

P-Y-R-A-D-E-V

INSTRUCTION BOOKLET

THE
COMPLETE
AMSTRAD
UTILITY

INCLUDES:
EDITOR
ASSEMBLER
MONITOR
DISC-NURSE
UTILITIES
SELF-TEACH

AMSTRAD CPC 464/664/6128

WHEN YOU HAVE MADE THE

DISCOVERY

NOTHING ELSE COMES CLOSE

USER MANUAL

TABLE OF CONTENTS

Section 1.

PYRADEV

| | |
|-----------------------------|---|
| ○ Introduction | 3 |
| ○ Getting Started | 4 |
| ○ New Users | 4 |
| ○ Working Discs | 4 |
| ○ System Restrictions | 5 |

Section 2.

EDITOR

| | |
|---------------------------------|----|
| ○ File Selection | 5 |
| ○ Edit Mode Display | 6 |
| ○ ESCAPE Help Page | 7 |
| ○ ABORT Edit Session | 7 |
| ○ EXIT and Save File | 7 |
| ○ Cursor Movement | 7 |
| ○ Screen Scrolling | 7 |
| ○ Character Insert/Delete | 7 |
| ○ Line. Insert/Delete | 8 |
| ○ Block Copy/Delete | 8 |
| ○ Setting Tabs | 8 |
| ○ Append Files | 9 |
| ○ Search, Stop, Replace | 9 |
| ○ Keystroke Memories | 10 |
| ○ Printing | 10 |
| ○ Directory Display | 11 |
| ○ Rename Delete Files | 11 |
| ○ Compression Mode | 11 |
| ○ Changing Ink Colours | 11 |
| ○ Environment Save/Load | 11 |
| ○ ZAP backup files | 12 |

Section 3.

ASSEMBLER

| | |
|-------------------------------|----|
| ○ Start Run Options | 12 |
| ○ Assembly Control | 14 |
| ○ Error Handling | 15 |
| ○ Language Definition | 15 |
| ○ Expressions | 16 |
| ○ Directives | 16 |
| ○ Conditional Assembly | 20 |
| ○ Macro Definitions | 21 |
| ○ EXTRA Op-Codes | 22 |
| ○ FREE Format | 23 |
| ○ Memory Management | 24 |
| ○ Assembly Statistics | 25 |
| ○ Summary of Directives | 25 |

| | | |
|-------------------|--------------------------------|----|
| Section 4. | ○ Screen Display | 27 |
| <i>MONITOR</i> | ○ Relocation | 27 |
| | ○ Load Test Code | 28 |
| | ○ Dis-Assembly Display | 28 |
| | ○ Changing Code, Memory | 28 |
| | ○ Setting Traps | 29 |
| | ○ GOTO address | 29 |
| | ○ Single Stepping | 29 |
| | ○ Double Stepping | 29 |
| | ○ Changing Registers | 29 |
| | ○ Screen Refresh | 29 |
| | ○ Query/Search Mode | 30 |
| | ○ ROM switch & display | 30 |
| | ○ Write Binary to Disc | 30 |
| | ○ Dis-Assemble to Source | 30 |
| | ○ Copy Memory | 30 |
| | ○ Screen Printing | 30 |
| | ○ Paper & Pen Colours | 31 |
| | ○ Stack Allocation | 31 |
| | ○ External Access | 31 |

| | | |
|-------------------|-----------------------------|----|
| Section 5. | ○ Disc LOGIN | 31 |
| <i>DISC-NURSE</i> | ○ Drive Select | 32 |
| | ○ Track Select | 32 |
| | ○ Sector Select | 32 |
| | ○ Read Sector | 32 |
| | ○ Modify Sector | 32 |
| | ○ Write Sector | 32 |
| | ○ Un-Erase File | 32 |
| | ○ File Selection | 33 |
| | ○ Query Search Mode | 33 |
| | ○ Screen Printing | 33 |
| | ○ Paper & Pen Colours | 33 |

