UNISYS

Product Information Announcement

o New Release　　• Revision　　o Update　　o New Mail Code

Title

System 80 Model 50 Capabilities Overview

This announces the release of a revision to this document.

The System 80 model 50 is the latest and most powerful member of the Unisys System 80 family. The model 50 is a high-performance system designed to meet the processing needs of the largest current System 80 users.

This document describes the capabilities and features of the model 50 - from the streamlined single-cabinet design to the variety of Operating System/3 (OS/3) software products it supports.

This revision includes additions, corrections, and expanded information pertaining to:
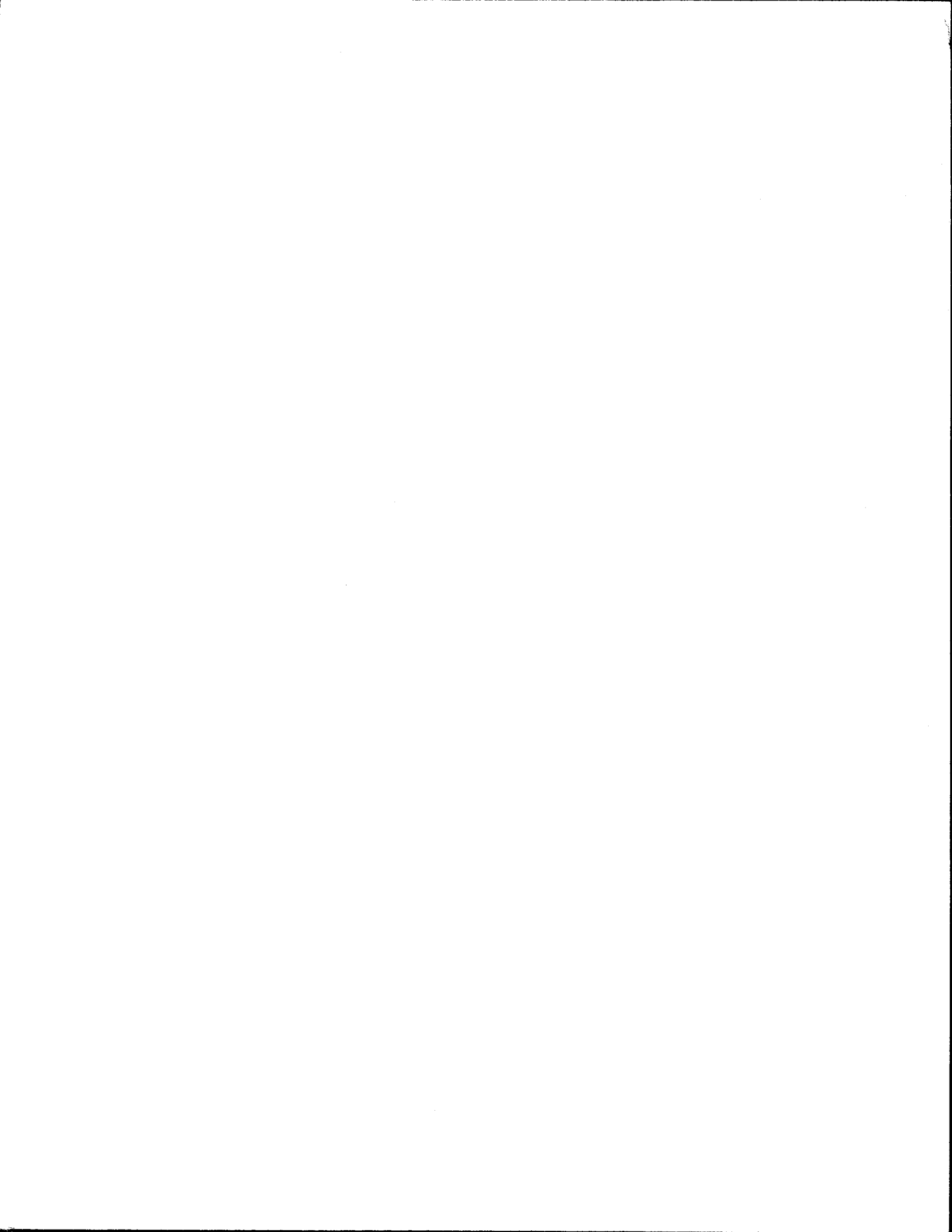
* USR9600-15 disk
* USR5073 tape

To order additional copies of this document:

* United States customers should call Unisys Direct at 1-800-448-1424.
* All other customers should contact their Unisys Subsidiary Librarian.
* Unisys personnel should use the Electronic Literature Ordering (ELO) system.

System 80
Model 50

Capabilities Overview

UNISYS

Contents

Introduction

The System 80 model 50 is the latest and most powerful member of the Unisys System 80 processor family.

The model 50 is a high-performance system designed to meet the processing needs of the largest current System 80 users. It provides four times the memory capacity and almost three times the processing power of a model 20.

But performance is only part of the model 50 story; there's also cost-effectiveness and investment protection.

The streamlined single-cabinet design of the model 50 significantly reduces system size, power consumption, and heat generation. And the flexible I/O design supports many existing System 80 peripheral subsystems, as well as new, low-cost peripherals.

The model 50 uses the latest version of the proven Operating System/3 (OS/3) software – providing efficient system control software as well as a full range of optional program product offerings, such as enhanced programming languages, fourth-generation language (4GL) information systems, and data base products.

The model 50 is also a participant in the Unisys Architecture. This assures you of a system that provides interoperability, a high-productivity application environment, and extensive transaction processing facilities.

Unisys Architecture provides the product and system integration that you want now – the kind of integration that will be critical in the future.

The System 80 model 50 addresses your needs of today, as well as the needs of tomorrow. It offers:

- Protection of your previous investments in hardware and software products

- The increase in processing power and system integration you need today

- A practical and necessary path for future growth

Some of the specific highlights of the model 50 follow. Then, this overview gives more detailed descriptions of the System 80 model 50 hardware and software.

Hardware Highlights

The System 80 model 50 offers many attractive hardware features, including:

- Between 2-1/2 and 3 times more processor power than the System 80 model 20

- Up to 64 MB of memory in a fully configured system

- Single-cabinet processor complex with a footprint under 8 square feet

- PC-based system console with a color VGA monitor

- Both selector and SCSI I/O interfaces

- Peripheral power unit for automatic power-on sequencing

- Unattended operation feature that allows you to automatically power on your processor at preprogrammed times without operator intervention

Software Highlights

The basic OS/3 software for the model 50 is a comprehensive new package called the System Platform Software, or SPS. SPS includes all the products formerly packaged as system control software (SCS) and extended system software (ESS). It also includes the integrated communications access method (ICAM) software.

Unisys Architecture

The Unisys Architecture addresses the complex needs of integrating the resources of today's growing, and increasingly diversified, networked computing systems.

The model 50 offers you the processing power and continued product support you expect from System 80. Unisys Architecture ensures that as the hub of your enterprise, your model 50 will also provide enhanced interoperation with other systems in your network, as well as increased product integration.

Through a combination of unique Unisys premium products, regional servers able to access an extensive range of application solutions, and a commitment to industry standards, the model 50 offers you effective use of your current resources and a flexible environment for future growth.

Transition

The model 50 is designed to be a convenient upgrade for the current System 80 user. Object program and data compatibility ensures that existing OS/3 data bases and most of your current applications can be used on the model 50.

A complete list of supported peripheral devices can be found in the Hardware section of this overview.

You'll see that many of your existing printer, tape, and disk subsystems, which connect to the selector channels, are supported. In addition, communications network devices like UNISCOPE® terminals, PCs with synchronous terminal emulation package (STEP), and many distributed communications processors are fully supported on the System 80 model 50.

Hardware

The processor complex of the model 50 is housed in a single cabinet containing the processing unit, main storage, I/O control, system control processor, maintenance hardware, and integrated disk and diskette drives.

The processor cabinet requires less than 8 square feet of floor space and is designed to conveniently fit into either an office or computer room environment.

Completing the model 50 hardware are the system console subsystem and the selection of disks, tapes, printers, and data communication devices that match your model 50 system to your data processing requirements.

The model 50 is designed for easy operation. This ease of use is enhanced by two optional hardware features: the peripheral power unit and the unattended operation feature.

This entire hardware system is monitored by a comprehensive set of confidence tests. They are used by both the customer and Unisys service representatives to ensure that the hardware remains in top operating condition.

Hardware Configuration

The following diagram illustrates a sample hardware configuration of a model 50 system. Everything above the shaded line is contained in the processor cabinet.

```
┌──────────MAIN STORAGE UNIT──────────┐   ┌────────────PROCESSING UNIT────────────┐
┌────────────┐  ┌────────────┐         ┌────────────┐ ┌────────────┐ ┌──────────────────┐
│  8-MB BMU  │  │  MEMORY    │         │ ACCELERATED│ │   BASIC    │ │INSTRUCTION CACHE │
├────────────┤  │  CONTROL   │         │ EXECUTION  │ │ PROCESSING │ ├──────────────────┤
│  8-MB BMU  │  │   UNIT     │         │   UNIT     │ │   UNIT     │ │  OPERAND CACHE   │
└────────────┘  └────────────┘         └────────────┘ └────────────┘ └──────────────────┘

┌──────────────────────────────── SYSTEM BUS ────────────────────────────────┐

┌────────────┐  ┌────────────┐      ┌────────────┐          ┌────────────┐
│  REMOTE    │  │  SYSTEM    │      │   POWER    │          │  CHANNEL   │
│MAINTENANCE │  │  CONTROL   │      │  CONTROL   │          │  ADAPTER   │
│ INTERFACE  │  │ PROCESSOR  │      └────────────┘          │   (CHA)    │
└────────────┘  └────────────┘                              └────────────┘
                            ┌────────────┐ ┌────────────┐
                            │ UNATTENDED │ │ PERIPHERAL │
                            │ OPERATION  │ │   POWER    │
                            │   (UAT)    │ │ UNIT (PPU) │
                            └────────────┘ └────────────┘
┌────────────┐  ┌────────────┐
│   40-MB    │  │  1.44-MB   │
│   DISK     │  │  DISKETTE  │
│   DRIVE    │  │   DRIVE    │
└────────────┘  └────────────┘
                ┌────────────┐      ┌──────────────┐ ┌────────────┐ ┌────────────┐
                │  SYSTEM    │      │SMALL COMPUTER│ │  SELECTOR  │ │  SELECTOR  │
                │  CONTROL   │      │   SYSTEMS    │ │  CHANNEL   │ │  CHANNEL   │
                │  PANEL     │      │  INTERFACE   │ │   (SEL)    │ │   (SEL)    │
                └────────────┘      │(SCSI) CHANNEL│ └────────────┘ └────────────┘
```

(shaded line)

```
            ┌────────────┐  ┌────────────┐  ┌────────────┐              ┌────────────┐
            │  SYSTEM    │  │   TAPE     │  │    DCP     │              │  PRINTER   │
            │  CONSOLE   │  │ SELECTION  │  │            │              │ SELECTION  │
            └────────────┘  └────────────┘  └────────────┘              └────────────┘
                            ┌────────────┐                              ┌────────────┐
                            │   DISK     │                              │   TAPE     │
                            │ SELECTION  │                              │ SELECTION  │
                            └────────────┘                              └────────────┘
                                                                        ┌────────────┐
                                                                        │   DISK     │
                                                                        │ SELECTION  │
                                                                        └────────────┘
```

- SVT 1120/23/24/25
- UTS 20/30/40
- U200/400
- PC (with STEP)
- X.25 PDN Network
- Async Terminals
- Bisync Terminal (2780/3780)
- U4000 Cluster Controller
- UNIX® with UNISCOPE Emulation
- BTOS Workstation with UNISCOPE Emulation
- X.21 Circuit Switched

┌──────────── SYSTEM MAXIMUMS ────────────┐
All maximums may not be
collectively achievable.

- Main Storage: 64 MB
- CHA: 2
- I/O Channels: 8
- SEL: 6
- SCSI: 4
- PPU: 2

Processor Complex

The processor complex consists of:

- A processing unit

- Main storage

- I/O channels

- System control processor

- Remote maintenance interface

- An integrated 40-MB disk and a diskette drive

Processing Unit

The processing unit consists of a basic processing unit (BPU) using high-speed, bit-slice microarchitecture and a six-stage instruction pipeline. The BPU is supported by separate instruction and operand caches (each 64K), which improve processor and system performance.

Performance is further improved by an accelerated execution unit (AEU), which performs several basic processing unit functions.

The model 50 processor supports the full OS/3 instruction set, as well as the new extended mode architecture associated with the system maximum of 64 MB of memory.

Main Storage

Memory is contained in 8-MB basic memory unit (BMU) modules. The system minimum of 16 MB can be expanded in a single main storage unit to 32 MB.

Each main storage unit also contains a memory control unit that performs memory control for the system. It provides double-bit error detection and single-bit error correction. It also logs soft/hard memory failures and additional system status.

The model 50 can be configured with two main storage units, providing both memory interleaving and a system maximum of 64 MB of memory.

Input/Output Subsystem

The I/O subsystem of the model 50 consists of two types of channels: selector channels and the optional small computer systems interface (SCSI) channels.

The selector channel supports communications devices and many existing Unisys peripherals – so you can continue to use many of your current devices. The optional SCSI channels let you expand your system with a new selection of peripherals utilizing the industry standard SCSI interface.

Both the selector and SCSI channels connect to the system bus via the channel adapter (CHA). The model 50 supports two CHAs. Up to four channels can connect to each CHA.

System Control Processor (SCP)

The SCP controls the system console, the remote maintenance interface, and the integrated peripherals. It also supports the system IMPL and IPL functions, the service frames, and the system clock.

Remote Maintenance Interface (RMI)

Remote maintenance for the model 50 is provided via the RMI. This feature allows the local support centers to quickly diagnose and resolve onsite problems as well as support remote operation.

Integrated Disk and Diskette Drives

The processor cabinet of the model 50 also contains a 40-MB hard disk drive and a 1.44-MB diskette drive. These devices are not for customer use. They support specific OS/3 and SCP functions like system error-logging and loadable microcode, microdiagnostics, and macrodiagnostics.

