any intermix of types 8430 and 8433. Multiple disk subsystems may be configured (up to the system maximum of 24 disk drives) and dual channel operating capability is available.

DISK SUBSYSTEM

8430	8433
	

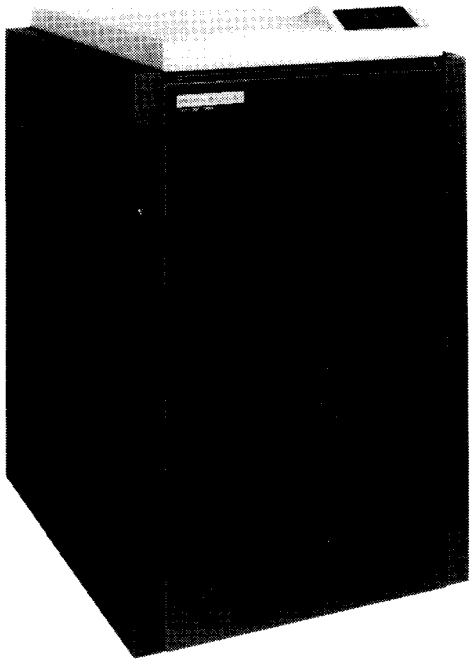
	8430	8433
Record Format	Variable	Variable
Recording surfaces	19	19
Data rate	806 KB/s	806 KB/s
Average access time	27 milliseconds	30 milliseconds
Disk media	Removable	Removable
Data capacity	100 MB	200 MB
Rotational speed	3600 rpm	3600 rpm
Number of cylinders	411	815

4.3.1.3. 8470 Disk Subsystem

The 8470 disk subsystem uses fixed media disk drives and is controlled by a disk controller located inside the Model 8 processor cabinet. The 8470 disk subsystem has as an optional feature dual access capability (by two controllers). Up to eight drives may be connected to one controller.

DISK SUBSYSTEM

8470



Record Format	256 bytes/record
Tracks per cylinder	32
Number of cylinders	625 + 5 alternate
Data rate	2.1 MB/s
Average access time	23 milliseconds
Data capacity	491 MB random access
Disk media	Fixed
Rotational speed	3600 rpm

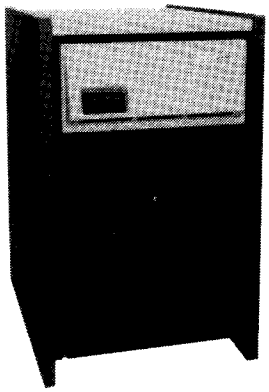
4.3.1.4. 8417 Nonremovable Disk Drive

The 8417 nonremovable disk drive is a freestanding disk storage device. It is capable of storing up to 118.2 MB of information. Up to three 8417 disk drives may be housed in a single cabinet.

An optional feature employing 60 fixed heads and an additional storage capacity of 860,160 bytes may be included to significantly reduce disk seek time and thereby improve its I/O response time.

DISK SUBSYSTEM

8417



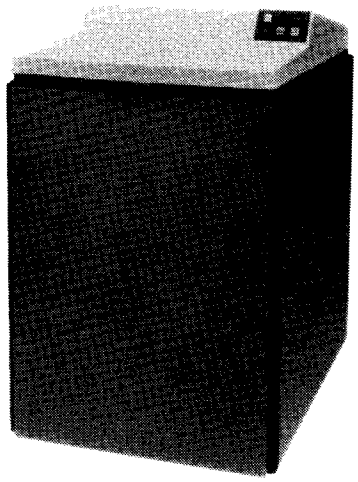
Number of drives per system	1–8
Capacity	
Record format (bytes)	256
Records per track	60
Tracks per cylinder	14
Cylinders per disk	550
Bytes per disk (MB)	118.2
Optional fixed-head capacity (MB)	0.86
Performance	
Data rate (MB/s)	1.1
Head positioning time (ms)	
Minimum	7
Average	35
Maximum	70
Rotational latency (ms)	
Average	8.8
Maximum	17.6
Fixed head average latency time	8.8

4.3.1.5. 8419 Removable Disk Drive

The 8419 removable disk drive is a freestanding disk storage device that includes its own power supply and operator control panel. It is capable of storing up to 72.3 MB of information.

DISK SUBSYSTEM

8419



Number of drives per system	1–8
Capacity	
Record format (bytes)	256
Records per track	50
Tracks per cylinder	7
Cylinders per disk	808
Bytes per disk (MB)	72.8
Optional fixed-head capacity (MB)	NA
Performance	
Data rate (MB/s)	0.78
Head positioning time (ms)	
Minimum	10
Average	33
Maximum	60
Rotational latency (ms)	
Average	10.7
Maximum	21.4

4.3.2. Workstation Subsystem

A workstation subsystem consists of a microprocessor controller and up to eight workstations. Additional workstation controllers may be attached to the system to expand the number of workstations in the system to a maximum of 120.

There are two types of workstations available with the System 80 Model 8: the Model 1 and the Model 2. Models 1 and 2 are in many ways similar; the Model 2 workstation offers an increased functionality through its capability to be programmed by the user and its expanded selection of peripheral devices. Models 1 and 2 are described in 4.3.2.1 and 4.3.2.2.

Workstations of both types may be located up to 5000 cable feet (1500 meters) from the processor cabinet. The controller uses dedicated buffers to allow data transfers at a rate of 19,200 bits per second.

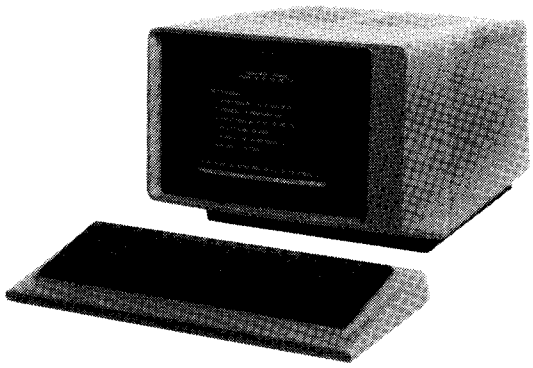
4.3.2.1. Model 1 Workstation

The workstation is designed for ease of operation. Many operating capabilities are included for maximum versatility and access to the system. Indicators inform the operator of the workstation mode of operation (system or workstation) when power is on. They also inform the operator when there are messages waiting to be transferred to the workstation from the system.

Text and system messages are provided on 24 vertical lines. A system status line indicates system status or operating mode. Each line allows up to 80 characters. The display screen is 12 inches (304.8 mm) diagonally. A natural viewing angle prevents glare from ambient light while the operator is seated. Large clear characters permit an excellent view.

The Model 1 workstation features follow.

MODEL 1 WORKSTATION



FEATURES

PROTECTED CHARACTERS

REVERSE VIDEO DISPLAY

WORKSTATION PRINTERS

- 80-Column Printer
- 132-Column Printer

CAPABILITIES

PROTECTED CHARACTERS

Characters displayed on the screen may be protected so that the operator cannot position the cursor on protected characters. If all character locations are protected, the cursor moves to the home position. Characters to be protected are selected by programming commands. Protected characters are identified on the display by a different intensity or reverse video.

