

NEW DIRECTORY DCR/1

Koen Van de Perre

5 août 1986

POROM	EQU	40H	
SPSAV	EQU	5BH	
LCasNm	EQU	28H	;Length Cassette-name
BASRUN	EQU	87H	;BASIC-command 'RUN'
TOPSCR	EQU	8AH	
ENDSCR	EQU	8CH	
MACC	EQU	0D5H	
;			
HSIZE	EQU	100H	;* Initialisation
SYSBOT	EQU	100H	;*
LOPVAR	EQU	104H	;*
STKGOS	EQU	113H	;*
RDIPF	EQU	117H	;*
RUNFL	EQU	118H	;Used if running an MLP-file
ERSFL	EQU	122H	;*
CONFL	EQU	126H	;*
STACK	EQU	127H	;*
;			
FILES	EQU	1B0H	;Save number of skip files
EBUF	EQU	13EH	;Tempory save area for filename
;			
;			
LBlk1	EQU	EBUF	;Temporary save area to cal-
LBlk2	EQU	LBlk1+4H	; culate the length of the
LBlk3	EQU	LBlk2+4H	; different blocks of a file
LUSER	EQU	LBlk3+4H	;Calculate length of directory
;			
IMPTAB	EQU	275H	;* Initialisation
PHBGN	EQU	29BH	;*
PHSIZE	EQU	29DH	;*
SHLK	EQU	2C3H	;Shift-Locked key
;			
WOPEN	EQU	2C5H	;* Used by saving Directory
WBLOK	EQU	2C8H	;*
WCLOS	EQU	2CBH	;*
;			

ROPEN	EQU	2CEH	;* Used by reading if loading
RBLOK	EQU	2D1H	;* a MLP file was request
RCLOS	EQU	2D4H	;*
;			
HBGN	EQU	2ECH	;* Initialisation
;			
ORIG1	EQU	0AA55H	
LORG	EQU	8FAH	;* ----- Length directory MARDAGA
BufEnd	EQU	ORIG1+LORG	;* (Keep same length)
;			
LINE3	EQU	0B577H	;* Line control bytes
LINE4	EQU	0B5FDH	;*
LINE19	EQU	0BDD7H	;*
LINE22	EQU	0BF69H	;*
LINE23	EQU	0BFEFH	;*
;			
RINIT	EQU	0C7A0H	;Goto BASIC monitor
MSGIN\$	EQU	0C7D3H	;"BASIC V1.1"
RUNBAS	EQU	0C892H	;Run suspended command(s)
SELB0	EQU	0CEE4H	;Select bank 0
SHINIT	EQU	0D101H	
CMPD\$H	EQU	0D121H	;Compaire 2 strings
;			; exit Z=0 if not ident
ERROR	EQU	0D2ADH	;Run loading error
RBUEX	EQU	0D387H	;Calculate checksum
;			; if reading a byte
KBINIT	EQU	0D560H	;* Initialisation
BYTCUR	EQU	0D7ECH	;Write byte on cursor pos.
KBEI	EQU	0D988H	;* Initialisation
CLKEI	EQU	0D9DBH	;*
PMSGR	EQU	0DAFFH	;Print string pointed by
;			; 2 next bytes
PSTR	EQU	0DB32H	;Print string pointed by H.L
;			; on exit H,L points after string
PINT	EQU	0DB53H	;Print a INT number of
;			; bytes in MACC
OTDAT\$	EQU	0DC33H	;'OUT OF DATA'
INLINX	EQU	0DD1FH	;Input a text line
OUTC	EQU	0DD60H	;Output a character
DADM	EQU	0DE39H	;Skip a string
;			; (H,L points after string)
FILL	EQU	0DE7CH	;Fill D,E till H,L with data A
RNEW	EQU	0DEB8H	;Run Basic cmmmand 'NEW'
;			
KEYTU	EQU	0E8C5H	;ASCII-table ROM bank 3
L3E159	EQU	0EA04H	;Return after 'GO'
Z2Res	EQU	0ECC1H	;Part of Z3 reset
GO2	EQU	0ED99H	;Part of G-commando
;			
REWIND	EQU	0F012H	;Rewind files (see FILES)
SKIP	EQU	0F015H	;Idem skip
CMPN2	EQU	0F29CH	;Part of check a fileblok
XTHL4	EQU	0F2D3H	;Delay routine (9 msec)
DRCLOS	EQU	0F31FH	;Stop DCR-motor
DOPEN	EQU	0F348H	;Start DCR-motor
DSTART	EQU	0F3D5H	;Read a byte from tape
RDADR	EQU	0F41DH	;Read length blok
	EQU	0F43DH	;Read recognition byte
RDHDR	EQU	0F458H	;Read header + filetype
STTOP	EQU	0F900H	