System Console Subsystem

The system console subsystem uses a Unisys PW^2 personal computer. The system console, which includes a 3-1/2-inch flexible disk, a 20-MB hard disk drive, and a VGA color monitor, provides faster and simpler console operations and an easy-to-use operator interface.

The model 50 allows you to use colors to highlight the operator console screen for maximum operator efficiency and screen visibility.
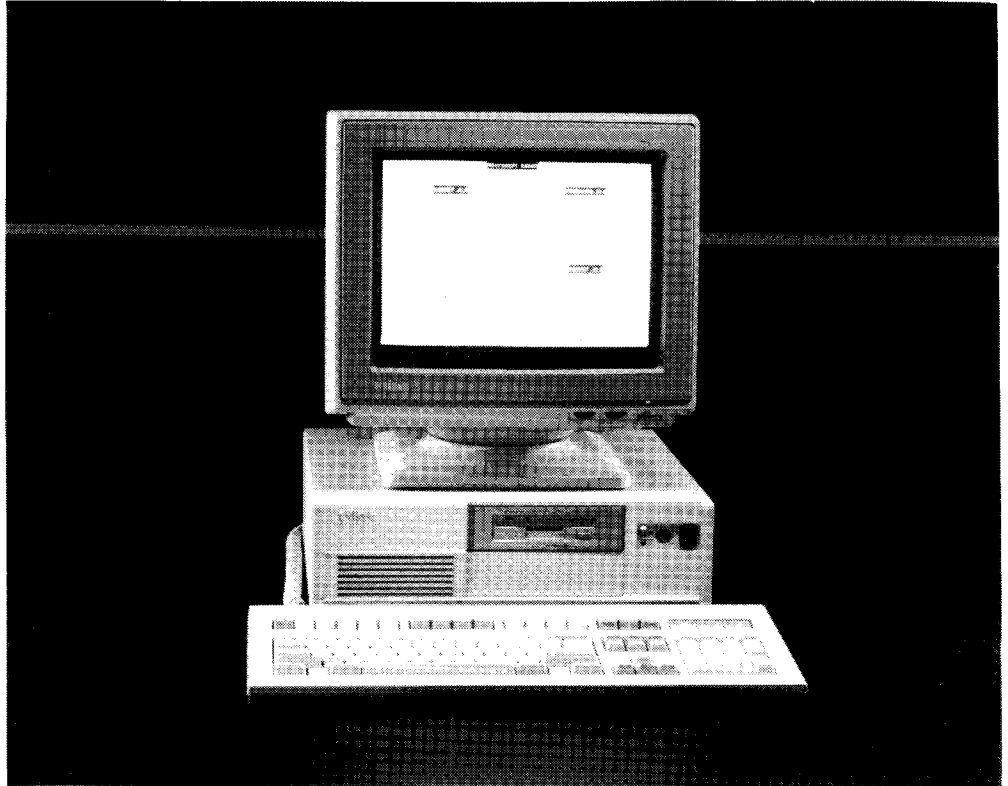
Data Communications Subsystem

Data communications for the model 50 is provided via distributed communications processors (DCPs).

All the communications devices and various terminals connect to the system through a DCP. The model 50 supports the DCP 5/15/25/30/35/40/50/55.

The DCP (except the DCP/5) connects to the model 50 via a dedicated selector channel. The DCP/5 is housed in a Unisys PW^2 personal computer and connects to the system through another DCP. All DCPs can be used as remote concentrators or as front-end processors, except the DCP/5, which functions as a remote concentrator.

Peripherals

The System 80 model 50 supports the following peripheral devices. More detailed descriptions of some of the latest devices follow.

Disk Drives

- M9720

- 8494

- USR9600-15

Tape Drives

- 2145-03

- USR5073

- BT3200

- UNISERVO 22/24/26/28 tape drives using 5055 controller

Printers

- 9246-25B

- 9246-14B

- 0770

- 0776

M9720 Disk Drive

The M9720 disk subsystem is a large capacity storage device that connects to the model 50 via the selector channel. The M9720 disk is available in two cabinet designs. From 4 to 8 disk drives can be housed in the 1-meter cabinet, and 4 to 16 drives can be housed in the 1.75-meter cabinet.
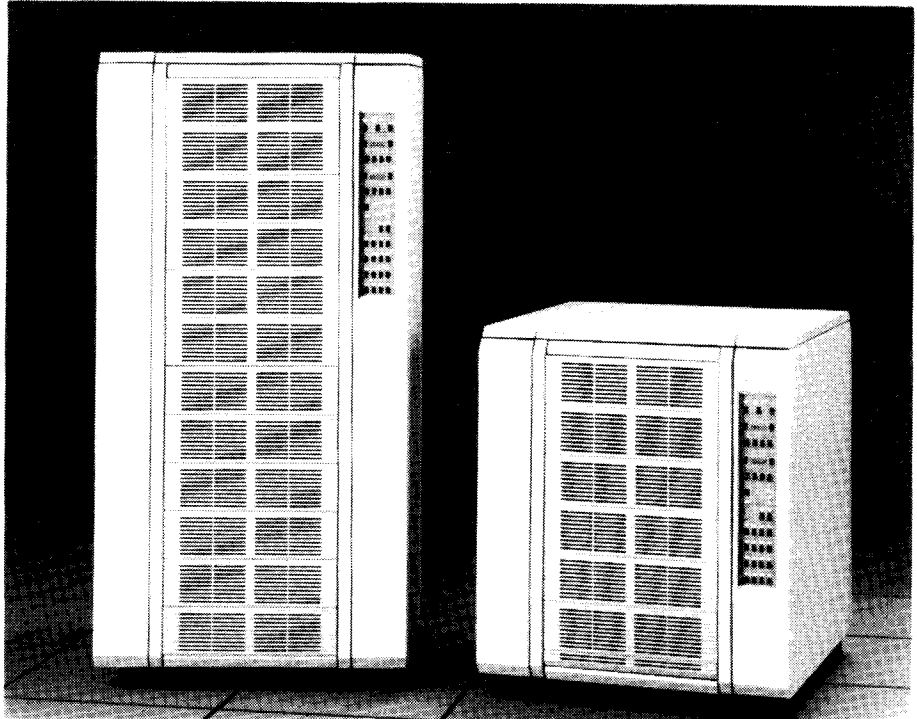
Record (blocks) format in bytes	512 (physical); 256 (logical)
Records per track	76
Tracks per cylinder	27
Number of cylinders	745
Data rate	3.0 MB/s
Average access time	18 milliseconds
Data capacity	782 MB
Disk media	Fixed
Rotational speed	3,600 rpm

USR9600-15 Disk Subsystem

The USR9600-15 disk subsystem is a freestanding subsystem that connects to the model 50 via the SCSI channel. The disks are contained in a 1-meter cabinet that can house up to 6 disk racks. Each rack contains up to 4 full high disks giving each cabinet a maximum capacity of 24 disks.

Data capacity (formatted)	597 MB
Rotational speed	3,600 rpm
Rotation period	16.66 milliseconds
Data rate	18-22 mega-bits per second
Track density	1,296 tracks per inch
Bytes per track	24,576
Sectors per track	96
Number of cylinders	1,214
Tracks per cylinder	20
Average access time	16 milliseconds

2145-03 Tape

The 2145-03 tape drive is a high-performance desktop tape drive that uses an open reel half-inch tape and a SCSI interface. It operates at 100 ips in streaming mode and 50 ips in start-stop mode. It is a 270-MB-capacity tape supporting 6250 GCR, 1600 PE, and 800 NRZI. This tape includes the significant features of read/write error detection and correction and automatic calibration.
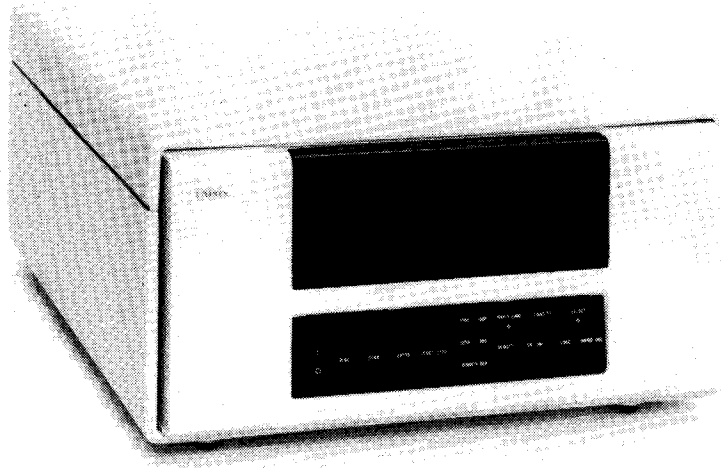
	Streaming Mode	Start-Stop Mode
Tape speed	100 ips	50 ips
Rewind time	190 seconds	190 seconds
Data rates	625 KB/s	312.5 KB/s
Data density	800 - 6250 bpi	800 - 6250 bpi
Recording format	9-track PE,GCR,NRZI	9-track PE,GCR,NRZI

USR5073 Tape

The USR5073 tape unit is rack-mounted in either UC-19 or UC-36 cabinets. It uses standard half-inch tape cartridges that hold approximately 200 MB of data. An optional automatic cartridge loader feature is available that provides automatic loading of tape cartridges into the tape drive. Magazines are available to hold either 5 or 10 tape cartridges.

The USR5073 tape unit connects to the model 50 via the SCSI channel. It can be configured to support up to 2 tape drives per controller.

Tape speed	78 ips
Rewind time	48 seconds
Data rates	3 MB/second
Data density	38000 bpi
Recording format	ANSI Standard X3B5

13

9246-25B Printer

The 9246-25B printer is a high-speed impact printer that features a power stacker, both front and rear control panels, and a variety of operator-replaceable print bands.
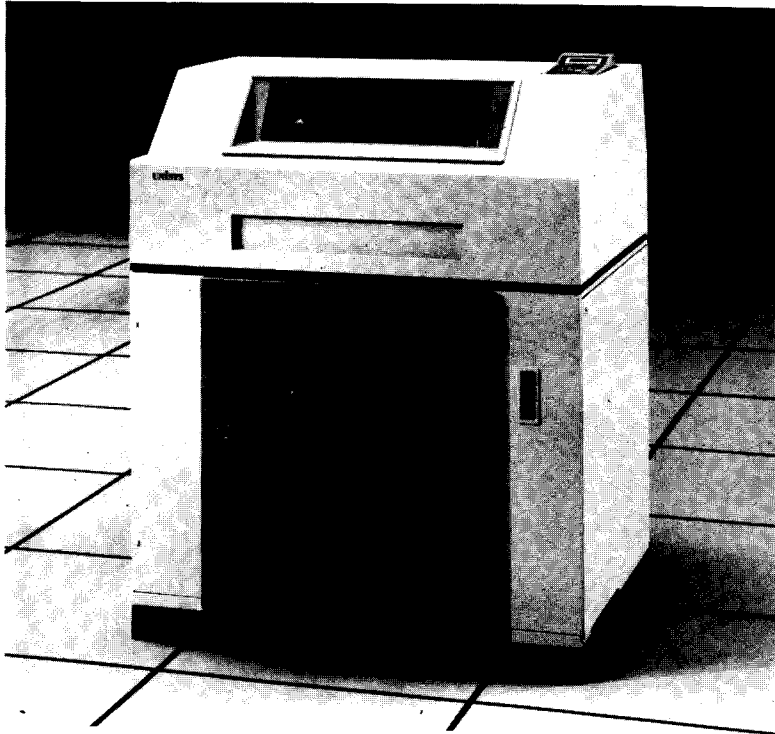
Print speed	2000 lpm for a 48-character set
Print line	136 columns wide
Printing density	10 columns per inch
Line spacing	6 or 8 lpi (operator selectable)
Forms width	4 to 17 inches (10.2 to 43.2 cm)
Forms length	8 to 14 inches (20.3 to 36.6 cm)
Number of copies	1 original plus 5 carbons

9246-14B Printer

The 9246-14B printer is a moderate-speed band printer that provides quality printing at a rate of up to 1,200 lpm with the 64-character-set print band. Standard features include operator-changeable print bands and component testing during the power-up cycle.
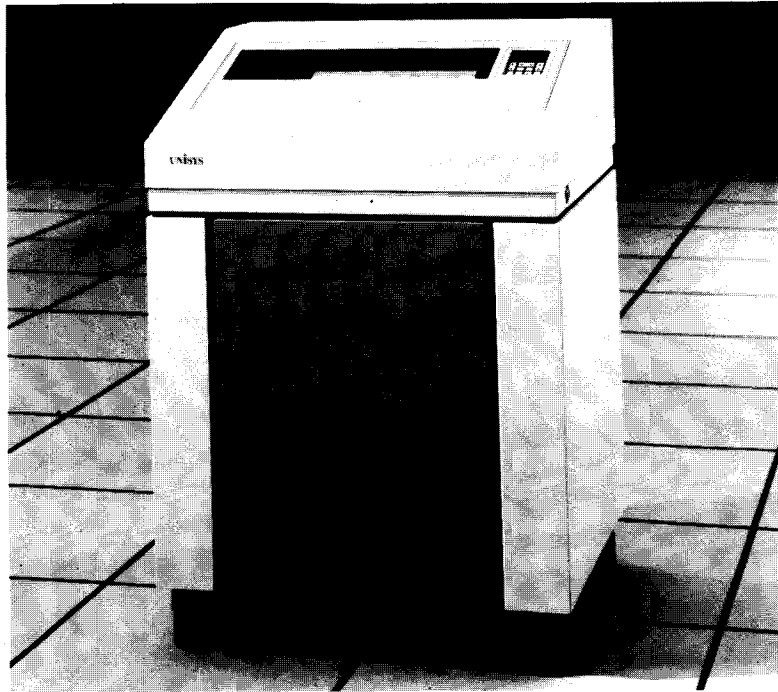
Print speed	1,200 lpm for a 64-character set
Print line	80, 132, or 136 columns wide
Printing density	10 columns per inch
Line spacing	6 or 8 lines per inch (operator selectable)
Line advance	14 ms at 6 lines per inch
Forms slew rate	15 inches per second (38 cm/s)
Forms width	3 to 16 inches (7.7 to 41 cm)
Forms length	3 to 14 inches (7.7 to 36 cm)
Number of copies	1 original plus 5 carbons

15

Ease-of-Use Options

There are two options that complement the ease-of-use design of the model 50:

- Peripheral power unit – This option permits you to control power to the peripherals as power is applied to the processor cabinet.