CURSOR SCAN

The operator can scan the cursor backward, forward, up, or down by pressing appropriate keys on the keyboard. Rapid corrections or changes may be made by entering the desired character location via the shortest path.

OPERATING MODES

The workstation operator maintains complete control of his operating environment by allowing use of console commands to the operating system without restricting use of the display screen. The operating modes are:

- Workstation Mode

  In normal operation, the workstation mode provides communications to applications programs.

- System Mode

  Communications directly to the OS/3 operating system are made via system mode. Workstation mode is suspended, and the top two lines of the display indicate system information when the particular workstation is in system mode. These two top lines are used for communication with the system. Lines 3 through 24 continue to display workstation mode information. The first line is used by the operator to enter inputs to the system, and the second line displays the system response messages.

  In system mode, the operator uses authorized OS/3 facilities to inquire on system operation, activate particular jobs, and perform other functions useful for operation. After using system mode, the operator may easily return to workstation mode simply by pressing a single key on the keyboard. The display returns to its original workstation status, including the original cursor position.

BLINK MARKER CHARACTERS

Special characters may be selected for blink (delta) display fields. This assists the operator in changing contents of unprotected locations within blink fields. The delta symbols indicate the beginning and end of these fields.

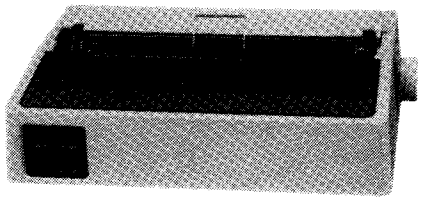
REVERSE VIDEO DISPLAY

Special areas of a workstation display may be designated by programming for reverse video display. This provides special attention to the operator by highlighting a particular point on the screen that should be noted. Programming of special areas may also be at a lower intensity so that either, but not both, highlighting methods may appear on a display. Using reverse video does not affect other operations of the workstation in either mode.

MAGNETIC STRIPE READER

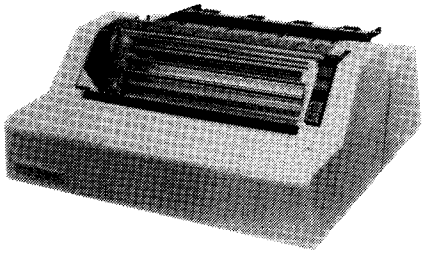
A magnetic stripe reader may be connected to the Model 1 or Model 2 workstation. The magnetic stripe may be on a bank card or similar media. The magnetic stripe reader reads data in either the American Banking Association or International Air Transport Association format. Data from the magnetic stripe reader is treated as if it were entered from the keyboard.

WORKSTATION PRINTERS



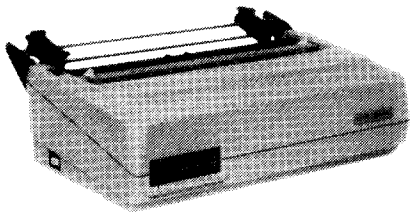
- 80-Column Printer

The 80-column printer is an inexpensive printer, lightweight and compact. A tension-loaded printhead forms 7- by 9-dot matrix characters. The printer prints at speeds up to 80 characters per second. The printer is available with a friction-feed platen to print cut sheets or with a pin-feed platen for printing on continuous-feed forms.



- 132-Column Printer

The 132-column printer maintains the economy of character-by-character serial printing while offering high-speed printing rates of up to 200 characters per second. The same character sets available for the workstation are offered for the 132-character printer. The USA character set is available as a standard 64-character set, a standard 96-character set, and a 96-character set with descenders. The latter includes certain letters (i.e., j, p, and q) formatted so that the tail of the letter descends below the normal print line.



- 0425 Printer

The SPERRY UNIVAC 0425 printer uses a dot matrix printing system to print at speeds of up to 160 characters per second. Two different dot matrices are available with the printer. The standard is a 9 x 7 dot matrix, with lower case descenders, for draft-quality printing. Available as a special feature is an 18 x 40 dot matrix producing high-resolution characters at speeds of up to 40 characters per second. The printer prints up to 132 columns. Paper may be fed either with a friction platen, capable of producing an original and up to three copies, or the printer may use sprocketed forms with a forms tractor, capable of producing an original and up to five copies. Variable spacing is selectable at either 6 or 8 lines per inch. Horizontal spacing may be set at 10, 12.5, or 16.6 characters per inch.

In addition to a standard keyboard arrangement, the workstation contains a number of keys that allow the operator maximum versatility in operations. Table 4–1 lists the keys, and Table 4–2 lists the workstation characteristics.

Table 4–1. Workstation Keys

Key No.	Function
1	Home Cursor
2	Scan Backwards
3	Scan Forward
4	Scan Up
5	Scan Down
6	Insert In Line
7	Delete In Line
8	Insert Line
9	Delete Line
10	Erase To End of Line
11	Erase Unprotected Display
12	Erase Display
13	Lock Keyboard
14	Cursor Return
15	Tabulate
16	Tab Set
17	Positioning Cursor

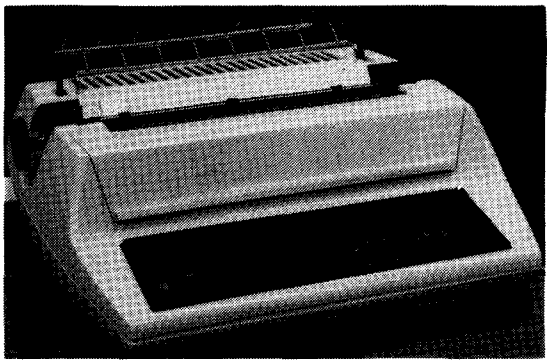
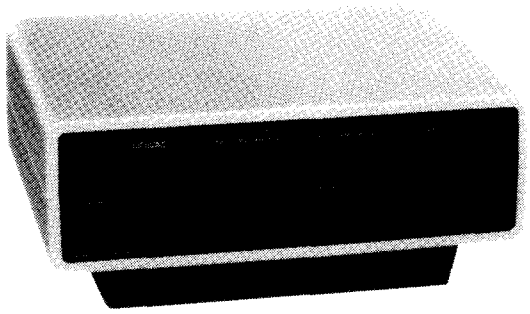
Key No.	Function
18	Position SOE
19	Message Waiting
20	Transmit Unprotected
21	Transmit Display
22	Start Blink Field
23	End Blink Field
24	Start Protect Field
25	Stop Protect Field
26	Insert in Display
27	Delete in Display
28	Erase Field
29	Unlock Keyboard
30	Data Message Header Byte (automatic firmware function)
31	System Mode
32	Workstation Mode

Table 4–2. Workstation Characteristics

Characteristics	
Type of display	Cathode ray tube (CRT)
Number of display lines	24 data plus 1 indicator
Characters per line	80
Character size	0.14-inch (3.5 mm) high, 0.07-inch (1.8mm) wide
Keyboard arrangements	Three, as follows: ■ Typewriter layout ■ Typewriter layout with numeric and function pads
Direct connect communications interface characteristics	■ Serial ■ Asynchronous ■ 19,200 bps ■ Optically isolated
Character sets	Eight, as follows: Domestic, United Kingdom, Germany, France, Spain, Denmark/Norway, Sweden/Finland, and Italy

4.3.2.2. Model 2 Workstation

The Model 2 workstation provides all the features of the Model 1 workstation and offers three important additional features; additional peripheral support, screen bypass, and user programmability.