PORO ;	EQU	0FD06H	
	ORG	ORIG1	
; INIT	LXI H	HBBN	;Set BASIC pointers
	SHLD	PHBBN	
	LXI H	HBBN	
	SHLD	PHBBN	
	CALL	RNEW	
;	LXI H	KEYTU	;Set keyboard pointers to
;			;default value (ASCII-table)
	CALL	KBINIT	
	CALL	SHINIT	;String handler
	CALL	KBEI	;Enable keyboards interrupts
	CALL	CLKEI	;Enable clock interrupt
;			
	MVI A	10H	;Set to Integer-type
	LXI D	IMPTAB	
	LXI H	IMPTAB+1BH	;#1B = 27 (Aantal chars)
	CALL	FILL	;Load bank with ident. data
;			
	EI		
;			
	CALL	SELB0	;Set to ROM bank 0
	NOP		
	NOP		
;			
	LXI H	STTOP	;Set stack pointer
	SHLD	STACK	
	SPHL		
;			
	XRA A		
	STA	CONFL	;Set flag no suspended prog.
	MOV H,A		
	MOV L,A		
	SHLD	SYSBOT	;no current line
	SHLD	LOPVAR	;no variables
	SHLD	STKGOS	;Stack level gosub's
	STA	ERSFL	;No encoding a stored line
;			
	LXI H	RDIPF	;Flag set while running input
	MOV M,A		
	DCR A		;A=0FFH
	INX H		
	MOV M,A		;Flag set while running prog.
;			
;	***** Start from BASIC or UTILITY *****		
;			
BEGIN	MVI A	0FFH	;Mode 0
	RST 5		
	DB	18H	
;			
	LXI H	TXTCOL	;Set text colors
	RST 5		
	DB	6H	
;			
	CALL	PMSGR	;Print CHR\$(12) +
	DW	Title\$; programm title
	MVI A	6AH	;Enlarged letters

	STA	LINE23	; programmtitle
	STA	LINE22	; and cassette name
;			
	LXI H	300H	;Cursor 0,3
	RST 5		
	DB	9H	
;			
	MVI A	0DFH	
	STA	LINE3-1H	;Colorbyte
	CALL	PMSGGR	
	DW	MENU\$;Print menu
;			
	LXI H	LINE19	;Screen division
	SHLD	TOPSCR	
	LXI H	LINE4	
	SHLD	ENDSCR	
;			
StrtPr	MVI A	0CH	;Print CHR\$(12)
	RST 5		
	DB	3H	
;			
;			Print cassettename + filenames
;			
	LXI H	1100H	;Cursor 0,17
	RST 5		
	DB	9H	
;			
	LXI H	CasNm\$	
	CALL	PSTR	;Print cassette name
;			
	LXI H	AanNm\$;Copy number of names
	RST 4		; into MACC
	DB	0CH	
	CALL	PINT	
;			
	CALL	PMSGGR	
	DW	Files\$	
;			
	LXI H	Bytes	;Copy total used Bytes
	RST 4		; into MACC
	DB	0CH	
	CALL	PINT	
;			
	CALL	PMSGGR	
	DW	Bytes\$	
;			
	LXI H	MACC	;Set MACC = 0
	RST 4		
	DB	69H	
;			
	LXI H	AanNm\$+3H	
	MOV A,M		;Number of names in A
	STA	FILES	;If accidently l or L pushed
;			; when buffer empty
	ORA A		;Is number = 0 ?
	JNZ	PNames	;No, print names
;			
EMPTY	PUSH H		; Save name-adres
	CALL	PMSGGR	;Print 'OUT OF DATA'
	DW	OTDAT\$	

