DataBoard

070-8025

Pascal, OS8MT

PASCAL working under the operating system OS.8MT, is a powerful language equally suitable for both technical and administrative applications i.e. calculation, process control (the code can be placed in PROM), data management, etc. PASCAL follows the UCSD recommendations.

OS.8MT — ORIENTATION

Multi-user Several users working at the same time is a function in OS.8MT. The users can work with both program development and application-execution at the same time.

Multitasking Supervised by OS.8MT several programs can run at the same time. More than one task can be started from the same terminal device. This makes it possible to, for example, control a machine at the same time as running a data base.

Real time All execution is done in a real time environment, with the possibilities of both time-sharing and giving tasks different priorities. A calender and a clock are included in OS.8MT. Tasks can also be time-synchronised.

INPUT/OUTPUT

Peripherals All I/O is totally independent of the physical characteristics of each device. Any combination of devices can be used as every type is controlled by driver routines contained in OS.8MT.
The most common types of peripherals are:

Terminal devices:	Intelligent and non-intelligent.
Printers:	Character and line oriented.
Mass-storage:	Floppy discs, winchesters, hard discs and magnetic tape.
Modems:	Syncronous and asyncronous
Process control:	Analogous and digital inputs and outputs.

Filing system Totally independent of a chosen combination of mass-storage devices i.e. floppy disc, winchester, etc. Every disc has its own name, independant of the drives. One master file directory and any number of element file directories exist on each volume.
Files and space for files are allocated and deallocated in a totally dynamic way.

Files Every file-type can be reached from PASCAL. The record length can be fixed or variable between 1 and 65535 characters and the files can consist of 1 to 16777215 records. The files can be contiguous with fixed size or indexed with variable size. Three access methods exist: on byte, record or sector level.

Data bases Data bases are managed by an ISAM program, which permits several users to work with one or more data bases at the same time. Index-sequential access method with fixed record length is used. Up to 10 indices are allowed in the same data base. While every record can be read, protection is made centrally. As all indices and keys are updated when writing, no sorting is needed. The data bases can be moved to different mass-storage devices without any limitations.

Networks Up to 32 systems can be used together. Both local and central peripherals can be used.

PROGRAM DEVELOPMENT

Source code PASCAL source code is prepared with a powerful editor in ASCII-code.

P-code The source code can be compiled into pseudo-code which is executed with the help of an interpreter.

P-code linking Modules can be linked together on P-code level.

Object code The P-code can be compiled into a fast and compact object code.

Multi-language Tasks can include modules written in different compiling languages.

PROMation Pascal programs (not including segmented procedures and functions) can be placed in PROM, either by promming the task-code or the P-code and interpreter. This option makes it possible to optimize the size and speed of the program.

Error handling As every type of error can be handled without stopping the program, very stable application programs can be written.

Debugging Debugging is made very easy by using TRACE and single step (CTRL-S) functions.

PROGRAMMING TOOLS

System programs A number of programs are available for compiling, executing, deleting, dumping, linking, interpretating and printing different kinds of files. It is also possible to create P-code libraries and cross reference lists.

Command streams In CSS (Command String Supervisor)-mode the user can instruct the interpreter to execute powerful CSS-commands in a user-specified file.

Control statements Defines the actions that a program is to perform on its defined data.

Data types Integers, floating points, booleans and characters. User defined, structured, pointer and file-data types are available.

Procedures and functions Accepts user written procedures and functions. A great number of pre-prepared procedures and functions are also available.

Segmentation Procedures and functions can be segmented.

ADVANCED PROGRAMMING

ISAM With the aid of special instructions, advanced, multi-user data base systems can be built.

Supervisor calls Every system function can be reached with supervisor-calls. Parameters are transferred using standardised parameter-blocks.

External programs Programs written in arbitrarily chosen languages can be loaded, started and manipulated from a PASCAL-program.

ASSEMBLER-routines User written ASSEMBLER-routines, loaded by OS8.MT, can be called from a PASCAL-program.

DATA INDUSTRIER AB

Box 2029, 18302 TÄBY. Tel 08-768 06 60. Telex 10978.

SYSTEM PROGRAMS

PASCAL	To execute pre-compiled user programs.
PASSYS	To execute the following system programs.
PASCOMP	To compile text files to P-code.
PASLINK	To link pre-compiled user programs or library modules.
PASOBJ	To convert P-code to object-code.
PASLIB	To create and update a P-code library.
PASCROSS	To create a cross reference list.
PASDEL	To delete files.
PASPRINT	To print text files.
PASDUMP	To dump files.

CSS-MODE

$-commands	Define the current CSS-mode and are interpreted by the CSS-processor.
$$-commands	Control the flow and execution of a CSS-command file. They can also be used to create permanent files and to operate on taskfiles. The $$-commands can be used in interactive mode.