MODEL 2 WORKSTATION PERIPHERALS	
CQP-1	8406 DISKETTE
	

In addition to the printers and magnetic stripe reader supported by the Model 1 workstation, Model 2 supports a diskette drive and correspondence-quality printer. The diskette drive provides the Model 2 with offline file accessibility and file-building capability. Files called from the System 80 to which the workstation is connected can be stored on diskette. The user may also prepare files and store them for later transmission to System 80. Diskettes may also be used to store programs for execution by the Model 2 workstation. One or two diskette drives may be attached to the Model 2 workstation.

The 132-column correspondence-quality printer produces clean, clear-cut characters on business letter head paper or continuously-fed, sprocketed, multi-part forms. The printer operates at a speed of 45 characters per second. When horizontal spacing is set at 12 characters per inch, the printer may print up to 158 characters per line. At 10 lines per inch, the printer prints up to 132 characters per line. Vertical spacing is selectable at either 6 or 8 lines per inch. Text length and width, margins, pitch and line spacing are determined at the workstation by means of a format statement. Parameters may also be set by the operator. Printing is bidirectional.

The screen bypass feature permits the user, when using certain components of the system control software, to specify a Model 2 workstation with a printer as two separate devices. This allows data to be output to a printer connected to the workstation at the same time the workstation is engaged by other system functions or user programs.

The most important feature of the Model 2 workstation is that it can be programmed by the user in COBOL. Model 2 can be programmed for such functions as data creation, formatting and validation, and various arithmetic operations

Programs for Model 2 are coded in a special subset of American National Standard COBOL X3.23 – 1974. The programs are compiled and linked on the System 80 to which the workstation is connected. The resulting load module is downline loaded into the peripheral diskette or directly into the workstation memory. The Model 2 may be configured with up to 64K bytes of storage. Storage is added to the Model 2 in increments of 32K bytes.

4.3.2.3. System Console

The system console permits the system operator to power up the system, control its operation, and power it down at the close of business. In addition, it serves as the interface between the system operator and system software, and provides diagnostic information for the operator and maintenance personnel. The system console operates in the following three modes.

CONSOLE WORKSTATION OPERATING MODE FUNCTIONS
SYSTEM CONSOLE MODE
Provides communications between operator and system software for system console functions.
SYSTEM CONTROL MODE
Provides communications between operator and system control functions, such as for initial microprogram load (IMPL) or initial program load (IPL).
MAINTENANCE MODE
Provides communications between the Sperry Univac customer engineer and system hardware for maintenance-panel functions.

4.3.3. Printer Subsystems

Printer subsystems are connected to the System 80 Model 8 via the paper peripheral controller and byte multiplexer. Character printers can also be attached to each workstation.

The 0789 printer, 0776 high performance line printer, and 0798 character printer are connected through the paper peripheral controller.

The byte multiplexer supports the 0776 and 0770 printers. They are both high-speed printers.

The 0797, 0798, and correspondence-quality workstation printers provide hard copy of data appearing on the workstation screen. See 4.3.2.1.

A remote printer attachment permits the connection of a remote printer, either the 0789 or 0798 line printers, or the 200 CPS character printer. There is no degradation in print speed when printers are connected as remote printers.

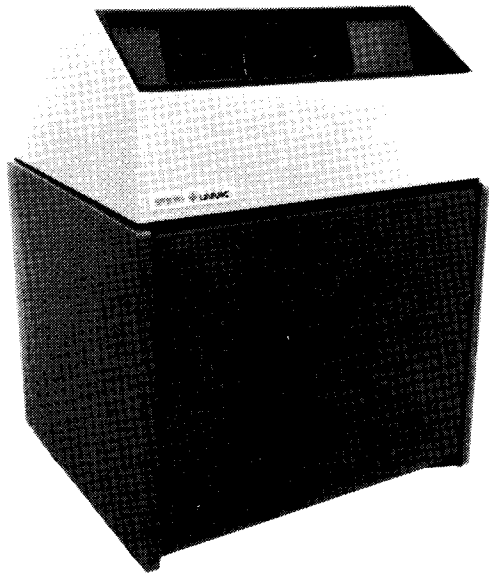
4.3.3.1. 0789 Line Printer

The 0789 printer is a freestanding impact line printer. It employs a horizontal moving band in a print cartridge assembly that permits easy interchange of print cartridges by the operator.

Three printer speeds are available with the 0789 line printer: 180, 300, and 640 lines per minute (lpm). The exact print speed is determined by the print band speed, paper advance time, data transfer rate, the number of times an individual character appears on a print band, and the character density (characters/inch) at which printing takes place.

LINE PRINTER

0789



Print speed	180, 300, 640 lpm for 48-character set
Print line	132 columns wide
Printing density	10 columns per inch
Line spacing	6 or 8 lines per inch (operator selectable)
Line advance	30 ms at 6 lines per inch
Forms slew rate	15 inches per second (38 cm/s)
Forms width	16 inches (40.6 cm)
Forms length	22 inches (56 cm)
Number of copies	1 original plus five carbons

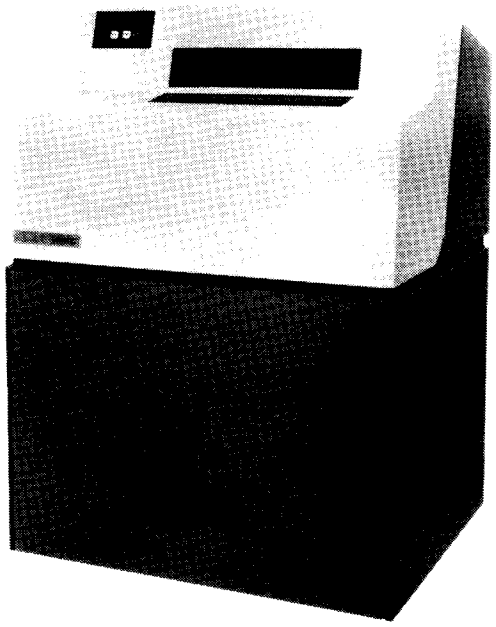
4.3.3.2. 0776 High-Performance Line Printer

The 0776 high-performance line printer is a freestanding impact line printer capable of print speeds up to 1200 lines per minute. It employs a horizontal-moving print band in a print cartridge assembly that permits easy interchange of print cartridges by the operator.

The print rate of the 0776 printer is depedent upon the character set used for printing. For example, the print speed for a character set containing 128 characters is 560 lines per minute, while the print speed for a character set containing 48 characters is 1200 lines per minute.