- Unattended operation – This option allows you to automatically power on your processor without operator intervention. In concert with the peripheral power unit, this feature lets you program your system to power on and initiate specific tasks at specified times.

Hardware Diagnostics

The efficient operation of the model 50 is enhanced by extensive hardware diagnostics. These not only routinely check the functioning of the hardware components, but they also make it easier to identify and correct hardware and software problems should they occur.

Several diagnostics packages are offered for the model 50:

- Resident diagnostics - perform basic hardware checking on the processor complex components automatically when the system is powered on. These include power-on confidence tests and initial microprogram load tests for individual components.

- Online diagnostics - are programs that check all peripheral devices to ensure proper system function and operation. These programs are under the control of the operating system and run concurrently with user jobs.

- Offline diagnostics - are programs that perform a variety of functions, including verifying the operating system load paths, checking system instructions, and checking the operation of the BPU, main storage, and I/O channels. These programs are initiated from the console or remotely by Unisys support personnel.

These tests are supplemented by the remote support provided by the Unisys customer support centers. Using the remote maintenance interface hardware, service personnel can run and monitor a number of tests to help evaluate and correct both hardware and software problems.

Hardware Summary

The following table summarizes the system hardware characteristics of the System 80 model 50.

System Characteristics

Component	Characteristics
Processing unit	Instruction processor (IP) Two sets of 16 general-purpose registers Four floating-point registers Six-stage pipelined processors Maximum system bus transfer rate of 40 MB/s
Instruction cache	64K bytes capacity
Operand cache	64K bytes capacity
Main storage unit	16 MB minimum, 64 MB maximum Maximum of two MSUs Single-bit error correction, double-bit error detection
I/O subsystem	Maximum of eight channels in any of the following combinations: • 6 SELs and 2 SCSIs • 5 SELs and 3 SCSIs • 4 SELs and 4 SCSIs Maximum of two channel adapters (CHAs)
Selector channel	Maximum of six SELs; supports data communications devices and the following peripherals: • M9720 disk • 8494 disk • BT3200 tape • UNISERVO tapes using 5055 controller • 9246-25B printer • 9246-14B printer • 0770 printer • 0776 printer
SCSI channel	Maximum of four SCSIs; supports the following peripherals: • USR9600-15 disk • 2145-03 tape • USR5073 tape

System Platform Software

The System Platform Software (SPS) provides the basic system software for the model 50. It ensures the efficient and flexible centralized control of System 80 activities. SPS includes all the OS/3 products formerly packaged as the system control software and the extended system software, as well as ICAM software including the distributed communications architecture termination system.

Current System 80 users will recognize many of the programs that form the basis of OS/3 operations, such as the supervisor, job control, interactive services, and data management.

OS/3 gives you a multiple job environment that allows concurrent operation of programs with immediate response to user inquiries and requests at terminals.

OS/3 also supports a wide range of peripheral hardware and makes optimum use of all available hardware facilities, resulting in efficient job turnaround time.

Supervisor

The supervisor provides the central control needed to make the system hardware, software, user programs, and interactive facilities work together efficiently. The supervisor services make many features possible, including:

- Multiple program processing

- Simultaneous interactive access

- Error control and recovery

- Automatic resource management

The supervisor manages and coordinates all system activities, handles randomly occurring events, initiates and coordinates execution of batch and interactive programs, and provides advanced programming facilities. The supervisor is one of the most complex components in the system, but it makes the System 80 an easy-to-use and efficient data processing system.

Supervisor Functions and Services

Task switcher	Determines the order in which the various tasks are given processing unit time.
Multitasking	Concurrently processes more than one task from one or more jobs.
Transient management	Schedules, locates, and loads transient (nonresident) supervisor routines.
Main storage management	Allocates and deallocates main storage as required by other system software.
System console and terminal management	Controls information to and from the system console and various terminals.
Timer and day clock services	Provide a system clock for timed, batch, or interactive activities.
Physical input/output control	Controls the dispatching, queuing, and interrupt processing for all I/O devices directly connected to the system.
Disk cache facility	Stores data in a reserved main storage cache to improve I/O performance.
Supervisor interrupt requests	Service the various interrupts generated by the system.
Error recovery and logging	Handles any program or machine error that causes an interrupt. Supports a variety of routines to isolate or correct diverse error conditions.
Security	Controls access to the system's interactive facilities.
Automatic boot procedure	Permits an automatic supervisor boot without operator intervention.

Task Switcher

A task is the smallest entity to compete for processing unit time. The system, as well as users, can initiate tasks. Tasks initiated by users 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 task switcher coordinates the processing of all current tasks and decides, based on established priority, which task should be processed next.

SUPERVISOR
CONTROLS PROCESSING THROUGH TASKS AND INTERRUPTS

JOB 1

| TASK | INTERRUPT (OPEN FILE) | TASK | INTERRUPT (CLOSE FILE) |

SUPERVISOR

JOB 2

| TASK | INTERRUPT (PRINT RESULTS OF PROGRAM) |

A18620
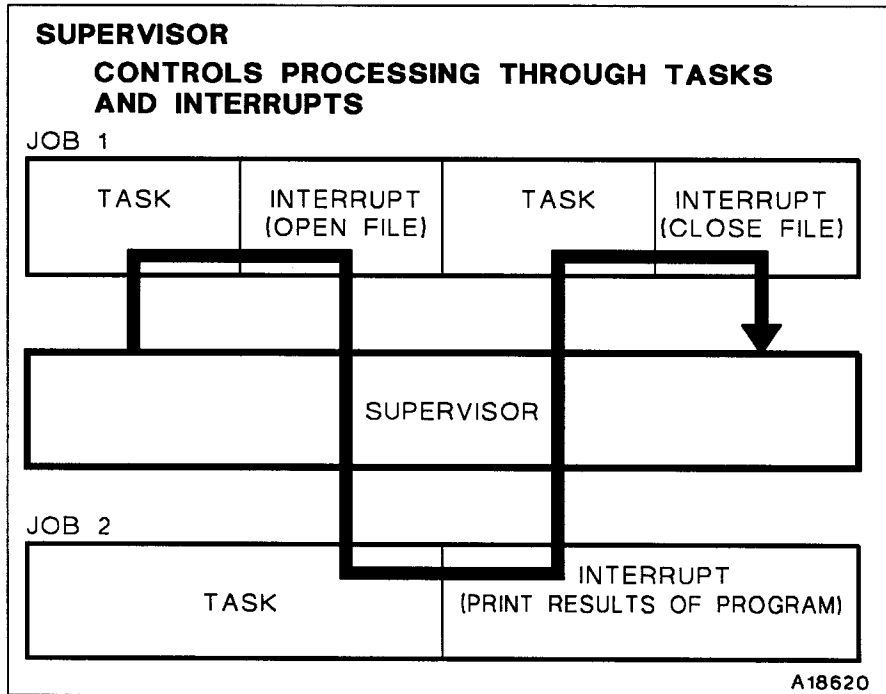
Multitasking

Multitasking is a programming technique that significantly reduces processing time. Each job consists of one or more tasks for the system to perform. Normally, each job step has only one task actively using the processing unit. Through multitasking, you establish a hierarchy of independent tasks so that several tasks from a single job step are active at the same time. Processor control passes from one task awaiting completion of an external event to another task. If the second task is from the same job, that job gets processed faster. The separate tasks defined for each job step are called subtasks. They vie with tasks from other jobs for processor time as independent tasks.

21

Main Storage Management

The supervisor ensures proper and efficient use of main storage by streamlining main storage operations and providing as much automatic storage management as possible. 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 times. In addition, the supervisor can:

- Dynamically expand the main storage regions of certain programs, jobs, and routines, when necessary

- Rearrange and consolidate job regions to provide as much contiguous free space as possible

- Temporarily suspend and remove jobs from main storage to make room for preemptive priority jobs

System Console and Terminal Management

System console management controls message displays and the operator's responses and commands. The screen images scroll 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.

Terminal management ensures the efficient operation of terminals when dedicated to a job. It provides the physical level support required to coordinate the terminal activities, such as data transfers, with the activities of the requested functions, such as interactive services.

Timer Service Management

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

The supervisor uses a simulated day clock to:

- Provide the time of day to tasks upon request

- Time-stamp messages and job accounting entries

Physical Input/Output Control System (PIOCS)

OS/3 performs all I/O operations with peripheral devices through the physical input/output control system (PIOCS). OS/3 PIOCS is composed of software routines that provide maximum throughput on all peripheral devices and allow expansion to support new devices.

PIOCS handles the queuing and initiating of all I/O commands and the processing of I/O interrupts in the following ways:

- Control – PIOCS receives control when a request is issued for an I/O operation. Control is not returned to the issuing task until the I/O request is completed. However, other tasks in the system can be activated while waiting for the I/O to complete.

- Queuing – Requests for I/O operations are initially queued, by priority, in device and channel queues.

- Dispatching – Dispatching follows queuing of an I/O order when a path to the device is available.

- Dispatch routines – The I/O dispatch routines perform needed service functions, 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 were queued.

- Completion of I/O request – Upon completion of an I/O order, an I/O status analysis is performed to determine whether an abnormal condition occurred.

Disk Cache Facility (DCF)

The DCF increases system performance by reducing the potential for I/O contention to disk devices. A portion of memory is allocated as a disk cache and managed by operating system software. Portions of disk information currently being accessed are transferred to disk cache. Selected I/O functions reference disk cache to determine if the information being requested resides in cache. If present in cache, the requested information is transferred to the requesting program without an I/O request to the disk.

When a particular area on a disk is read, the disk cache facility writes all or a portion of it (depending on the size of the cache entered when DCF was initialized) to a reserved cache storage area. A physical I/O is unnecessary in subsequent reads of that same disk area since that data is already present in storage.

Reducing the number of required I/Os results in the following reductions:

- Queuing delays – I/O requests must wait to be issued in turn.

- Disk seek time – The disk accessor moves to the proper cylinder.

- Latency – The disk rotates so that the disk read/write heads reach the proper record.

- Data transfer time from disk to storage.

DCF is normally initialized during initial program load (IPL). Commands also let the operator do the following:

- Initialize DCF

- Override defaults

- Activate and deactivate drives to DCF

- Change the cache segment and buffer sizes

- Shut down DCF

- Monitor DCF activity

- Select files to be stored in cache

Supervisor Interrupt Requests

A supervisor interrupt stops the normal processor flow and must be completed in some way before processing can continue. Requests are generated by the various hardware and software components. OS/3 recognizes eight types of interrupts.

Supervisor 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.
Input/output	Occurs in response to signals from I/O channels.
Exigent machine check	Indicates a malfunction in the processor from which the supervisor cannot recover.
Repressible machine check	Indicates a malfunction in the processor from which supervisor recovery is possible.
Timer interrupt	Generated by the processor interval timer.
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	Provides dynamic monitoring of executing programs by storing information about the current instruction when a specified event occurs.
Restart	Occurs when a restart command is issued from the system console and can be used to put a stopped processor in the operating state.

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

Error Recovery

When an error causes a program interrupt, the supervisor determines the type of error, such as program check or protection violation, and the type of job involved.

If the supervisor detects an error, it halts the system and attempts a restart from that point. If recovery fails, the supervisor collects information for an orderly abnormal termination.

The supervisor examines machine check interrupts to determine if the error is recoverable. If the error is not recoverable or recovery fails, the supervisor abnormally terminates the system. The supervisor collects and logs all information pertinent to the abnormal termination.

Error Logging

The error logging facility records hardware and software errors in the system error log file. You can retrieve records from this file and use them to prepare statistical reports. The error logging facility is loaded at system generation time. The supervisor then determines the type of errors to be recorded and the devices to be monitored. Error logs are collected for every device configured into the system.

Error log records are collected for four types of errors:

- Peripheral device

- Communications

- Machine check

- User specified

During the initial program load (IPL) of a new session, you can retain the error log file from the previous session or reset the file with the same set of collection parameters or a new set. During operation, you can alter the collection of record types.

Diagnosing Problems

The supervisor provides a number of services for diagnosing system and program problems:

- Monitor routines trace the execution of a program to locate and correct errors. The routine interrupts each instruction before it executes and tests for conditions specified in the monitor input. You can monitor an entire program or a portion of it.

- Snapshot displays of main storage provide partial main storage printouts to aid in solving main storage problems. The area of main storage 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 provide program event information under the following conditions:

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

  - Program or operator request dump – This routine lets the operator or any program request a main storage dump in the same format as the abnormal termination dump.

  - System failure dump – This routine is used when an abnormal condition occurs and other dump programs cannot be used.

- Standard system error message interface provides complete and specific error messages for display on the system console.

- Error codes returned by the supervisor to user jobs are standardized to provide a uniform interface for all system services.

Security Maintenance Utility

The security maintenance utility (SMU) controls access to the system's interactive facilities through security information located in the system security file. The system administrator determines who is able to access the system and assigns identification to each system user. The security maintenance utility also accounts for computer time used by interactive system users and automatically executes predefined sets of interactive commands when users log on.

The security maintenance utility performs its tasks by creating three types of profiles:

- A user profile contains security and accounting information, as well as execution and command profile names.

- An execution profile contains the commands for automatic execution at logon, as well as the execution profile name.

- A command profile lets the system administrator specify the system commands a user can enter from a terminal.

The system administrator decides what to include in each of these profiles and can display or modify them.

In addition, the security maintenance utility may restrict access to the system by using a logon procedure that:

- Requires the user to enter a password

- Requires a user profile definition for each individual

- Can allow access only during a specific time or date range

- Prevents a user from logging on after a specified date

Automatic Boot Procedure

The automatic boot procedure (ABP) feature performs a supervisor boot of the system without operator intervention.

When ABP is in effect, the system is automatically booted with a default supervisor if there is no response to the IPL's supervisor/boot-dvc query within a specified time.