| | | |
|-------------------|---------------------------|----|
| Section 6. | ○ Deleting Files | 33 |
| <i>UTILITIES</i> | ○ Renaming Files | 33 |
| | ○ Copying Files | 34 |
| | ○ Copy Restrictions | 34 |

| | | |
|-------------------|--------------------------------|----|
| Section 7. | ○ Demonstration Program | 35 |
| <i>SELF-TEACH</i> | ○ Assembler Test Program | 35 |
| | ○ General Information | 35 |

| | | |
|-------------------|------------------------------|----|
| Section 8. | ○ Control Keys Summary | 35 |
|-------------------|------------------------------|----|

1.0 INTRODUCTION.

PYRADEV is a set of five programs which collectively provide a secure and comfortable environment for the development of AMSDOS and CPM software on the AMSTRAD 464, 664 and 6128 micro computers using the DD1 disc system. The system is AMSDOS based and does not require extra ROM or RAM. It will support large programming projects on single or double drive systems and consists of the following programs:-

- A very fast full screen SOURCE FILE EDITOR.
- A multi-file Z80 MACRO ASSEMBLER/Linker.
- A powerful debug MONITOR and DIS-ASSEMBLER.
- A friendly and easy to use DISC-NURSE.
- A comprehensive FILE MANAGEMENT utility.

The EDITOR uses its own special disc read and write routines enabling large files (up to 32,300 bytes) to be quickly accessed and edited. Code can be searched, replaced, entered, deleted, moved, copied, merged (across files). In addition, files can be erased, renamed, deleted and so on. Space compression is used throughout to make the best use of the AMSTRAD memory and disc capacity.

The ASSEMBLER allows up to 32 files to be selected in a single assembly. They can be on different discs (if two drives are used) allowing upto 992K of source code (about 5 discs) to be processed into a single binary file. Video or Print output can be toggled on/off as required. Symbol and Cross-Reference maps may also be produced. The Assembler is very fast, processing up to 40,000 source code characters a minute on 64K systems. On larger systems, the additional RAM is fully utilised for increased performance.

The self-relocating MONITOR will load, trap, single step, double step, modify and write code back to disc. In addition, ROMS can be selected and studied. During debug mode registers can be modified, and screen prints can be taken at any time. Code dis-assembly can be output to a printer or ASCII files.

The DISC-NURSE handles system, data and IBM style discs. File sectors can be searched, viewed, printed, modified and re-written. The search can be extended to the whole disc if required. Whenever a sector is read the file-owner is displayed. Previously deleted files can be re-claimed if subsequent sector allocation has not corrupted data. An extended Directory feature enables the disc directory to be printed with a header message, in a detailed format suitable for disc housekeeping lists.

The UTILITY (file-manager) program provides directory display, erase, rename, copy to/from discs and tape all in a single and easy to use program. Input file headers are always displayed, and protection can be inserted or removed in all copy operations. This is the easy way to manage ALL standard AMSDOS files.

1.1 SETTING UP.

PYRADEV is supplied on a MASTER DISC which must never be written to. It should only be used to make WORKING DISC copies with the CPM utility DISCCOPY or DISCKIT3. Refer to the appropriate manual and use the PYRADEV MASTER DISC as the Source Disc to create a WORKING (destination) Disc.

After the copy is complete, remove the PYRADEV MASTER DISC and store it away somewhere safe.

1.2 GETTING STARTED.

Place the WORKING DISC in drive A and clear the machine by pressing CTRL-SHIFT-ESCAPE. Enter RUN"PYRADEV and press the large enter key. The SYSTEM-MENU will appear. The amount of memory present is displayed on the bottom right of the screen.