LINE PRINTER

0776



Print speeds	1200 lines per minute for 48-character set 980 lines per minute for 64-character set
Print line	120 columns or 136 columns by feature
Print density	10 columns per inch
Line spacing	6 or 8 lines per inch (operator selectable)
Line advance	16 ms at 6 lines per inch (single line)
Forms slew rate	50 inches per second (127 cm/s)
Forms width	18.75 inches (47.6 cm)
Forms length	18 inches (45.7 cm)
Number of copies	1 original plus 5 carbons

4.3.3.3. 0770 Line Printer

The 0770 printer is a freestanding impact line printer. It is capable of print speeds of up to 2000 lines per minute. The 0770 printer contains its own controller and is connected to the system through the byte multiplexer.

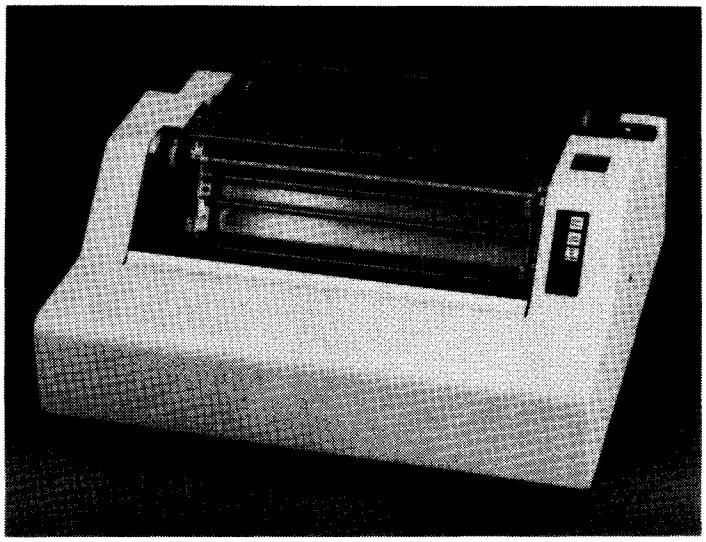
LINE PRINTER

0770

Print speeds	800 lpm, 1400 lpm, or 2000 lpm, depending on model
Print line	132 columns or 160 columns by feature
Print density	10 columns per inch
Line spacing	6 or 8 lines per inch (program selectable)
Forms advance rate	50 inches per second (127 cm/s) or 75 inches per second (190.5 cm/s) depending on model
Data codes	EBCDIC, ASCII, or any code of 6, 7, or 8 bits
Character set	48 standard, up to 384 by feature
Vertical format	Program control
Forms width	3.5 to 18.75 inches (8.89 to 47.6 cm)
Forms length	7 to 24 inches (17.8 to 61 cm)
Number of copies	1 original plus 5 carbons

4.3.3.4. 0798 Character Printer

The 0798 printer is a 132-column matrix character printer that may be connected to the paper peripheral controller, remote printer controller, or to a workstation. The printer is available with a standard 64-character print set, standard 96-character set, and 96-character set with lowercase descenders.

CHARACTER PRINTER	
0798	
	
Print speed	37.5 lines per minute, equivalent to 75 lines per minute when printing is bidirectional (200 characters per second)
Print line	132 columns
Print density	10 columns per inch
Line spacing	6 or 8 lines per inch (operator selectable)
Print matrix size	7 x 7 or 7 x 9 by feature
Forms slew rate	6.7 inches per second (17 cm/s)
Forms width	15 inches (38 cm)
Number of copies	1 original plus 5 carbons
Paper feed	tractor control

4.3.4. Magnetic Tape Subsystems

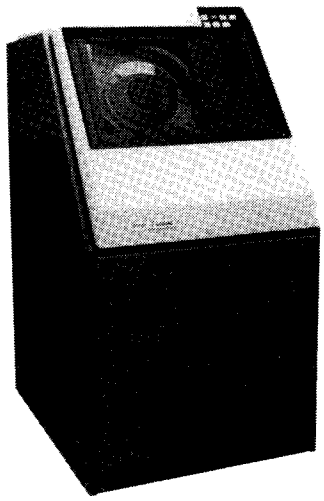
Many different magnetic tape subsystems may be connected to System 80 Model 8 in many different ways. The Series 90 UNISERVO 10/14, 12/16, and 20 magnetic tape subsystems may be connected to System 80 Model 8 through the selector channel. The UNISERVO 22/24 magnetic tape subsystems may also be connected through the selector channel. The UNISERVO 22 and streaming magnetic tape subsystems may be connected to Model 8 through the integrated tape controller, through the IOMP. The System 80 UNISERVO 10 (Type 0871) magnetic tape subsystem may be connected through a dedicated controller, through the IOMP. A maximum of eight UNISERVO 10 tape drives may be connected in this way.

4.3.4.1. UNISERVO 10 (Type 0871) Magnetic Tape Subsystem

The System 80 UNISERVO 10 Magnetic Tape Subsystem is a low-cost unit using magnetic tape conforming to standards specified by the American National Standards Institute (ANSI). The UNISERVO 10 provides features, such as automatic tape loading, and uses the ANSI-standard, industry-compatible wraparound tape cartridge.

UNISERVO 10 MAGNETIC TAPE SUBSYSTEM

0871



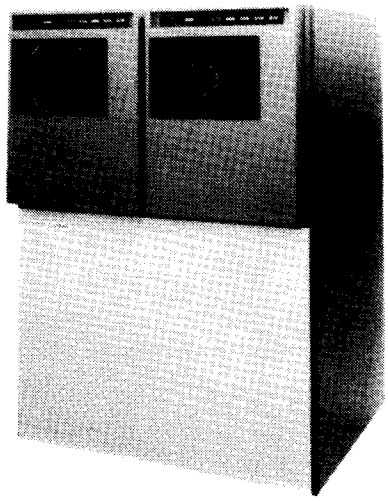
Tape speed	25 inches per second (63.5 cm/s)
Read access time	17 milliseconds
Start/Stop time	17 milliseconds
Reversal time	16 milliseconds
Rewind time	80 seconds for 2400 feet (731.5 meters) of tape
Data rate	40 KB/s
Recording density	800 and 1600 bits per inch in NRZI or PE mode recording 9 tracks; 200, 556, and 800 bits per inch in NRZI recording 7 tracks

4.3.4.2. UNISERVO 10/14 Magnetic Tape Subsystem

The UNISERVO 10/14 Magnetic Tape Subsystem offers data rates of up to 96 KB per second. A basic UNISERVO 10/14 subsystem contains two tape drives and a controller in one tape drive cabinet. Up to eight tape drives are supported by one controller and may be included in one subsystem.

