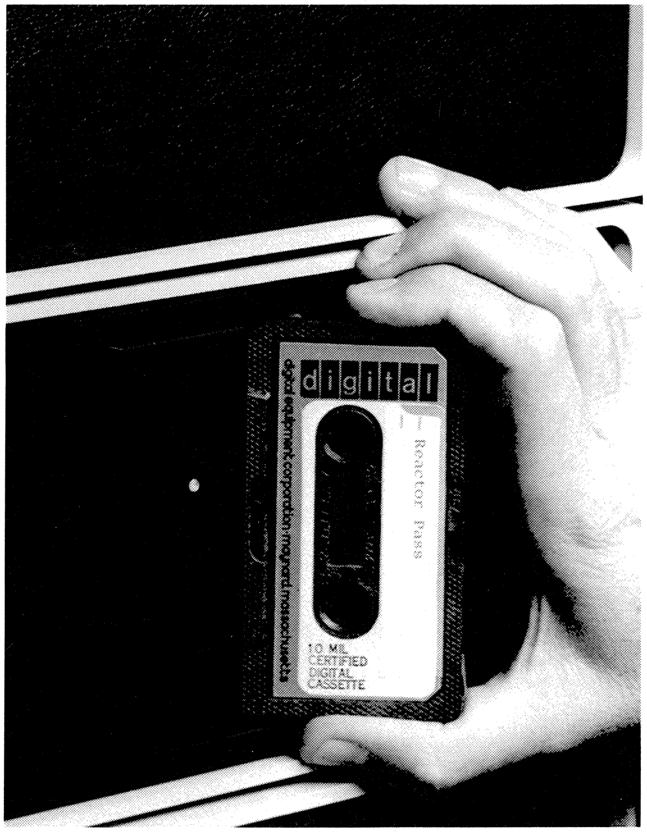
PDP-11
CAPS-11 (programutvecklingssystem.)

Cassett programming system

digital

JULY 1976

CAPS-11 is a small, cassette-based, program development single-user system with true operating system characteristics. CAPS-11 includes a monitor, an editor, assembler, linker, on-line debugging module, and file interchange utility. As an option, CAPS-11 supports a BASIC language processor, BASIC/CAPS.



FEATURES

- Small, magnetic cassette tape-based operating system
- Upward compatible with PTS-11 paper tape software
- Users can develop programs which use as many or as few of the monitor facilities as they wish
- Users can create and load stand-alone programs to use all but 400 words of available memory

MONITOR FUNCTIONS

The monitor includes a keyboard listener, command string interpreter, cassette loader and resident monitor. The keyboard listener allows the operator to request the monitor to load and start user programs, run system programs, set the date, etc. The command string interpreter decodes the operator's commands, including input and output file specifications, and passes the information to system or user programs. The cassette loader brings binary format programs into memory from cassette and starts their execution. The resident monitor performs all cassette, console and line printer input and output operations. All devices are interrupt-driven.

I/O SERVICES

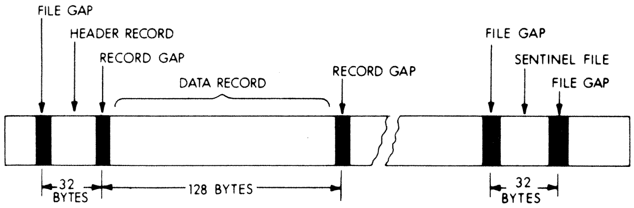
CAPS-11 requires an operator's console and a dual cassette drive and control. It supports an optional line printer and high-speed paper tape reader/punch. The paper tape reader/punch is used to transfer files from cassette to the punch or from the reader to cassette only. Under the BASIC/CAPS language, the user can access graphic display and laboratory data peripherals.

The CAPS-11 Monitor is compatible with the IOX executive of the PTS-11 paper tape software, but uses magnetic tape cassettes for program and data storage:

FILE SYSTEM

A cassette consists of a sequence of one or more files, separated from each other by a single file gap. The first file on the cassette must be preceded by a file gap; the last file must be followed by a file gap and a sentinel file or by clear trailer. The figure below illustrates the cassette tape format.

Each file consists of a sequence of a header record plus zero or more data records separated from each other by record gaps. The first record of a file is called the file header record, or file label. The header record is 32 (decimal) bytes long.



CASSETTE TAPE FORMAT

USER INTERFACE

The user communicates with CAPS-11 by typing in commands from the console. These commands request monitor services such as RUN, LOAD, or START executing a program, or DIRECTORY, which asks the monitor to list the directory of a cassette on the operator's console.

The user may also request monitor services from within a program.

The command string interpreter is used by all system programs except EDIT and ODT and can be used by any user program which is loaded and started by the RUN command. When the user runs a program, it responds by printing an asterisk (*) at the left margin of the console terminal page. The user responds by entering all device and file I/O information needed by the program. The command string interpreter then constructs a table which contains the information needed by the program.

SOFTWARE ENVIRONMENT

BASIC Language Processor

As an option, CAPS-11 supports a BASIC language processor which is a single-user subset of the RT-11 MU BASIC language processor.

BASIC/CAPS provides sequential file capabilities and allows the user to save and retrieve files from cassettes. BASIC/CAPS allows user-defined functions, user-written assembly language routines, and chaining between BASIC programs with data passed in cassette files and/or in memory.

For systems with more than 8K of memory, support is also provided for the Laboratory Peripheral System and the GT40 display processor. Support is provided in the form of subroutines that may be linked with BASIC.

Assembler

The CAPS-11 Assembler is a modified version of PAL-11S, the PTS-11 assembler. Its features are:

- error listing on the console terminal
- double-buffered and concurrent I/O
- alphabetized, formatted symbol table listing
- relocatable object modules
- global symbols for linking between object modules
- conditional assembly directives
- program sectioning directives

Utilities

The CAPS-11 operating system utilities include an interactive editor, linker, on-line debugger and file interchange utility. These programs are loaded and executed by issuing the Monitor's RUN command when the Monitor is loaded into memory and the system cassette is mounted.

HARDWARE ENVIRONMENT

Minimum Hardware

- PDP-11 Central processor (except the PDP-11/03)
- 8K words of memory
- Console terminal
- Dual cassette tape system

Optional Hardware

- Additional memory up to 28K
- Paper tape reader/punch
- Line printer

PDP-11
RT-11

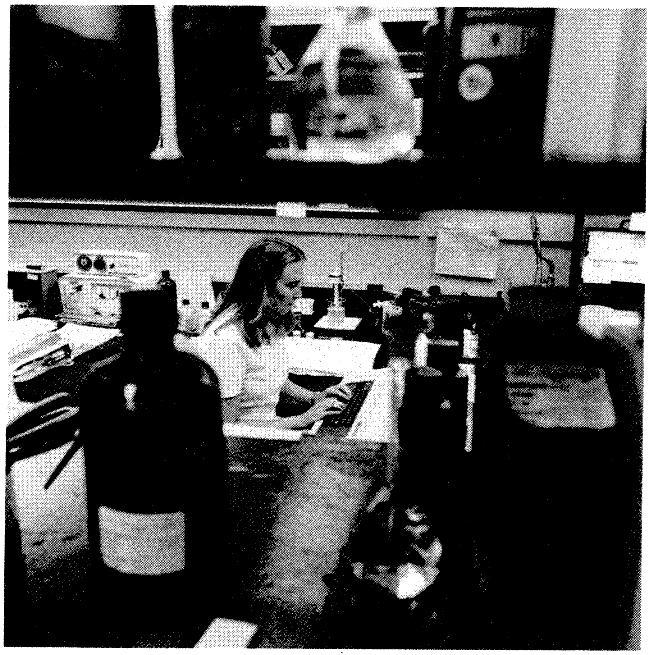
Real tids operativsystem

RT-11

digital

JULY 1976

RT-11 is a small, single-user disk operating system designed for interactive program development and/or on-line applications on the PDP-11. RT-11 supports both single job (S/J) and foreground/background (F/B) modes of processing. In addition to a variety of system and program development utilities, RT-11 offers optional support of a number of high-level language processors, including FORTRAN IV, FOCAL, and BASIC.



FEATURES

- Ease-of-use
- Contiguous file structure
- Configuration independence
- Flexible real-time I/O
- Low system overhead
- Ease of expansion
- Industry compatible magnetic tape
- Batch processing

MONITOR FUNCTIONS

The RT-11 operating system is delivered with both the single job (S/J) monitor and the foreground/background (F/B) monitor. The F/B monitor allows two programs to operate: a foreground program and a background program. The real-time function is accomplished in the foreground, which has priority on system resources. Functions which do not have critical response time requirements, e.g., program development, are accomplished in the background, which operates whenever the foreground is not busy.

I/O SERVICES

RT-11 has been designed to satisfy a wide variety of input/output requirements by providing three modes of I/O operation:

- Synchronous I/O–processing is suspended until the completion of the I/O event.
- Asynchronous I/O–an I/O event is started, and processing continues until a user-defined point is reached. Processing is then suspended until the I/O event is completed.
- Event-driven I/O–an I/O event is started, and processing continues until the I/O event completes. Processing is then interrupted to service the completed I/O event.

FILE SYSTEM

To facilitate throughput and minimize system overhead, the file management system of RT-11 is centered about a contiguous file structure. This system requires the minimum number of accesses to locate a given item of information. In addition, the monitor processing routines are small and compact.

USER INTERFACE

Keyboard commands and special function keys allow the operator to communicate with the RT-11 Monitor to allocate system resources, manipulate memory images, start programs, and use foreground/background services.

SOFTWARE ENVIRONMENT

BATCH

RT-11 BATCH is a complete job control subsystem which provides batch-mode processing of user jobs. In the F/B environment, BATCH processes job streams in the background partition, allowing real-time or other user jobs to run in the foreground. The BATCH run-time support package (which is resident only when BATCH is running) requires only 1K words of memory. RT-11 BATCH may be used in either S/J monitor configurations of 12K or more words of memory, or in any F/B configuration.

BASIC/RT-11

BASIC/RT-11 is an exceptionally fast incremental compiler which takes advantage of the familiar high-level language in order to afford convenient access to RT-11's file structure.

MU BASIC/RT-11

The multi-user version of BASIC/RT-11 supports up to 4 users in 16K and up to 8 users in 24K words of memory running under the S/J monitor. The language features of the single-user version of BASIC/RT-11 are available to the individual users of MU BASIC/ RT-11, including CHAINing, overlays, sequential and direct-access file support, and assembly language capabilities through the CALL subroutine feature.

FOCAL/RT-11

FOCAL is an easy-to-learn interactive language. The RT-11 implementation is as an interpreter, which provides both stored program and immediate mode operations. FOCAL uses the same floating point package as does FORTRAN/RT-11, so all arithmetic options are supported.

Any peripheral supported under RT-11 is available to the FOCAL user. The LIBRARY command allows the user to access any RT-11 file-structured device.

Other features include scheduling up to eight asynchronous tasks from the clock; processing interrupts in the FOCAL language; user-controlled error processing, and the facility for one or more user-written assembly language functions.

FORTRAN/RT-11

FORTRAN/RT-11 is a compiler and run-time system which fully uses the features of the RT-11 Operating System. The FORTRAN/RT-11 language is an extension of the FORTRAN IV language based on the ANSI X3.9 1966 Standard.

Scientific Subroutine Package

The Scientific Subroutine Package is a collection of FORTRAN IV subroutines which provide the user with a large cross-section of those mathematical and statistical routines commonly required in scientific programming. They are all I/O free and are provided in source form. The source listings give a brief outline of the method used, as well as a bibliographical reference in the more complex routines.

Lab Application-11 Library

The Lab Application-11 Library is an integrated set of program modules designed to accomplish standard signal processing functions, particularly those found in laboratories.

SPARTA is a complete program supplied with the Lab Application-11 Library that uses these functions to provide the scientist with a general-purpose data acquisition and manipulation tool. Although not required, SPARTA can use a graphics display processor to display collected analog data on the screen and allow the scientist to examine, store and manipulate that data.

GAMMA-11 F/B

GAMMA-11 F/B is a hardware/software system specifically designed for nuclear medicine and the needs of a nuclear medicine department. Used in conjunction with a gamma camera, it is designed to acquire, store, display, and manipulate images from the gamma camera in order to supply meaningful clinical information for more detailed diagnostic reports.

Graphics Support

For systems with a VT11 or VS60 Graphics Display Processor, both BASIC/RT-11 and FORTRAN/RT-11 utilize the system's Display File Handler to provide a complete set of graphic routines, allowing the full utilization of the display processor's hardware features—features such as vectors, alphanumerics, points, multi-intensities, and blinks. Additional commands perform tasks such as creating and tagging subpictures (graphic subroutines), and displaying figures and arrays.

HARDWARE ENVIRONMENT

The RT-11 operating system runs on any PDP-11 processor. A console terminal and at least 8K words are required for single-job operation. If the BATCH facility is desired, at least 12K words are required. If foreground/ background operation is desired, at least 16K words of memory and a line frequency clock are required. In addition to the processor, memory and console terminal, a random access mass-storage device is required for the system device, and a second I/O peripheral device is required for software distribution. For example, a floppy disk system with two disk drives.

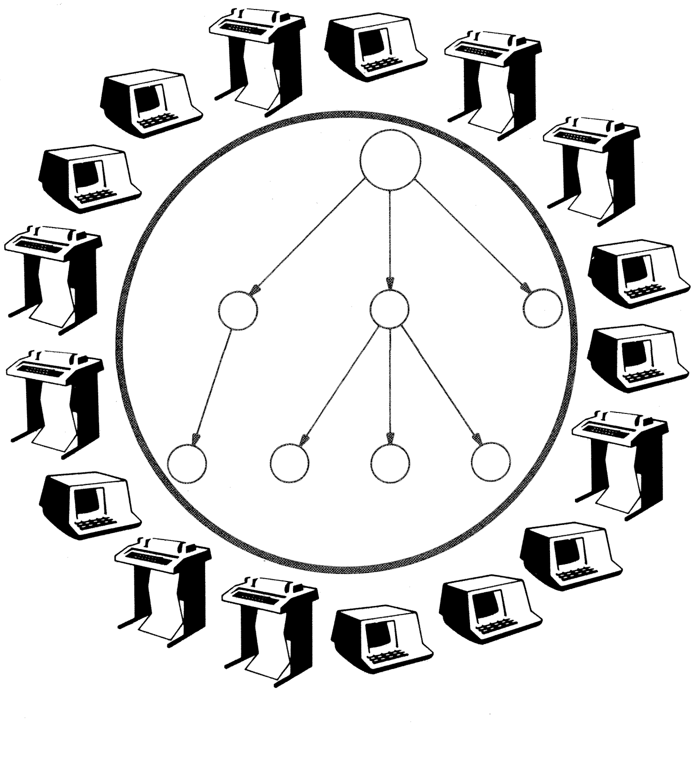
RT-11 supports a wide variety of optional hardware, including:

- Hardware bootstrap loader
- Additional memory to a system total of 28K words.
- Line printer
- Card reader
- Paper tape reader/punch
- Additional disk drives
- Additional dual DECtape drives
- Industry-compatible magtape
- Laboratory Peripheral Systems (LPS-11) supported by BASIC, FORTRAN, and LA-11
- Analog real time system (AR11) supported by FORTRAN or LA-11
- Graphics display system (VT11) with graphics capability using BASIC, FORTRAN or LA-11
- Electrostatic printer/plotter (LV11) with plotting supported by BASIC and FORTRAN.
- Digital I/O option (DR11-K) supported by FORTRAN.

PDP-11
MUMPS-11

JULY 1976

MUMPS-11 is an interactive multi-user data base management operating system. The high level MUMPS language supports string manipulation and a hierarchical file structure. The system is optimized for data base management functions, including the random retrieval of string oriented data from large data files. Timesharing is accomplished by a memory partition system to minimize turnaround time between jobs.



FEATURES

Data management

- hierarchical file structure
- string manipulation
- random retrieval of string data
- storage allocated only for actual data

High level language

- string oriented
- easy to use
- complete set of arithmetic and logic operations

Multiprogramming

DESCRIPTION

The unique feature of MUMPS-11 is its file management operation. MUMPS-11 files are structured in a hierarchical or tree form. The trees may have any number of levels and may have virtually unlimited branches at each level. The trees are accessed via a symbolic notation of multidimensional arrays. Only elements for which data is explicitly defined exist, and only the space required for the specific element is allocated.

The MUMPS language is a high level English-like language. It consists of relatively few but powerful commands and functions, and contains extensive call and overlay facilities. The language is heavily string-oriented with operators and functions for such string operations as pattern verification, collating, extracting, searching and concatenating. In addition it contains a complete set of arithmetic and Boolean operators.