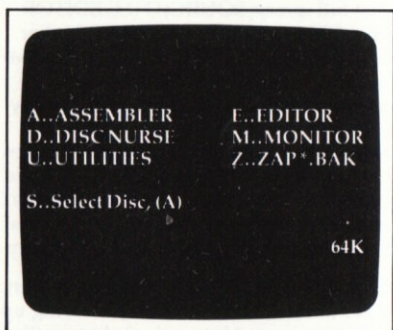


Figure 1.0 System-Menu

The system programs are selected by pressing the appropriate single key «A, E, D, M, U». They all return to the above menu upon Exit.

The «Z» option will delete all backup files (*.BAK) on the current Default Disc. Use it with care.

The «S» option allows you to change the default disc drive setting for Data/Source files to (A) or (B). The PYRADEV disc must always be in drive (A) as it expects to find its programs there, regardless of the disc default. The «S» option should NOT be used on single drive systems.

1.3 NEW USERS.

When the SYSTEM MENU appears, press «E» to select the Editor. Enter the name PROGRAM.001 as your input file and press the space bar twice. The file will be read into memory. Browse through the file with the UP and DOWN arrow keys. The file is a small program with lots of comments and examples about using the Editor and Assembler.

1.4 NEW DISCS.

When a WORKING DISC is created there may be system files on it which are not required. Press «U» to select the Utilities program, then press «A» or «B» to see the disc directory. The «D»elete command can be used to remove files. This list describes the system files.

| | |
|-------------|--|
| PYRADEV.BIN | SYSTEM-MENU, required on ALL Working Discs. |
| PYRAMED.BIN | Full Screen Editor for Source Program Editing. |
| PYRAMAS.BIN | Z80 Multi-File Macro Assembler. |
| PYRAMON.BIN | Test Monitor and Dis-Assembler. |

PYRADSC.BIN
PYRAUTL.BIN

Disc Nurse for Modifying Sectors.
File Utilities for General Copying.

PROGRAM.001
PROGRAM.002
PROGRAM.003

Self-Teach Demonstration Source Program.
Assembler Code Examples and Test Files.
Simple ASCII file copier program.

After deleting non-essential files, the new Working Disc is Ready for use. File deletion is also possible within the Editor.

SYSTEM RESTRICTIONS.

PYRADEV uses locations 40 hex through to 4F hex to pass variables amongst its programs. Please avoid using these areas during program development and testing.

When started, PYRADEV will initialise all background ROMS by calling the KL.ROM.WALK routine. It then sets its own HIMEM value based on the returned HL value. Ideally, only the DISC.ROM should be connected, however, other ADD-ONS should not cause problems, unless they use large areas of RAM. The Editor work area is normally about 32,300 bytes.

When using the MONITOR, be aware that all background ROMS connected to the AMSTRAD have been initialised. Initialising a ROM twice may cause un-desirable effects.

2.0 SOURCE FILE EDITOR.

The Editor is designed for very fast program development and editing. It is selected from the System Menu by pressing (E). Once loaded it will display the default disc directory, and then wait for Input and Output file names to be entered. The Editor always returns to the Idle Screen after the edit session is complete.

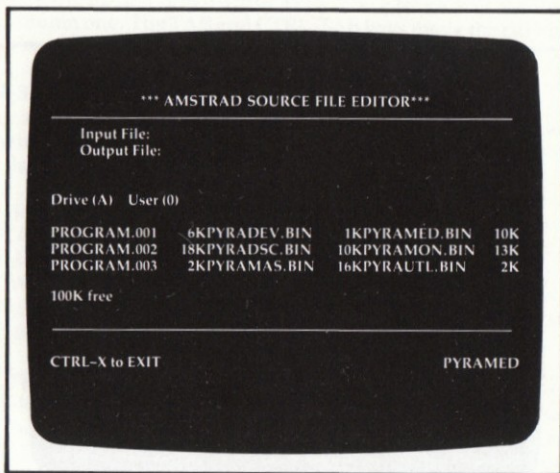


Figure 2.00 Idle Screen

Line 25 of the Edit Screen always displays status information:— The current Input and (probable) Output file names, current record number and cursor column, Free-Bytes and default Disc Drive are shown.

ESCAPE KEY.