MAGNETIC TAPE SUBSYSTEM

UNISERVO 10/14

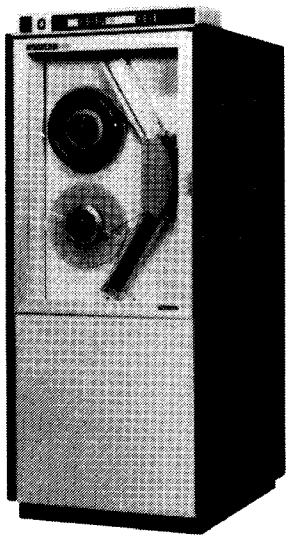
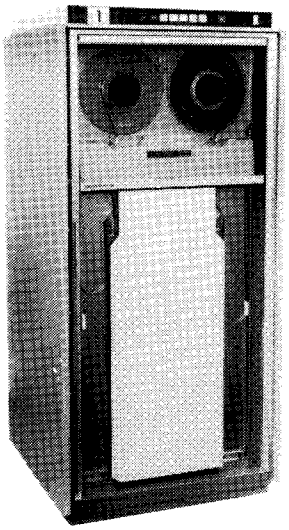


	UNISERVO 10	UNISERVO 14
Tape speed	25 ips (63.5 cmps)	60 ips (152.4 cmps)
Read access time	17 milliseconds	7 milliseconds
Interblock gap passing time	25 milliseconds	12.6 milliseconds
Reversal time	17 milliseconds	6.3 milliseconds
Rewind time	170 seconds	170 seconds
Data rates:		
9 track	40 KB/s	96 KB/s
9 track NRZI	20 KB/s	48 KB/s
7 track NRZI at:		
200 bpi	5 KB/s	12 KB/s
556 bpi	13.9 KB/s	33.4 KB/s
800 bpi	20 KB/s	48 KB/s
Bit density	800 and 1600 bpi in NRZI or PE 9 track; 200, 556, 800 in NRZI 7 track	

4.3.4.3. UNISERVO 12/16 Magnetic Tape Subsystem

The UNISERVO 12/16 Magnetic Tape Subsystem consists of a controller and up to 16 tape drives. The drives may be all UNISERVO 12 or UNISERVO 16 drives, or any combination of the two.

MAGNETIC TAPE SUBSYSTEM

UNISERVO 12	UNISERVO 16
	

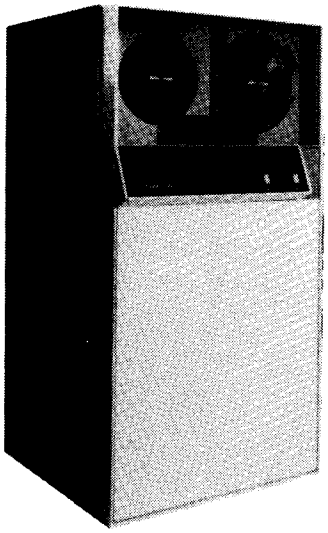
	UNISERVO 12	UNISERVO 16
Tape speed	42.7 ips	120 ips
Reversal time	25 milliseconds	10 milliseconds
Rewind time	180 seconds	120 seconds
Data transfer rates:		
9-track PE	68 KB/s	192 KB/s
9-track NRZI	34 KB/s	96 KB/s
7-track NRZI at:		
200 bpi	6.4 KB/s	18 KB/s
556 bpi	17.8 KB/s	50 KB/s
800 bpi	25.6 KB/s	72 KB/s

4.3.4.4. UNISERVO 20 Magnetic Tape Subsystem

The UNISERVO 20 Magnetic Tape Subsystem is a high-performance tape handling subsystem, consisting of a control unit and from 1 to 16 magnetic tape units. The UNISERVO 20 control unit may be configured with mixed tape units; i.e., a mixture of UNISERVO 12, 16, and 20 tape units.

MAGNETIC TAPE SUBSYSTEM

UNISERVO 20



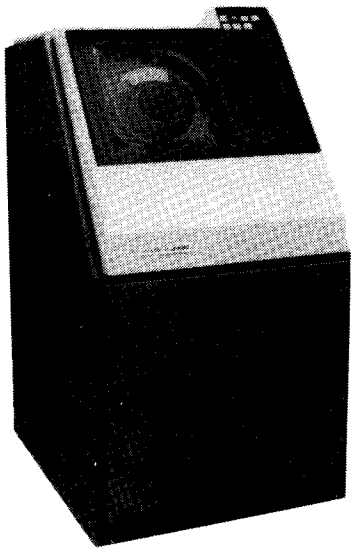
Tape speed	200 ips
Data transfer rate	320 KB/s
Recording density	1600 bpi
Reversal time	16 ms
Rewind time	1 minute
Recording mode	9-track PE

4.3.4.5. UNISERVO 22/24 Magnetic Tape Subsystem

The UNISERVO 22/24 Magnetic Tape Subsystem offers the user a high-speed magnetic tape subsystem in a compact cabinet. The UNISERVO 22/24 subsystem is connected through either the block multiplexer or the byte multiplexer. Each UNISERVO 22/24 subsystem consists of a controller and one to eight tape drives. UNISERVO 22 and 24 drives may be intermixed in a subsystem.

MAGNETIC TAPE SUBSYSTEM

UNISERVO 22/24

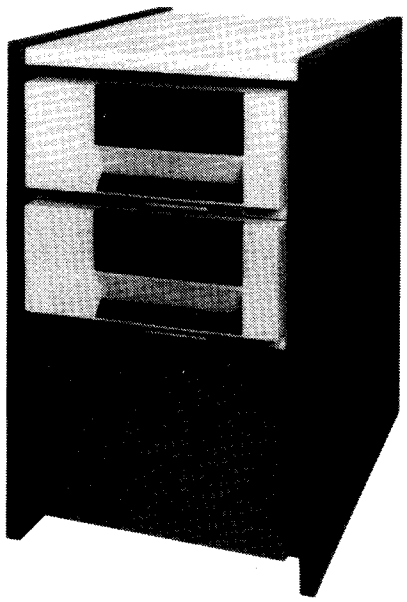


	UNISERVO 22	UNISERVO 24
Tape speed	75 ips	125 ips
Recording density	1600 bpi PE 800 bpi NRZI	
Reversal time	10 ms	10 ms
Rewind time	120 seconds	120 seconds
Transfer rate	PE 120 KB/s NRZI 60 KB/s	PE 200 KB/s NRZI 100 KB/s

4.3.4.6. Streaming Magnetic Tape Subsystem

The streaming magnetic tape subsystem is primarily used as a tape backup for high-capacity disk drives. It is connected to System 80 Model 8 through the integrated tape controller. Up to four streaming tape drives may be configured on System 80 Model 8. A total of eight drives may be attached to the integrated tape controller: any combination of streaming tape drives and UNISERVO 22 tape drives, with a maximum of four streaming tape drives.

STREAMING MAGNETIC TAPE SUBSYSTEM



	High Speed	Low Speed
Tape speed	100 ips	25 ips
Interblock gap passing time	6 ms	24 ms
Rewind time	190 seconds	190 seconds
Data rates:		
Streaming mode	160 KB/s	160 KB/s
Low speed mode	40 KB/s	40 KB/s
Data density	1600 cpi	
Recording format	9-track PE	9-track PE

4.3.5. Diskette Subsystem

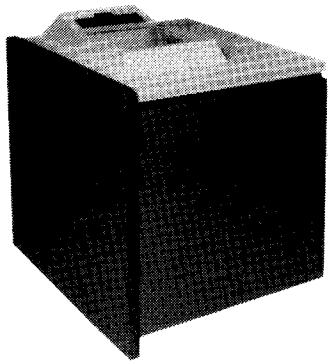
The diskette subsystem supports up to either two auto-load drives and two manual drives or four manual drives. At least one diskette drive is required in the minimum Model 8 configuration.