When ABP is used with the unattended operation and peripheral power unit hardware features, the system administrator can configure the system to automatically:

- Power on the processor and associated peripherals at a specific time

- Boot the default supervisor

- Execute the console execution profile

28

The system administrator uses interactive services to specify the default supervisor, set the timeout interval, and define the console execution profile.

During IPL the system administrator can override the ABP and specify a different supervisor by responding to the supervisor/boot-dvc query within the time limit.

System Service Programs

OS/3 provides a number of system service programs, including:

- System librarians

- Linkage editor

- Catalog manipulation utility

- Disk and tape initialization routines

- Dump routines

- Disk dump/restore

System Librarians

OS/3 includes two librarian utility programs to maintain system access technique (SAT) and multiple indexed random access method (MIRAM) libraries.

Programs and elements, such as language processor source modules, language processor output (relocatable object) modules, and system-executable load modules, reside within a library, which is either a system or private file in the OS/3 environment.

The two librarians have a different set of supported functions; however, both librarians can perform the following operations:

- Copy modules from one file to another

- Delete modules from a file

- Print listings of modules

- Rename modules

Both librarians produce a printed listing of all operations performed during a given librarian session. This listing is called a librarian map.

Linkage Editor

Compilers produce object modules as output. You can tailor these modules into an executable program by using the OS/3 linkage editor. The linkage editor can structure object modules from various language processor compilations into a single loadable program. Additionally, the object modules can contain cross-references to each other for the specific purposes of program execution and communication. The linkage editor resolves such cross-references between separately compiled modules when it collects these elements and constructs a loadable program.

Catalog Manipulation Utility

The catalog manipulation utility maintains the system catalog file that holds the device assignment sets for often-used user files. Once a file device assignment set is placed in the system catalog file, it is then accessed through one job control statement. The catalog manipulation utility lets the system administrator assign a protective password to the catalog file, make backup copies of it, 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.

Disk and Tape Media Initialization Routines

The disk and tape initialization, or prep, routines check the condition of the magnetic storage media and prepare them for use by the system. The disk prep routine is used to control alternate track assignments, defect skipping, and surface analysis functions. The tape prep routine is used to control tape density and format (ASCII or EBCDIC), in addition to creating a standard label set.

Disk Dump/Restore

The disk dump/restore utility lets you make backup copies of a disk volume. It executes as an interactive program or runs as a batch job. For interactive execution, you enter control specifications for the program in response to a dialog displayed on the terminal screen. Once you provide all appropriate information, the program executes and performs the specified operation.

If the disk dump/restore utility executes as a batch job, through job control, you prepare a job control stream that includes the appropriate program control statements.

Dump Routines

OS/3 has a variety of dump routines to aid in the diagnosis of subtle hardware and software problems. These dump routines include the following:

- JOBDUMP – This dump lists all the information in the system relating to the job. Two routines can produce a job dump: the job dump routine and the user end-of-job (EOJ) dump routine. Both routines produce the same information, but the formats are different.

- SYSDUMP – This dump displays the entire contents of storage in hexadecimal format with English explanations of many of the system's tables and structures. The system dump listing is divided into several parts, each part corresponding to a system component.

- SYSDUMPO – This dump performs the same function as SYSDUMP, but allows you to specify what portions of storage are displayed.

- MINIDUMP – This is a dump of only the actual main storage regions involved in the system error or HPR.

System Activity Monitor (SAM)

The system activity monitor (SAM) evaluates system efficiency by monitoring and recording various system activities. It is optionally configured during system generation.

The monitor measures:

- Processor and main storage use

- Disk use

- Input/output operations

- Supervisor interrupts

- Various communications activities

The information gathered by the monitor allows you to:

- Detect possible production bottlenecks

- Develop an optimum job mix

- Change system variables to enhance system performance

- Determine the optimum use of the system's peripheral devices

- Make use of capacity planning

Job Control

Job control manages the system resources and prepares jobs submitted for execution.

OS/3 performs job control services at three different times:

- Before execution of the initial job step

- During the transition between job steps

- At the conclusion of the job

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

Job Control Services Capabilities List

Automatic job scheduling and initiation

Automatic main storage allocation

Device assignment

Initiation of interactive services for user input

Volume and file label processing

Retrieval of cataloged control streams for subsequent modification and execution

File cataloging

Program restart from a checkpoint

Query of terminal for run-time variables

Scheduling of additional jobs from within a job

Control stream and data storage

Dialog for Job Control Stream and Jproc Preparation

The job control dialog, supplied by Unisys, is a product that leads you step by step through building a job control stream or a jproc from a terminal. A control stream is a group of sequenced statements, written in JCL, that defines a job and directs its execution.

The job control dialog lets you choose the statements and system jprocs required from lists of menu items. It requests parameter values when necessary. An experienced programmer can use this dialog to build a syntactically correct control stream quickly. A novice can use the HELP screens to learn about job control while building a valid job control stream.

The job control dialog automatically stores the control streams and jprocs in either the system job control library file or an alternate library file that you specify.
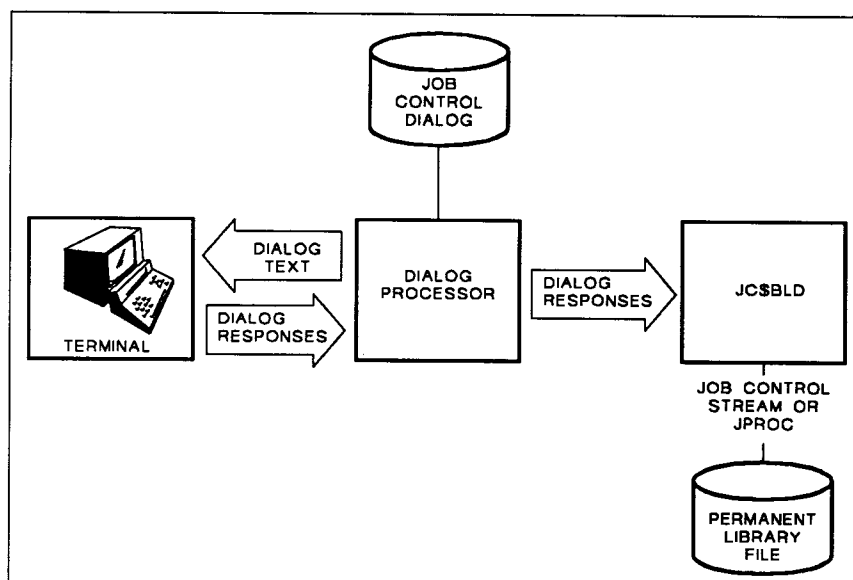
A18622

File Cataloging

The file cataloging facility lets you control file usage and restrict use of selected files to only authorized individuals by assigning protective read and write passwords. File cataloging also builds and maintains system generation files. It is a convenient facility because it reduces the amount of job control required to access a file.

The file cataloging facility also lets you alter a file while retaining a copy of it as it was before alteration. This process creates generation files. For example, three generations of a payroll file could be kept: present payroll, week-old payroll, and 2-week-old payroll.

A catalog manipulation utility is provided for the system administrator to perform certain maintenance functions on the catalog.

Interactive Services

OS/3 offers an array of interactive programming features that improve programmer efficiency and help make the system easier to use for both experienced and novice personnel.

Interactive services offers the following features and benefits:

- Increased productivity – Faster response time and simpler data entry and retrieval methods result in more efficient system usage.

- Easy access – Every authorized member of an organization can access the system through easy-to-use facilities.

- Security – Protective passwords and user identification and command restriction features provide effective security.

- Simplified programming – Interactive communication with the system control software simplifies the entire programming effort.

- Interactive access – Variable data can be entered into jobs currently running in the system.

- Help for novice programmers – Job control dialogs assist novice programmers in building a valid job control stream.

- Help for operators – Soliciting SYSGEN parameters through a SYSGEN dialog assists operators in system generation. A data utilities dialog similarly prompts operators to perform system utilities.

- Screen formats – You can create fill-in-the-blank screens to input, maintain, and display data.

- Multiple jobs – The System 80 model 50 supports 48 job slots, of which 47 are available for user jobs and 1 for interactive services.

Much of the model 50's interactive capability is provided by the following processors:

- Command processor

- Dialog processor

- Screen format processors

Command Processors

The command processors provide the interface between the terminal and the various components of the OS/3 software. These processors allow you to use a set of interactive commands directly from a terminal to:

- Direct operation of the system or a system component

- Query the system for specific information

- Direct job execution

- Create and modify source programs, data files, and job control streams

You can also use a terminal either as an input/output device dedicated to a job or system component or as a program development tool using various interactive facilities, such as Editor, TPS^{TM}, ESCORT^{TM} and BASIC. These interactive facilities interface with:

- Information management system (IMS)

- Data base management system (DMS)

- Integrated communications access method (ICAM)

- Distributed data processing (DDP)

Dialog Processing Services

OS/3 dialog processing services lets programmers and operators communicate easily with programs and procedures by providing interactive, conversational dialogs. This is particularly beneficial to novice operators and nonprogramming users.

Dialogs consist of a series of queries to which the user responds with the appropriate information. The dialog processor displays the dialog text, and users respond to the dialog through the keyboard. This input is routed back to the program that the dialog complements. For example, a dialog is written to ask the user the type of report desired. The responses cause the program to produce the requested report.

Unisys supplies certain dialogs for interactive system functions. You can also write your own for use with your application programs. User-written dialogs are written in dialog specification language (DSL). You can create dialog sessions for application programs written in any of the OS/3 programming languages. You can add to or change the dialog information.

DSL source code is submitted to the dialog specification language translator, which compiles the source code to produce the desired dialog, stores the dialog in a permanent file, and produces a compilation listing.

A18625
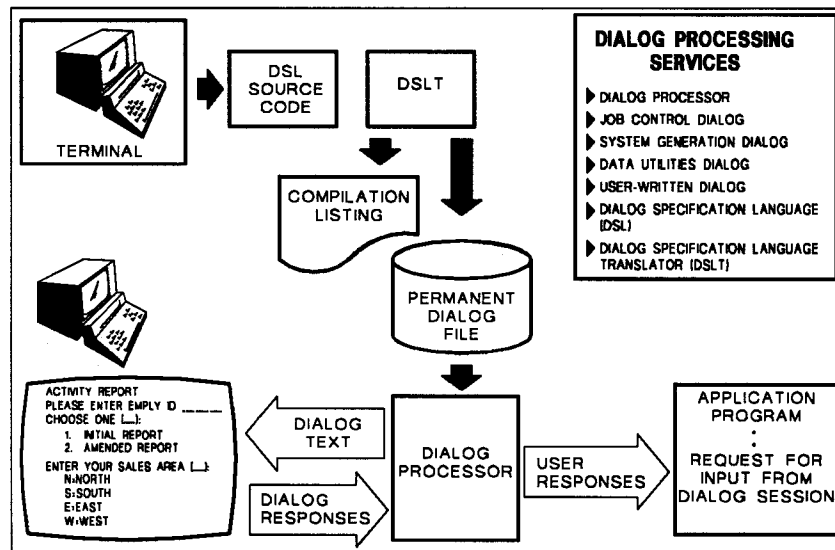
The dialog processor produces a printed summary and an audit file of each dialog session. You can use these as guides to change responses in subsequent sessions.

Screen Format Services

OS/3 screen format services simplifies variable data entry by using prestored displays called screen formats. Using screen formats makes data entry much easier because it presents users with a form on the screen which can be filled out by entering data in designated spaces.

Screen format services lets you:

- Design personalized fill-in-the-blank forms (screen formats)

- Route data entered in response to a screen format back to the application program that requested it
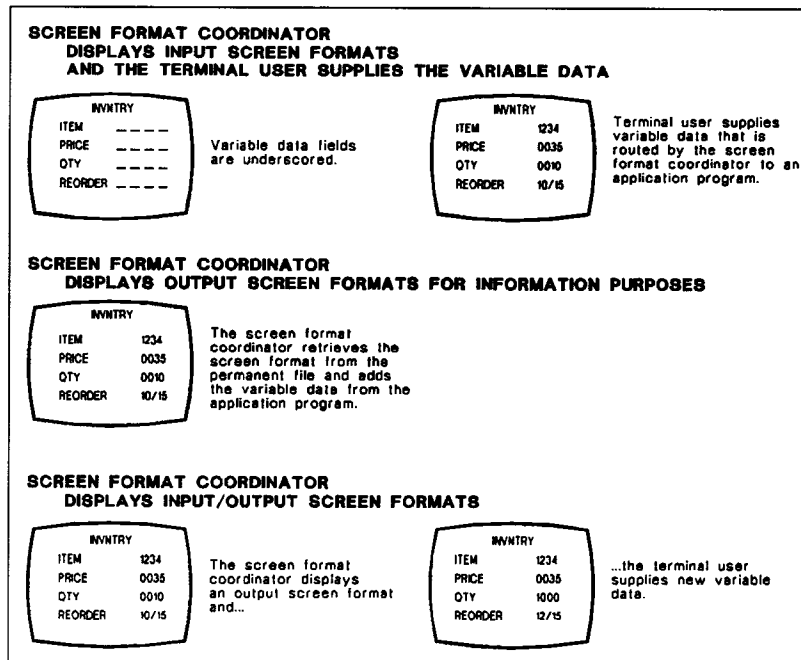
- Display filled-in screen formats to a terminal for informational purposes

37

Screen format services consists of two software processors – the screen format generator, which helps you create screen formats, and the screen format coordinator, which manages screen formats once they are created.

You use the screen format generator to interactively create the template-like screen formats. The screen format generator prompts you for information and provides HELP screens to simplify creating and modifying screens.

The screen format coordinator is activated in response to a program request. It retrieves the appropriate screen formats from the permanent file and displays them with either blank input fields or fields filled in with variable data. The variable data is stored with the program. If a program calls a screen format that serves 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 terminal 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 TERMINAL USER SUPPLIES THE VARIABLE DATA

INVNTRY
ITEM   _ _ _ _
PRICE   _ _ _ _   Variable data fields
QTY   _ _ _ _   are underscored.
REORDER   _ _ _ _

INVNTRY
ITEM   1234   Terminal user supplies
PRICE   0035   variable data that is
QTY   0010   routed by the screen
REORDER   10/15   format coordinator to an
                         application program.