When used in Edit-Mode, the escape key will cause the Help-Page screen to be displayed. This is just an 'aide memoire' and describes very briefly the functions below. Pressing the escape key a second time returns the screen to Edit-Mode.

ABORT & EXIT:

The (CTRL-A) key may be used to ABORT the Edit-Session. when used all files are abandoned and no disc changes take place. The character (Y) must be entered to confirm the Abort Request. The Editor returns to the Idle Screen and displays the Disc Directory.

SAVE & EXIT:

The (CTRL-X) key is the normal exit. The output file name will be displayed and can be overtyped as required. The file will be written to disc and control will return to the Idle Screen. If the File-Write fails, the message ** FILE SAVE FAILED ** will be shown and Command Mode will be selected so you 'action' the disc (ie delete non-required files) or use another disc. After a successful save the Editor returns to the Idle Screen.

The Output file name can be prefixed with A: or B: to set a new disc drive default.

CURSOR-MOVEMENT.

This is controlled by using the UP, DOWN, LEFT and RIGHT ARROW keys or by entering text. Forward and Reverse scrolling is automatic as row-1 or row-24 are reached. Text is ALWAYS overkeyed at the cursor position. Records are added to the front or back of the file if the cursor is taken past the first or last records using the up and down arrow keys. The ENTER key always goes to the next line, column one. The TAB and CTRL-TAB keys move the cursor to the next tab or previous tab stop respectively.

CURSOR-SPEED.

The cursor speed can be toggled between fast and slow by using the (CTRL-Y) key. Each time it is changed, the prompt (Flash?) will appear. Answer (Y) if you require a flashing cursor, or (N) for a non-flashing cursor.

SCREEN-SCROLLING:

The SHIFT-DOWN-ARROW and SHIFT-UP-ARROW will scroll the screen one line up or down respectively whilst maintaining the cursor position. The CTRL-DOWN-ARROW and CTRL-UP-ARROW will scroll forwards and backwards 24 lines at a time. These four functions will not scroll past the beginning or end of the file.

CHARACTER-INSERT-DELETE:

The SHIFT-RIGHT-ARROW and SHIFT-LEFT-ARROW provide these functions. Note that the insert function places a single blank in the text which can then be overtyped. The Insert and Delete actions both occur at the cursor position. The (CLR) and (DEL) keys also provide character delete and reverse delete capabilities (as they do in BASIC).

LINE-INSERT-DELETE:

The CTRL-RIGHT-ARROW and CTRL-LEFT-ARROW provide these functions, and they both operate on the current line, indicated by the cursor. The line insert action places a blank line into the text which can be overtyped. As text is typed of the end of the line another blank line is inserted. The same occurs if the ENTER key is pressed. Line Insert Mode is cancelled by the use of CTRL-LEFT-ARROW (to delete a newly created blank line) or by scrolling down past row 24.

OPEN SPACE.

The (CTRL-O) key can be used to open or split a line at the cursor position into two new lines. The first will contain text up to the cursor position. The second line will contain text from the cursor position. This is useful when inserted new code at a line which already has a label.

BLOCK COPY:

To copy a block of code, move the cursor to the first line of the code and press (CTRL-B) to set the BEGIN marker. Move the cursor to the last line of the code and press (CTRL-E) to set the END marker. The message 'Block Saved' will be displayed. Position the cursor where the block is to be copied to and press the (COPY) key. The block will be COPIED-INSERTED at the following line. The saved block will stay in memory until another Block is marked. It can be copied repeatedly anywhere in the file, or to another file or to another disc.

BLOCK-DELETE:

Mark the Block with the (CTRL-B) and (CTRL-E) keys as for the Block Copy function. When the 'Block Saved' message appears, press the (CTRL-D) key. This will display the marked block in reverse INKS. Press (Y) to permit the delete operation. Note after the delete, the block is still saved and can be copied back if required with the (COPY) key.

BLOCK MOVE:

This is the BLOCK-DELETE function followed by use of the (COPY) key as described above.