The Operating System

The operating system contains all software necessary to operate MUMPS-11 in the hardware environment of a PDP-11. It is highly modular and resides permanently in memory. The system uses between 22K and 48K bytes of memory, depending on the hardware configuration and system software options selected during system generation. The memory remaining after the system is loaded is subdivided into user partitions.

Data Base Structure

The data management features of the system allow local data used by a program to be referenced symbolically. Storage space for this data is allocated by the system as needed. Local data is that set of variables established within the domain of a particular partition and is defined only for programs within that partition. This form of storage is used for scratch or transient data.

These local data arrays are treated as if they are intended to be sparse. That is, only subscripts for which data are defined are allocated space. Other implied subscripts of the array are not allocated space. A symbolic variable used in a program can be given either a numeric or a variable length string value. When it has the string value, only that space actually required to store the string is allocated.

This local data storage philosophy is extended to the management of data on the random access disk system. All elements stored in data files are referenced symbolically. The file name is similar to that of a symbolic local variable name in a program. Records in the file are treated as array elements and are referenced by subscripts. Subrecords (or fields) are referenced by appending additional subscripts. Files on disk thus comprise an external system of arrays, which provide a common data base available to all programs within a given user class. The arrays which make up this external system are called global arrays. Each global array is identified by a unique name.

The structure of global arrays is hierarchical. Any element within an array tree can contain a numeric or string data value and (or) be a pointer to a lower subscripting level in the tree. Data can be stored at any level. There are no constraints on the dimension or the size of any array. In addition, the number of subscripts in an array is dynamic, so that its content and structure can change during usage.

Additional features include:

- variable-size data elements and logical records
- random access of data using multiple keys
- inter-task memory-to-memory communication facilities
- choice of ANSI standard and EBCDIC magnetic tape labeling and capability to use several standard tape formats

File Access and Security

Each user of the MUMPS-11 system gains access to the system's programs using a special log-in sequence which involves one or two access codes (depending on the privileges of the user). These codes, provided by the system manager, are the User Class Identifier code or UCI, and the Programmer Access Code or PAC.

The MUMPS-11 system can have up to 16 UCIs (classes of user). The UCI code must be entered by everyone who wishes to use the system. It allows access to the programs and globals listed in the program and global directories for that UCI. A user who is permitted simply to run programs needs to know only the UCI and the name of the programs for that UCI.

Users who are allowed to create or modify programs and global files must know the system's PAC. This code permits system operation in direct mode. In direct mode, a programmer can issue MUMPS commands at the keyboard, as well as create,modify and delete global data and programs associated with the UCI under which the user logged-in.

The system manager may designate any terminal as "tied" to a given program. During operation, simply striking one key on the tied terminal will cause it to begin executing the assigned program. It is then impossible for the user to go through the log-in procedure.

A tied terminal can never be used for program development, examining, modifying filed programs, or affecting the data base in any way other than the application's normal program operation. But the tied program can use the CALL and OVERLAY commands to branch to other programs. With the tied terminal facility, the tied program can be an executive-type "root" program that identifies the user via any sort of password scheme that the system implementor deems appropriate, then branches to the necessary programs via the OVERLAY or CALL commands. Such programs are designed to allow only authorized access to the files. And since the user is unable to modify the programs from a tied terminal, all operation can be under the control of the system manager.

The Mumps Language

The basic orientation of MUMPS is procedural, much like FORTRAN or COBOL. However, because of the interactive nature of the system, programs are written and fully debugged in a fraction of the time required by other high level languages. Its capabilities are primarily directed toward the processing of variable-length string and array data. In addition, algebraic, Boolean and assembly-like bit manipulation operations are available.

Language processing is in every sense interpretive. Each line of MUMPS code undergoes identical processing each time it is executed. The language interpreter has two operating modes: program execution mode (Indirect Mode) and program creation mode (Direct Mode). In Direct Mode, programs can be created, modified, debugged, stored and executed in whole or in part. Indirect Mode permits the execution of these programs.

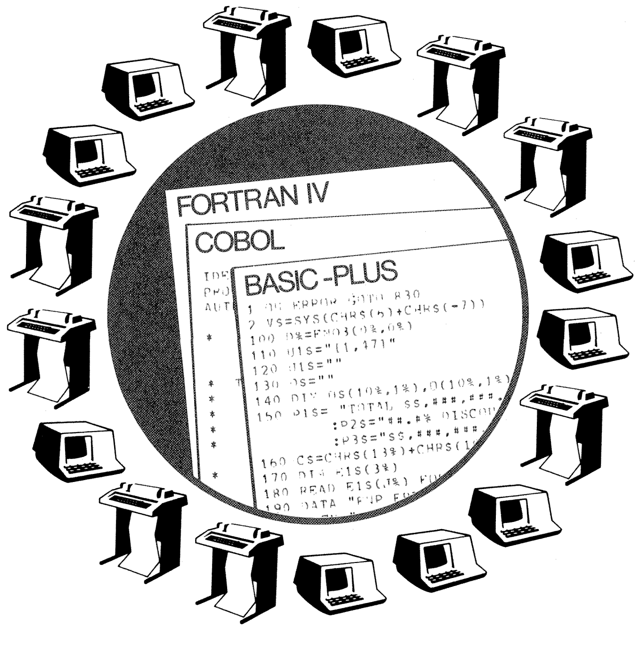
MUMPS-11 HARDWARE ENVIRONMENT

MUMPS-11 runs on a wide range of hardware configurations, from a small 2-user 24K word PDP-11/10 system to a large 40-user 124K word PDP-11/70 system. MUMPS-11 requires a console terminal, a disk system and a magnetic tape system in the minimum configuration. The user can add user terminals up to a system total of 64, which can be connected by as many as 16 remote or local lines and as many as 48 multiplexed local lines. The system supports additional disk controllers (up to a maximum of eight drives per controller), up to four magnetic tape drives, and up to four dual DECtape transports. A line printer, card reader and paper tape reader/punch are also supported.

PDP-11
RSTS/E

JULY 1976

RSTS/E, the Resource Sharing Timesharing System/Extended, is an operating system for the PDP-11 which allows multiple users to interactively process large amounts of data easily and efficiently. RSTS/E supports up to 63 simultaneous users processing data interactively using the BASIC-PLUS language processor. As options, the system can support BASIC-PLUS-2, ANSI'74, COBOL, FORTRAN IV, and RPG II language facilities.



FEATURES

- Powerful, flexible file system
- BASIC-PLUS, BASIC-PLUS-2, FORTRAN IV, RPG II, and ANSI'74 COBOL
- Efficient and controlled resource sharing
  Up to 63 users simultaneously on-line
- Commercial extensions
- Dynamic scheduling algorithm on a best fit-best throughput basis

MONITOR FUNCTIONS

The RSTS/E operating system automatically and dynamically assigns one of 255 job priority levels to each timesharing job. These priority levels are based on such criteria as job size, computing requirements, current time since last quantum of run time for the job, and input/output requirements. They may also be altered by the System Manager.

RSTS/E attempts to keep as many jobs in memory as possible. When more memory is required to run a job than is available, the system temporarily swaps some jobs out of memory and stores them in one of the swap files defined by the system manager. When it is again their turn to run, the jobs in one of the swap files are swapped back into memory.

I/O SERVICES

Under BASIC-PLUS, RSTS/E provides three access methods:

Formatted ASCII	For standard sequential I/O operations
Virtual Arrays	For random access of large data files that are too large to be contained in memory at one time. A virtual array is stored on disk and can contain string, integer and floating point matrices.
Record I/O	Allows the user to have complete control over I/O operations.

Since RSTS/E is a resource sharing system, every terminal user has access to all the system peripherals and resources. Line printers, card reader, disks, tapes are all available to any terminal user on-line.

FILE SYSTEM

The data file system of RSTS/E provides a wide range of on-line processing capabilities.

- Files can be random or sequential, numeric or alphanumeric.
- Files can be created, updated, extended and deleted interactively from the user's terminal or under program control.
- Files can be protected from access on an individual, group or universal basis.

- Files may be accessed by many terminal users simultaneously and updated on-line.
- Data can be stored on removable disk cartridges, disk packs, or magnetic tape. Industry-compatible magnetic tape files prepared on another computer can be read on RSTS/E, and tape files generated on RSTS/E are readable by other computer equipment.
- Programs and data may be totally or selectively back-up, on-line or off-line.

SYSTEM ADMINISTRATION

RSTS/E provides the System Manager with accurate and effective control of system use. The System Manager can, for each user, specify the programmer and project number, the password, the maximum logged-out disk space, and the maximum number of files.

Access to peripheral devices is generally open to all users under the resource sharing concept on a first-come, first-served basis. However, the capability is available to the System Manager to intervene in peripheral assignment and permit assignment as he sees fit.

SOFTWARE ENVIRONMENT

BASIC-PLUS—AN EXPANDED LANGUAGE

Timesharing users interact with RSTS/E using BASIC-PLUS. The language is easy to learn and use. Its immediate mode of operation enables the terminal to be used for simple calculations. Dynamic debugging is faster since programs may be interrupted at any point, checked, corrected, and operation resumed.

Immediate Mode Of Execution

Normal timesharing use of RSTS/E consists of typing program text using a keyboard terminal and at the end of the program typing a RUN command at which time the program executes. A second mode of using RSTS/E, called immediate mode, consists of typing program statements on the keyboard and having them executed immediately. Program statements are identical in either case except that, in immediate mode, they are typed without line numbers.

A comprehensive group of string operations is provided in BASIC-PLUS. Strings can be appended to one another. Strings can be compared to one another to see, for example, if a keyboard response is correct or to alphabetize a list of names.

Functions are available to extract, examine, or search for a string of characters contained within a larger string.

Matrix Operations

The user of RSTS/E can improve processing and programming efficiency by organizing numeric data into one- and two-dimensional arrays or matrices. Both numeric and character string matrices can be input, read, and printed with single commands. The BASIC-PLUS virtual memory facility can be used as an extension of main memory as needed.

Extended Program Statement Coding

The addition of numerous extensions previously found only in advanced scientific languages like ALGOL, permit more concise expression of complex program steps.

Formatted Output

Many applications, such as business data processing, require more flexible control of the printing format than Dartmouth BASIC allows. BASIC-PLUS includes a PRINT USING statement which may be used to achieve precise definition of printed data format. PRINT USING allows character, decimal, and exponential data field lengths and positions to be defined, and mixed, in a line of output. In addition, leading dollar sign or asterisk symbols may be "floated" to automatically precede the most significant digit of decimal fields. Also, trailing minus signs may be specified for compatibility with accounting report standards.

COBOL

PDP-11 COBOL is a fully implemented low-level compiler conforming in language element, representation, symbology, and coding format to American National Standard COBOL, Specification X.3.23-1974.

FORTRAN IV

PDP-11 FORTRAN IV provides substantially improved performance for the entire PDP-11 Family. New optimizations make programs small and fast, on any PDP-11 configuration. And the unmatched compilation speed of FORTRAN IV minimizes program development time.

Commercial Extensions to RSTS/E

In order to provide those system facilities common to most commercial data procesing operations, following:

Disk sort/merge program
Indexed file access method
Decimal arithmetic capability
Line printer spooling package

2780 Support

The RSTS/2780 software package enables a RSTS/E system to act as a powerful remote job entry terminal. Using RSTS/2780, RSTS/E users can queue data and job control files for transmission to a host IBM 360 or 370 system or to another PDP-11 Remote Computer System.

HARDWARE ENVIRONMENT

RSTS/E runs on a PDP-11/35 or 11/40 with the Extended Instruction Set, or on a PDP-11/45 or a PDP-11/70. The system requires a console terminal, real time clock, and 32K words of parity memory with the memory management option (at least 64K words of memory are required to support FORTRAN IV or COBOL). In addition, the system requires a disk pack system or a dual-drive fixed-head disk or disk cartridge. Magnetic tape is also generally required for software distribution.

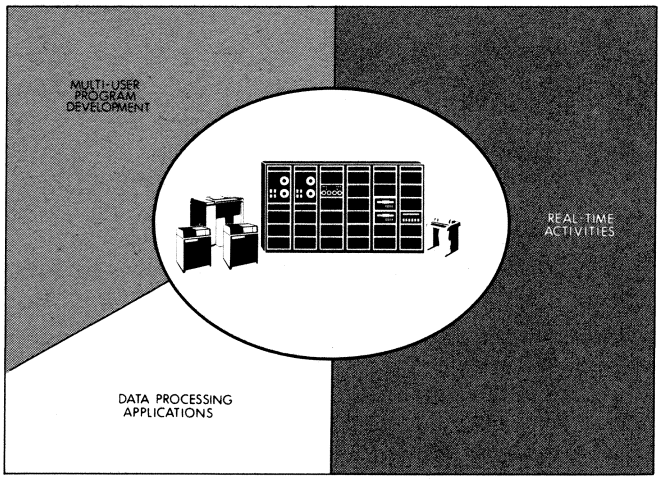
Optional Hardware Supported:

- Floating Point Processor
- Floating Instruction Set
- Additional memory up to 124K words on PDP-11/35, 40, 45, up to 1024K words on PDP-11/70
- Additional disk drives (up to 8 per type)
- Up to a maximum of 63 terminal line interfaces, which can be local, auto-answer dataset, or multiplexed
- Up to eight line printers
- Card reader
- Paper tape reader/punch
- Local line and auto-answer dataset interfaces
- Terminal multiplexers
- Video, hard-copy and IBM 2741-compatible terminals

PDP-11
RSX-11D

JULY 1976

RSX-11D is a complete multi-user operating system for concurrent real-time applications execution, program development and general data processing. Tasks to run in the priority schedule event-driven multiprogramming system can be written in MACRO assembly language, FORTRAN IV, FORTRAN IV-PLUS or COBOL.



FEATURES:

- Real-time multiprogramming
  - job/task accounting
  - event-driven priority scheduling
  - batch processing
  - loadable disk resident device handlers
- Disk-based operation
  - comprehensive file system
- Dynamic memory allocation
  - task checkpointing
  - system and user controlled partitions
  - re-entrant tasking
  - shared data areas
- User-mode peripheral diagnostics

MONITOR FUNCTIONS

The Executive

The executive includes the code that controls the multiprogramming environment, performs task scheduling and checkpointing, and interfaces with the monitor console routine. The executive also provides programmed services through directives issued by tasks.

Multiprogramming

Multiprogramming is the concurrent processing of two or more tasks (program images) residing in memory. In the RSX-11 family, multiprogramming is accomplished by logically dividing available memory into a number of named partitions. Tasks are built to execute in a partition, and all partitions in the system can operate in parallel.

Event-Driven Priority Scheduling

Task scheduling in the RSX-11 family is primarily event-driven, in contrast to systems which use a static scheduling mechanism for determining a task's eligibility to execute. The basis of event-driven task scheduling is the software priority assigned to each active task.

When a significant event (such as an external interrupt or I/O completion) occurs, the executive interrupts the executing task and searches for the highest priority task then capable of executing.

Task Checkpointing

Once a task is in memory, the executive normally allows it to run to completion in a multiprogrammed fashion even if its memory is required for the execution of a higher priority, non-resident task. However, if it is desirable to free memory for execution of a higher-priority task, a task can be declared checkpointable when it is created.

A checkpointable task currently active in a partition can be interrupted and rolled out of memory to disk when a higher priority task requests the partition in which it is active. Later, after the higher priority task has completed its execution, the checkpointed task will be rolled-in and restored to active execution at the point where it was interrupted.

Dynamic Memory Allocation

In the RSX-11D system, tasks are built with a base address of zero. This allows the user to load and execute a task in any partition large enough to contain it. When the task is loaded into memory, the system sets relocation registers to indicate where the task actually reside in memory.

I/O SERVICES

Device handlers are not strictly part of the executive in RSX-11D. They are implemented as system tasks using all the conventions of any other task in the system. Because they are tasks, device handlers are easily developed, installed and modified. Intimate knowledge of the executive is not required to develop a specialized device driver.

The handler tasks can be loaded into memory while the system is running, instead of being built into the executive during system generation. They can also be unloaded when their services are no longer needed, and the partition they occupy is freed for use by another task.

FILE SYSTEM

Files-11 is a general purpose file system that provides a facility for the creation, extension and deletion of files. Files are normally accessed by the logical names assigned to them. Multiple user file directories are possible. Designed into Files-11 is a scheme for volume and file protection which allows the owner of a volume or file to deny all access or certain kinds of access to all users, groups of users, or particular users in the system.

In addition to the RSX-11 Files-11 file system, RSX-11D supports ANSI standard Level 3 format for single- or multi-volume, multi-reel magnetic tape files.