	POP H			
	JMP	SELECT		
;				
PNames	LXI H	BgnNm\$;Adres begin names in HL
	MVI C	0H		;Consecutive number in C
;				
PrName	INX H			;Calculate length-adres
	INX H			
	MOV A,C			;Consecutive number in A
	STA	FILES		;Store it for skip n files
	INR A			
	STA	MACC+3H		;And prepare for Print number
	CPI	0AH		;Print a space if number<10
	CC	PSpace		
;				
	MOV A,M			;A=length filename
	ORA A			
	JZ	StrtPr		;If no next name
;				
	PUSH H			;Save length-adres
;				; current name
	CALL	PINT		;Print number
	CALL	PSpace		
	POP H			;Get length-adres
	PUSH H			; save it again far
;				; loading or printing
;				
	DCX H			;Set number of bytes
	MOV A,M			; into MACC
	STA	MACC+3H		
	DCX H			
	MOV A,M			
	STA	MACC+2H		
	CALL	PINT		
	MVI A	9H		; (tabulator)
	CALL	OUTC		
	LXI H	MACC		;Clear MACC
	RST 4			
	DB	69H		
;				
	RST 5			;Ask cursor position
	DB	0CH		
	MVI L	0BH		;Set X-pos on 11
	RST 5			;Set new cursor pos.
	DB	9H		
;				
	POP H			;Get length-adres
	SHLD	NmPtr		;Save begin name for LdFile
	CALL	PSTR		;Print filename (on entry
;				; HL points to string - on
;				; exit HL points after string)
	INR C			;Update number
;				
SELECT	XRA A			
	STA	SHLK		;Set shift unlocked
	RST 1			;Get character
	DB	15H		
;				
	JC	BASIC		
	CPI	' '		

	JZ	PrName	
	CPI	'L'	
	JZ	NORUN	
	CPI	'I'	
	JZ	LDRUN	
	CPI	'S'	
	JZ	SkFile	
	CPI	'U'	
	JZ	UPDATE	
	JMP	SELECT	
	;		
	;***** Set no RUN pointers *****		
	;		
NORUN	XRA A		;Set no BASIC-run
	STA	BasRN	; if reading a BASIC progr.
	LXI H	L3E159	;Set Return to Utility
	SHLD	RNMLP2+1H	
	JMP	LdFile	
	;		
	;***** Set RUN pointers *****		
	;		
LDRUN	MVI A	BASRUN	;Evt. reset 'RUN'- pointers
	STA	BasRN	
	LXI H	GO2	
	SHLD	RNMLP2+1H	
	;		
	;***** Load a file *****		
	; On entry: FILES = number of files to skip		
	; NmPtr= adres filename to search		
	;		
LdFile	CALL	SKIP	;Number stored in FILES
	;		
	LHLD	NmPtr	;Adres length current name in H,L
	PUSH H		;Save it for retrieve length
	MOV A,M		;Calculate orig. length
	SUI	4H	; = minus 4 bytes :
	MOV M,A		; filetype,space,+/- and CR
	;		
	LXI B	0FFH	;Any filetype,Any name
	CALL	ROPEN	;Read header
	CALL	REW1	
	;		
	RST 5		;Ask cursor position
	DB	0CH	; (H,L=y,x)
	MOV B,L		; X-pos in B
	;		
	LXI H	EBUF+1H	;Adres to store readed name
	MVI C	0H	;First character pos. + counter
GETCHR	RST 5		;Get character from line
	DB	15H	
	MOV M,A		;Store char.
	INX H		;Points to next memory location
	INR C		;Next position
	MOV A,C		
	CMP B		;All chars stored?
	JNZ	GETCHR	
	;		
	DCR C		;Set C to last pos.
	LXI H	EBUF	
	MOV M,C		;And store length

