DataBoard

070-8015

Basic II, OS8MT

BASIC II, which works under the operating system OS.8 MT, is equally suitable for both technical and administrative applications i.e. calculation process control, data management etc.

ORIENTATION

Multi-user Several users working at the same time is a function in OS8 MT. The users can work with both program development and application/ execution at the same time.

Multitasking Supervised by OS.8 MT 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.8 MT. Tasks can also be time-synchronised.

Compiling Programs are compiled from ASCII-code to an internal BASIC-code containing all the references to variables and instructions. The result is a very fast execution of programs, as all searching is eliminated.

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.8 MT.
The most common types of peripherals are:

Terminals 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. 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 BASIC. The record length can be fixed or variable between 1 and 65535 characters, while the files can consist of 1 to 16777215 records. The files can be continuos 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

Programming All programming is made in an interactive way with immediate syntax control and error monitoring. Programs are easily structured by using multi-line functions that permits both parameters and results to be transferred. To promote readability loops are tapered when listing a program.

Segmentation Programs larger than 40 Kb are segmented and later chained together. Variables are transferred between the segments.

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.

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 BASIC-program.

ASSEMBLER-routines User written ASSEMBLER-routines, loaded by OS.8 MT can be called from a BASIC-program.

PROGRAMMING

Commands The immediate actions are controlled with 14 commands.

Instructions 68 instructions, most of which can be used as commands, are available. Several instructions on each line are permitted.

Functions 46 functions including user defined functions with the possibility of local variables and recursion.

Data-types Integers, strings and floating points.

Variables Long names and multiple dimensions.

Operations Logical and arithmetical expressions with a precision of up to 126 digits.

DATA-TYPES

Integers Between −32768 and 32767 (16 bit)

Floating-points A precision of 6 digits (32 bit) or 16 digits (64 bit).

Character-strings No limitation in length.

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

VARIABLES

Names	Up to 32 characters long.
Vectors/ matrix	Every data-type can be used with any number of dimensions and chosen intervals for each dimension.

OPERATIONS

Arithmetic	Exponentiation (**), Multiplication (*), Division (/), Additions (+) and Subtraktion (–).
Logical	AND, EQV, IMP, NOT, OR, XOR.
Relations	–, <>, >, <, >=, <=.
String-arithmetics	A maximum size of 125 characters.

COMMANDS

RUN	Starts a program.
CONT	Continues program execution.
NEW or SCR	Clears storage.
AUTO	Automatic line numbering.
RENUMBER	Changes the numbering of the lines.
ERASE	Erases one or more lines.
ED	Starts editing of a line.
LOAD	Load a program into the working storage of the computer.
MERGE	Merges program files.
LIST	Lists the program.
SAVE	Saves the program in compiled form.
UNSAVE	Erases a program from a disc.

DATA-INSTRUCTIONS

LET	Assigns a value to a variable.
READ	Assigns a value from a DATA statement to a variable.
DATA	Assigns values to variables (used with read).
RESTORE	Resets the DATA-pointer.
DIM	Allocates space for strings and vectors.
COMMON	Declares the variables, whose values are to be transferred to another program (CHAIN).
SINGLE	Sets floating-point numbers to single precision (6 digits)
DOUBLE	Sets floating point numbers to double precision (16 digits)
EXTEND	Allows extended variable names to be used.
NO EXTEND	Terminates work in EXTEND-mode.
INTEGER	Sets the input and lists format to integer for variables.
FLOAT	Sets the input and list format to float for variables.
OPTION BASE	Denotes the lowest vector index value.
RANDOMIZE	Sets a start value for the RND function.
SET TIME	Sets date and time.