USER INTERFACE

The Monitor Console Routine (MCR) is the interactive terminal interface between the user and the RSX-11 operating system. MCR includes: initialization commands, informational messages, task control commands, and system maintenance commands. Some commands are privileged and can be invoked only by privileged users, as defined by the system manager. The organization of MCR allows users to add commands to meet special application needs.

SOFTWARE ENVIRONMENT

BATCH

RSX-11D also supports a single-stream batch capability that allows users to submit jobs to the system for background processing. The user can create a file (or punched card deck) containing a series of commands and, optionally, data for batch processing. Once a job has been submitted, its tasks can execute without operator intervention unless the job specifically requests operator action.

Languages

FORTRAN IV

The PDP-11 FORTRAN IV language conforms to the specifications for the ANS FORTRAN x3.9-1966. The FORTRAN IV language includes many enhancements, such as mixed-mode expressions, up to seven array dimensions, default FORMAT widths, and byte data types.

FORTRAN IV-PLUS

The FORTRAN IV-PLUS compiler produces highly optimized object code which fully utilizes the floating point processor. Language extensions to the ANSI standard include: INTEGER*4 data type, generic function selection, formatted direct access I/O, and list-directed I/O.

COBOL

The PDP-11 COBOL language conforms to the low-level ANS-74 COBOL specifications, with high level implementations. It includes a SORT utility, a report generation utility, and a source program reformatting utility.

HARDWARE ENVIRONMENT

RSX-11D runs on the PDP-11/45, 55 and the PDP-11/70. The minimum configuration requires a central processor with Memory Management Unit (11/45, 55) and at least 48K words of memory, a console terminal, a disk system, and a magnetic tape system.

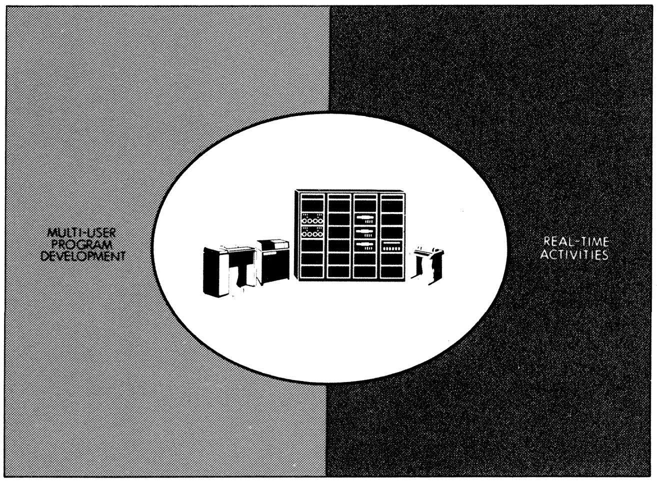
If concurrent program development and applications execution is desired, at least 56K words of memory are required. Memory can expand to 124K words on PDP-11/45 or 55 systems, or to 1024K words on the PDP-11/70.

RSX-11D supports a variety of laboratory, industrial control and communications equipment, including the AD01 Analog/Digital Converter, LPS11 Lab Peripheral System, ICS/ICR Industrial Control Systems (local and remote), and modem control multiplexers. Standard peripheral devices, such as a card reader, paper tape reader/punch, and line printer, are also supported.

PDP-11
RSX-11M

JULY 1976

RSX-11M is the performance leader of the RSX-11 family of real-time operating systems. Designed for minimum size and overhead, it can be generated to run in a variety of hardware and application environments: from small, dedicated laboratory or industrial control systems to large, multi-user transaction processing and information management systems.



FEATURES

- Real-time multi-programming
  event-driven priority scheduling
- Disk-based operation
  comprehensive file system
- Dynamic memory allocation
  automatic memory compaction
  task checkpointing
  system and user controlled partitions
- Configuration flexibility

MONITOR FUNCTIONS

Multiprogramming

Multiprogramming is the concurrent processing of two or more tasks (program images) residing in memory. In the RSX-11 family, multiprogramming is accomplished by logically dividing available memory into a number of named partitions. Tasks are built to execute in a specific partition, and all partitions in the system can operate in parallel.

RSX-11M systems that have the memory management unit provide automatic memory protection. The memory area assigned to a task is protected from other tasks executing in the system. Each task has a specific address range in which to execute. A task can reference and alter memory only within that area which it owns. In an RSX-11M system without the memory management unit, a task cannot be prevented from referencing all available memory.

Event-Driven Priority Scheduling

Task scheduling in the RSX-11 family is primarily event-driven, in contrast to systems which use a static scheduling mechanism for determining a task's eligibility to execute. The basis of event-driven task scheduling is the software priority assigned to each active task.

When a significant event (such as I/O completion) occurs, the executive interrupts the executing task and searches for the highest priority task capable of executing.

Task Checkpointing

Once a task is in memory, the executive normally allows it to run to completion in a multiprogrammed fashion even if its memory is required for the execution of a higher priority, non-resident task. However, if it is desirable to free memory for execution of a higher-priority task, a task can be declared checkpointable when created.

A checkpointable task currently active a partition can be interrupted and swapped out of memory to disk when a higher priority task requests the partition in which it is active. Later, after the higher priority task has completed its execution, the checkpointed task will be rolled-in and restored to active execution at the point where it was interrupted.

Dynamic Memory Allocation

As an option in systems with the hardware memory management unit, the RSX-11M executive can dynamically allocate available memory in system controlled partitions. The executive keeps a list of the available areas of memory and all tasks ready to execute in the partitions. Tasks are brought in to memory on a priority basis until all the requests are satisfied or there is no memory available in the partitions. When a task terminates, the memory it occupies becomes available again.

The Executive

The basic executive includes the code that controls the multiprogramming environment, performs task checkpointing and power fail recovery, and handles system traps.

I/O SERVICES

The file control services enable the user to perform record-oriented and block-oriented I/O operations and to perform additional functions required for file control such as open, close, wait and delete operations. The file services enable the user to read and write files on file-structured devices and to process files in terms of logical records. Both sequential and direct access modes are supported.

FILE SYSTEM

Files-11 is a general purpose file system that provides a facility for the creation, extension and deletion of files. Files are normally accessed by the logical file names assigned to them. Many users can have file directories. Designed into Files-11 is a scheme for volume and file protection which allows the owner of a volume or file to deny all access or certain kinds of access to all users, groups of users, or particular users in the system.

USER INTERFACE

The monitor console routine (MCR) is the terminal interface between the user and the RSX-11 operating system. MCR includes: initialization commands, status, messages, task control commands and system maintenance commands. Some commands are privileged commands which can be invoked only at privileged terminals, as defined by the system manager. The MCR organization makes it possible for users to add commands to meet special application needs.

SOFTWARE ENVIRONMENT

FORTRAN IV

PDP-11 FORTRAN IV language conforms to the specifications for the ANS FORTRAN X3.9-1966. It includes many enhancements, such as mixed-mode expressions, up to seven array dimensions, default FORMAT widths, and byte data types.

FORTRAN IV-PLUS

The FORTRAN IV-PLUS compiler produces highly optimized object code which uses the floating point processor hardware. Some of its extensions to the ANSI standard are: INTEGER*4 data type, generic function selection, formatted direct access I/O, and list-directed I/O.

HARDWARE ENVIRONMENT

RSX-11M runs on any of the PDP-11 processors except the LSI-11 based processors. The minimum configuration requires a central processor with a clock and at least 16K words of memory, a console terminal, a disk system, and a magnetic tape system. The minimum system requires 8K words of memory for the operating system; 8K words of memory are available for user tasks.

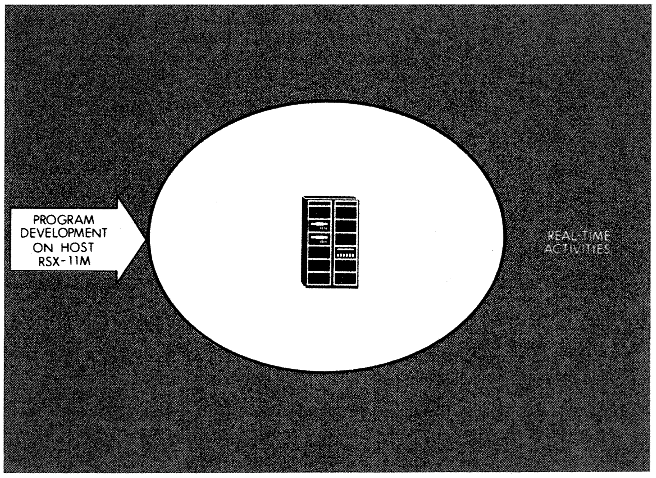
If concurrent program development and applications execution is desired, at least 24K words of memory are required. Memory can expand to 28K words on systems without the memory management options, to 124K words on systems with the memory management unit, or to 1024K words on the PDP-11/70.

RSX-11M supports a wide range of laboratory, industrial control and communications equipment, including the AR11 Analog Real-time Subsystem, LPS11 Lab Peripheral System, ICS/ICR Industrial Control Systems (local and remote), modem control multiplexers, and the DMA UNIBUS link. Standard peripheral devices, such as floppy disks, cassettes, a card reader, paper tape reader/punch, and line printer are also supported.

PDP-11
RSX-11S

JULY 1976

As the smallest member of the RSX-11 family of real-time multiprogramming operating systems, RSX-11S provides a dedicated, execute-only environment for monitoring and controlling many real-time processes concurrently. Program development and RSX-11S system generation take place on a host RSX-11M system. This means that all of the RSX-11S system's resources are devoted to supervising real-time applications execution.



FEATURES:

- real-time multiprogramming
- memory-only
- dedicated multi-task system
- event-driven priority scheduling
- multiple task partitions
- on-line task loading

MONITOR FUNCTIONS

Multiprogramming

Multiprogramming is the concurrent processing of two or more tasks (program images) residing in memory. In the RSX-11 family, the multiprogramming of tasks is accomplished by logically dividing available memory into a number of named partitions. Tasks are built to execute in a specific partition, and all partitions in the system can operate in parallel.

The user can define the partitioning scheme to match the needs of the particular application tasks that are to be concurrently executing. Partition location and size are determined by the user during system generation. In RSX-11S systems, partitions are user controlled.

If the RSX-11S system configuration includes the memory management unit, the system provides automatic memory protection. The memory area assigned to a task is protected from other tasks executing in the system. A task is assigned a specific address range in which to execute. A task can reference and alter memory only within the areas it owns. In systems without memory management, a task can not be prevented from referencing all of available memory.

RSX-11S is implemented as a memory-based subset of RSX-11M. It is not dependent on any mass-storage device for execution. Also, because RSX-11S is a subset of RSX-11M, complete upward compatibility is provided. Any application task that executes under RSX-11S will execute under RSX-11M without change. The I/O device driver interfaces are identical. Device drivers written to execute under RSX-11M execute under RSX-11S without modification, and vice versa.

As a real-time multiprogramming system, RSX-11S provides most of the RSX-11 family's services, with the exceptions of dynamic memory allocation, task checkpointing, program development facilities and a file system. The primary RSX-11 family characteristics which RSX-11S displays are multiprogramming, event-driven task scheduling, and power failure recovery.

Event-Driven Priority Scheduling

Task scheduling in the RSX-11 family is primarily event-driven, in contrast to systems which use a time slice mechanism for determining a task's eligibility to execute. The basis of event-driven task scheduling is the software priority assigned to each active task.

When a significant event (such as I/O completion) is declared, the Executive interrupts the executing task and searches for a task capable of executing.

OPERATING SYSTEM COMPONENTS

The RSX-11S system software is completely modular: software components can be included or omitted during system generation to suit the needs of a particular application or hardware environment. The user has virtually complete control over the size and capabilities of the operating system.

In spite of its many services, the RSX-11S operating system itself uses very little of the system's resources. RSX-11S can be generated to operate in systems with as little as 8K words of memory. The minimum RSX-11S software system includes the executive and I/O device drivers. The executive can be as small as 2.5K words, and two to four device drivers can be included in the system adding as little as 1.5K words. In an 8K word system, therefore, approximately 4K words would be available to application tasks.

The user can also add support for a console terminal if operator communication is desired. If the system has at least 16K words of memory, the on-line task loader and system image preservation utilities can be added. At least 16K words of memory are required to execute tasks written in FORTRAN.

The Executive

The RSX-11S executive controls the multiprogramming environment, performs power fail restart, and handles system traps. In addition, the executive includes a subset of the RSX-11 family's file control services which provide record I/O support services, but contain no file-structured services.

On-line Task Loader Utility

The on-line task loader (OTL) can be included in an RSX-11S system if on-line task loading is desired. OTL can load task images into memory partitions from paper tape, cassette, floppy disk, DECtape, or magnetic tape. Tasks are created on the host RSX-11M system, copied onto the load media and then loaded into the RSX-11S system using OTL.

System Image Preservation Utility

The system image preservation program (SIP) is an on-line utility task that can save the image of a running system on a load medium such as paper tape, cassette, floppy disk, DECtape or magnetic tape. The saved system image can subsequently be restored by bootstrapping it from the load device.

Operator's Console Interface

In RSX-11 systems, the monitor console routine (MCR) acts as the console terminal interface between the operator and the operating system. If desired, a subset of the RSX-11 MCR commands can be included in an RSX-11S system to support operator/system communication. The MCR commands include: system information commands, task control commands, and system maintenance commands.

SOFTWARE ENVIRONMENT

RSX-11S AND NETWORKS

As an option, an RSX-11S operating system can support DECnet-11S. DECnet-11S extends the capabilities of the RSX-11S operating system by enabling RSX-11S systems to be interconnected with other DECnet systems. DECnet-11S can be used as a component of distributed networks, resource sharing networks and communications networks. Some of the features of a networked RSX-11S system are:

- The initial memory image of an RSX-11S system can be stored on a remote RSX-11 system device, and down-line loaded into the RSX-11S system.
- Task images can be stored on a remote RSX-11 system device and be loaded into the RSX-11S memory partitions in response to programmed or operator requests in the RSX-11S system.
- Tasks written in FORTRAN or MACRO can request the execution of programs in other systems in the network, and can cause programs executing in remote systems to be terminated.
- Tasks written in FORTRAN or MACRO can exchange data with other tasks executing in the DECNET environment, on the same and other processors in the network.

To support DECnet-11S, an RSX-11S system must have an additional 8K words of dedicated memory and a communications line.

HARDWARE ENVIRONMENT

RSX-11S runs on any of the PDP-11 processors, from the LSI-11 based products to the PDP-11/70. The minimum RSX-11S system consists of a CPU with at least 8K words of memory and a load device such as paper tape, cassette, DECtape, magnetic tape or floppy disk. The system can support a console terminal if operator communication is desired. At least 16K words of memory are needed to support on-line task loading or the execution of tasks written in FORTRAN.

RSX-11S requires an RSX-11M system for system generation and program development. An RSX-11S system is generated from the RSX-11M system using the standard system generation process. The maximum RSX-11S hardware and software configuration is the same as that of an RSX-11M system with the exceptions of non-resident tasks, task checkpointing, dynamic memory allocation and a file system. RSX-11S does, however, support non-file-structured data storage on all devices supported by RSX-11M.

Optional devices supported include a variety of laboratory, industrial control, and communications equipment. Memory can expand to 28K words on systems without the memory management option, to 124K words on systems with the memory management unit, or to 1024K words on the PDP-11/70.

PDP-11

IAS

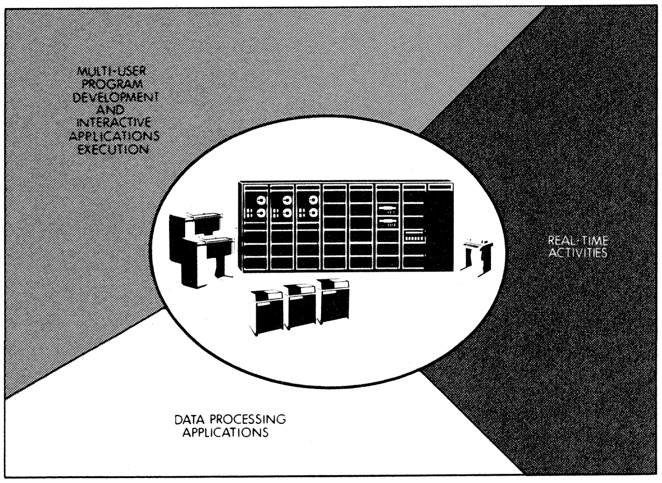
interactive application system - (niet kompliceerd)

JULY 1976

IAS is a complete, general-purpose operating system for the PDP-11/45 or PDP-11/70 computers. IAS provides a *mixed-mode* processing environment which enables real-time applications to execute concurrently with timeshared multi-user interactive and batch processing applications.