SCREEN FORMAT COORDINATOR
  DISPLAYS OUTPUT SCREEN FORMATS FOR INFORMATION PURPOSES

INVNTRY
ITEM   1234   The screen format
PRICE   0035   coordinator retrieves the
QTY   0010   screen format from the
REORDER   10/15   permanent file and adds
                         the variable data from the
                         application program.

SCREEN FORMAT COORDINATOR
  DISPLAYS INPUT/OUTPUT SCREEN FORMATS

INVNTRY
ITEM   1234   The screen format
PRICE   0035   coordinator displays
QTY   0010   an output screen format
REORDER   10/15   and...

INVNTRY
ITEM   1234   ...the terminal user
PRICE   0035   supplies new variable
QTY   1000   data.
REORDER   12/15

A18623

38

Consolidated Data Management (CDM)

The consolidated data management (CDM) system is an interface between user programs and the I/O facilities of the supervisor. CDM helps you access data files on peripheral devices without coding routines for blocking/deblocking, buffering, and communicating with PIOCS. The following features are offered:

- A single-access method for disk files

- Shared data management modules

- Device independence

- Support of the interactive features and terminal data transfers

Logical Input/Output Control System

The logical input/output control system (IOCS) modules that control each access method are sharable subroutines. When referenced within a user program, these subroutines are dynamically loaded into main storage and available to many user programs.

Disk Access Method

The CDM access method for disk files is the multiple indexed random access method (MIRAM). MIRAM simplifies data management requirements for disk files by offering a single-access method that provides several ways to access a disk file. The records of a MIRAM file can be accessed as follows:

- Sequentially in order of placement

- Sequentially by ascending key

- Randomly by multiple keys

- Randomly by relative record number

MIRAM conserves main storage space by providing a single data management module for disk access shared by every user program, rather than having several data management modules for each access method. A fast-loader utility (MILOAD) allows CDM users to efficiently load keyed MIRAM files.

Workstation Access Method

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

- Automatic support for the terminal as an interactive programming device, including support of the interactive features that cause screen displays and management of the screen displays

- Transfer of application data and messages between the terminal and user job

- Console-like capability between the user program, system console operator, and terminal operators

- Support of terminal function key capabilities

- Device independence

The workstation access method is enabled automatically. You can also control it through a set of declarative and imperative macroinstructions that connect terminals at a logical level through the common data interface.

Magnetic Tape and System Printer Access Methods

I/O operations are performed in sequential order on magnetic tape subsystems and printers. Records are handled from first to last, according to physical placement. CDM provides sequential access method modules to handle the access requests of user programs for these device types.

Integrated Communications Access Method (ICAM)

ICAM is designed for the beginning as well as the advanced 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.

An ICAM configuration includes:

- Software modules specified during system generation

- System utilities required by the ICAM network

- DCP interface control

- Terminal devices

- Programs that interface with the network through the DCP

ICAM features include the following:

- Message queuing

- Multiple destination routing

- Activity scheduling and priority control

- Timer service

- Checkpoint/restart

- Journal control and statistics accumulation

- Trace facility

- Dynamic buffer expansion

- Dynamic session control

ICAM supports end users utilizing the following protocols:

- UNISCOPE

- UDLC

- Async

- Bisync

- X.25 PDN Network

- X.21 Circuit Switched

ICAM Components

- Distributed communications architecture termination system (DCA TS) – coordinates 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 manages outgoing message queues. DCA TS also protects messages from being lost when communications links break down and provides data throttling and recovery services.

    ICAM supports the DCP/5, DCP/15, DCP/25, DCP/30, DCP/35, DCP/40, DCP/50, and DCP/55 running Telcon as remote DCA termination systems, allowing you to function in several modes. The DCP is connected to the model 50 by a selector channel (except for the DCP/5, which connects only via another DCP).

- Communications control area (CCA) – contains 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 terminal and queue tables within the CCA. These flags control polling, indicate the operational status of communications hardware, and reflect the current disposition of message queues.

    Each CCA contains a pool of network buffers under 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 queuing is specified. When disk storage queuing is specified, these network buffers provide intermediate storage during the active input or transmit phase of a message.

- Message user service task (MUST) – provides a message staging service that isolates a message processing program from device dependence. MUST copies input message data from the network buffer pool into designated work areas. It also copies output message data into the network buffers.

  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) – is a series of ICAM overlays performing functions that are neither time dependent nor frequently used. DUST performs the following functions:

  - ICAM initialization

  - Network initialization

  - Program termination

  - System console message processing

- Global user service task (GUST) – allows formation of a global, or nondedicated, communications network that permits several user programs to be accessed concurrently. The GUST routine controls network and line requests and line releases in a global environment.

ICAM Interfaces

In addition to providing a global communications network for the System 80, ICAM also supports the following interfaces for individual applications:

- Standard message control program interface (STDMCP) for user programs (including COBOL 74)

- Transaction control interface (TCI) for IMS and TPS

- Demand mode interface (DMI) for interactive services, distributed data processing, remote batch processing, and RPG II telecommunications

Data Utilities

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

- Make a copy of a data file on the same or different type of storage device

- Reformat the records of a data file

- Select or delete specific areas of a file

- 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 processes files stored on either disk or tape. Files can be transferred from one medium to the other, and printed listings can be made from files on either. You can also perform such device-related functions as label checking, tape rewinding, sequence checking, and writing protected files.

The DATA program executes as an interactive program initiated through a simple keyin command, or it runs as a batch job under job control. The DATA program has the same capabilities regardless of how it executes.

Sort/Merge

OS/3 sort/merge operates as an independent sort/merge program or as a modular subroutine. As an independent sort/merge program, defined and initiated by job control, it is constructed as a processor with interrelated modules operating within the framework of a system driver program. As sort/merge executes, 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 gives you flexibility and freedom. It can be called from a language processor program. The modular structure gives you efficient operation despite variations in hardware configurations and data requirements.

Sort/Merge Capabilities

An interface that permits disk or magnetic tape to be used as work area	Input and output that can be associated independently with disk 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	Specification of up to 255 key fields
• Binary (signed or unsigned)	
• EBCDIC data in ASCII collating sequence	Sorting of noncontiguous key fields in ascending or descending sequence
• Decimal (signed zoned and unsigned zoned)	
• Leading and trailing sign numeric	User specification of alternate collation sequence
• Overpunched leading and trailing sign numeric	
• Floating point (single/double precision)	Sorting of two or more different characters with the same collating
Execution of input and output code	value (multicharacter sort)
Shared input and reserved output devices	Performing of data validity and data integrity checks during sorting
A merge-only capability	
	Automatic extended sort output file
Convenient restart procedure	
	Optimized IRAM/MIRAM file handling
Printing of VOL and LBL information	

SORT3

Unisys also supplies an IBM® compatible sort, SORT3. SORT3 is an OS/3 program initiated through job control that is compatible with sort programs provided by IBM for its System/3, System/32, and System/34 data processing systems.

SORT3 can process disk or tape files and:

- Rearrange the records in a file

- Reformat the records in a file

- Summarize fields in the records

- Select specific records from a file

SORT3 does the following:

- Displays messages at terminals and on the printer

- Prints LBL information

- Provides keyword specifications for month, day, and year

- Optimizes IRAM/MIRAM file handling

- Provides a maximum overflow field length of 256 bytes

SORT3 performs three different types of sorts: address out sort, tag sort, and summary sort. The output from the address out sort is a 10-byte (binary) relative record number of the records in the input file. The tag sort output is a file of sorted records that contains control fields and data, control fields only, or data only. The output of the summary sort can be any of the following:

- Control fields only

- Data fields only

- Summary data fields only

- Data fields and summary data fields

- Control fields and summary data fields

- Control fields, data fields, and summary data fields

Spooling and Job Accounting

Spooling increases the throughput of System 80. Data from low-speed peripheral devices is transferred to disk storage independently of the program that uses the data. When a user program logically retrieves data from a low-speed peripheral, it physically retrieves 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 are later transferred to a low-speed device under system control. Spooling of output allows concurrent use of a specified device by multiple programs.

OS/3 spooling consists of the following routines:

- Spooler cooperative – The spooler stores data output from a user program in the spool file until a printer is available.

- Output writers – Output writers provide local or remote users with output from user programs that receive identical service regardless of final destination.

Spooling is supported for a printerless System 80. It generates printer files for subsequent printing on an OS/3 system that has a physical printer configured. Spool print files can be routed to auxiliary printers. These printers can be connected to a UTS 400, UTS 4000, or SVT terminal. Additionally, spool output can be routed to a PC (using STEP software) or to your UNIX system.

In a normal 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, you can save all the accounting data generated for each job in a spooling LOG file.

The system console and job log accounting utilities gather selected information from the system spool LOG file, sort it 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, or selected portions, of the spooling LOG file onto a disk or tape file for use as input to the joblog report program or the user accounting routines. You initiate the system log accumulation routine 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 information and sorts it according to user specifications. The joblog report program is also initiated by running a canned job control stream. You select the type of accounting records to include in the report and the manner in which they are sorted.

Software Installation Facilities

Unisys delivers the basic OS/3 software on a set of release tapes. Additional software, like system console software and optional OS/3 program products, is delivered on 3-1/2-inch diskettes.

Software installation involves the transfer of this software to a system disk. Once the software is copied onto the disk, the disk is referred to as the system-resident volume, or SYSRES. The SYSRES disk holds all system software and must remain online when the system is in operation.

Unisys provides installation routines as part of the standard OS/3 release. These routines help install the following:

* Initial release of OS/3 software

* Any new software received between releases

* Updated software as new releases of OS/3 become available

* System maintenance packages and changes

System Generation Facilities

System generation, or SYSGEN, is the process of defining the system's hardware configuration to OS/3 and generating, or creating, the control elements required to meet particular processing requirements. To simplify the procedure, Unisys provides the following SYSGEN facilities:

* Dialog

* Parameter processor

* Job control streams

The SYSGEN dialog is an easy-to-use facility for preparing and processing required parameters. You select SYSGEN parameters in response to queries displayed by the system. OS/3 provides default values for omitted or incorrectly specified parameters to ensure that a usable system is generated.

The dialog accepts your choices and passes them to the SYSGEN parameter processor, which then validates the selections and generates a series of job control streams.

You then execute the actual system generation by interactively running the generated job control streams through simple keyins.
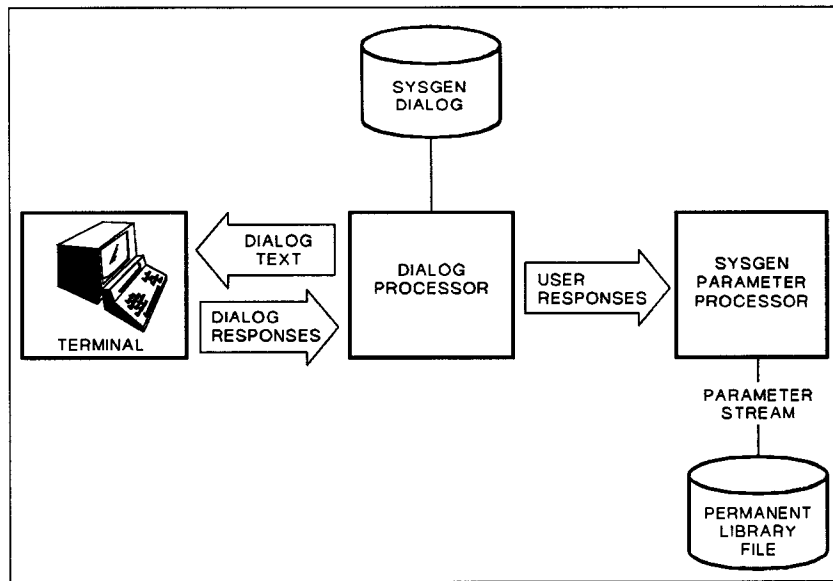
A18624

Installation Verification Programs

Unisys supplies a set of programs to test the functional capability and operation of the various components included on the resident disk unit. These programs are supplied as part of the supervisor functions and should be run after SYSGEN is completed. These installation verification programs (IVPs) verify that the software was installed correctly on your system.
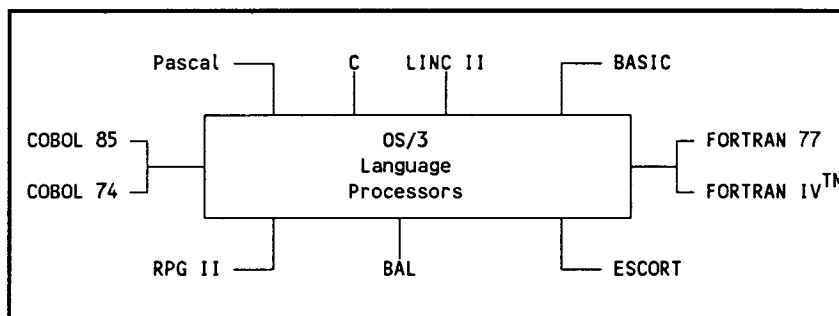
Program Products

The System 80 model 50 offers a wide range of program products. Among these products are a variety of programming languages, an interactive general editor, information and data base management facilities, distributed data processing, and interactive system communications.

Many of these program products are Unisys premium products, specifically designed to provide the System 80 user with convenient and efficient processing tools. The fourth-generation language LINC™ II is a prime example of the kind of powerful solutions System 80 can provide.

Languages

OS/3 supports a number of enhanced language processors for use with the System 80 model 50. These include COBOL 85, FORTRAN 77, C, and Pascal, which represent the application interface supporting the Unisys Architecture. The wide range of available language processors provides flexibility in the preparation of a variety of business, scientific, and report-oriented application programs.

```
                Pascal ─┐      C    LINC II      ┌─ BASIC
                        │      │      │          │
  COBOL 85 ─┐   ┌───────┴──────┴──────┴──────────┴───┐   ┌─ FORTRAN 77
            ├───┤               OS/3                  ├───┤
  COBOL 74 ─┘   │             Language                │   └─ FORTRAN IV™
                │            Processors                │
                └───────┬──────────┬──────────┬───────┘
                        │          │          │
                RPG II ─┘         BAL          └─ ESCORT
```

COBOL 85 and COBOL 74

COBOL is a powerful, general-purpose, English-like language designed for business applications, such as payroll, accounting, inventory, and personnel management.