	XCHG		;Readed name in D,E
	POP H		;Get adres present name
	CALL	CMPD\$H	;Same name ?
	PUSH PSW		;Save zeroflag
	MOV A,M		;Retrieve length
	ADI	4H	; of present name
	MOV M,A		
;			
	MVI A	0FFH	;Mode 0
	RST 5		
	DB	18H	
;			
	POP PSW		;Get zeroflag
	JNZ	RINIT	;If it was not the filename
;			; to be searched for
	INX H		;Points to filetype
	MOV A,M		;Get filetype
	CPI	'0'	;Was it a BASIC-file ?
	JNZ	LdMLP	; no,
	CALL	SELB0	; yes, print
	DW	MSGIN\$;"BASIC V1.1"
	LXI B	BasLD	;Pointer Basic commands
	JMP	RUNBAS	;Execute command(s)
BasLD	DB	8BH,19H,0H	;Basic command for LOAD
BasRN	DB	87H,0H,0H	;Basic command for RUN
;			
LdMLP	CPI	'1'	
	JNZ	RINIT	;If no MLP-file
;			
	LXI H	0H	
	LXI B	3100H	;Filetype 1 + print no names
	CALL	ROPEN	;Read Headers File-type en -name
	LXI H	EBUF	
	LXI D	EBUF+2H	
	CALL	RBLOK	;Read beginadres in ENBUF
	LXI H	0F900H	;Max adres to write data into
	XCHG		
	LHLD	EBUF	
	CC	RBLOK	
	CALL	RCLOS	
	JNC	ERROR	
;			
RUNMLP	LDA	POROM	;Select bank 3
	ORI	0C0H	
	STA	POROM	
	STA	PORO	
;			
	MVI A	2H	;Z2 Reset
	ORA A		
	CALL	Z2Res	
;			
	LXI H	STTOP	
	SPHL		
	SHLD	SPSAV	
;			
	LHLD	EBUF	
RNMLP2	JMP	GO2	;Part of GO-routine
;			; Or part of RETURN AFTER 'GO'
;			

;***** Skip & goto BASIC *****

	;	SkFile	CALL	SKIP	
	;	BASIC	MVI A	0FFH	;Set mode 0
			RST 5		
			DB	18H	
			JMP	RINIT	;Goto BASIC
	;				
	;				***** UPDATE DIRECTORY *****
	;				
		UPDATE	XRA A		;Clear entire pages
			LXI D	CasNm\$+5H	; from 5th byte Cassettename
			LXI H	EndNm\$	
			CALL	FILL	
	;				
		UPD2	CALL	PMSGR	
			DW	InpNm\$	
	;				
			MVI A	0DH	;Prompt type = CR
			CALL	INLINX	;Input a text on screen
	;				
			JC	BASIC	;If 'BREAK' key pressed
	;				
			XRA A		
			MOV C,A		;First character pos. = 0
			LXI H	CasNm\$+5H	;Adres cassettename
	;				; (+ 5H = bytes for
	;				; tabulation)
		GChrLn	RST 5		
			DB	15H	;Get character from line
	;				
			MOV M,A		;Store char in memory
			INX H		; and update memory
			INR C		;Update next char-pos
			CPI	0DH	;Was it a CR ?
			JNZ	GChrLn	; No, get next character
	;				
		LastCh	MOV A,C		;Pos. last char in A
			CPI	LCasNm-3H	;Test length
			JNC	UPDATE	;If to long, Re-enter text
			ADI	4H	;Update length, CALL PSTR used:
	;				; 4th byte tabulation=0
	;				; (If CALL PMSGR used, printing
	;				; stops at ZERO-value)
			STA	CasNm\$	
	;				
			CALL	REWALL	;Set DCR to begin tape
	;				
			MVI A	1H	
			STA	FILES	
			CALL	SKIP	;Skip 1 file ('USER')
	;				
			MVI A	0CH	;Clear screen
			RST 5		
			DB	3H	
	;				
	;		Read + check :		
	;				
			LXI H	BgnNm\$;First filename adres
			SHLD	NmPtr	