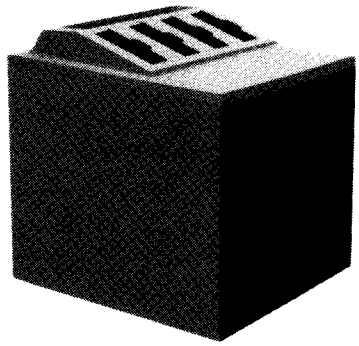
The diskette controller is buffered and performs label checking in data-set mode. The controller uses microprocessors and interfaces the IOMP through shared direct memory access.

Both the manual and auto-load drives read 1- and 2-sided diskettes recorded in either single or double density. Diskettes may be recorded in IBM-compatible basic data exchange (BDE) format for data interchange between IBM and SPERRY UNIVAC systems.

A diskette drive cabinet holds up to either one auto-load and one manual drive, or four manual drives.

DISKETTE SUBSYSTEM





Auto-load hopper/stacker capacity	20 diskettes
Load/unload time per diskette	5 seconds maximum (60 Hz model) 6 seconds maximum (50 Hz model)
Data read/write heads	2/drive
Access time	3 milliseconds track-to-track
Tracks per diskette	77 tracks/side
Tracks per inch	48
Sector size	128, 256, 512, dependent on density method
Data transfer rate	31 KB/s at 256 bytes/sector 62 KB/s at 512 bytes/sector dependent on sector sequence arrangement
Bit density	3400 or 6800 bpi (8704 or 17,408 bpcm)
Storage capacity (approximate)	Double density, 2 sides, 1 MB Double density, 1 side, 500 KB Single density, 2 sides, 500 KB Single density, 1 side, 250 KB

4.3.6. Punched Card Subsystems

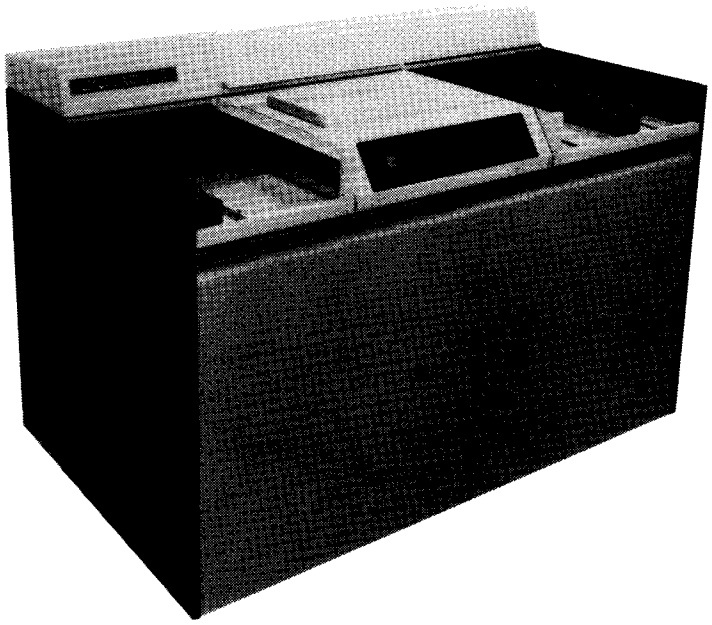
System 80 Model 8 offers several peripheral devices that provide the user with the familiarity and convenience of punched cards. Two card readers and one card punch are offered. One card reader is integrated with the system and is controlled by the paper peripheral controller. The other card reader and the card punch are freestanding, and contain their own controllers.

4.3.6.1. 0716 Card Reader Subsystem

This card reader subsystem contains its own controller and synchronizer. The synchronizer regulates the flow of data and control signals to and from the reader mechanism. The controller connects to the byte multiplexer.

CARD READER SUBSYSTEM

0716



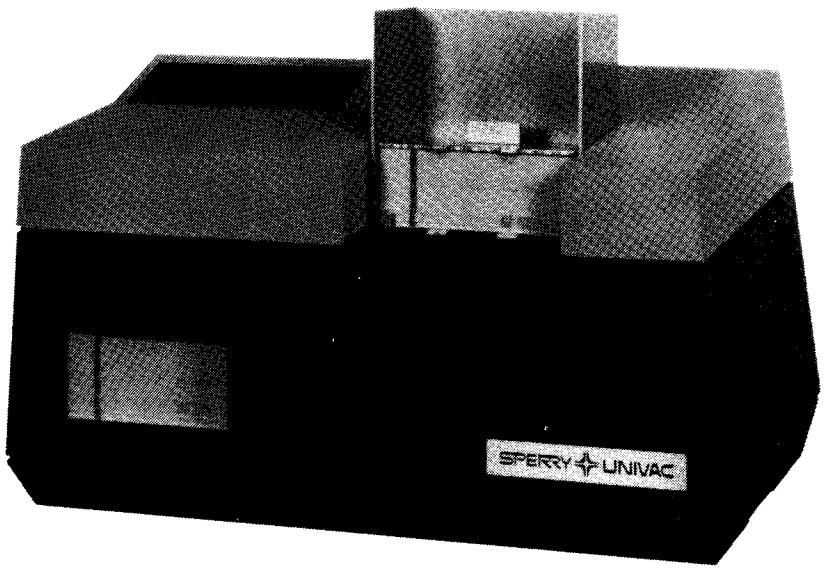
Reading speed	600 or 1000 cards per minute
Hopper capacity	2400 cards
Stacker capacity	2000 cards each (two stackers)
Read modes	Image mode at 160 charactes per card; Translate mode at 80 characters per card for EBCDIC, ASCII, and compressed codes
Card sizes	51-column and 66-column short cards, 80-column standard cards
Verification	Multiread validity checking

4.3.6.2. 0719 Card Reader Subsystem

The 0719 card reader subsystem is a table-top reader that connects to the system through the paper peripheral controller and then to the IOMP through the shared direct memory access. No other controller is required in the reader cabinet.

CARD READER SUBSYSTEM

0719



Reading speed	300 cards per minute
Hopper capacity	1000 cards
Stacker capacity	1000 cards
Read modes	Image mode at 160 characters per card Translate mode at 80 characters per card
Card size	80-column standard
Read technique	2-column sensors, dual redundant column checking
Read-stating sensing	Column by column

4.3.6.3. 0608 Card Punch Subsystem

The 0608 card punch subsystem contains its own controller. It is connected to the IOMP through shared direct memory access. The card punch contains an auxiliary stacker for holding cards in which errors were detected.