The major components of the IAS operating system are:

- multiprogramming executive with real-time and timesharing scheduling
- Files-11 file system
- general-purpose program development user interface
- system operator's interface
- program development and system maintenance utilities


MULTI-USER PROGRAM DEVELOPMENT AND INTERACTIVE APPLICATIONS EXECUTION
REAL-TIME ACTIVITIES
DATA PROCESSING APPLICATIONS

FEATURES

- Concurrent real-time interactive and batch processing
  - priority scheduling for real-time tasks
  - timeshared scheduling for interactive and batch tasks
- Comprehensive file system
  - file protection
  - user file directories
- Interactive application program development
  - easy-to-use command language
  - multi-language support
  - inter-task communications
- Extensive system management facilities
  - job/task accounting and control
  - dynamic system tuning
  - user defined command languages
  - system manager controls
  - on-line device diagnostics
- Dynamic memory allocation
  - shared user data areas
  - reentrant tasking for user programs
  - loadable device drivers

MONITOR FUNCTIONS

The IAS operating system provides the ideal processing environment for users who need the ability to service a wide range of processing activities in one computer system. The primary functions of the IAS executive include memory and disk management, file management, and maintenance of overall system integrity. To provide system flexibility, the IAS executive manages multi-programmed operations using two schedulers: the event-driven priority scheduler for privileged real-time tasks, and the timesharing scheduler.

The timesharing scheduler controls both interactive and batch processing. The executive schedules interactive and batch tasks on the basis of multi-queued, response-optimizing, selection algorithm when real-time activities do not take precedence. Batch processing normally receives the processor time available after interactive tasks are serviced. The timesharing scheduling algorithm and batch service parameters can be modified by the system manager when the needs of the system change.

Real-Time Executive

The heart of the IAS operating system is the real-time executive kernel. Real-time tasks, when active, require top priority on all the system's resources: from memory space to CPU time. The IAS executive provides these services through memory partitioning and event-driven priority scheduling. In addition, I/O handlers for real-time peripheral devices can be selectively loaded in the system when the real-time tasks using the devices need them, and only when they need them. To increase system reliability for a real-time environment, IAS provides power failure recovery, on-line device diagnostics, and hardware/software protection mechanisms.

FILE SYSTEM

IAS includes a general purpose file system that enables the user to create, extend and delete files. Files are normally accessed by the logical names assigned to them. Multiple user file directories are possible. The file system has a volume and file protection scheme which allows the owner of a volume or file to deny all access or certain kinds of access to all users, groups of users, or particular users in the system.

The executive's file control services enable the programmer to perform record-oriented and block-oriented I/O operations and to perform additional functions required for file control, such as open, close, wait and delete operations. The file services enable the user to read and write files on file-structured devices and to process files in terms of logical records. Both sequential and direct access modes are supported.

In addition to the file I/O system, IAS supports ANSI standard Level 3 format for single- or multi-volume, multi-reel magnetic tape files.

INTERACTIVE PROGRAM DEVELOPMENT SYSTEM

IAS includes an interactive timesharing user's interface program called PDS (Program Development System). PDS controls access to the system by allowing only valid users to log in at a terminal and gain access to user file accounts assigned by the system manager. PDS accepts and interprets commands typed on the user's terminal and performs the requested operation. The commands are English words that describe the operation to be performed. They can be abbreviated to as many letters as make the command name unique.

INTERACTIVE APPLICATION INTERFACES

IAS has special provisions for user-written terminal interface programs. Any program written to interact with a user at a terminal, such as an order entry, inventory control, bank teller, engineering computation, or on-line file update program, can be identified to the system as a command language interpreter (CLI). Many different terminal interface programs can be installed in the system and operated concurrently to provide an application-specific work-station capability at a remote terminal.

BATCH PROCESSING

As a batch system, IAS services a single queue of jobs which can be dequeued by multiple batch command language processors. The timesharing scheduler services the batch queue as if it were a low-priority timesharing terminal. System control parameters permit the system manager to adjust the level of batch throughput IAS will provide.

SYSTEM MANAGEMENT AND ACCOUNTING FACILITIES

The system manager authorizes users by assigning them system-recognizable user names and passwords. A user name and password enable the user to log into the system under the PDS (Program Development System) user interface. IAS enables the system manager to account for the user of the system and to control the kinds of activities each user can perform.

SOFTWARE ENVIRONMENT

BASIC

BASIC includes all of the features of the standard Dartmouth BASIC, plus extensions. Interactive in nature, it is an ideal language for the timesharing environment of IAS.

FORTRAN IV-PLUS

FORTRAN IV-PLUS is an extended version of ANSI-standard FORTRAN IV designed for high-speed execution. FORTRAN IV-PLUS generates highly optimized, sharable machine code for all hardware features. It is implemented with software virtual memory allowing the efficient compilation of large programs in a small user space.

COBOL-11

COBOL-11 is the PDP-11 business data processing language. It is a fully implemented Level 1 ANSI COBOL-74 standard compiler with many Level 2 features. Included with the COBOL compiler are a Report Program Generator and a SORT program.

HARDWARE ENVIRONMENT

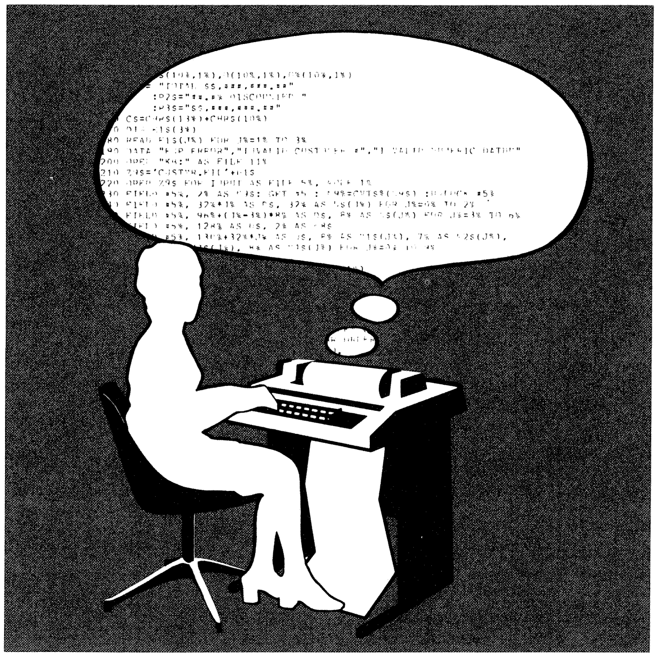
IAS runs on a PDP-11/45 or 55 with the memory management option or a PDP-11/70. The system must include at least 64K words of memory, two terminals, a disk system and a magnetic tape system. A 96K-word PDP-11/70 with RP04 system disk is recommended for support of 10 to 12 terminal users performing interactive and batch concurrent with real-time processing.

IAS supports a variety of laboratory, industrial control and communications equipment, including the AD01 Analog/Digital Converter, LPS11 Lab Peripheral System, ICS/ICR Industrial Control Systems (local and remote), and modem control multiplexers. Standard peripheral devices, such as a card reader, paper tape reader/punch, cassettes, DECtape and line printer, are also supported.

PDP-11
BASIC

JULY 1976

BASIC-11 is a high-level, easy-to-learn language, compatible with Dartmouth standard BASIC. It is an immediate response, interactive language, giving the user the capability to develop and debug a program in a minimum amount of time. BASIC-11 can be used for executing large data processing tasks as well as performing quick, one-time calculations.



```
  S(10%,1%),D(10%,1%),P%(10%,1%)
= "TOTAL $$,###,###.##"
   :P2$="##.#% DISCOUNTED "
   :P3$="$$,###,###.##"
 C$=CHR$(13%)+CHR$(10%)
0 DIM E1$(3%)
80 READ E1$(J%) FOR J%=1% TO 3%
90 DATA "[illegible] ERROR","INVALID CUSTOMER #","INVALID NUMERIC DATUM"
200 OPEN "KB:" AS FILE 11%
210 Z9$='CUSTMR.FIL'+[illegible]
220 OPEN Z9$ FOR INPUT AS FILE 5%, MODE 1%
230 FIELD #5%, 2% AS C9$: GET #5 : C9%=CVT$%(C9$) :UNLOCK #5%
  FIELD #5%, 32%*I% AS D$, 32% AS S$(I%) FOR J%=0% TO 2%
  FIELD #5%, 96%+(I%-3%)*8% AS D$, 8% AS S$(J%) FOR J%=3% TO 6%
  FIELD #5%, 128% AS D$, 2% AS [illegible]
  #5%, 130%+32%*J% AS D$, 8% AS M1$(J%), 7% AS M2$(J%),
  [illegible](J%), 8% AS M1$(J%) FOR J%=0% TO 3%
```

FEATURES

- Optional string capability
- Assembly language routines
- Immediate execution mode
- Convenient editing
- Interrupt driven I/O
- Modular programs

DESCRIPTION

Optional String Capability

Dartmouth-compatible string support, complete with string arrays and functions, is available. Memory-conscious users can delete the string support and reclaim the memory space for their programs.

Calling Assembly Language Routines

A "CALL" statement allows easy interfacing of assembly language routines. The routines can be called by name and several arguments can be passed. This technique is especially useful when interfacing to unique peripherals.

Immediate Execution Mode

This powerful feature allows the user to examine, modify, and control sections of the running program that have been stopped at selected points. This is a tremendous aid in program debugging. A single statement or a small part of a program can be executed immediately after being input. The computer can be used with the ease of a simple desk calculator.

Convenient Editing

A complete compilation of the input source program is not performed before the program is run, so that the source program (with only formatting changes) still exists in memory for editing or manipulation. Editing is done conveniently using the line numbers assigned to each statement.

Interrupt Driven I/O

Fast real-time response is achieved since the input/output peripherals signal by interrupts when service is required, or when an action has been completed. This allows the user to perform real-time programming in the convenient, easy-to-use BASIC-11 language.

Modular Programs

BASIC-11 is a modular implementation which allows the selective exclusion of unused functions and language features to provide more user memory space. The user can make memory size/feature trade-offs that reflect a particular application.

Optional Peripheral Support

Support is available for the LPS11 Laboratory Peripheral System and GT42, GT44, VT11 and VS60 Graphics Display Systems, as well as the standard line of PDP-11 disks, DECtape, paper tape, cassettes, and line printers. Support for the LV11 Electrostatic Printer is available as an option.

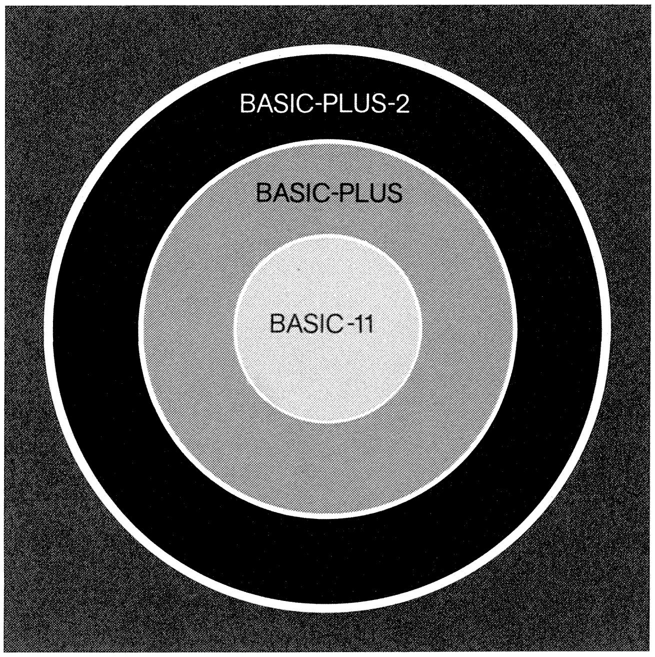
BASIC COMPARISON TABLE

	BASIC/PTS	BASIC/CAPS	BASIC/RT-11	MU BASIC/RT-11	IAS BASIC
Number of users:	1	1	1	1 to 8	1 to 20 (Shared)
Min. memory size:	8K words (16K for LPS11)	8K (12K for string support)	8K (12K for string support)	16K	12K partition
Max. memory size:	28K	28K	28K	28K	partition up to 28K
Features:	core-only	Sequential cassette files under CAPS-11, CHAIN, OVERLAY, CALL, ON-GOTO, ON-GOSUB, PRINT-USING, COMMON	Sequential and direct access files under RT-11, CHAIN, OVERLAY and CALL	Sequential and direct access files under RT-11, multi-user support, CHAIN, OVERLAY, CALL, ON-GOTO, ON-GOSUB, PRINT-USING, COMMON	FCS supported device in a timesharing environment. Language features same as MU-BASIC except for CALL, OVERLAY, and direct access files.

PDP-11
BASIC-PLUS-2

JULY 1976

BASIC-PLUS-2 is an optional compiler and object time system for the RSTS/E, RSX-11M and IAS operating systems. It is upward compatible with BASIC-11 and BASIC-PLUS and includes both new functions and enhanced performance.



FEATURES

Increased functionality

- Record I/O
  - LOCATE mode
  - MOVE mode
  - Sequential and relative access in LOCATE and MOVE mode
  - Implicit data conversion
  - Automatic record blocking
- A CALL statement which allows program access to external subroutines
- Long variable names (up to 30 characters)
- Optional indexed sequential access method (multi-key ISAM)
- Expanded debugging capability such as breakpointing

Increased performance

- True compilation capability with code optimization
- Significantly better run-time performance in a compute bound environment
- Lower core requirements
- Fast edit, compile and execute turnaround time

DESCRIPTION

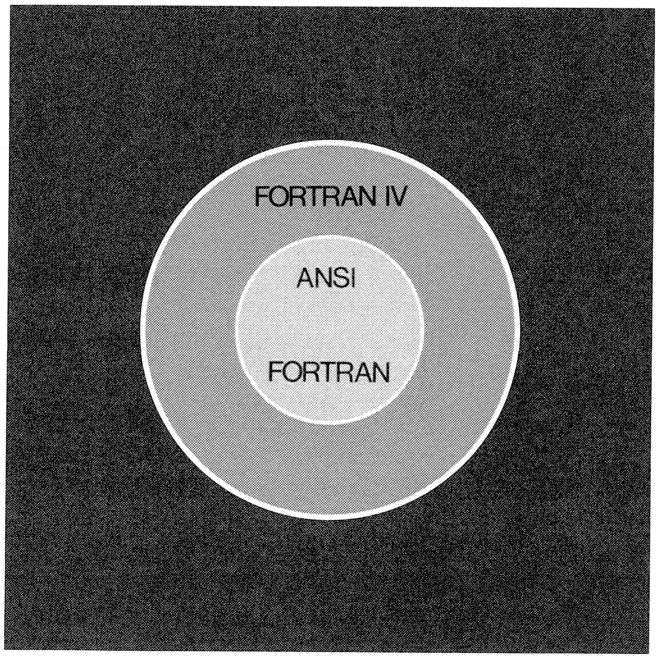
BASIC-PLUS-2 provides superior performance for the high end of the PDP-11 family. Compute bound and sophisticated data access programs will execute faster under BASIC-PLUS-2 since machine code will be executed rather than BASIC-PLUS code. Core requirements are minimized since the object code and run time support are smaller, and the symbol table is not normally kept at run time.

BASIC-PLUS-2 has a built-in editor (compatible with BASIC-PLUS) and the ability to directly output a runnable file. Thus the user can often avoid the need to link the program. The conditions which will force the user to use the task builder will be a CALL to an external routine, a program too big to compile (more than 256K characters) or the need to use overlays to run a large program. The compiler speed itself (when not doing optimization) is of the same order as BASIC-PLUS. Finally, the compiler does immediate syntax checking on input entered from a terminal so that many errors will be found without doing a compilation.

PDP-11
FORTRAN IV

JULY 1976

PDP-11 FORTRAN IV provides substantially improved performance for the entire PDP-11 Family. New optimizations make programs small and fast, on any PDP-11 configuration. And the unmatched compilation speed of FORTRAN IV minimizes program development time.



FEATURES

Speed

Typical 300-line programs compile in less than 9 seconds on a PDP-11/45. Large-scale 2,200-line programs compile in just one minute on a PDP-11/45. Run-time speed can be enhanced by using the extended arithmetic and floating point hardware options. The FORTRAN system uses specially-optimized code for each option.

Size

Optimizations and unique design allow a typical 250-line FORTRAN program to compile and execute on an 8K-word PDP-11. But small size does not mean fewer features or optimizations. All the benefits of FORTRAN IV are available, regardless of memory size.