	RdChck	INX H		;Calculate length-adres
		INX H		;(to compaire name in ChkEnd)
		PUSH H		; Save length-adres
	;***** Set up name-pointers,			
				clear bytes length blocks
				start DCR and read & check a filename
	RdFlnm	INX H		;Calculate name-adres
		INX H		;(to Read Name)
		PUSH H		;Save NAME-adres
		DCX H		;Calculate lenth-adres again
		DCX H		;(to Read Length)
		PUSH H		;Save LENGTH-adres
		INX H		;Calculate Type-adres
				;(to Read Header)
		PUSH H		;Save TYPE-adres
		XRA A		
		LXI D	LBlk1	;Set length of different
		LXI H	LUSER+3H	; blocks = 0
		CALL	FILL	
		LXI H	LBlk2+2H	;Set adres length-storage
		SHLD	SLBlok+1H	; in ChkBlk routine on 2nd blok
	RDCR1	XRA A		;Start DCR
		MOV D,A		
		CALL	DOPEN	
		JNZ	RDCR1	
		CALL	RDHDR	;Read header + filetype
		POP H		;Get TYPE-adres
		MOV M,A		;Store type
		CALL	BYTCUR	;Display type on screen
		CALL	RDADR	;Read name length
		XCHG		;Length in E
		POP H		;Get LENGTH-adres
		MOV M,E		;Store Length of name
		POP H		;Get NAME-adres
	RdName	MOV A,D		
		ORA E		
		JZ	RdFile	;If all bytes read
		CALL	DSTART	;Read next byte
		CALL	RBUEx	;Checksum
		JC	BAD2	
		ANI	7FH	; (Some chars are #80 higher ???)
		MOV M,A		;Place char in memory
		CALL	BYTCUR	;Display char on screen
		DCX D		
		INX H		
		JMP	RdName	;Get next byte

RdFile	PUSH H		;Save endadres name
	LXI H	0H	;Check without name
	PUSH H		
	LXI B	0H	;Check any filetype, no print
	XRA A		
	CALL	ChkBlk	
	LXI H	LBlk3+2H	;Set for 3e blok
	SHLD	SLBlok+1H	; in ChkBlk-routine
	POP H		
	CZ	ChkBlk	
	JNZ	BAD1	
; OK	POP H		;Get endadres name
	CALL	MSpace	
	MVI A	'+'	
	JMP	FileRD	
BAD1	POP H		
BAD2	CALL	MSpace	
	MVI A	'.'	
FileRD	MOV M,A		
	CALL	BYTCUR	
	INX H		;Endmarker name = CR
	MVI A	0DH	
	MOV M,A		
	CALL	DRCLOS	
	RST 5		;CRLF
	DB	3H	
ChkEnd	LXI H	MACC	;Clear MACC
	RST 4		
	DB	69H	
	POP H		;Get length-adres
	INR M		; Count +1 for Filetype
	LXI D	USER1\$	
	CALL	CMPD\$H	;Is Name = "1USER"
	JNZ	ChkNxt	; if no
UsrRd	XRA A		; yes then 0 bytes
	MOV M,A		;in name-adres
	CALL	REW1	
	JMP	WRDIR	
ChkNxt	LXI D	END\$	
	CALL	CMPD\$H	;Is name = "2 END"
	PUSH PSW		;Save Zeroflag
	*** Update name-length ***		
	This routine is used every time a file is read,		
	exept the USER-file.		
UpLngt	PUSH H		;Save length-adres
	MOV A,M		;Length in A
	DCR A		;Subtr. filetype
	STA	LBlk1+2H	;Length name
	MVI A	3H	;Add 3 chars :
			; space, +/-, and CR (Total 4 extra byes