CARD PUNCH SUBSYSTEM

0608

Punch speed	75 cards per minute with 80 columns punched or 160 cards per minute with first 28 columns punched; 120 columns per second advance speed
Punch code	Image or compressed
Card size	80-column standard
Checking method	Punch motion check
Feeding method	Column serial, on demand
Hopper capacity	700 cards
Stacker capacity	Primary stacker: 700 cards Auxiliary stacker: 100 cards
Read/validity check	Reads prepunched cards at 160 cards per minute

4.3.7. Intercomputer Controller

The intercomputer controller (ICCU) is a printed circuit card that may be located in either the I/O or expansion cabinet. The intercomputer controller permits bi-directional data flow between your system and other System 80 Models 4, 6, or 8 data processing systems.

The ICCU connects with the IOMP in the I/O or expansion cabinet. IOMP control is through shared direct memory access.

The microprogrammed ICCU contains microcode that is loaded through the IOMP during system initialization and controls the system it interfaces. Fiber optics cable is used to connect data processing systems through the intercomputer controller. Cable length between systems may range from 50 to 3000 feet. The data transfer rate varies from 145 KB/s to 25 KB/s, depending on the length of the connection cable.

4.3.8. Communications Subsystems

System communications capability is provided by the single-line communications adapters (SLCAs). The multiple-line communications multiplexer (MLCM) component of the IOMP supports up to 14 half-duplex SLCAs operating at high, medium, or low speeds, or up to 7 full-duplex SLCAs operating at high speed only.

Remotely located workstations, terminals, and other communications devices are supported through the SLCAs. An autodial capability may be included with each SLCA. Many different types of communications lines may be used with the SLCAs, including:

- Digital data network for distributed data processing
- Public data networks
- Private leased lines
- Automatic answering
- Telegraph
- Military communications networks

Communications line characteristics and the communications device attached to the line are accommodated and controlled by the SLCA. Each line can accept devices operating at speeds of up to 56K bits per second. SLCAs also perform special character recognition, integrity checks, data transfer control between main storage and devices, and other required control functions.

Table 4–3 lists the basic protocols used with the SLCA. The following subsections briefly describe communications devices available from Sperry Univac that may be attached to the system communcations lines.

Table 4–3. Communications Functionality

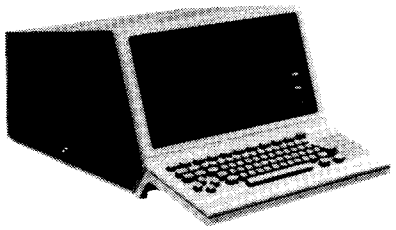
Protocol/Function	Line Speed	Mode	Interface
UNISCOPE 100/200, UTS 400, UTS 4000	2000 – 9600 bps	Half duplex, synchronous	RS-232-C or MIL-STD-188 or UNISCOPE and UTS 400 and 4000 only
BSC	2000 – 9600 bps	Half duplex, synchronous	RS-232-C or MIL-STD-188
UTS 10/TTY (or equivalent)	50 – 9600 bps	Half duplex, asynchronous	RS-232-C or MIL-STD-188
Universal Data Link Control (UDLC)	9600 – 19,200 bps	Full duplex, synchronous	RS-232-C/X.21. BIS
NTR	4800 – 56K bps	Full duplex, synchronous	RS-232-C or MIL-STD-188 or CCITT V.35

4.3.8.1. UNISCOPE Display Terminals

The UNISCOPE 100/200 Display Terminal is a 2-way remote terminal device that is keyboard-operated and contains a cathode-ray tube for message display.

The terminal provides input/output message buffering, refresh storage, character generation, and control logic. Special interfaces for hardcopy output are available. A variety of presentation formats provides display capacities of 960 or 1024 characters for the UNISCOPE 100 Display Terminal, or 1536 or 1920 characters for the UNISCOPE 200 Display Terminal. The terminal can be used as a data entry device or as a display device.

UNISCOPE DISPLAY TERMINAL



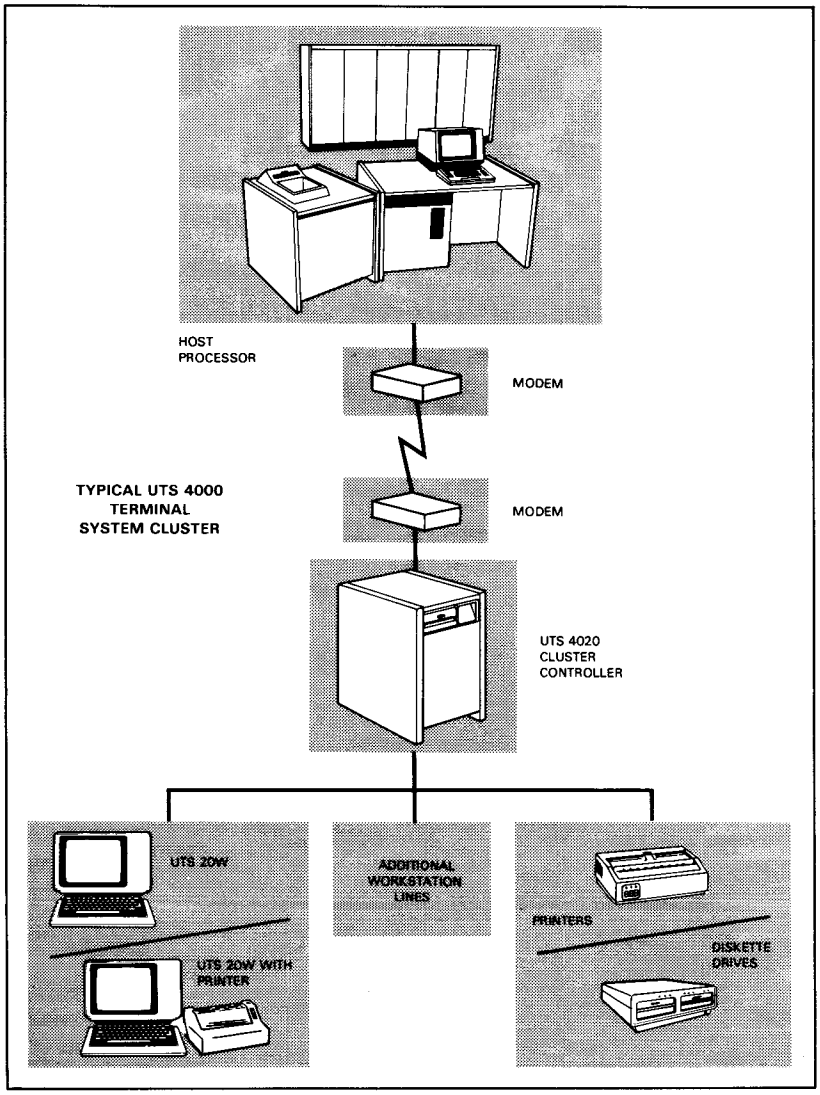
Displays	System output messages
	Operator input messages for composition and editing before transmission to system
Keyboard	Alphanumeric keys
	Cursor control
	Editing keys
	Characters are generated by a read-only storage digital stroke generator. Each character entered by the generator is displayed and stored in the display hardware.