Portability

Program portability is ensured because the same FORTRAN IV processor is available on five PDP-11 operating systems–RT-11, RSTS/E, IAS, RSX-11D and RSX-11M. Further, the FORTRAN IV language, with its

DESCRIPTION

Language Extensions

PDP-11 FORTRAN is a superset of ANSI-standard FORTRAN, allowing standard conforming programs written for other computer systems to run unmodified under FORTRAN IV.

Powerful extensions included within the PDP-11 FORTRAN IV make for markedly easier program coding. Some of the enhancements are:

General Expressions

Because of the restrictions on the usage of expressions, ANSI FORTRAN programs are often cluttered with assignment statements and temporary variables. PDP-11 FORTRAN IV eliminates these restrictions. General expressions are permitted in most places that a variable or constant is called for by ANSI FORTRAN.

Mixed Mode Arithmetic

Using PDP-11 FORTRAN IV, there is no need to worry about data conversions by assignment. Mixed-mode arithmetic is supported in all arithmetic contexts and the use of a variable in an expression implies that the appropriate conversion will be done.

Bit Manipulation

The LOGICAL operators can be applied to INTEGER quantities, to perform masking operations and bit manipulation.

Data Formatting

The FORTRAN IV programmer has the use of extended I/O facilities that completely use the power and flexibility of the PDP-11 and its operating systems. Formatted (character), unformatted (binary) and direct-access unformatted (record-oriented binary) are all supported.

End of File and Error Condition Transfer

The specifications END = n and ERR = n (where n is a statement number) can be included in any READ or WRITE statement to transfer control to the specified statement upon detection of an end-of-file or error condition.

Facilities

Overlay Facility

The overlay facility allows large programs to be executed in relatively small memory areas and is virtually invisible to the FORTRAN programmer.

Libraries

The FORTRAN programmer can create a library of commonly-used assembly language and FORTRAN functions and subroutines. The operating system's librarian utility provides a library creation and modification capability.

Debugging

Two debugging facilities are available to the FORTRAN programmer: 1) The FORTRAN object time system provides the traceback feature for fatal run-time errors. This feature locates the actual program unit and line number of a run-time error; 2) A "D" in column one of a FORTRAN statement allows that statement to be conditionally compiled. These statements are considered comment lines by the compiler unless the appropriate debugging lines switch is issued in the compiler command string. In this case, the lines are compiled as regular FORTRAN statements.

Optimization

Many techniques are used to increase the execution speed of an object program. Redundant expressions which occur in the same basic block of a program are located and eliminated.

Expressions which are constant (such as multiple constant subscripts of an array) are calculated at compilation time. Constant portions of subscripts are absorbed into the array address to require no additional operations during execution.

Branch structure optimizations improve program speed and size.

A unique automatic "array vectoring" feature of FORTRAN IV eliminates the time-consuming multiply operations required in array subscripting. And precompiled FORMAT statements make formatted input and output conversions faster and smaller.

To give a final polish to each program, FORTRAN IV does extensive local (or "peephole") optimization, examining each sequence of operations output and substituting a shorter and faster group if possible.

ENVIRONMENTS

The FORTRAN IV compiler and OTS is available as an optional language processor for the RT11, RSTS/E, and IAS operating systems. The FORTRAN IV compiler is included with the RSX-11M and RSX-11D operating systems. The compiler operation and facilities under each of these systems are essentially identical.

Each operating system provides additional features particular to the environment. For example, the monitor programmed requests or executive directives are usually available as a library of FORTRAN-callable routines.

UNDER RT-11

The RT-11 System Subroutine Library (SYSLIB) is a collection of FORTRAN-callable routines which allow a FORTRAN programmer to use various features of the RT-11 Foreground/Background (F/B) and Single-Job monitors. SYSLIB also provides various utility functions, a complete character string manipulation package, and 2-word integer support SYSLIB is provided as a library of object modules to be combined with FORTRAN programs at link-time. SYSLIB allows the RT-11 FORTRAN user to write almost all application programs in FORTRAN with no assembly-language coding.

Also available under RT-11 are:

- A library of FORTRAN-callable graphics routines supporting the VT11 and VS60 graphics hardware hardware systems.
- Plotting support for the LV11 electrostatic printer/plotter.
- Laboratory data acquisition and manipulation routines used in conjunction with the LPS11 and AR11 laboratory peripheral hardware.
- The Scientific Subroutine Library, providing FORTRAN-language routines for statistical applications.

UNDER RSTS/E

RSTS/E FORTRAN IV operates in interactive or batch mode under the RSTS/E monitor. The FORTRAN IV system includes the FORTRAN IV compiler, the object time system (OTS), and several utility programs. RSTS/E FORTRAN IV provides assembly language subprogram support, using the MACRO assembler. Although the assembly language subprogram can not issue any monitor calls, MACRO provides the experienced user with a tool to further enhance computational performance.

UNDER RSX-11 and IAS

In RSX-11M, the FORTRAN IV compiler runs in a minimum partition of 8K words. If run in a larger partition it uses the extra space for program and symbol table storage. In RSX-11D and IAS, the compiler task requires 8K words minimally and can be extended when it is installed. As with RSX-11M systems, the additional space allows the processing of larger FORTRAN programs.

An RSX-11/IAS library consists of object modules. Two types of libraries exist, shared and relocatable. Relocatable libraries are stored in files. Object modules from relocatable libraries are built into the task image of each task referencing the module. Shared libraries are located in main memory and a single copy of each library is used by all referencing tasks.

The RSX-11/IAS system relocatable library provides FORTRAN-callable forms of most executive directives. The FORTRAN programmer can schedule the execution of tasks, communicate with concurrently executing tasks, and manipulate system resources through these calls.

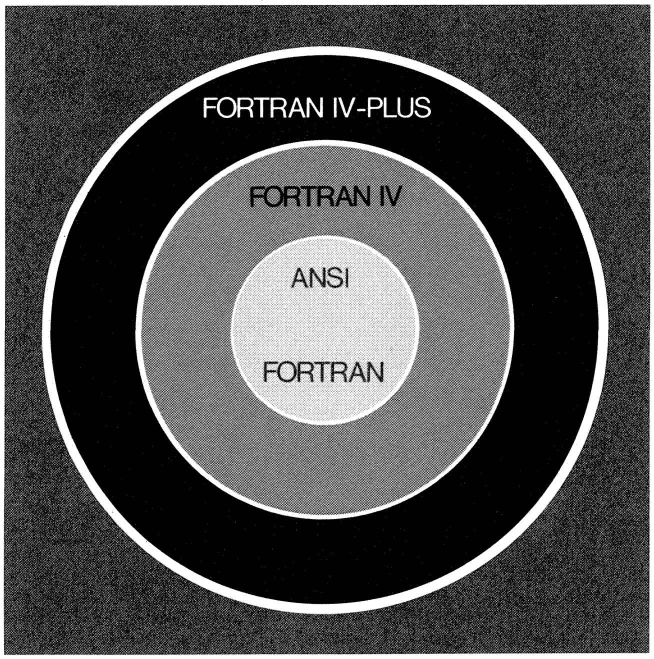
Industrial Society of America (ISA) extensions for process I/O control are available in FORTRAN-callable format under RSX-11. Support for laboratory and process control peripherals is also included.

PDP-11
FORTRAN IV-PLUS

JULY 1976

FORTRAN IV-PLUS is a high performance, optimizing compiler and object time system for the RSX-11D, RSX-11M and IAS operating systems. It is upward compatible with PDP-11 FORTRAN IV, with the same enhancements to the ANSI standard.

The primary differences between the FORTRAN IV compiler and the FORTRAN IV-PLUS compiler are that the FORTRAN IV-PLUS compiler produces highly optimized PDP-11 machine language code, creates code which uses the floating point processor option, and can produce shareable code.



FEATURES

- Language extensions
  OPEN & CLOSE
  Formatted direct access read and write
  List-directed I/O
  INCLUDE
  PARAMETER
  ENTRY
  EXTERNAL *
- Highly optimized machine code
- Uses floating point processor option
- Can produce shareable code

DESCRIPTION

Language Extensions

The FORTRAN IV-PLUS language is upward compatible with the PDP-11 FORTRAN IV language, and supports the same enhancements to the ANSI standard. Some of them are:

- General expressions, which are permitted in most places that a variable or constant is called for by ANSI FORTRAN.
- Mixed mode arithmetic is supported in all arithmetic contexts and the use of a variable in an expression implies that the appropriate conversion will be done.
- Bit manipulation is provided by applying the LOGICAL operators to INTEGER quantities.
- End-of-file and error condition transfers allow program control of I/O and data format errors as well as end-of-file conditions.

In addition, FORTRAN IV-PLUS also includes the following extensions:

Input/Output Statements

- OPEN and CLOSE–OPEN establishes a connection with a new or existing file and specifies its attributes, such as record size. CLOSE breaks the connection with the file.
- Formatted direct access READ and WRITE–Direct access files can be read or written under format control.
- List-directed input/output–Eliminates the need for FORMAT statements. In WRITE, TYPE and PRINT statements the data type of the variable in the I/O list determines the format of the elements. With READ and ACCEPT the data is converted into an internal format and is assigned to the variables in the I/O list.

Specification Statements

- INCLUDE–Allows FORTRAN source text from a separate file to be included in a FORTRAN program. It can appear anywhere in the program, and the included file can itself contain an INCLUDE statement.
- PARAMETER–Allows program variables to be defined in one statement. Thus variables which are used throughout a program can be changed by modifying just one adjustment.
- ENTRY–Permits multiple entry points in a program. It can be used within functions and subroutines.

Subprogram Statements

- EXTERNAL • name–permits function and subroutine names to be passed as arguments in a call. Otherwise the function name would be treated as a variable.

Optimizations

On typical compute-bound programs, a FORTRAN IV-PLUS compiled program executes two to three times faster than the same program compiled by the FORTRAN IV compiler on identical configurations. In addition the program may also be smaller. Some of the optimizations are:

Compile-Time Operations on Constants

The FORTRAN IV-PLUS compiler performs the following compile-time computations on expressions involving constants, including PARAMETER constants.

- Negation of Constants
- Type Conversion of Constants
- Integer Arithmetic on Constants

In addition, array subscripts involving constants are simplified at compile-time where possible.

Elimination of Common Subexpressions

Often the same subexpression appears in more than one computation. If the values of the operands of a common subexpression are not changed between computations, the value of the subexpression can be computed once and its result can be substituted where the subexpression appears.

Removal of Invariant Computations from DO Loops

The speed with which a given algorithm can be executed is increased if computations are moved from frequently executed program sequences to less frequently executed program sequences. In particular, computations within a loop involving only constants and variables whose values are not changed within the loop can be moved outside the loop.

Binding of Variables and Expressions to Registers

Frequently referenced variables are retained, if possible, in registers to reduce the number of load and store instructions. Frequently used variables and expressions are also assigned to registers across DO loops.

Shareable Code

The compiler produces shared object code when the read-only compilation switch is specified. Shared tasks are then created by using the multi-user linker switch. This ensures more efficient memory utilization since many users can share the one memory resident task.

ENVIRONMENTS

The FORTRAN IV-PLUS compiler is available as an optional language processor for the RSX-11M, RSX-11D and IAS operating systems. The compiler's operation and facilities under each of these operating systems are essentially identical. One of the special features is that FORTRAN IV-PLUS can produce shareable code.

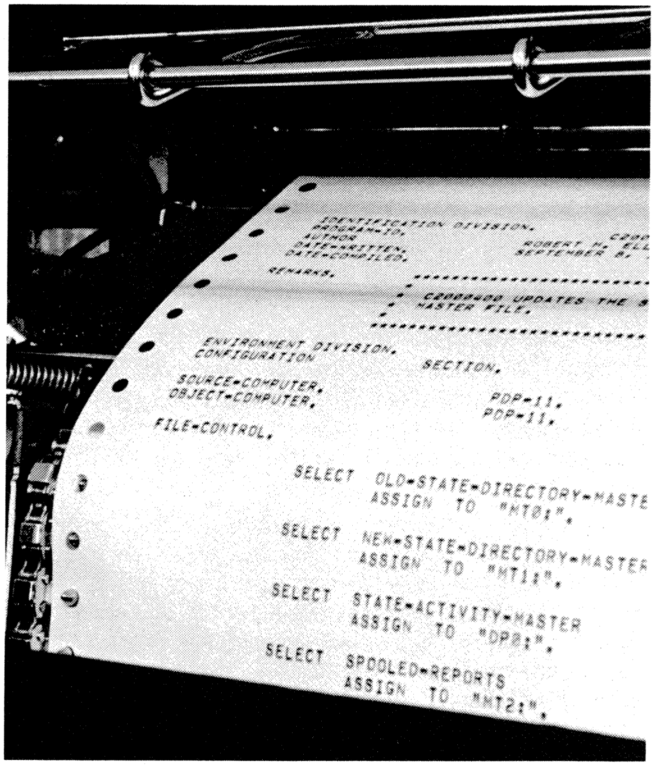
In all operating systems, the hardware configuration must include the FPP Floating Point Processor. The FORTRAN IV-PLUS compiler requires a minimum partition size of 17K words to execute in an RSX-11D or IAS system not including the system library. Under RSX-11M, the compiler and system library require a minimum partition size of 18K words.

PDP-11
COBOL

digital

JULY 1976

The PDP-11 COBOL compiler and object time system is an optional language processor for the RSTS/E, RSX-11D and IAS operating systems. It is a fully implemented low-level compiler conforming in language element, representation, symbology and coding format to ANS specification X.3.23-1974. Compilation and execution of COBOL programs are characterized by a high rate of throughput and efficient memory utilization.



```
IDENTIFICATION DIVISION.
PROGRAM-ID.                                     C20
AUTHOR                                ROBERT H. ELL
DATE-WRITTEN.                         SEPTEMBER 8,
DATE-COMPILED.
REMARKS.          **************************
                  *
                  *   C20040D UPDATES THE S
                  *   MASTER FILE.
                  *
                  **************************
ENVIRONMENT DIVISION.
CONFIGURATION          SECTION.
SOURCE-COMPUTER.                  PDP-11.
OBJECT-COMPUTER.                  PDP-11.
FILE-CONTROL.
            SELECT  OLD-STATE-DIRECTORY-MASTE
                    ASSIGN  TO  "MT0:".
            SELECT  NEW-STATE-DIRECTORY-MASTER
                    ASSIGN  TO  "MT1:".
            SELECT  STATE-ACTIVITY-MASTER
                    ASSIGN  TO  "DP0:".
            SELECT  SPOOLED-REPORTS
                    ASSIGN  TO  "MT2:".
```

FEATURES

- Utilities for file sorting, source reformatting, and report generating
- On-line interactive program execution
- String manipulation capabilities
- Sequential and relative I/O
- Efficient memory utilization

DESCRIPTION

The Language

On-line Interactive Program Execution

The Procedure Division ACCEPT and DISPLAY statements allow terminal-oriented interaction between a COBOL program and a user. Using these statements, a COBOL program can exercise interactive operation with a user running the program.

File Organization

Both the Sequential I/O and Relative I/O modules meet the full ANS-74 high-level standards and include the following COBOL verbs:

OPEN EXTEND	Add records to a previously-created sequential file without recopying the file.
DYNAMIC ACCESS	Process relative file both randomly and sequentially in the same program.
START	Logical positioning within a relative file for subsequent record retrieval.
REWRITE/WRITE	Logically replace a record in a mass storage file and generate a new record.
CLOSE LOCK	Protect a file from being opened second time by the currently processing program.
LINAGE	Specify logical page format.

String Manipulation

PDP-11 COBOL has the capability to manipulate data strings. It offers INSPECT, STRING, and UNSTRING–powerful COBOL verbs for character string handling–to search for embedded character strings, with tally and replace. Plus, they have the ability to join together or break out separate strings with various delimiters.

Debugging Features

With PDP-11 COBOL, programming need not be complicated by device specifications because device assignments can be made at execution time–a benefit for test and later production runs.

Debugging large source programs is made still easier by the use of the optional Data Division allocation map and the modular programming techniques offered by the segmentation facility.

The Compiler

Performance

Compilation and execution of COBOL programs by the DIGITAL PDP-11 systems are characterized by a high rate of throughput and efficient memory utilization. Depending on the size of the compiler generated in a particular system, COBOL source programs can contain up to 6,000 statements. Running on a PDP-11/45 processor, COBOL can compile up to 400 lines a minute. On a PDP-11/70 processor, COBOL can compile at a rate between 1000 and 1200 lines per minute.

Operation

PDP-11 COBOL is a compile-and-go system. This means that the normal processing flow is from source program input to object program creation to object program execution. There is no intermediate step of linking. The COBOL system can, of course, be requested to compile only, or to execute a previously compiled object program.