TAB STOPS:

Pressing (CTRL-T) will display the current tab settings as small triangles on line 24. Tabs are set or cleared by positioning the cursor and pressing the (TAB) key until the triangle symbol appears or disappears. Pressing «CTRL-TAB» will clear all tab stops. When the new tab stops are ready, press the large Enter key to record them. Tab marks can be saved for future use, see Command Mode.

MARK-FIND-LINE:

A single LINE can be marked by using the CTRL-L key. The line can be brought back to the screen with the CTRL-F key (FIND). These two operations are useful in that they allow a rapid return to code being entered after studying code elsewhere in the file.

VIEW-BEGIN-END:

Press CTRL-V to select 'View' then press (B) to see the beginning of the file (record-1) or (E) to see the end of file (last record).

GO TO RECORD:

Pressing CTRL-G allows a record number to be entered. The Editor will go to the record immediately. Use of zero or numbers greater than the last record will cause the Beginning or End of File to be displayed respectively. The CTRL-G function is helpful in locating program code from the Assembler Listing.

NEW-INPUT:

Use of the CTRL-N function allows another INPUT file to be appended to the current memory workspace file. In this way it is possible to build up new programs or documents from previous code or text. Once the new file has been added into the memory, the Block Delete and Move functions can quickly bring code or text to where it is required.

This function may only be used if at least 1028 bytes are still free.

SEARCH SYSTEM:

CTRL-S selects the search system and two types of operation are provided.

Search and Stop.

Enter a search string in single or double quotes then press the enter key twice, leaving the replace string empty. The search will start. Each time the string is found, the record containing it will be displayed together with the preceding and following record in its own small 'window'. The following options can then be used:-

- Enter Key Ignore this match and continue searching.
- Escape Key Terminate, and return to original text.
- (G) Key Goto the record where the match was found.
- Up/Down Scroll the records where the match was found.

Search and stop enables rapid positioning to program code using source code labels which are (usually) unique.

Search and Replace.

Enter a search string in single or double quotes. Then enter a replace string in single or double quotes. Be sure to use upper or lower case as required for the replace string. After the replace string is entered, answer the question to select Automatic or Conditional replacement and the search will start.

If the Auto-Replace option is chosen, the process will run to completion without further action. If the Conditional option is used, each time the string is found it will be shown with the preceding and following records. Press (R) to do the REPLACE, or press (C) to CONTINUE with no action.

All search functions operate from the current line to end-of-file. Use CTRL-V B to select record 1 if you intend to search the whole file. A good keystroke memory sequence to use prior to using the search function is CTRL-L CTRL-V B. This marks the current line and goes to the beginning of the file. After using the search function CTRL-F will return to the marked line.

Any search can be cancelled with the ESCAPE key.

Note: Truncation will occur if a replace string is longer than the source string and characters are forced past column 80.

KEYSTROKE-MEMORIES:

The editor contains nine keystroke memories which can each record 32 keystrokes. Each of them may be used to record text or control keys. These can then be 'replayed' as required.

Display:

To view the contents of the keystroke memories press the numeric island key 0 (f0). The memories will be displayed. Press the ESCAPE key to return to Edit-Mode.

Record:

Press the SMALL-ENTER key then select a memory using the island keys 1 thru 9. Once selected the message 'Keystroke Recording ON' will be displayed. To terminate the recording press the small enter key again. If a recording exceeds the 32 keys limit, the next memory will be attached and used. If the end of keystroke-memory nine is reached, recording will be turned off, and the KS-sequence-saved message will be displayed.

Replay:

To replay any recorded sequence press the numeric island key 1 thru 9 which was used to make the recording. The keys will be taken and used as though they had just been entered at the keyboard. The replay can be stopped by pressing the ESCAPE key.

Erase:

Press CTRL-SMALL-ENTER and select a memory with the numeric island keys 1 thru 9. The memory will be cleared ready for use once again.