CONTROL STATEMENTS

BEGIN...END	A compounded sequence of statements.
Assignments	(), NOT, ★, /, DIV, MOD, AND, +, —, OR, =, < >, <, >, IN.
WHILE...DO	Executes a statement or compound statement repeatedly until the condition being tested becomes false.
REPEAT... UNTIL	Executes a statement or a list of statements repeatedly until a desired condition is met.
FOR...TO (or DOWNTO)... DO	Executes a simple or compound statement for a predetermined number of times.
IF...THEN... ELSE	Evaluates an expression and choses between two possible actions.
CASE...OF	Transfers control to one of several statement labels depending on the variable's name.
GOTO	Unconditionally transfers control from one portion of the program to another.

STANDARD DATA TYPES

Integers	Between —32 767 and 32 767.
Floating points	A precision of 6 digits.
BOOLEAN	Logical variables.
Characters	Any symbol from the ASCII-set.

USER DEFINED DATA TYPES

TYPE	A set of constant values which a variable may assume.
SET OF	A collection of values which are all of the same type.

STRUCTURED DATA TYPES

ARRAY...OF	A fixed number of components referenced by the same name.
PACKED ARRAY...OF	An array with minimal memory waste.
STRING	A packed array of characters.
RECORD	A type with a user defined structure that incorporates several components.
PACKED RECORD	A record with minimal memory waste.
WITH...DO	References a record once for one or more field accesses.

POINTER AND FILE DATA TYPES

↑	Declaration of a pointer variable.
TYPE...FILE OF	Defines a record file.
TYPE...TEXT	Defines a text file.
TYPE...FILE	Defines a physical file.
TYPE-ISAM-FILE	Defines an ISAM file.

PROCEDURES AND FUNCTIONS

PROCEDURE	Affects programming situations.
FUNCTION	Returns a value of a specified kind.
FORWARD	Allows functions and procedures to be accessed before they are defined.

STRING INTRINSICS

CONCAT	Concatinates one or more strings together.
COPY	Returns a string copied from another string.
DELETE	Removes characters from a string.
INSERT	Inserts one string into another.
LENGTH	Returns the length of a given string.
POS	Returns the position of the first occurrence of a character sequence within a string.

INPUT/OUTPUT INTRINSICS

BLOCKREAD	Transfers blocks of data from a file to an array and returns the byte count.
BLOCK-WRITE	Transfers blocks of data from an array to a file and returns the byte count.
CLOSE	Closes and deletes files.
EOF	Returns a true value when end of file is reached.
EOLN	Returns a true value when end of line is reached.
GET	Reads data from a file.
INP	Reads data from a port.
IORESULT	Holds the result codes of the I/O operations.
OUT	Writes a value to a port.
PAGE	Sends a page carriage control to a text file.
PUT	Writes data to a file.
READ	Reads data from a file or the keyboard.
READLN	Reads a line of input until the first position of the next line.
RESET	Prepares a file to be read.
REWRITE	Prepares a file to be written.
SEEK	Changes the order in which data is accessed from a file.
WRITE	Outputs variables and strings.
WRITELN	Outputs a line and carrage return.

CHARACTER ARRAY MANIPULATION INTRINSICS

FILLCHAR	Places a character into a array a specified number of times.
MOVELEFT	Moves characters from the left end of one string to the left of another.
MOVERIGHT	As MOVELEFT but in the opposite direction.
SCAN	Finds the distance between a character and a starting point.

MATHEMATICAL FUNCTIONS

ABS	The absolute value of a value.
ARCTAN	The arctangent of a value.
COS	The COS of a value.
EXP	"e" powered to a value.
LN	The natural logarithm of a value.
LOG	The common logarithm of a value.
MOD	The remainder of an integer division of the arguments.
ODD	The BOOLEAN value specifying wether an integer is odd.
ROUND	The integer representation of a REAL number.
SIN	The sine of a value.
SQR	The square of a number.
SQRT	The square root of a number.
TRUNC	The integer representation of the decimal portion of a REAL number.

MISCELLANEOUS ROUTINES

DATE Gives the current date.

DISPOSE Returns allocated memory.

EOLCHR The integer value representing a termination.

EXIT An orderly exit from a program, function or main program.

GOTOXY Sends the cursor to specified coordinates.

HALT Terminates the execution of a PASCAL program.

MARK Sets a pointer to the current top-of-heap of the available free memory.

NEW Allocates free memory from the heap.

OPTION Returns the starting switches.

RELEASE Sets the pointer to the memory location of the pointer variable.

SIZE OF Returns the number of bytes in memory that are assigned to an identifier.

STARTPAR Returns the characters that are written after the code-file name when a program is executed.

SVC Executes Supervisor calls.

TIME Returns the time since the system last was booted.

ISAM STATEMENTS AND PROGRAMS

CREINDEX Allocates and creates an ISAM Index file and its associated data file.

ISAM... DELETE Removes a particular data record's key from an ISAM index file.

ISAM... READ... Accesses by key or sequentially, records contained in the data file associated with an ISAM index file.

ISAM...UPDATE Replaces a specified record in the ISAM data file.

ISAM...WRITE Enters a new record into the ISAM data file.