I/O Flexibility

The disk-resident compiler can accept source program input from cards, consoles, and disks—including the capability of accepting input from source text library files stored on disks. In addition, COBOL programs can also create ANSI standard format magnetic tape files if magnetic tape systems are included in the system's hardware configuration.

Library Facility

With PDP-11 COBOL the user has a full ANS-74 low-level Library facility, plus high-level extensions. All frequently-used data descriptions and program text sections can be held in library files available to all programs. These files can then be copied in source programs at compile-time.

Virtual Memory

A work file is used as a virtual extension of processor memory. The COBOL Work File System allows the COBOL compiler to compile and execute much larger programs than would normally be expected of a minicomputer with limited available memory. The system makes an efficient use of processor memory by allowing working data to overflow to disk; in addition, it monitors the use of data kept in memory to minimize accesses made to disk, based on the activity of data in memory and on disk.

Utility Programs

SORT File Storing Program

The user arranges the data base for special processing techniques using the SORT program. The SORT program allows the user to:

- rearrange, delete and reformat records in a mass-storage file
- select a sorting process and input device that best suits the processing environment
- create new file indices to access a large data base
- batch stream several sorting jobs
- define sort specification files in a standard fixed format

REFORMAT allows the programmer to enter source programs in the simpler terminal format and then, if compatibility is ever required for those programs, provides a simple method for conversion to conventional format.

COBRG Report Generating Program

The COBRG (COBOL Report Generator) utility program provides a fast, simple mechanism for producing printed reports from data files.

ENVIRONMENT

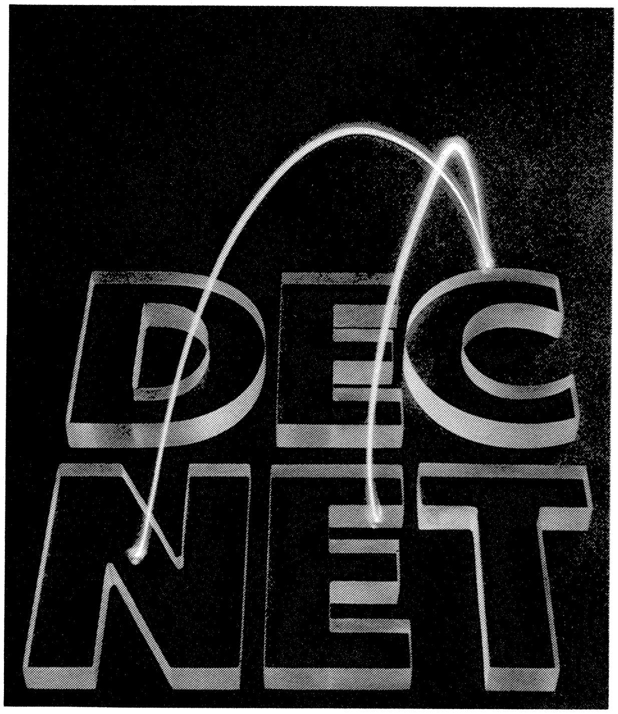
PDP-11 COBOL is available under three operating systems: RSTS/E, RSX-11D and IAS. PDP-11 COBOL is completely independent of the operating systems on which it is available. It uses the standard file system available under the operating system. The commands used to invoke the compiler and run-time system are the same from system to system.

The hardware configuration supporting COBOL under RSTS/E, RSX-11D or IAS operating systems must include a PDP-11/45, 11/55 or 11/70 central processor and memory, a console terminal, a line printer, and disk storage. Under RSTS/E or IAS, at least 64K words of memory are required to run COBOL. An RSX-11D system must have at least 48K words of memory. The recommended minimum disk storage is either two RK05 disk drives or an RP03 or RP04 disk drive. Optional hardware supported includes a card reader and magnetic tape systems.

PDP-11
DECnet

JULY 1976

DECnet is the set of software products which extend various DIGITAL operating systems so they may be interconnected with each other to form computer networks. The DECnet user can configure a variety of networks, satisfying a variety of constraints, by choosing the appropriate CPU's, line interfaces (and speeds) and operating systems software.



FEATURES

Digital Network Architecture, implemented across a wide range of operating systems and hardware architectures, enables users to build a variety of networks. Such networks have some common attributes, and individual systems in the network may have certain system-specific attributes. The common attributes include:

- Inter-Program Communication: Programs on one system can exchange data with programs on other systems in a real-time fashion.
- Inter-System Resource Sharing: Programs on one system can utilize files and devices physically attached to other systems in the network.

Additionally, many DECnet systems support other features which are useful in the network environment. These include:

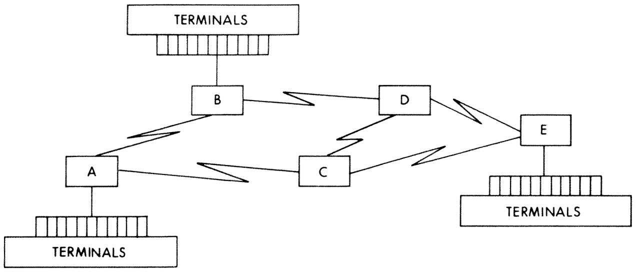
- Down-Line System Loading: Initial core images for other systems in the network may be stored on the local system, and loaded on request into other systems in the network. The remote systems usually require the use of a network bootstrap loader, implemented in read-only memory.
- Down-Line Program Loading: Programs to be executed on other systems in the network may be stored on the local system, and loaded on request into other systems, under the joint control of the operating systems at both ends of the network. This and the preceding feature simplify the operation of network systems without mass storage devices, by allowing such systems to use remote mass storage devices in a convenient and straightforward manner.
- Down-Line Program Commands: Programs running on one system in the network may cause program commands to be executed at remote systems in the network. This feature allows remote programs to be started and stopped.
- Inter-System File Transfer: This facility allows an entire data file to be moved between systems, at either program or operator request.
- Cross-System Support: This facility allows program development activities to be performed on a system different from the one where the programs will be executed.
- High-Level Language Interface: This facility allows programs written in high-level languages (FORTRAN, COBOL, BASIC) to access some or all of the network facilities.

DESCRIPTION

Networks typically fall into one of three classes:

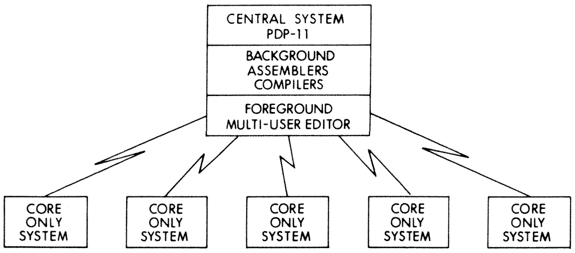
Communications Networks

These networks move data from one physical location to another, often distant, location. The data may be file oriented (as is often the case for remote job entry systems) or line oriented (as occurs with the concentration of interactive terminal data). Interfaces to common carriers, using both switched and leased-line facilities, are normally a part of such networks. Such networks are often characterized by the concentration of all user applications programs and data bases on one or two large 'Host Systems' in the network.



Resource-Sharing Networks

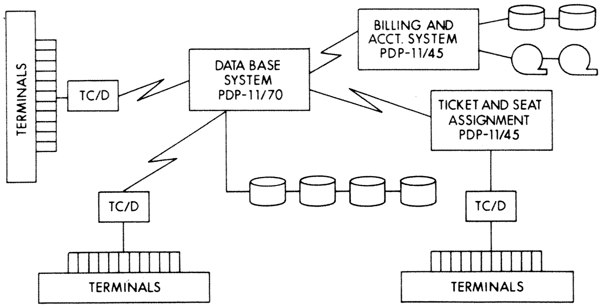
These networks permit the sharing of expensive computer resources among several computer systems. Shared resources can include powerful central-processors, mass storage devices, unit record equipment, and other peripherals. The resource to be shared may be a logical, rather than physical, entity, as in a data base which is stored centrally and made available to other machines in the network. Such networks are often characterized by the concentration of large peripherals, large data bases, and large programs on one or two 'Host Systems' in the network, while 'Satellite Systems' have smaller peripherals and programs.



Distributed Computing Networks

These networks coordinate the activities of several independent computing systems toward some larger goal. Networks of this nature may have specific geometries (star, ring, hierarchy), but often have no regular arrangement of links and nodes. Such networks are usually built to put computational power close to the users of such power. Distributed computing networks are usually characterized by a multiplicity of comparably sized computers and applications programs and data bases distributed throughout the network.

In order to satisfy these widely varying constraints, DECnet allows the user to build networks from a range of system and communication components.


TERMINALS
TC/D
DATA BASE SYSTEM PDP-11/70
BILLING AND ACCT. SYSTEM PDP-11/45
TICKET AND SEAT ASSIGNMENT PDP-11/45
TC/D
TC/D
TERMINALS
TERMINALS

DECnet includes a set of network protocols, each of which is designed to fulfill specific functions within the network. Collectively, these protocols are known as the Digital Network Architecture, or DNA. The major protocols, and their functions, are:

Digital Data Communications Message Protocol (DDCMP)

DDCMP handles the link traffic control and error recovery within DECnet. DDCMP has been designed to operate over full and half-duplex facilities, using synchronous, asynchronous, and parallel facilities.

Network Services Protocol (NSP)

NSP handles network management functions within DECnet, including the routing of messages between systems, and the routing of messages within any given system.

Data Access Protocol (DAP)

The DAPenables programs on one node of the network to utilize the I/O services available on other network nodes. Each operating system in DECnet provides facilities for translating its own unique I/O calls into the DAP standard, and vice-versa. DAP thus allows remote file access, including OPEN, READ, WRITE, CLOSE and DELETE for sequential and random files, and remote device access, for unit record devices.

ENVIRONMENT

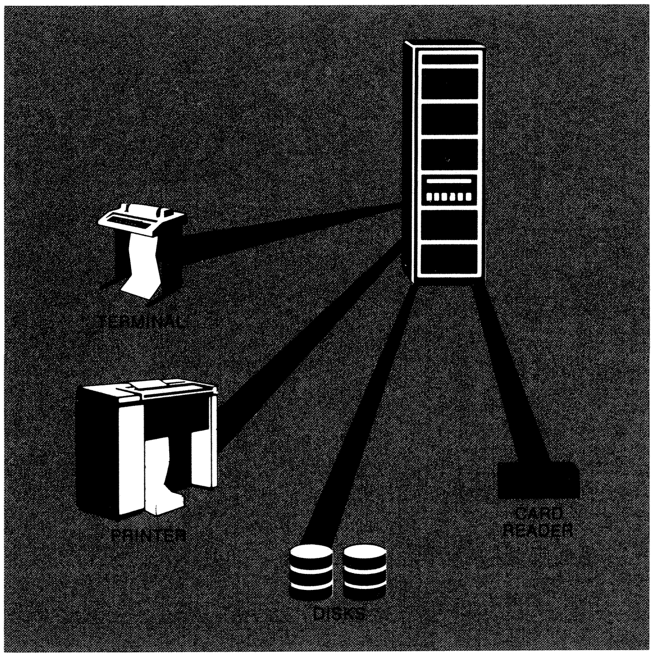
DECnet comes as an enhancement to the following DIGITAL operating systems:

Operating System	Processor
RTS-8	PDP-8
RT-11	PDP-11
RSX-11S	PDP-11
RSX-11M	PDP-11
RSX-11D	PDP-11
RSTS/E	PDP-11
IAS	PDP-11
TOPS-10	DECsystem-10

PDP-11
2780 Emulator

JULY 1976

The 2780 Emulator programs allow a DIGITAL operating system to emulate the features of an IBM Model 1 or 2 2780 Data Transmission Terminal with the multiple record option. This permits communication between the PDP-11 and the following IBM Remote Job Entry programs: OS/HASP, OS/ASP, OS/RJE, and DOS/POWER. In addition the PDP-11 may communicate with a second PDP-11 with 2780 support. PDP-11 operating system which have 2780 support are: RT-11, RSTS/E, RSX-11 and RSX-11M. As with IBM 2780's, BISYNC (Binary Synchronous Control) protocol is used for communication.



FEATURES:

- Transmits data from card readers and mass storage devices such as disk
- Prints received data on line printer
- Writes files on disk
- Character or binary transmission

DESCRIPTION

Transmission

All 2780 emulators can transmit data from card readers and mass storage devices such as disk. The physical unit transmitted is a block, which is in turn divided into logical units called records.

The user can transmit data in character or binary (transparent) mode. If the user selects character format, the 2780 emulator converts the ASCII characters to their EBCDIC equivalents before transmission.

Reception

All 2780 emulators can print received character data on a line printer or write it as files on a mass storage device such as a disk.

All 2780 emulators can receive binary data as well as the subset of EBCDIC that has an ASCII equivalent. They automatically convert the EBCDIC data to ASCII, unless the user chooses to have the EBCDIC data written to a file in binary format.

HARDWARE

MODEMS AND DATA LINKS

All 2780 emulators support operations over synchronous data links, in point-to-point contention mode, at speeds up to 4800 baud. Supported modems include Bell 201 or 208 or equivalent. Data link control characters are automatically supplied.

PDP-11 HARDWARE

Specific hardware requirements are discussed under each operating system in the chart below. The following are required by all:

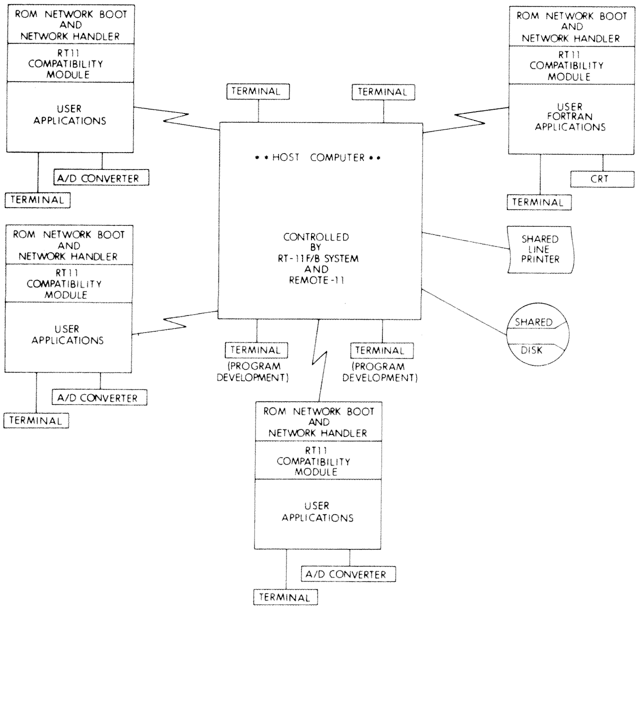
- DP11, DU11, or DUP11 Synchronous Data Link
- KG11-A Communications Arithmetic Element
- KW11-L or -P Real Time Clock

PDP-11
REMOTE

JULY 1976

REMOTE/RT-11 is an RT-11 based communications system using DECnet compatible messages. It allows users having a minimum 16K (28K word for foreground/background) RT-11 system to communicate with a maximum of eight satellite PDP-11s. A satellite system has the facility to develop programs (edit, assemble, compile and link) using RT-11 foreground/background in the host system. Any PDP-11 processor may be a satellite. Additionally, a satellite system may down-line load absolute programs.

REMOTE SYSTEMS WITH
FOUR SATELLITES AND FOUR TERMINALS


ROM NETWORK BOOT AND NETWORK HANDLER
RT11 COMPATIBILITY MODULE
USER APPLICATIONS
A/D CONVERTER
TERMINAL
TERMINAL
TERMINAL
ROM NETWORK BOOT AND NETWORK HANDLER
RT11 COMPATIBILITY MODULE
USER FORTRAN APPLICATIONS
CRT
TERMINAL
• • HOST COMPUTER • •
CONTROLLED BY RT-11F/B SYSTEM AND REMOTE-11
SHARED LINE PRINTER
SHARED DISK
ROM NETWORK BOOT AND NETWORK HANDLER
RT11 COMPATIBILITY MODULE
USER APPLICATIONS
A/D CONVERTER
TERMINAL
TERMINAL (PROGRAM DEVELOPMENT)
TERMINAL (PROGRAM DEVELOPMENT)
ROM NETWORK BOOT AND NETWORK HANDLER
RT11 COMPATIBILITY MODULE
USER APPLICATIONS
A/D CONVERTER
TERMINAL

FEATURES:

- A low-cost resource sharing network.
- A multi-user program development system.
- Easy to build, easy to use network.
- A multi-user editor.

DESCRIPTION

A REMOTE network allows researchers to place satellites next to their experiments and to dedicate a CPU for data acquisition, data reduction, and process control. Furthermore, each REMOTE node has access to all host resources. For example, a satellite may use the central disk to store or retrieve both programs and data. The fundamental theme for REMOTE is: to provide a straightforward, inexpensive yet responsive network tool for achieving systematic integration of diverse laboratory processes.