NOTE. 1: Memory functions may only be recorded or initiated for playback in normal Edit-Mode. Recorded sequences may for example select the Search system and start a search for a character string then revert to Edit-Mode. However it is not possible to start a recording or keystroke memory playback if the Search system has already been selected manually.

NOTE. 2: The keystroke memories can be saved. See Command Mode.

PRINTING:

Pressing the decimal point key on the numeric island will cause printing to start from the current line. Pressing the key twice will stop line-feeds being sent with carriage-returns. Printing will continue until End Of File. Pressing the ESCAPE key once pauses the print which can be continued with any key. Pressing the ESCAPE key twice cancels the print activity.

Special print codes may be embedded in the text being printed in order to control the printer as follows:-

During printing, the up-arrow symbol (on the pound sign key) indicates that the following character is a control symbol. The value 33 is subtracted from ASCII value of the character. If it is minus, the character is ignored, else it is sent to the printer. An up-arrow symbol at the end of a line causes the invisible CR character to be skipped, allowing two screen lines to make one long line of printing.

Examples.

⊠«E represents ESCAPE-E, ⊠«S represents ESCAPE-S,

⊠ represents code 15, ⊠/ represents code 14.

⊠∅ represents code 15.

COMMAND MODE.

Command mode is selected with the CTRL-Z key whilst in Edit Mode. The Edit Session is temporarily suspended while the File Management and Option routines become available. Exit from Command Mode with the Escape Key.

A small menu is displayed at the bottom of the screen and Functions can be selected by pressing the appropriate key as follows:-

| | |
|---------------------|-----------------|
| A,B: Select Disk | C: Compression |
| D,R: Delete, Rename | Z: Zap BAKups |
| L,S: Load Options | S: Save Options |
| J,K: Paper Pen Inks | |

A,B: Select Disc A or B.

Pressing (A) or (B) changes the default disc drive and displays the new disc directory, file-names and free space.

C: Compression ON/OFF.

If you are Editing and Assembling code within the PYRADEV system, you should leave this option ON. The Editor and Assembler will use data compression to keep your files as small as possible. If you use the Editor to edit BASIC files, before using CTRL-X to write the file back to the disc you should turn compression OFF. The file will be expanded to normal and can then be used by the BASIC system.

D: Delete File.

This is similar to the BASIC/CPM directive 'ERA'. It can be used to delete a single file or group of files. A specific disc drive can be selected by using an A: or B: prefix.

Examples:

| | |
|--------------|---|
| TEMPFILE.001 | - Delete the file. |
| TEMP.* | - delete all TEMP files. |
| TEMP????.* | - delete file names starting with TEMP. |
| B:TEMP.FIL | - delete TEMP.FIL on drive B. |

R: Rename File.

Any existing OLD file can be renamed to a NEW file if the new file does not already exist.

J: Jump PAPER Ink.

Each time the (J) key is pressed the Paper Ink will be advanced to its next value. Pressing (CTRL-J) does the opposite.

K: Jump PEN Ink.

Each time the (K) key is pressed the Pen Ink will be advanced to its next value. Pressing (CTRL-K) does the opposite.

S: Save User Options.

Pressing (S) causes the current Tabs, Paper and Pen Ink Values, and Keystroke Memories to be written to a disc file for future use. A file-name must be entered and can be any name. Suggested file extension (suffix) is -OPT. Several different Option Files can be used to define different 'environments'.

L: Load User Options.

Use of the (L) key allows an 'option file' created with the (S)ave command to be re-instated for the current Edit Session. After the file is read, the new Tabs, Inks, and Keystroke Memories will be in operation.

*Z: Zap *.BAK files.*

Removes all backup files from the default disc directory. It is the same as using the (D)elete function with the file name *.BAK.

3.0 MACRO ASSEMBLER.

The PYRADEV Macro Assembler supports the ZILOG Z80 programming language (with a few alterations and extensions) and will process one or more source program files to create a binary load file, directly on disc. It is selected from the SYSTEM MENU by pressing (A). Once loaded it displays the default disc directory and waits as the various RUN OPTIONS are entered....