				; (Filetype was add in ChkEnd)
		ADD M		
		MOV M,A		
		*** Update number of Bytes ***		
		(RST 4 + data 4EH : MACC = MACC + inh H,L)		
		LXI H	LBlk1+2H	
		CALL	ADDByt	
		LXI H	LBlk2+2H	
		CALL	ADDByt	
		LXI H	LBlk3+2H	
		CALL	ADDByt	
		LHLD	NmPtr	;Set length file
		LDA	MACC+2H	; in first 2 title-bytes
		MOV M,A		
		INX H		
		LDA	MACC+3H	
		MOV M,A		
		LXI H	Bytes	;Update number of bytes
		RST 4		
		DB	4EH	;MACC=MACC+Bytes
		RST 4		
		DB	0FH	;Bytes=MACC
		Set Name-pointer to net Filename-adres		
		POP H		;Get namelength-adres
UpdPtr		CALL	DADM	;Calculate next pointer
		MVI M	0H	;Set length net name = 0
		SHLD	NmPtr	;Store new adres for AddByt
		PUSH H		
UpAant		LXI H	AanNm\$+3H	;Update number of files
		INR M		
		POP H		
		POP PSW		;Get zeroflag
		JNZ	RdChck	;If no "2 END" was readed
		CALL	REWALL	
LUSER\$	SET	9H		;1 (Recognize-byte) +
				;1 (Filetype)
				;2 (Length filename) +
				;4 (Filename) +
				;1 (Checksum byte)
LUSER2	SET	7H		;1 (Recognize-byte)
				;2 (Length blok2 (=adres)) +
				;1 (Checksum byte) +
				;2 (adres) +
				;1 (Checksum byte)
				;1 (Recognize-byte) +
				;2 (Length blok3 (=data)) +
				;1 (checksum byte) +
LUSER3	SET	BufEnd-ORIG1		; Length Directory +

LUSER3	SET	LUSER3+5H	;1 (Checksum byte)
;			
TotUsr	SET	LUSER\$+LUSER2+LUSER3	
;			
WRDIR	LXI H	MACC	;Clear MACC for endaddition
	RST 4		
	DB	69H	
;			
	LXI H	TotUsr-4H	; (see ADDByt routine)
	SHLD	LUSER+2H	
	LXI H	LUSER+2H	
	CALL	ADDByt	;Prepare Bytes for addition,
;			; MACC = MACC + length Directory
;			
	LXI H	Bytes	;Updating bytes (+ USER)
	RST 4		
	DB	4EH	;MACC=MACC+Bytes
	RST 4		;Copy MACC into Bytes
	DB	0FH	; H,L = adres Bytes
;			
;	***** Save directory *****		
;			
	LXI H	INIT	
	LXI D	EndNm\$	
	MOV A,E		;
	SUB L		;
			;
	MOV E,A		;
	MOV A,D		;
	SBB H		;
			;
	MOV D,A		;
	PUSH D		;
			;
;			
;	Write header		
;			
;			
	MVI A	31H	;File type
	LXI H	USER\$;Filename
	CALL	WOPEN	
;			
;	Write beginadres		
;			
;			
	LXI H	IniPtr	;Pointer Start progr
	LXI D	2H	;Length pointer
	CALL	WBLOK	
;			
;	Write data		
;			
;			
	POP D		;Get EINDadres
	LXI H	INIT	;Get STARTadres
	CALL	WBLOK	
;			
	CALL	WCLOS	
	JMP	BEGIN	
;			
;	***** Subroutines *****		
;			
;			
;	*** Rewind file(s) ***		
;			
REW1	MVI A	1H	
	STA	FILES	
	JMP	REW	