REMOTE-11 systems form a star topology, which means that each satellite computer sits at an experimental station and is physically connected back to the host system at the star's center. A typical REMOTE node would consist of a terminal and a core-only PDP-11 with A/D converters. Their job is to gather data from an instrument, process it and ship it back to the host via an asynchronous serial data link. In addition to real-time tasks, any node may serve as a program development station by utilizing the host's multiprocessor editor, or may perform other functions such as graphics applications. The host processor runs the system and REMOTE software manages the central resources such as the disk, and in certain cases controls real-time experiments or processes batch streams.

Components

REMOTE-11 runs under the RT-11 disk operating system and consists of

- A multi-user editor: LETTER.
- Communication routines to communicate with satellite computers using the Digital Data Communications Message Protocol (DDCMP).
- Facilities for down-line loading of programs into satellite computers.

Hardware Configurations

The basic hardware system consists of a central computer system (called a host computer) operating under the control of the RT-11 monitor with one to eight terminals connected to it. The terminals may be connected directly through DL11 asynchronous interfaces or may be connected to satellite computer systems which are connected to the host computer through DL11 interfaces. A maximum of eight terminals and/or satellite computers with terminals may be connected to the host. Maximum simultaneous throughput is about 78K baud (7800 characters per second).

RT-11 Compatibility

Satellite computers connected to the host may also utilize an RT-11 compatibility module which will allow programs to execute in satellite computers under a simulated RT-11 environment. The module converts all necessary monitor calls issued by a satellite program into DDCMP communication calls and sends them to the host computer which actually performs the indicated operations. Satellite computers utilizing the compatibility module will be able, for example, to run FORTRAN/RT-11, BASIC/RT-11, or FOCAL/RT-11 programs with random access file capability, and execute assembly language programs with up to four files open at any one time. Satellite programs, relinked without the compatibility module, will operate directly under RT-11.

Host Operations

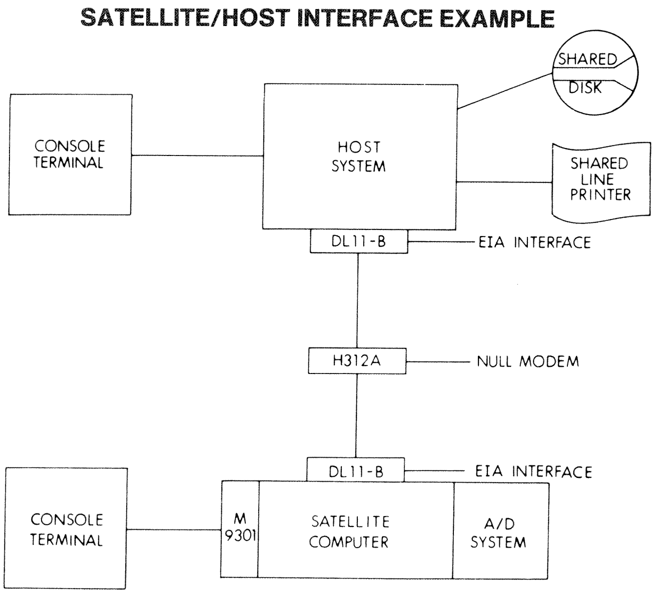
The REMOTE-11 software runs under the RT-11 operating system in the host computer. REMOTE-11 may be used with either the single-job monitor (SJ) or the foreground/background monitor (F/B). When using the SJ monitor, REMOTE-11 is a multi-user editor, a down-line loader of programs into satellite processors, or a handler for file requests from satellite systems. The SJ system cannot be used to simultaneously assemble, link, and compile.

Satellite Operations

Satellite computers connected to a host running the REMOTE-11 software can operate in three modes:

- A terminal connected to the satellite can be used as a normal terminal to the REMOTE-11 software and perform all operations that local terminals connected to the host can perform.
- The user can request programs to be down-line loaded into the satellite and automatically started.
- Programs loaded into the satellite can communicate with the host to open and close files, transfer data, and simulate an RT-11 operating system in the satellite. This allows programs to be developed for satellite processors with only a small amount of memory, a terminal, and a communications interface. Additionally, if the satellite processor is upgraded to an RT-11 system, software written previously will execute with no changes.

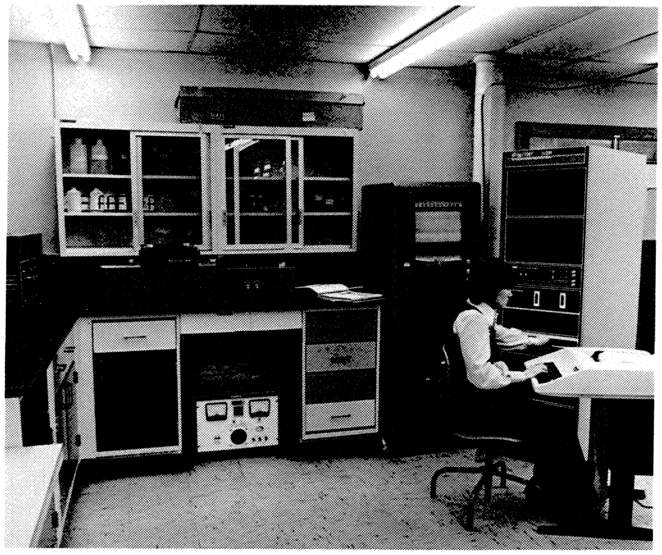
SATELLITE/HOST INTERFACE EXAMPLE



PDP-11
Lab Applications-11

JULY 1976

Lab Applications-11 is a package of application software modules specifically designed to solve most lab automation problems. It is a software library comprised of a broad range of assembly language routines, each of which has been streamlined to perform a unique function while maintaining its flexibility to interface with any combination of companion modules. In essence, DIGITAL provides the ingredients and leaves the rest to you.



FEATURES

- Implements a wide variety of functions common to laboratory computer needs, such as:
  data acquisition
  instrument control
  data display and manipulation
  file storage and retrieval
- Easily modified for specific needs
- Supported by RT-11
- Supports a wide variety of devices such as:
  LPS-11 (Laboratory Peripheral System)
  graphics subsystems
  magnetic tapes
  disks
  paper tape

DESCRIPTION

Lab Applications-11 modules are available to perform operator console interaction, data acquisition, data editing, fast Fourier transformation, output printing, and displaying. Lab Applications-11 allows many variations of these functions. The library of modules will be enhanced as time goes on, and as application needs are defined, more and more of the requirements for laboratory computing will be supplied by DIGITAL.

Signal Processing and Real-Time Analysis (SPARTA)

SPARTA consists of five major components: console interface, data acquisition, interactive graphics, data reduction, and data storage/retrieval. The console interface passes command input from the researcher and controls the execution of the other SPARTA components. For example, the command BIN (Buffer INtegrate) activates the integration routines in the data reduction module. SPARTA can sample analog signals concurrently from multiple channels at aggregate rates in excess of 28 kHz, a single channel at 40 kHz, and up to 150 kHz under DMA (Direct Memory Access).

Interactive Dialogue Module

The Interactive Dialogue Module provides the researcher with a standard interface to facilitate console question-and-answer dialogue for such activity as initiating or controlling an experiment and providing parameters for manipulation routines.

Output Formatter Module

The Output Formatter converts internal data types and prints them in a user specified format on the console terminal. The output formatter performs floating point and octal conversions, outputs character strings, handles carriage control, and column and line formats.

Analog Data Acquisition Modules

This group of modules samples data from up to 64 analog channels simultaneously. It provides the flexibility to continuously buffer each channel independently, assimilate the data from different sensors at variable time intervals, and acknowledge both external and programmed start.

Spike Train Analysis Modules

These modules allow the clinical researcher to acquire and analyze data from both spontaneous and stimulated spike train signals. The acquisition module will record elapsed time between subject responses subsequent to a given stimulus, measure spike train activity in synchronization to an external stimulus or act as the stimulus source itself. The processing module can generate stimulus time or a latency histogram.

Envelope Profile Analysis Modules

The acquisition module extracts peaks and the time of their occurance from an incoming data stream; specifically, only samples above a user-defined threshold and the time the peak occurred with respect to start of sampling. The envelope profile analysis module will optionally retain input from two additional sensors for each threshold penetration.

Envelope Profile Processing Modules

The Envelope Profile Processing Module will generally operate on an envelope of peak profiles, reducing such blocks of information to a compact parameter stream. The routine will optionally subtract out baseline from intensity to offset environmental biases, and will delete user-defined minimum width peaks to eliminate noise spikes and enhance spectral analysis. All peak profiles are reduced to time occurrence of peak, peak area above threshold, peak centroid, peak width, and optionally, two additional sensor readings per peak.

Peak Reduction Processing Module

This module will process peaks on-line from multiple independent instruments. Each data stream is reduced asynchronously with respect to other experiments; the processor will retain the following peak characteristics: area, height, width, position, and baseline information. The peak reduction module will separate fused peaks by dropping a perpendicular at the valley between them or perform automatic tangential "skimming" of small peaks following large ones.

Graphics Display Module

This module will plot byte, single precision, double precision or floating point data on a CRT under the control of an independent asynchronous hardware graphics processor which can also display alphanumeric characters and vectors for axis labelling and decimal readout of XY coordinate pairs. The routine will display one or two wave forms simultaneously either superimposed or isolated. Up to 1,024 points can be plotted with options to plot arbitrary points of a wave form and to begin plotting at variable positions along the X-axis to facilitate display windowing. The module can apply linear transformations to Y data, display data which is in either Y or XY format and support user-defined interactive cursors in fixed and free formats.

FFT Module

The Fast Fourier Transform Module computes both forward and inverse complex transformations on any number of data points from 8 to 8,192 in powers of two. The algorithm operates on single precision data using scale correction factors and quarter wave sine tables to minimize execute time while maximizing precision. The transform for 1K points executes under one second, and supports hardware multiply/divide options.

Point Plot Display Module

This module will display single precision integer, byte, and double precision data on an XY point plot scope; the routine can display one or more spectra simultaneously with a window display and cursors. The data can be in Y or XY format with a Y offset.

Throughput Module

This module will throughput data from core memory into a file on an auxiliary storage medium; it will be common for this routine to execute in cooperation with other Lab Application modules. The throughput routine will support multiple and asynchronous data streams in real-time with options to divert incoming spectra to separate data files.

MINIMUM SYSTEM REQUIREMENTS

1. Any UNIBUS PDP-11 central processor with an ASCII compatible terminal.
2. 16K of main frame memory.
3. Auxiliary storage devices:
   a. Dual-drive DECtape.
   b. Dual-drive floppy disk.
   c. Dual-drive, removable-cartridge DECpack.
   d. Single-drive, removable-cartridge DECpack with either a dual-drive cassette or a dual-drive floppy disk.
4. RT-11 binary license.

Supported Options

1. Up to 28K of main frame memory.
2. Extended arithmetic element.
3. Additional removable-cartridge DECpack and DECtape drives.
4. Fixed-head disks.
5. Industry standard 7- and 9-track magnetic tapes.
6. High-speed paper tape reader/punch.
7. Line printer.
8. A/D converter, multiplexer, and programmable switch gain.
9. Programmable real-time clock with Schmitt triggers.
10. Digital I/O.
11. Asynchronous display processor.
12. Point plot display controller.

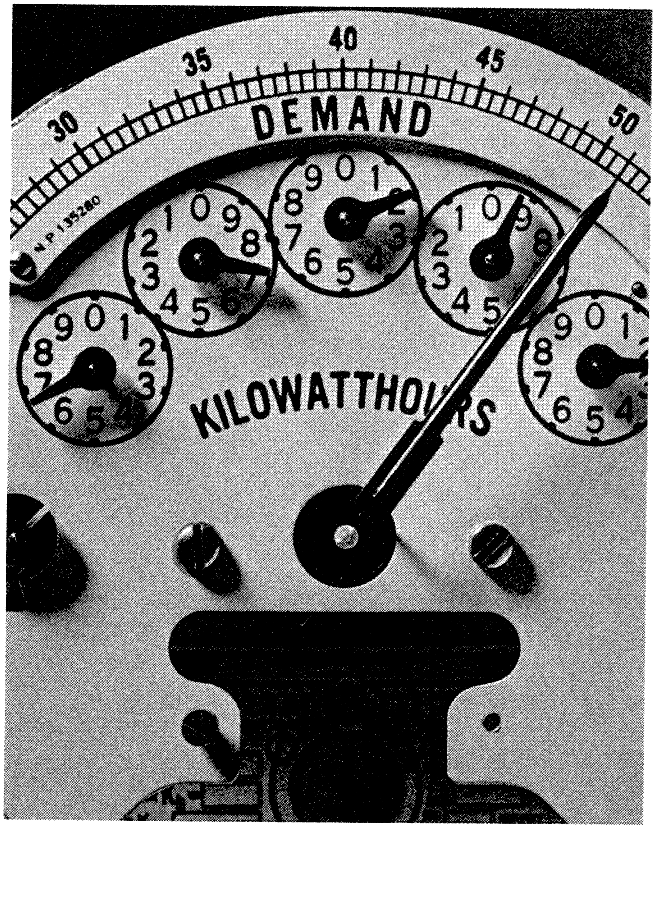
PDP-11
Power Management Systems

JULY 1976

A significant portion of your power bill is frequently for the power demand charge, which power companies calculate on the basis of the peak power your facility consumes.

Abnormally high power peaks, therefore, can prove to be very costly. They can result for a variety of reasons—several large pieces of equipment running simultaneously, or many small pieces of equipment running simultaneously.

DIGITAL's Power Management Systems combine proven industrial computer systems with the power management programs to provide a cost effective tool for reducing power demand and energy usage. All the programming is ready-to-use. Simplified messages and commands provide flexible system control without the need for a computer programmer or expensive operator training.



FEATURES

- Reduces peak power demand and energy requirements
- English-like operator commands simplify operation
- Control based on user-entered parameters
- Parameters permit selection of order in which equipment is shut off and how long it's off
- Six daily time periods provide flexible equipment control
- All parameters easily entered and changed
- Cassette tapes provide compact media for preserving operating data and for loading programs
- Simulated power meter included for trying parameters
- Expansion capability to 96 loads
- Built around standard software/hardware components

DESCRIPTION

DIGITAL's Power Management System works by reducing peak power demand without adversely affecting productivity or comfort levels. It compares actual power consumption with a specified Target Demand and shuts off (sheds) selected equipment or loads when it is necessary to reduce peaks. The system is linked to power company metering equipment and monitors meter signals, constantly projecting power usage to the end of the demand interval. If demand continues to increase, additional selected equipment is shut down. Equipment is later restored to full operation. Overall energy usage can be further reduced by continuously cycling selected loads.

System Parameters

Setting up the Power Management System is a quick and relatively simple procedure. You describe each piece of your equipment and how you want the system to control it. Since each load or piece of equipment has individual characteristics, a fill-in-the-blanks format lets you tune system operation and equipment control to your plant operations. We call these individual characteristics "load parameters". These load parameters allow you to specify (for each load):

- Load ID or name
- Load priority level, where high priority loads are shed last
- Minimum "off-times" for each of 6 daily time periods
- A minimum cycle time to prevent excessive equipment restarting
- A time-of-day for equipment to be automatically turned on and off, such as lights or fans
- Recovery from a plant power outage. Upon return of electrical service you can select whether you want a load to remain off or on until the system has active control again
- Kilowatt power consumption of each load. Load size is taken into account as loads are shed. Other loads are held in reserve for possible later load shed requirements

Saving System Parameters

After the system has been set up by entry of the parameters for your operation, the parameters can be saved on cassette tape. Several sets of different parameters can be entered and saved. For example, plant work schedules may be different for summer and winter and permit different equipment operation.

Status Reports

English-like operator commands simplify system setup and operation. Reports can be requested for up-to-date status, or scheduled for later printout.

Savings Potential

The Power Management System reduces demand charges by reducing power demand peaks. Reduction of overall energy consumption from demand control and load cycling will also result in savings. Using the utility company rate schedule and a copy of a recent power bill, an estimate of savings can be made.

THE SYSTEM COMPONENTS

The system can be implemented as a dedicated power management application or expanded to include other applications. Pre-packaged standard configurations are available which include the minimum required system software and hardware. Only the Industrial I/O assemblies need be added for a complete power management system. The power management software is an applications program especially designed to operate with the RSX-11 Real-Time Executive software product on a PDP-11 system. RSX-11 controls system activity and acts as taskmaster for the various system resources, including memory, processor and peripherals.