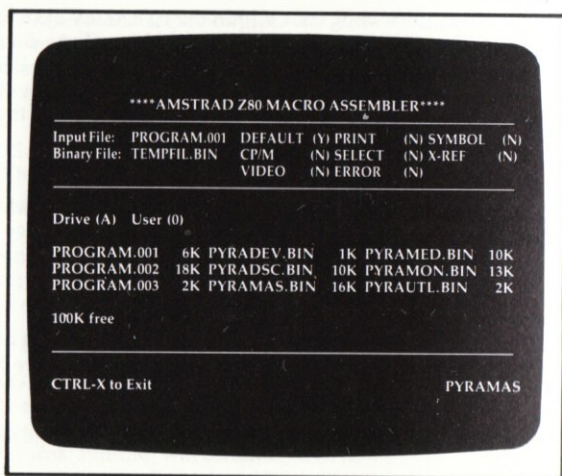


Figure 3.0 Assembler Run Options

INPUT-FILE.

This names the first source file for the input stream. It may be a complete program or the first of many which are to be assembled together to make up the binary load program. SELECT statements embedded in the source code files can be used to append other files to the input stream. One way of using the SELECT system is for the first file to contain nothing but SELECT statements. A good file suffix for these files is .CTL as it represents a CONTROL file. A default binary file name with a .BIN suffix is created from the input file name. An END statement in the input stream is optional, but if present will end the input stream immediately, even if there are more files.

BINARY-FILE.

This is the name which is to be used when the binary code file is catalogued on disc. It is built from the input file name (by adding a .BIN suffix) and may be overkeyed. Pressing the ENTER key leaves it as it is. If no binary file is required, this field should be set to blanks.

DEFAULTS.

If the Enter key is pressed or (Y) is pressed the Assembler will use the displayed defaults and start the assembly process. If any of the run options are to be altered, press (N) for this first option.

CP/M-OUTPUT.

If set to (Y), the binary file is written to disc without a file header, suitable for use as a CP/M transient program. No other checks are made to ensure that the program conforms to CP/M standards.

VIDEO-DISPLAY.

If set to (Y), the source record being processed will be displayed in pass 1. During pass 2, the object code and source record will be displayed.

PRINT-OUTPUT.

Setting the PRINT option to (Y)es instructs the Assembler to produce an assembly listing during pass 2 object code generation. This option works together with the SELECT option....

SELECTIVE PRINT.

The SELECT option can be set to (Y)es or (N)o and instructs the Assembler to honour (YES) or ignore (NO) the LIST and NOLIST directives contained in the source code.

LIST ERRORS.

When set to (Y)es, this option forces ERROR messages to be listed on the printer for subsequent use when correcting coding mistakes with the EDITOR. It works independently of the PRINT and SELECT options described above, however, errors are always listed if the PRINT option is (Y)es.

| PRINT | SELECTIVE | ERRORS | Printed Code | Printed Errors |
|-------|-----------|--------|--------------|----------------|
| NO | NO | NO | NONE | NONE |
| NO | NO | YES | NONE | ALL |
| NO | YES | NO | NONE | NONE |
| NO | YES | YES | NONE | ALL |
| YES | NO | NO | ALL | ALL |
| YES | NO | YES | ALL | ALL |
| YES | YES | NO | SELECTIVE | ALL |
| YES | YES | YES | SELECTIVE | ALL |

Figure 3.1 Print Options

WARNING: The use of the VIDEO or PRINT options will SIGNIFICANTLY slow the Assembly Process down. For fastest results reply (Y) to defaults. An exception is to list ERRORS which you will find very useful during the first few Assemblies of large programs.

SYMBOL-TABLE.

If set to (Y), a list of the symbols (labels) used in the source program(s) will be output with their values. Output will be to VIDEO or PRINTER depending on the options chosen above. Each symbol may be preceded by a (U), (D) or (N) denoting Undefined, Doubly-Defined or Not-Referenced.