The COBOL supplied by Unisys also offers a powerful sorting facility that lets users sort data files on multiple keys. In addition, COBOL users can readily interface with IMS and DMS.

The COBOL 85 compiler is an implementation of the COBOL language as specified in the *American National Standard for Programming Language COBOL (ANSI X3.23-1985, ISO 1979-1985)*. The COBOL 74 compiler conforms to the specifications of the *American National Standard COBOL (ANSI X3.23-1974)*.

COBOL 85 is an enhanced and modified version of the previous COBOL standard *(ANSI X3.23-1974)* and has many additional features not available in COBOL 74, notably improved structured programming constructs.

Unisys extensions to the standards for COBOL include floating-point data items, format editing functions, and terminal support.

Pascal

Pascal uses structured programming techniques to improve program development and maintenance. It is designed for professional programmers, students, scientists, and engineers.

The Unisys OS/3 Pascal compiler conforms to the *American National Standard IEEE Standard Pascal (ANSI/IEEE X3.97-1983)*. Because it involves a small number of fundamental programming concepts, Pascal is suitable for teaching programming as a logical and systematic discipline, while providing a practical language for writing systems and application programs.

Unisys OS/3 Pascal also adds several extensions to the standard, including:

• String manipulation

• Relative I/O

• Initial values for variables

• Use of separately compiled functions and procedures

• Conditional compilations

• Multiple exit statements (EXIT, NEXT, RETURN)

C Language

The Unisys OS/3 C compiler is an implementation of the proposed ANSI standard for the C programming language, as described in the "Information Bulletin" published by CBEMA (the April 1985 draft of the X3-J11 Committee Standard).

The C language facilitates the application of structured programming techniques for general data processing problems. Its key features are its small object code size, fast execution speed, and flexibility in building applications.

Unisys extensions include:

- Extensions to the ANSI math library

- Additional signals

- Substitution of macro arguments

- Type conversion options

- Storing in string constants

- Special OS calling sequence

BASIC

BASIC is an interactive programming language that meets the requirements of both business and scientific programming. The BASIC language available on OS/3 complies with the *American National Standard Minimal BASIC (X3.60-1978)* and includes Dartmouth features and compatibility. It has a powerful set of commands that allows the novice to learn the language quickly, yet gives the experienced programmer an extensive list of features for various applications.

BASIC source statements entered directly at the screen are checked for syntax errors as they are entered. A message appears on the screen if a line is in error. Also, BASIC source programs can be compiled directly at the terminal and errors corrected immediately. During an interactive BASIC session, you can input, modify, execute, and save programs.

The OS/3 BASIC compiler performs arithmetic operations, data file processing, matrix generation and processing, and logical operations. You can use subroutines and string operations in a BASIC program.

52

FORTRAN 77 and FORTRAN IV

FORTRAN is a powerful, mathematically oriented programming language geared primarily toward scientific and business applications that require extensive mathematical computing.

FORTRAN language expresses an algorithm in a natural way. You do not need to be concerned with the particular system characteristics. FORTRAN can reference procedures written in another language by name and make them an implicit part of the program.

FORTRAN 77 conforms to the *American National Standard FORTRAN (X3.9-1978)*. 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. These systems feature code optimization, high-performance I/O, and extended functional capabilities.

The primary enhancement of FORTRAN 77 over the previous version is the addition of improved structured programming constructs.

RPG II

RPG II is a high-level language for producing reports and maintaining files in a business environment. The RPG II cycle provides the basic logical flow of a program. You specify the details of processing by using the input format, output format, calculations, and other specifications. You use formatted displays to enter source program statements. Auto report, which accepts simplified RPG II source statements and creates a complete RPG II program, is also available.

To define a report program for generation, you list the requirements on specifications forms. The information coded on these forms becomes the input to the RPG II compiler. The compiler generates an object module for input in the linkage editor. The subsequent output is a loadable program module. You can display the RPG II formats on the terminal screen and complete the proper entries. You can also enter unformatted RPG II source statements.

The following enhancements let you develop more sophisticated report programs within the existing framework of the language:

- Telecommunications interfaces, supplied through an RPG II specifications form, offer extensive communication capabilities to the RPG II user.

- Eight control stream user switches (indicators) set calculations, input files, output files, or specific output records.

- IMS action programs are written in RPG II.

- Terminals are accessible through programmed operations.

- OS/3 RPG II supports single-key and multikey MIRAM files.

- Error diagnostic messages can optionally be written by the RPG II compiler to a module accessed by the editor's error file processor.

- The currency sign (Japanese or other type) can replace the dollar sign in edit words on the control specifications form.

- The RPG II indicator table tests the processing of screen formats. Indicators that are set on when the terminal function keys are pressed are added to the RPG II language.

- Data structures allow multiple definitions of internal data, subdivision of data files, and grouping of fields, with support for packed and zoned decimal data.

- OS/3 RPG II handles multiple terminal and terminal error processing.

- The COPY statement includes RPG statements from a library.

- Structured programming operations

BAL

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

The assembler recognizes a set of directives used to direct operations. These directives 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 reduces the effort required to write patterns of coding, either repeated in one program or common among several programs. The macro facility allows a macro to be written so that the pattern of coding generated can vary widely, depending on the parameters supplied with the call. Macro definitions are specified in two formats: MACRO and PROC.

Output from the assembler run consists of a complete listing of symbolic coding, generated object coding, diagnostic messages, and a cross-reference listing. BAL produces a relocatable object module that can link to other modules before it is loaded for subsequent execution.

ESCORT

The ESCORT language is a high-level, interactive programming language that generates file processing, data handling programs, and report generation. It is easy to use, requires minimal training, yet has the power of more complex data handling languages. ESCORT language uses English verbs and clauses to express programming operations. Thus, even a programming novice can quickly begin producing useful programs.

Following is a typical ESCORT screen display:

```
                      CHANGE DATA STATEMENT
CHANGE DATA OF file-m (structure-m)  /FROM/file-i (structure-i)        \ \
                                    (      WS (structure-i)              )
                                     \     WS (structure-i,formname)    / /
FROM clause, structure-names and formname ARE OPTIONAL PARAMETERS
file-m IS NAME OF DATA FILE BEING CHANGED
file-i IS NAME OF FILE CONTAINING CHANGE CRITERIA
IF WORKSTATION IS USED AS CHANGE CRITERIA DEVICE, THEN
    KEYIN WS FOR file-i parameter
KEYIN PARAMETERS
        file-m--------      structure-m--------
        file-i--------      structure-i--------      formname--------
```

An ESCORT program creates, sorts, or merges files; adds, deletes, sorts, or replaces records within a file; updates selected records from a terminal; issues prompting messages to a terminal for input data; and validates the input data. Available computational operations include addition, subtraction, multiplication, division, and modulo, a division operation in which the result is the remainder only.

You conduct ESCORT programming sessions directly at the terminal. Jobs and programs initiate from the ESCORT call. The ESCORT compiler uses prompting messages for data entry. The amount of prompting depends on the programming expertise of the user.

A novice can conduct a session in interactive tutorial mode that creates an ESCORT program using a question-and-answer technique. HELP displays explain various choices offered by the question-and-answer display screens. In tutorial mode, the full ESCORT set of string relational operators is enabled. Each menu has a default option, and you can save programs without exiting the session.

The experienced user can use the interactive program mode to create ESCORT programs faster, with less prompting. Program mode is used to:

• Create programs by filling in the blanks on menu screens displayed by the ESCORT program

• Enter ESCORT statements directly onto a blank screen

Direct entry is the quickest way to create an ESCORT program. The compiler offers immediate syntax checking of ESCORT statements and sends messages when it detects errors. You can correct these errors immediately through the terminal.

LINC II

Unisys Logic and Information Network Compiler II (LINC II) is an advanced fourth-generation system generator. With LINC II, you define your information needs in business terms rather than in technical data processing terms.

LINC II uses three basic building blocks:

- The fundamental components of your organization

- The day-to-day business events that occur

- The different ways in which these are viewed by management and user departments (profiles)

Complementing this design philosophy is an implementation language that facilitates the high-level business definition of the information system. A sophisticated interactive development environment assists as the system is defined. LINC II automatically generates every aspect of the system, including the data base, transaction management, network management, and application programs. As a result, analysts can focus on the business problem to be solved rather than on technical implementation details.

LINC II provides the following features:

- Interactive development environment

- Communication between application systems

- Multithreading and synchronized recovery

- Sophisticated reporting capabilities

- Source-level portability across Unisys mainframes

- Efficient resource utilization

MAPPER 80

MAPPER 80 generates reports from a data base management system (DMS) data base. Data on the DMS can only be accessed by MAPPER 80 software. However, MAPPER 80 utilities can capture MAPPER 80 data from external files (spool or other OS/3 data files) and transfer report data to DMS files.

The MAPPER 80 system also uses screen format services. Some available screen characteristics are:

- Lowercase translation

- Original and overlay formats

- Special editing characters

- Nondisplayed fields

The MAPPER 80 system has an internal system of locks to prevent data inconsistencies from occurring when more than one user tries to update the same report at the same time. Two or more users can look at the same report simultaneously, but it can be updated by only one user at a time.

General Editor (EDT)

The OS/3 general editor, commonly known as EDT, is a user-oriented interactive program that lets you:

- Create and update library modules, data, and text files from a terminal

- Interactively create and update source programs, job control streams, and data files

- Copy, delete, and concatenate files

- Create and call procedures

EDT offers the following versatile features:

- Command language – EDT provides an easy-to-learn interactive command language.

- Procedure files – EDT lets you divide the EDT workspace file into a maximum of 10 subfiles: a main work file and 9 other procedure files. EDT procedure files are basically like the main work file; they consist of data and/or EDT commands. However, unlike the main work file, EDT procedure files can be executed. The commands or data entered in them can be executed against the main work file or any of the other defined procedure files.

- Subeditors – Two subeditors, the RPG editor and the COBOL 74 editor, provide syntax checking on source entry for the RPG and COBOL 74 languages.

Screen commands let you:

- Enter multiple commands and data

- Request help with EDT error messages

- View and update lines and the EDT environment parameters

- Display the EDT commands and their HELP screens

- Return to the EDT session from a screen command

The editor has file protection facilities that prevent a file from being inadvertently destroyed or altered, either by direct user action or through some system failure.

The general editor operates in two modes: line mode and screen mode. Each mode offers full editorial functions.
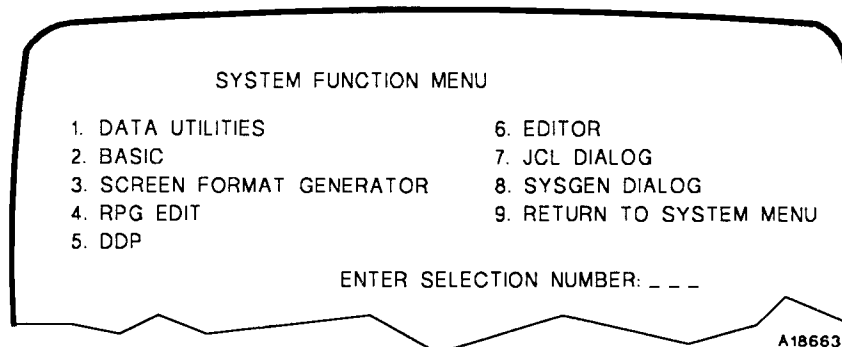
- Line mode – You create or edit material one line at a time.

- Screen mode – You can enter up to 14 lines at a time while using specially formatted screen displays to enter source programs in the COBOL, FORTRAN, and RPG II programming languages.

59

Menu Services

The menu services software supports menus to aid you in data processing activities. Menus are screen displays that list, by number, the options available to you, such as system programs, interactive commands, and other system activities. OS/3 menu services software consists of two components:

- Menu Generator – lets you create, modify, and display menus. You engage in a dialog to create the menu and the HELP screens for all or some of the items offered by the menu. The menus you create can execute programs and interactive services commands, call other menus, and return constant data to a program.

- Menu Processor – retrieves and displays menus on terminal screens, handles user input from the menu, and arranges for program or command execution. It also keeps track of multiple menu sequences. The processor displays the menus in the order specified and responds to user commands to display other menus. Menu processing is available to user jobs through JCL or interactive services.

A Unisys system menu is shown below. Menus list easily executed system programs and interactive services commands. You enter the appropriate selection number to execute the corresponding option.

```
                    SYSTEM FUNCTION MENU

        1. DATA UTILITIES               6. EDITOR
        2. BASIC                        7. JCL DIALOG
        3. SCREEN FORMAT GENERATOR      8. SYSGEN DIALOG
        4. RPG EDIT                     9. RETURN TO SYSTEM MENU
        5. DDP
                            ENTER SELECTION NUMBER: _ _ _
```
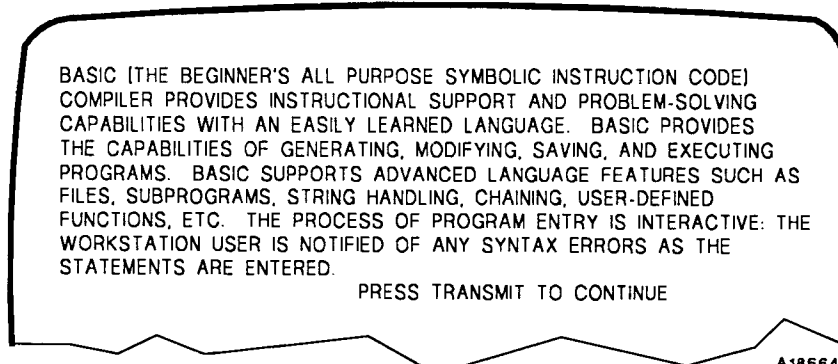A18663

Menus also have online HELP screens that provide supplemental information regarding a menu option. The illustration below shows a HELP screen explaining the programming language BASIC, one of the choices on the menu shown above.

```
   BASIC (THE BEGINNER'S ALL PURPOSE SYMBOLIC INSTRUCTION CODE)
   COMPILER PROVIDES INSTRUCTIONAL SUPPORT AND PROBLEM-SOLVING
   CAPABILITIES WITH AN EASILY LEARNED LANGUAGE. BASIC PROVIDES
   THE CAPABILITIES OF GENERATING, MODIFYING, SAVING, AND EXECUTING
   PROGRAMS. BASIC SUPPORTS ADVANCED LANGUAGE FEATURES SUCH AS
   FILES, SUBPROGRAMS, STRING HANDLING, CHAINING, USER-DEFINED
   FUNCTIONS, ETC. THE PROCESS OF PROGRAM ENTRY IS INTERACTIVE: THE
   WORKSTATION USER IS NOTIFIED OF ANY SYNTAX ERRORS AS THE
   STATEMENTS ARE ENTERED.
                        PRESS TRANSMIT TO CONTINUE
```
A18664I

Transaction Processing

For the System 80 model 50, OS/3 offers a new Transaction Platform System (TPS), which is an integrated online processing and development system. TPS fully supports the development and execution of IMS and TIP/30® action programs and provides the transaction interface to LINC II, OFIS,® and CAPIR, the IBM CICS Application Interface Routine. It contains all the software tools necessary for system maintenance and program development.