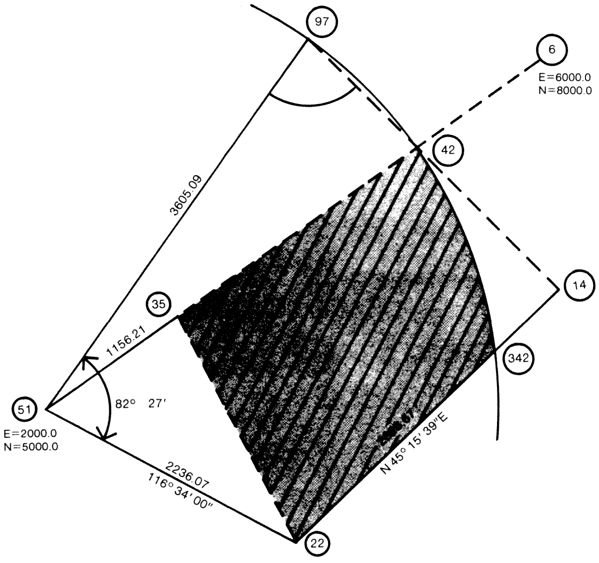
Industrial Hardware

The Industrial Subsystem provides the electrical interface for interconnecting the Power Management System to both the watthour demand metering equipment (typically supplied by the power company) and to the controllable plant equipment. The metering equipment is isolated with optically coupled isolation circuits. Relay contacts insure electrical isolation from the controllable plant equipment. Solid state interconnecting circuits are optionally available for use with low-level DC control circuits.

PDP-11
COGO-11

JULY 1976

COGO-11 is an easy to use, but powerful, programming system used to solve coordinate geometry problems. While there is virtually no limitation to the kinds of geometric problems that may be solved with COGO-11, typical applications center on land surveying, sub-division work, highway design and construction layout.



FEATURES

- Engineering terminology
- Program structure follows engineering methods
- Data points and solutions may be saved for later use
- May be used interactively or in batch mode
- Written in FORTRAN IV so it may be easily modified

DESCRIPTION

COGO-11 is designed for use by civil engineers with little or no computer experience. Using it, coordinate geometry problems can be stated in normal engineering terminology. Working from as little as a sketch of his problem, the engineer writes a description of the problem as though he were solving it by hand, except that each of the problem-solving steps is stated as a COGO-11 command.

The summary of these commands becomes the program for solving the problem. Data points are stored in tables and can be recalled and used directly as required without having to re-enter data for each calculation. As a result, previous calculations can be built upon to solve very complex problems.

COGO-11 may be used in the batch mode or interactively by one or several users. In the interactive mode the engineer works directly at a terminal and types in his commands, either in a complete or abbreviated form. Output appears at the terminal directly after each command is typed and, if a command has been typed incorrectly an error message is returned to the terminal. The engineer may correct his mistake immediately and continue with the problem.

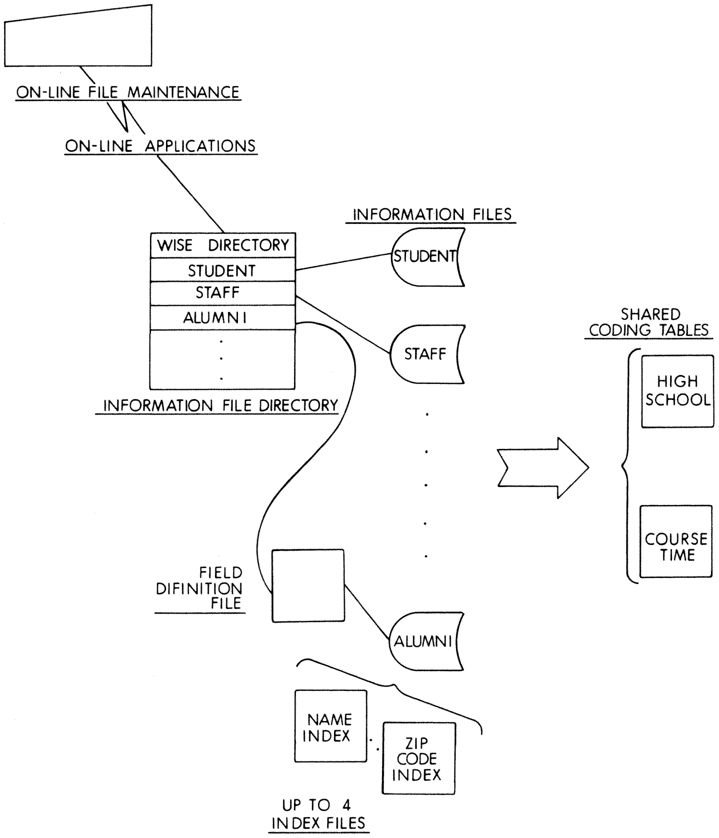
In the batch mode, the engineer writes down the entire command sequence for solving his problem(s), then has this command sequence keypunched. This job, or several of them, is processed sequentially with no interaction with the user. If an error occurs, a diagnostic message is printed and the remaining commands are checked for validity, but no more processing is done on that job. An error, however, will not affect other jobs in the same batch stream.

Modification of COGO-11 is easily accomplished. Because COGO-11 has been written modularly and entirely in FORTRAN IV, existing functions can be modified and new ones can be added with a modest knowledge of operating system software and without the use of an assembly language.

PDP-11
WISE

JULY 1976

WISE is an integrated, easily expandable data management system designed within an educational timesharing framework. WISE provides the means necessary to handle the data processing requirements of today's colleges and universities while also allowing simultaneous instructional timesharing.



FEATURES

Flexibility

- The choice of data items to be maintained and their size and format is determined by the individual institution.
- New records, additional information within a record, and even entirely new files may be added to the system without the need to reprogram existing applications
- Data base applications may be extended, added or modified as desired.

Efficiency

- Maintains file security and data accuracy
- Handles variable-length data within a record
- Eliminates redundant data storage through cross-referencing (linking) of records within a file or among different files
- Accesses the data base using indexed sequential access techniques
- Performs file maintenance (add, modify, delete) for for all files using a single update program
- Operates applications programs concurrently on any file in the system

DESCRIPTION

WISE consists of three categories of administrative application programs plus utilities which run under RSTS/E (Resource Sharing Timesharing System/ Extended), DIGITAL's most popular PDP-11 based timesharing system. These programs provide each administrative user with a direct means of creating and maintaining his own information base; provide application programs to solve his most common information-handling problems; and allow development of additional applications which utilize this information with a minimum of programming effort. This change from traditional data base management techniques, where data processing personnel are responsible for maintaining the information base, follows the philosophy that data reliability and accuracy increase when each user is responsible for the accuracy of his own department's data.

Briefly, the total system may be divided into four components: the Data Management System, the Student Information System, the Admissions Information System, and the Alumni Information System.

Data Management System

- Multi-key indexing
- On-line file inquiry and updating
- Easy-to-use report generator

Student Information System

- On-line file maintenance
- On-line course registration

  Class lists

  Student schedules
- On-line grade entry
- Grade reports

Admissions Information System

- On-line file maintenance

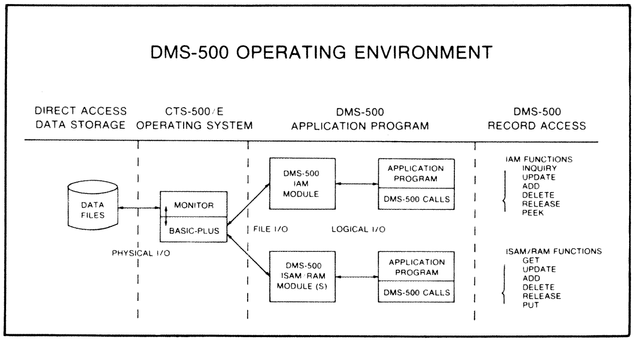
Alumni Information System

- On-line alumni data file maintenance
- Mailing labels
- Donor analyses

PDP-11
Data Management Service

JULY 1976

DMS-500 is a series of software modules under the CTS-500/E (Commercial Timesharing-Extended) of the DEC DATASYSTEMS 500 series. It provides general methods for organizing and processing logical data records, which may be stored in indexed random, indexed sequential, and relative access file structures. Interactive utility routines define, allocate, and organize data in the file structures. Control of direct access device input/output, data buffering, and error handling is accomplished by selected software modules appended to user BASIC-PLUS programs. Application programs access stored data by identifying specific logical records to be added, retrieved, modified, or deleted from these file structures. System-level routines are used to control multiple, concurrently operating user programs which access the same data file structures.


DMS-500 OPERATING ENVIRONMENT

FEATURES

- Data file management services for BASIC-PLUS application programs.
- Indexed random, indexed sequential, and relative access mode data file structures.
- Simplified input/output user programming with software control of logical record blocking and deblocking.
- Comprehensive set of record management functions to retrieve, modify, add, and delete data records.
- Random, sequential, and dynamic random sequential record retrieval access in indexed sequential and relative mode files.
- Automatic retrieval and update access control for multiple programs concurrently interacting with the same data files.
- System control of distributed free space for data record addition and deletion.
- Common status and error control functions.
- Range of supporting utility programs to verify, reallocate, and reorganize active data file structures.
- Disk sort utility for sequencing large, multi-volume files.

DESCRIPTION

The major components of DMS-500 include:

- Indexed Access Method (IAM);
- Indexed Sequential/Relative Access Method (ISAM/RAM); and
- Extended Disk Sort Facility.

This software provides efficient, easily used data file handling capabilities to meet the requirements of DEC DATASYSTEMS 500 installations. The functions, structures, and access methods included have been chosen from well established and proven file management techniques. The design of DMS-500 is oriented toward ensuring the integrity of data stored and accessed in interactive usage modes.

IAM and ISAM/RAM are two separately structured groups of software modules. Both groups are made available to users in source level language modules appended to BASIC-PLUS application programs. Also provided are utility programs to define, allocate, and initialize data file structures.

An extended disk sort facility sequences data maintained in any DMS-500 file format or equivalent CTS-500/E input/output file containing fixed length logical records blocked within 512 byte physical records. A multi-volume sort input/output capability is included.

DMS-500 Usage

The CTS-500/E system user can select the DMS-500 access method most appropriate to the specific data file storage and retrieval requirements of a given application. IAM file structures are most appropriate for access to data records in random order by a symbolic key value. The IAM facility allows a small initial file to be created and then updated continuously or periodically with a relatively large number of new data records.

An example of the use of an IAM file structure would be the creation and maintenance of a customer information file where individual records are stored and retrieved in a random sequence by a name field value.

An ISAM data file structure can be used when both random and sequential access modes to data records are required. The ISAM capability to sequentially store and retrieve a group of ordered data records facilitates the processing of files organized by the value of a key field; e.g., a customer order number or a part number. Certain applications are most efficiently processed by randomly accessing a specific record, then sequentially retrieving a group of related records. The ISAM sequential access mode may also be used for large volume updates in an off-line batch processing mode. The ISAM duplicate, generic, and approximate key handling capabilities are relevant in applications where a unique key is not necessarily associated with each data record to be stored (e.g., an employee medical claim file containing multiple data records with the same employee name and number).

By eliminating the index file and utilizing a relative record number, the RAM facility provides the most efficient direct access to data records. In most cases, this is a good method for rapidly retrieving records since only one disk read must be performed to find the correct record. Sequential access is also permitted. Full DMS-500 data record input/output buffering, concurrent access control, and error handling facilities are provided with the RAM structure and access method.

The extended disk sort facility is a valuable tool for the user in organizing data records for a particular use. It is designed to facilitate data processing oriented applications programmed in BASIC-PLUS.

In summary, the combined facilities of the DMS-500 data file structures and access methods allow the CTS-500/E system user to select a processing capability best suited to the individual requirements of a given application.

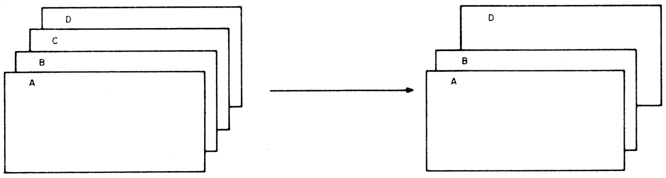
DMS-500 ACCESS METHOD FEATURE SUMMARY

	IAM	ISAM	RAM
SYMBOLIC KEY ACCESS	Yes	Yes	No
DUPLICATE KEYS	No	Yes	No
GENERIC AND APPROXIMATE KEYS	No	Yes	No
RELATIVE RECORD NUMBER ACCESS	No	Yes	Yes
FIXED LENGTH RECORDS	Yes	Yes	Yes
MAXIMUM RECORD SIZE	511 Bytes	508 Bytes	508 Bytes
RANDOM ACCESS BY KEY	Yes	Yes	Yes
SEQUENTIAL ACCESS	No	Yes	Yes
DYNAMIC RANDOM AND SEQUENTIAL	No	Yes	Yes
DISTRIBUTED FREE SPACE	Index File	Data Files	Data Files
KEY FIELD IN DATA RECORD	Yes	Yes	No
ORIGINAL INPUT DATA FILE REQUIRED	No	Yes	No
IMMEDIATE RETURN OF DELETED DATA SPACE	No	Yes	Yes
MULTI-VOLUME DATA FILE SUPPORT	No	Yes	Yes

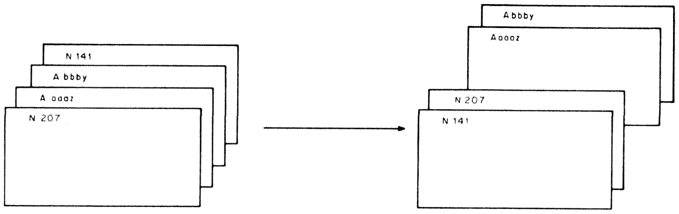
PDP-11
SORT-11

JULY 1976

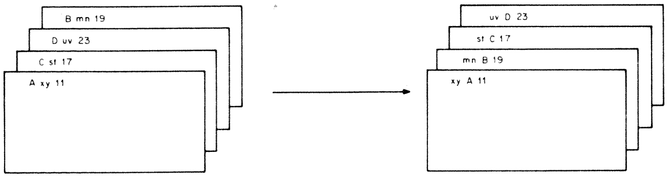
SORT-11 is a utility which enables the user to reorder data from any ASCII input file into a new file, in a sequence based upon control or key fields with the input data records. It is supported by RSTS/E, RSX-11D, RSX-11M and IAS. SORT will accept as input a file from any device supported by the host operating system, and output the reordered file to any acceptable output device.



RECORD SELECTION



INPUT FORMAT VARIATION



OUTPUT FORMAT VARIATION

FEATURES

- Rearrangement, deletion and reformatting of records in a mass-storage file
- Selection of a sorting process and input device that best suits the processing environment
- Creation of new file indices to access a large data base
- Ability to combine several sorting jobs in a batch stream
- Definition of sort specification files in a simple fixed format

DESCRIPTION

SORT-11 provides four sorting techniques:

Record Sort (SORTR)

Produces a reordered file by manipulating all records in their entirety. It can be used on any acceptable input device; i.e., cards, magtape, DECtape or disk. SORTR is able to process variable length records.

Tag Sort (SORTT)

Produces a reordered file that conserves temporary storage by manipulating only the key position of each record. It then randomly reaccesses the input file to write out a newly sequenced output file.

ADDress ROUTing Sort (SORTA)

Produces an address file enabling the user to maintain only one data file but to have access to it through several different addressing files appropriate to alternative sequential processing needs. The programmer can later use the output file as an index to access the data in the desired sequence; e.g., a customer master file may be referenced by customer-number index or sales-territory index for different reports. SORTA is the fastest of the four methods.

Index Sort (SORTI)

Provides an indexing capability for sequential and direct accessing from a random-data file via the Index Access Method (IAM). SORT produces a separate index file containing the key field of each data record and a pointer to its location in the data file.

File Format

The simple specification file format is compatible with other disk sorts. It also performs input-record selection based on user-defined conditions. In addition, the input records may vary in format and the output records may be restructured. If the user's file does not require selection or restructuring, the user may control SORT using only a command string. Input can be entered interactively through the console or prepared off-line for future batch processing.

File Organization

Two types of file organization are handled by the sort: sequential and relative.

Sequential files are organized so that each record in the file except the first has a unique predecessor record, and each record except the last has a unique successor record. These predecessor-successor relationships are established by the order of write statements when the file is created. Once established, the predecessor-successor relationships do not change except in the case where records are added to the end of the file.

Relative file organization is permitted only on directory devices. A relative file consists of records which are identified by relative record numbers. The file may be thought of as being composed of a serial string of areas, each capable of holding a logical record. Each of these areas is denominated by a relative record number. Records are stored and retrieved based on this number. For example, the tenth record is the one addressed by relative record number 10 and is in the tenth record area, whether or not records have been written in the first through the ninth record areas.