PROGRAM-CONTROL STATEMENTS

PAUSE	Pauses the execution of a program.
STOP	Stops the execution of a program.
END	Terminates a program.
BYE	Exits BASIC and gives control to the operating system.
CHAIN	Loads and executes a program from the currently executing program.
DEF FN	Defines a multiple line function.
RETURN	Returns from a subroutine or multiple line function.
FNEND	Terminates a multiple line function.
FOR... TO... STEP	Starts a program loop.
NEXT	Terminates a program loop.
GOSUB	Jump to a subroutine.
ON... GOSUB	Conditional jump to one of several subroutines.
GOTO	Jump to a given line number.
ON... GOTO	Conditional jump to one of several lines.
IF... THEN... ELSE	Conditional execution.
WHILE	Specifies condition for the branching out of a program loop.
WEND	Terminates loop.
ON ERROR GOTO	Specifies a routine for error handling.
RESUME	Return from error handling.
ON... RESUME	Conditional return from error handling.
ON... RESTORE	Conditional resetting of the DATA-pointer.
TRACE	Prints the line number of the executed program lines.
NO TRACE	Terminates execution in TRACE-mode.

I/O INSTRUCTIONS

DIGITS	Specifies the number of digits to be printed.
OPTION EUROPE	Specifies the format when using PRINT USING.
INPUT	Fetches data for the current program.
INPUT LINE	Accepts a line of characters.
PRINT	Prints data in ASCII format.
PRINT USING	Prints formatted data.
GET	Reads one or more characters from the key-board.
PUT	Writes a string variable in binary format.
NAME	Changes the name of a file.
KILL	Deletes a file.
OPEN	Opens a file.
PREPARE	Creates and opens a file.
CLOSE	Closes one or more files.

MATHEMATICAL FUNCTIONS

ABS(x)	The absolute value of x.
FIX(x)	The truncated value of x.
INT(x)	The greatest integer <=x.
MOD (x,y)	The remainder of an integer division of the arguments.
PI	The constant 3.14159
RND	A random value between 0 and 0.999999.
SGN(x)	The sign of x.
SQR(x)	The square root of x.
EXP(x)	The value e**x.
LOG(x)	The natural logarithm of x.
LOG10(x)	The common logarithm of x.
ATN(x)	The arctangent of x in radians.
COS(x)	The cosine of x in radians.
TAN(x)	The tangent of x.
HEX$(x)	The hexadecimal string representation of x.
OCT$ (x)	The octal string representation of x.

STRING FUNCTIONS

ADD $	Adds two numerical strings.
DIV $	Divides two numerical strings.
MUL $	Multiplicates two numerical strings.
SUB $	Subtracts two numerical strings.
COMP %	Compares two numerical strings.
NUM $	Converts a value to a numerical string.
VAL	Converts a numerical string to a value.
ASCII	The ASCII value of the first character of a string.

INSTR	Seaches for the occurrence of one string in another string.
LEN	The length of a string.
MID $	Returns a part of a string.
LEFT $	Returns a part of a string.
RIGHT $	Returns a part of a string.
CHR $	A string of ASCII values corresponding to the arguments.
SPACE	A string containing spaces.
STRING	A string of ASCII characters.

REMAINING INSTRUCTIONS AND FUNCTIONS

CUR	Positions the cursor on the screen.
ERRCODE	The latest generated error code.
FN	Calls a user defined function.
REM or !	Inserts comments in the program.
SLEEP	Delays the execution for a number of seconds.
TAB	Tabulates to a chosen position.
TIME	A string of the current date and time.

ADVANCED INSTRUCTIONS AND FUNCTIONS

CALL	Calls an assembler program.
CVT	Converts integers and floats to and from binary numbers.
ISAM OPEN	Opens a data base for read and write.
ISAM READ	Reads a record from a data base.
ISAM WRITE	Writes a record and updates the indices.
ISAM UPDATE	Changes a record and updates the indices.
ISAM DELETE	Deletes a post and updates the indices.
OPEN... MODE	Opens a file for access on byte or record level.
PREPARE... MODE	Creates and opens a file.
OUT	Writes data to a chosen port.
INP	Reads data from a chosen port.
PEEK	Reads a byte from the primary memory.
PEEK 2	Reads two bytes from the primary memory.
POKE	Loads a byte in the primary memory.
SVC	A call requesting a system resource controlled by OS.8MT.
SWAP %	Returns an integer with the first and second bytes transposed.
SYS	Returns system information like program size etc.
VAROOT	The address of the description of a variable.
VARPTR	The address of a variable.