TPS provides a comprehensive, system-wide transaction interface to the OS/3 operating system, allowing all end users to access OS/3 in a consistent and efficient manner regardless of the components utilized.

For applications development programmers, it is designed specifically to improve online system performance and reduce online development time by significantly increasing programmer productivity.

TPS:

- Provides efficient, multithread control of your transaction processing environment

- Provides a hardware-independent interface to your terminal

- Provides efficient, secure access to your data files, libraries, and various data base management systems

- Offers a wide variety of features designed to help end users and program developers, including security, printing, program development and test, and system maintenance

System Security

TPS provides full system security through user-ids, passwords, assigned security levels, and profile-like applications groups to restrict and control user access to programs and files.

Security is controlled by the Catalog, a file that the system administrator maintains via the online Catalog Manager program. The Catalog contains entries for all authorized users, for all available files, and for all online programs.

Each user has a Catalog entry that specifies the user-id, password, security level, and the application groups that the user can access. Each online program and every online file has an entry that specifies the application group to which it belongs and the security level required to access it.

A user logging on has access only to those facilities available by application group and security level.

Printing Service

TPS provides complete printing services. In addition to sending printer output to the standard OS/3 spool facility for printing, you can redirect data to an online terminal printer, to a Unisys U5000 system, to an OFIS document, or to an MS-DOS® file on a personal computer.

PC-to-Host Data Transfer (PCXFER)

Through PCXFER, TPS allows you to design applications that fully integrate PC-based applications and data into your online environment. These applications can be incorporated into existing online and batch systems without the need to redesign the systems.

With PCXFER, the PC becomes an additional input source or output destination. Records can be transferred in character or hexadecimal format so that binary data records or even program records can be transferred with no restriction on record length.

A user-developed host file maintenance program can update host files with data read directly from an MS-DOS file. The online file maintenance program has access to the full file journaling and recorder features so that the integrity of the host files is maintained.

Text Editor

TPS offers a powerful and versatile full-screen text editor with a simplified user interface that makes it easy to learn and use. With the full-screen editor, a user can alter displayed text directly or through a complement of online commands.

Integrated Screen Definition, Testing, and Maintenance

TPS includes a facility to define, compose, test, and archive screen formats. You can quickly develop and fine-tune your screen formats to maximize end-user utilization.

Utilities

TPS provides a complete catalog of utility routines to perform a variety of specialized functions. Just a few are:

- CAT – The system administrator uses the online Catalog Management program to maintain entries in the Catalog for all users, programs, and files active under TPS.

- CCA – This utility makes ICAM statistical information available to the online user via a series of screen displays or as a printed report. The types of statistics reported include buffer pool penetration, line and terminal traffic, and LOCAP usage for a global network.

- JCL – Users can quickly create and submit one-time-only jobs via the JCL job submission utility. JCL statements are entered at the terminal and submitted directly to the run processor, eliminating the need to create a JCL library element.

- SPL – This utility allows spool file queues to be examined and subfiles to be listed at the terminal. The listing can be manipulated by the user with subfiles printed at the terminal, released to the system output writer, or deleted. In addition, reports created by batch jobs can be easily printed at terminals having printers.

Information Management System (IMS)

IMS is a transaction-oriented processing system. Each time you enter an input message, such as a transaction code, action programs process the input message, access data files, and provide an output message response to the terminal. IMS operates in a multithread environment, where IMS actions from several terminals are processed concurrently. The action program that processes the input message can be written by the user or provided by IMS.

Action Definition

The entire process, from entering an input message to completion of the programming function, is called an action. An action is the basic unit of work in IMS. One or more actions comprise a transaction. When one action accomplishes a task, it is a simple transaction. When you need a sequence of two or more actions to complete a task, it is a dialog transaction. In a dialog transaction, the first action program tells IMS the name of the action program that processes the second message.
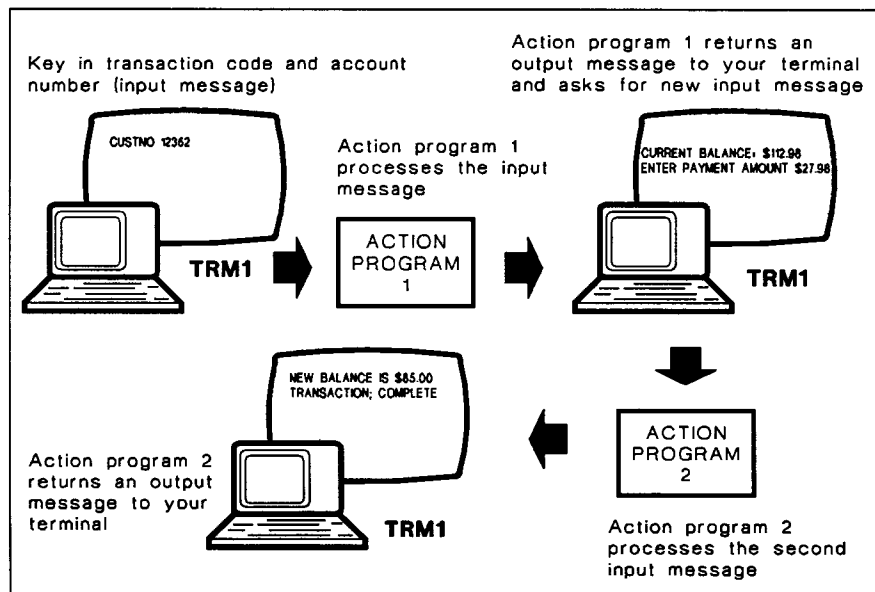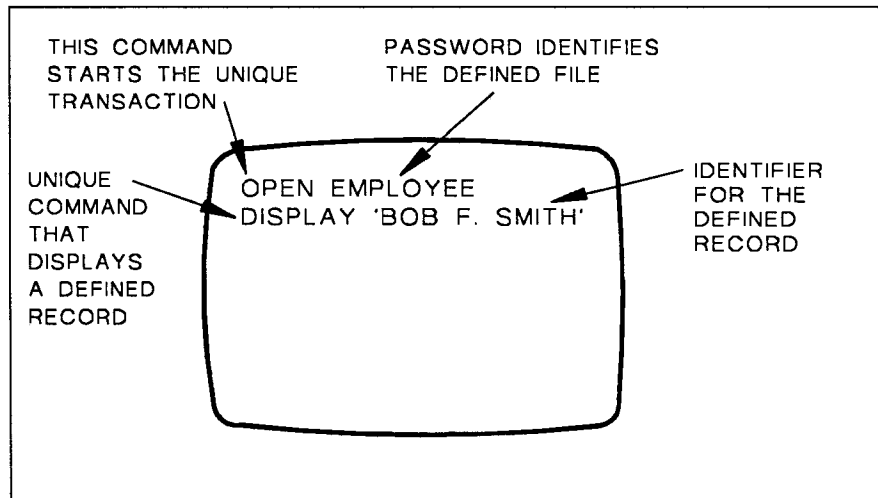

A18627

IMS File Processing Package – UNIQUE

IMS supplies a set of action programs called the uniform inquiry update element (UNIQUE). UNIQUE is a general-purpose language that enables extensive file processing capabilities. Users initiate file activity through UNIQUE commands. In addition to UNIQUE, you can write IMS action programs in BAL, COBOL, or RPG II. This gives you more flexibility in designing IMS programs.
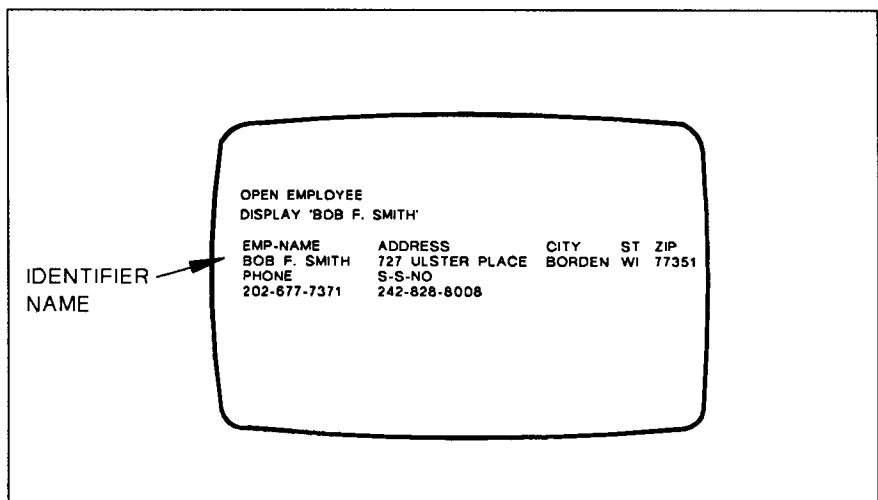
UNIQUE lets you access, retrieve, and update files. It also lets you calculate statistics on those files. To use UNIQUE, first create defined files from elements of existing files by using an IMS utility called the data definition processor. A password is defined in the data definition that controls access to a defined file. The password is the same as the defined file name, and all configured terminals can use it.

An Input Screen to be Processed by UNIQUE

THIS COMMAND STARTS THE UNIQUE TRANSACTION

PASSWORD IDENTIFIES THE DEFINED FILE

UNIQUE COMMAND THAT DISPLAYS A DEFINED RECORD

OPEN EMPLOYEE
DISPLAY 'BOB F. SMITH'

IDENTIFIER FOR THE DEFINED RECORD

A18661

Output Screen Generated by UNIQUE

OPEN EMPLOYEE
DISPLAY 'BOB F. SMITH'

EMP-NAME	ADDRESS	CITY	ST	ZIP
BOB F. SMITH	727 ULSTER PLACE	BORDEN	WI	77351
PHONE	S-S-NO			
202-677-7371	242-828-8008			

IDENTIFIER NAME

A18662

Data File Types

IMS can access any type of user disk data file, including:

- MIRAM files – These are used for sequential, direct, or indexed file processing.

- Files generated by DMS – These can also interface with COBOL action programs.

- User-written data definition files – These let you logically redefine a file without actually altering the physical file structure. Redefined files can then be used by IMS action programs.

User-Written Action Programs

IMS also supports user-written action programs in BAL, COBOL, and RPG II. Action programs operate under IMS control and access files through 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.

Both UNIQUE and user-written action programs can access defined files that derive from DMS data base records.

IMS Interface Capabilities

- IMS/DMS – IMS action programs can concurrently access DMS data bases through data manipulation statements embedded in COBOL action programs. DMS modules support 48 job slots.

- IMS/ICAM – ICAM presentation services is a group of modules that build a chain of linked buffers for holding messages at the ICAM terminal. This facility makes efficient use of ICAM resources when a terminal is used with a data mode session to IMS.

- IMS/CDM – IMS, in consolidated data management (CDM) mode, lets action programs access records in a DMS data base. IMS loads fixed- and variable-length MIRAM files into a common storage area during system start-up. This eliminates disk reads and writes whenever action programs access records in these files.

- IMS/DDP – In a multithread environment, IMS/DDP gives you flexibility in designing transaction routing to the directory, to action programs, and to the operator.

Advanced IMS Capabilities

- Interactive commands - These assist with the administrative and operational aspects of IMS. You can control processing by designating a master terminal to control the processing activity of the entire network through a master terminal command set.

- Language section - You can define a command set for use with UNIQUE. Non-English commands, abbreviated commands, or business-oriented terms are examples of the types of command sets you can define.

- Multikey support - IMS supports multikey, duplicate key, and key change features to MIRAM files during operation in the CDM environment. You can retrieve records from an indexed MIRAM file by using any key of reference. Deletion of records through CDM is also provided for MIRAM files containing single-key or multikey records.

- Statistical reporting - IMS provides statistics, such as the total number of file accesses on a transaction basis, terminal statistics on an IMS session basis, and number of I/O messages per transaction.

- Multiple job environment - IMS can run in a multiple job environment because IMS executes as a program under control of OS/3.

- Dynamic storage allocation - Online IMS provides dynamic main storage allocation at action scheduling time. This lets you modify the size of an action program without reconfiguring it.

- Reliability - IMS provides extensive features to prevent accidental alteration or destruction of data files. Record locking and recovery facilities are also provided.

Data Base Management System (DMS)

DMS accesses and controls data bases stored on direct access devices. There are two main areas of responsibility in DMS:

- Data base administration:

    - Defines data structures (logical and physical) and chooses access methods

    - Monitors system performance

    - Reorganizes the data base as needed to improve system performance

    - Operates DMS utilities to maintain a stable data base

    - Performs start-up/shutdown of the DMS job

- Data base application

    The applications programmer is responsible for writing COBOL/data manipulation language programs using a defined data base structure to interface with the data base.