4.3.8.2. UTS 4000 Universal Terminal System

The SPERRY UNIVAC UTS 4000 terminal system is a comprehensive family of communications devices that offers a wide range of interactive communications capabilities. The UTS 4000 system can be used for a variety of applications, from simple data entry and retrieval to sophisticated distributed data processing. Through use of interchangeable components, the system can be configured to meet the needs of any communications user.

The UTS 4000 system employs a terminal cluster architecture using a programmable controller unit attached to the host processor. Terminals and other peripheral devices are connected through the controller. All available terminal types are video display devices and include editing and programmable terminals. Peripheral devices include a desk-top diskette unit supporting 1-megabyte diskettes and a variety of terminal printers.

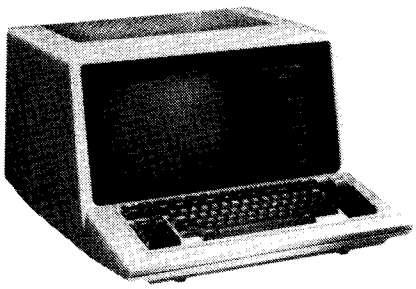
The UTS 4000 control unit (the UTS 4020) is a programmable, microprocessor based unit capable of acting as a stand-alone processor, as well as a communications controller. Provided with an integrated diskette unit, random access storage, and a complement of software elements, it is one of the most advanced communications devices available. The stand-alone processing capabilities of the UTS 4020, coupled with its advanced communications features, make the UTS 4000 system ideally suited for inclusion in a distributed data processing network.



4.3.8.3. UTS 400 Universal Terminal System

The UTS 400 Universal Terminal System is a powerful, programmable terminal display device.

UTS 400 UNIVERSAL TERMINAL SYSTEM

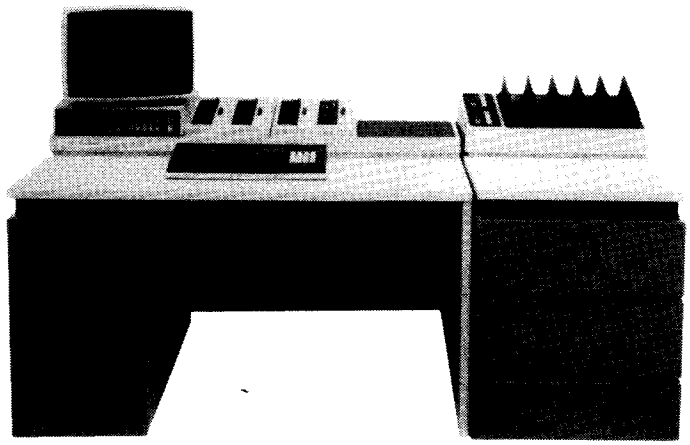


A variety of presentation formats is offered. They provide a total display capacity of 960, 1024, 1536, or 1920 characters. Due to its modular construction, the UTS 400 can operate as a data entry device or as a display device and can be conveniently located at the central computer site or at a remote location. Various configurations are possible. The UTS 400 can be configured as a master terminal with up to two slave terminals, or as a controller with up to six slave terminals. It can be used with the UNISCOPE 100 or UNISCOPE 200 Display Terminals in multidrop or multiplexer configurations.

4.3.8.4. BC/7 Business Computer System

The BC/7 Business Computer System is a low-cost, powerful terminal system built into a desk configuration. Emulators allow the BC/7 to be used in existing or new communications networks involving remote batch operations.

BC/7 BUSINESS COMPUTER SYSTEM



Emulators are programs that are loaded into the main storage area of the system and make the system appear and function as another type of communication device. For example, when the system is initialized as a remote batch terminal (RBT) by using the DCT 2000 emulator, the system function is virtually the same as if it were a DCT 2000 in a communications environment. Consequently, you can use a BC/7 in any communications network that would have supported a DCT 2000. Using emulators allows one hardware device configuration (the BC/7) to have a wide range of applications in communications without any changes in the basic hardware or to the programs currently available with the host computer system.

4.3.8.5. Additional Communications Devices

In addition to the SPERRY UNIVAC devices listed, the following devices can also be attached as communications terminals:

- IBM 3270 terminal system
- TELETYPE Models 33, 35, 37, and 38

While these devices can be attached as communications terminals, all services available to the normal communications user may not be available to users of these devices.

4.4. HARDWARE SUMMARY

Table 4–4 summarizes the system hardware characteristics:

Table 4–4. System Characteristics (Part 1 of 3)

Characteristics	
System orientation	Disk, with intermix of interactive and batch processing
System control	Central processing unit
Control storage	10K 80-bit words
Main storage unit	1 MB expandable to 4 MB in 1-MB increments; further expansion to 8 MB in 2-MB increments
Selector channel	Maximum of five selector channels supporting the following: ■ Up to six integrated disk control units, controlling a maximum of 24 Type 8416, 8417, 8418, 8419, 8470 disk drives. ■ Additional peripheral controllers, supporting the following peripherals: – UNISERVO 10/14, 12/16, 20, 22/24 tape drives – Type 8430/8433 disk drives

Table 4–4. System Characteristics (Part 2 of 3)

Characteristics	
Input/output microprocessor	Maximum of two IOMPs per system, each supporting up to eight device controllers consisting of: ■ Diskette controller having one, two, or four manual drives, or one or two autoload diskette drives plus one or two manual load drives. No more than a total of four diskette drives per system. ■ Workstation controller to support eight workstations. Maximum of 15 workstation controllers per system, for a maximum of 120 workstations. ■ Paper peripheral controller supporting: – Two type 0719 card readers, or one card reader and one type 0608 card punch – One or two local printers, in any mix from 200 characters per second to 1200 lines per minute with a total aggregate not exceeding 1500 lines per minute; types 0789, 0776, 0798 – One 0719 card reader and one printer, type 0789, 0776 or 0798 ■ Maximum of 12 paper peripheral controllers per system ■ Remote printer controllers. each supporting one type 0789 or 0798 printer ■ Integrated tape controller capable of supporting a total of eight magnetic tape drives in the following combinations: – Up to four streaming tape drives – Up to eight UNISERVO 22 tape drives – Any combination of the two not exceeding four streaming tape drives or eight tape drives total. ■ Dedicated tape controller channel, supporting up to eight UNISERVO 10 (Type 0871) tape drives. ■ Intercomputer control unit (ICCU) interfacing another System 80
Byte adapter channel	One byte adapter channel through which are connected the system console and the following peripherals: ■ Type 0770 printer ■ Type 0716 card reader ■ Type 0776 printer
Central processing unit	Has two sets of 16 general-purpose registers Four floating-point registers Priority controller for access to any two units on a priority basis Aggregate transfer rate of 8 MB/s

Table 4–4. System Characteristics (Part 3 of 3)

	Characteristics
Main storage processor	Has 64 relocation registers for address conversion (48 for user programs, 16 for operating system use) 48 protection keys Error correction Storage protection (2K granularity)