REWALL	MVI A	0FFH	
	STA	FILES	
REW	CALL	REWIND	
	RET		
;			
;	*** Print a space ***		
;			
PSpace	MVI A	' '	
	CALL	OUTC	
	RET		
;			
;	*** Set a space on NmPtr + update NmPtr ***		
;			
MSpace	MVI A	20H	
	MOV M,A		
	CALL	BYTCUR	
	INX H		
	RET		
;			
;	*** First part of Check a Blok		
;	(routine starts on adres 0F287H)		
;			
ChkBlk	PUSH B		
	PUSH PSW		
	CALL	RDPAMB	
	POP PSW		
	POP B		
	CALL	XTHL4	
	NOP		
	NOP		
	NOP		
	PUSH B		
	PUSH H		
	MOV B,A		
	PUSH D		
	PUSH H		
	CALL	RDADR	;Read length blok
SLBlok	SHLD	0H	;Store length blok 2, depends
;			; on adres set in RdFile
	JMP	CMPN2	
;			
;	*** Add a number of bytes (H,L = pointer) in MACC		
;			
ADDByt	MOV E,M		;Change 2 bytes
	INX H		
	MOV D,M		
	MOV M,E		
	DCX H		
	MOV M,D		
	DCX H		;And set for ADD MACC
	DCX H		
	RST 4		
	DB	4EH	
;			
	LXI H	Plus5	;Add 5 extra bytes (rec.byte,
;			; 2 length bytes, + 2 C.S bytes)
	RST 4		
	DB	4EH	
	RET		
;			

***** Data buffer *****

;

TXTCOL	DB	8H,0H,8H,0CH	
Title\$	DB	0CH	
	DB	'DIRECTORY DCR/1'	
	DB	0H	
Files\$	DB	' Files -'	
	DB	0H	
Bytes\$	DB	' Bytes used'	
	DB	0DH,0DH,0H	
MENU\$	DB	8EH,56H	; 'SPACE'
	DB	'..'	
	DB	8CH,0D9H	; 'NEXT'
	DB	' FILE L..'	
	DB	8DH,1BH	; 'LOAD'
	DB	' L..'	
	DB	8DH,1BH	; 'LOAD'
	DB	:'	
	DB	8BH,0F2H	; 'RUN'
	DB	' S..'	
	DB	0B7H,95H	; 'SKIP'
	DB	' U..UPDATE'	
	DB	0H	

;

; Data buffer Updating part

;

InpNm\$	DB	0CH	
	DB	0DH	
	DB	'NAME + DATE (max 35 characters) .'	
	DB	0H	
END\$	DB	5H	
	DB	'2 END'	
USER1\$	DB	5H	
	DB	'1USER'	
USERS\$	DB	4H	
	DB	'USER'	
IniPtr	DW	INIT	
Plus5	DB	0H,0H,0H,5H	

;

CasNm\$	DB	5H	;Length-byte
	DB	1BH	;Escape
	DB	'D'	; tabulation
	DB	0BH	; on lith position
	DB	0H	; end tab.

;

; *** From here all data is cleared

; if updating Directory

;

	DB	0DH	
	DS	LCasNm-5H	

;

AanNm\$	DB	0H,0H,0H,0H	
Bytes	DB	0H,0H,0H,0H	
NmPtr	DB	0H,0H	

;

ORIG2	ORG	\$+0FH&0FFF0H	;Set next adress
-------	-----	---------------	------------------

;

; end with 0

;

BgnNm\$	DB	0H,0H,0H	
	DS	BufEnd-\$;Spare for filenames
EndNm\$	END		