DMS Components

The various DMS components are:

- Four language processors:

    - Schema processor

    - Subschema processor

    - Device media control language processor (DMCLP)

    - Data manipulation language preprocessor

- The DBMS run-time software component

- System support modules that include utility processors, library files, and procedure files

Data Base Languages

There are two data base languages for data dictionary access:

- Data description language (DDL) – develops a schema to describe the logical structure of the data base and develops subschemas to describe data base subsets that relate to specific applications. A DMS data dictionary can contain more than one schema and its related network. DDL is the source input for the schema and subschema processors. The device media control language (DMCL) defines physical characteristics of a user data base and data dictionary. DMCL is the source input for the DMCLP.

- Data manipulation language (DML) – statements are included within conventional COBOL Procedure Division statements for batch processing.

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 written in a high-level language (data description language), which is comparable to the data declaration language of COBOL.

User's View of a DMS Data Base

A18626

69
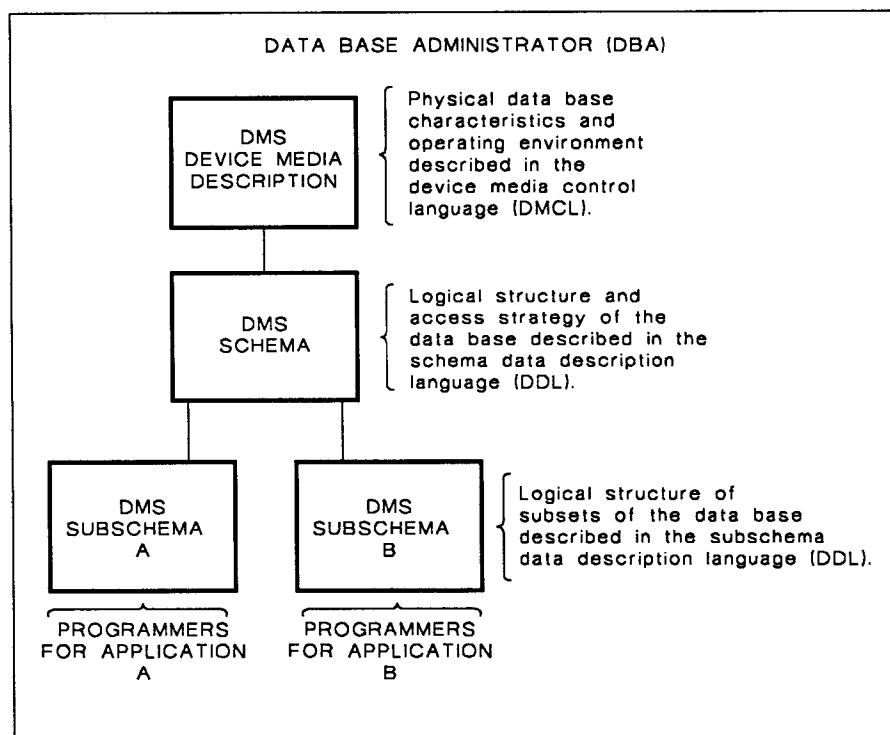
Run-Time Software

The DMS data base management system (DBMS) is the main set of routines that actually performs the functions requested by a run-unit (either a DMS processor or a user application program). It retrieves and updates data base records conforming to the user-defined data bases and creates a journal file for use in offline manual recovery.

The DBMS is a single reentrant run-time component running as a privileged job under OS/3.

System Support Functions

The system support functions create, establish, and maintain a data base. There are three categories of system support functions:

- Configuration function – creates an installation version of DMS that fulfills specific processing requirements.

- System generation function – creates and initializes a data dictionary and a data base description.

- System maintenance function – is a recovery-related utility processor. The DMS utility processor language lets you specify the utility functions to be performed and the portion of the data base they operate on. DMS recovery utility functions include journaling, security dumping, automatic and manual recovery methods, and DBA processors that print reports and statistics.

Advanced DMS Capabilities

- You can exclusively lock a record as soon as it is retrieved to prevent record update conflicts.

- The action program can test the availability of a data base record before retrieving it. If the record is not available, a minor end is returned rather than a fatal thread cancellation.

- You can sequentially process the same data base using two or more DMCLs.

- DBPAG prints a compromised data base page instead of terminating with an error message. This allows analysis of hardware or software problems suspected of causing the compromises.

- DMS allows 32 index keys per record.

- You can specify a unique journal file for each data base. This allows concurrent operation of journaling and use of multiple data bases that have different page sizes.

- Disk journal files can extend over more than one DBMS session.

- You can obtain a system dump when an unexpected data management error occurs.

- The DBMS log records the activity of the DBMS.

- DBMON provides a *window* into the DBMS run-time component during execution through 12 monitor screens. Data on the monitor screens allows you to analyze and improve DBMS performance.

Distributed Data Processing (DDP)

In large and complex networks, it is critical that your data processing is efficiently distributed and managed throughout the systems.

OS/3 DDP lets you:

- Distribute files between remote data processing systems

- Engage those systems to cooperatively process jobs

- Access files and other programs on remote systems

- Perform transaction processing between remote systems

DDP functions in OS/3 are divided into three software packages, each offering different elements. You can tailor distributed processing software to include only the particular features your installation requires. The software packages are:

- DDP transfer facility

- DDP file access

- IMS-DDP transaction facility

DDP Transfer Facility

The DDP transfer facility lets you view each system in the DDP network as an available resource for scheduling and executing your work. Using simple commands, you can initiate job distribution and file transfer within the system. You can do this without concern for the requirements of the hardware and software of each system and without concern for the communications protocols needed to initiate and monitor the distribution of a job.

The DDP transfer facility provides the following capabilities:

- Site-to-site data file and program library transfers

- Remote job initiation and control

- Operator console control over remote sites and routing of messages to remote consoles

- Routing print output files to different host computer systems

- Routing print output files to remote auxiliary printers

The DDP transfer facility consists of two pieces of software, the job transfer manager and the file transfer manager.

Job Transfer Manager	File Transfer Manager
Provides commands to:	Permits you to:
- Submit jobs to the DDP network	- Generate a file directory to catalog the characteristics of files
- Monitor execution of submitted jobs	- Transfer copies of data files and program libraries
- Cancel a submitted job	
- Communicate with a remote operator's console	- Delete a file from the file directory
- Issue instructions to a remote operating system	- Obtain a listing of a file's characteristics from the file directory
- Respond to messages issued by a job executing on a remote system	

Files Transmitted between Systems

- Data files – Files transmitted from OS/3 are MIRAM files. Files transmitted to OS/3 are created in MIRAM format.

- Program libraries – Any directly accessible module in a program library or an entire library can be transferred between systems.

DDP File Access

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

DDP file access provides the following capabilities:

- User programs can access disk files residing on a remote OS/3 host through job control statements and use them as if they resided on the local system.

- Programs running on remote hosts in a multihost environment can access each other as files.

IMS-DDP Transaction Facility

The IMS-DDP transaction facility lets you perform transaction processing on any local or remote files within a DDP network. The transactions are processed either by IMS user action programs written in BAL, COBOL, and RPG II or by the UNIQUE inquiry language.

The IMS-DDP transaction facility also lets you perform inquiries and updates on files residing at remote terminals anywhere within the DDP network.

IMS-DDP software requires a configured communications network to physically link the DDP processors at each site. The processors interpret all DDP-related commands and perform the requested functions. The processors are designed to operate in an interactive environment and respond to commands issued from a terminal.

There are three ways in which IMS can route a transaction to a remote system:

- Directory routing – A terminal operator at the primary IMS enters a transaction code that identifies a transaction at a particular remote system. The primary IMS routes the message to the secondary IMS, where UNIQUE or action programs process the transaction. Once the transaction begins, a communications link is established between the terminal operator and the remote system. This allows a dialog transaction consisting of multiple input and output messages.

- Operator routing – Operator routing is similar to directory routing. In this case, a special character, rather than a transaction code, is associated with the remote system. When the terminal operator enters a transaction code prefixed by the special character, IMS routes the message to the secondary IMS, where the transaction is processed in the same manner as in directory routing.

- Action program routing – The terminal operator enters a transaction code that initiates a transaction at the primary IMS. A COBOL or BAL action program at the primary IMS issues an ACTIVATE function call to IMS, identifies the remote system in its output message header, and generates a message containing a transaction code. IMS routes this message to the remote IMS, where action programs process the transaction and return a message to the originating action program or its successor. The action program at the primary IMS can then return a message to the terminal operator or can issue another ACTIVATE call to initiate another remote transaction.

PC-to-OS/3 Connectivity

The OS/3 connectivity products, such as the three PC-to-OS/3 utilities, allow OS/3 to act as a regional server providing access to hundreds of applications. The following products let the Unisys personal computer (PC) user access System 80 data bases and software applications for daily business use:

- File transfer (PCTRAN)

- On-line transfer (OLTRAN)

- PC on-line disk

PCTRAN

PCTRAN permits transfer of data files, library modules, and source program files between a disk or diskette on a PC and a disk or tape on System 80.

When you transfer OS/3 object or load modules to a PC, the data is translated by PCTRAN and maintained on the PC as character data. When the module is retransmitted to the host system, PCTRAN translates it into OS/3 object code format.

You can select character or PC hexadecimal mode data transmission. When hexadecimal mode is used, the PC terminal emulation software and PCTRAN translate the data during file transfer. This option allows PC object code and binary data to be transferred to and from the OS/3 host system.

OLTRAN

OLTRAN is an OS/3 program product that offers you an easy way to transfer files between a PC and a System 80 mainframe. You can transfer OS/3 MIRAM files and SAT file elements to and from PC ASCII data files. You can transfer ASCII files from a PC to the OS/3 mainframe with or without translation to EBCDIC code. A complete error-checking protocol is provided to ensure error-free data transfer.
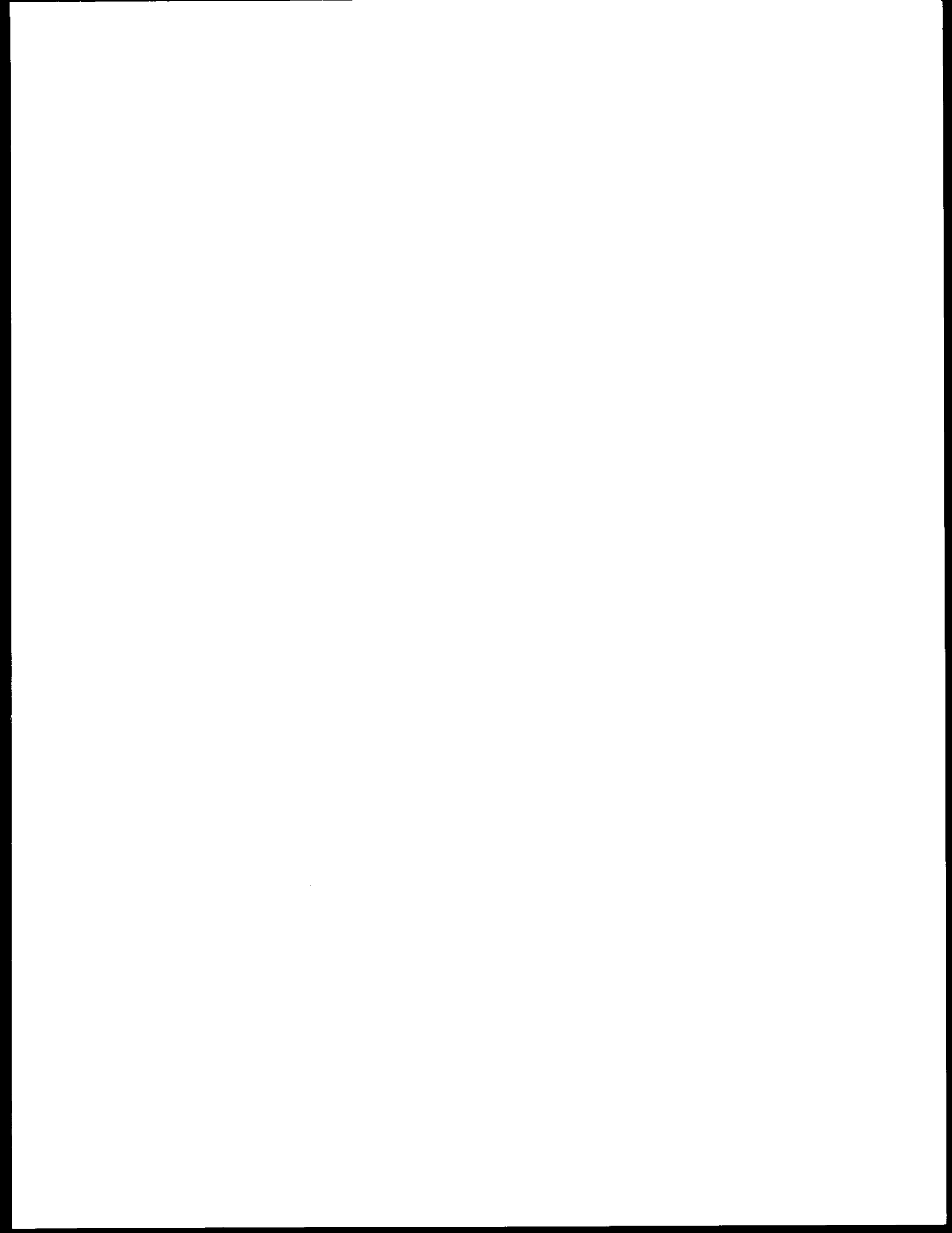
PC On-line Disk

PC on-line disk lets you allocate and format OS/3 files as Unisys PC diskettes or hard disks. Multiple PCs can use the same disk files, allowing users to share data, keep backup copies of data diskettes on an OS/3 host, and assign an area of random access memory (RAM) for use as a diskette.

This utility supports most MS-DOS PC operating commands and MS-DOS compatible software.

File Placement Analyzer (FIPLAN)

FIPLAN can improve system performance by projecting optimum file allocation. FIPLAN can analyze and project allocations for up to 1000 files at a time. The allocation of files recommended by FIPLAN can result in the relocation of user files across volume boundaries to achieve optimum load balancing. FIPLAN does not analyze dynamic file extensions and new files created during the monitor session.

UNISYS