CROSS-REFERENCE.

If set to (Y) a cross reference map will be output to the Video Display or Printer after the assembly process is complete. The listing will display all source files read, numbering them as files 1,2,3 and so on. The symbol table is printed, together with all file numbers and line numbers of statements that reference them.

Please note that if they are used, the SYMBOL or X-REF options will wait for the VIDEO or PRINT options to be set to ON before the assembly process can output the information requested.

ASSEMBLER RUN TIME KEYS.

Once started, the Assembly process can be controlled by 'toggling' the RUN OPTIONS described above. This is done by pressing single keys. The screen run options display will change as the keys are used.

The following keys may be used:-

- | | | | |
|-----|----------------------------|-----|----------------------------|
| (P) | Reverse the PRINT option. | (E) | Reverse the ERRORS option. |
| (S) | Reverse the SELECT option. | (V) | Reverse the VIDEO option. |

In addition the Escape key may be used to PAUSE the Assembly. This is useful in order to study object code being displayed on the screen. Press any key to resume the assembly process.

Pressing the CTRL-A key will abort the assembly process. All files will be abandoned and no disc changes will take place. The Assembler will ask for any key to be pressed so that it can return control to the System Menu....

```
****AMSTRAD Z80 MACRO ASSEMBLER****
-----
Input File: PYRAMD.CTL      DEFAULT  (N) PRINT  (N) SYMBOL (N)
Binary File: PYRAMD.BIN    CP/M      (N) SELECT (N) A-REF  (N)
                               VIDEO      (Y) ERROR  (N)
-----
Pass 2: Generating Object Code          Reading SCID34.ASM          Rec'd: 0397
-----
0355 121B CDD225          CMI 1-DMP-AM
0356 121 C36104          IP 1-DIRR
0357 1211 0204374 WWR1QD DB 99 Ctl-WReqd' 37
0358
0359                               Word-Processor Resnake Test Routine
0360                               Resnake current line to end of Paragraph
0362 1302 CDDC12 CERER CMI CHK-WB-ON CHECK-CH-AMT-G
0363 1305 CERER-2 IQL S RETURN-IF-OK
0364 1305 DD2A972D ID IX-CR-PTB IX-REC-PTB
0365 1308 010003 ID BC-m33 INST-COUNT
0366 130C DD*100 CRI-2X ID A-IX*+6 GET-NAI-CHAR
-----
CTRL-A to Abort! !PYRAMAS
```

Figure 3.1 Run Time (VIDEO-ON)

ASSEMBLY ERROR HANDLING.

The detection of errors by the assembly process will be handled in one of three ways according to the output options set above.

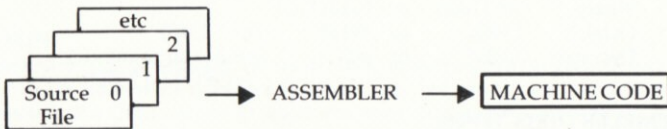
- a) If any of the PRINT options or ERROR options are set to (Y)es, the Assembler will assume a printer device is attached. The error code and description will be printed. The assembly process will NOT pause.
- b) If the VIDEO option is set to (Y)es, the error code and the description will be displayed. The assembly process will pause, and can be resumed by pressing any key.
- c) If the PRINT, ERROR and VIDEO options are all set to (N)o, the assembly process will be running at its fastest setting. Errors will be sent to the Video Screen, but the assembly process will only pause after ten error messages have been displayed. If this occurs, press any key to resume the assembly process.

Error Codes.

| | | | |
|----|---------------------|----|-------------------------|
| B | - Branch/Jump Error | N | - Numeric Expression |
| C | - Conditional Error | OE | - Operand(s) Error |
| D | - Doubly Defined | OM | - Operand Missing |
| F | - File Error | SF | - Too many SELECTs |
| G | - Org Usage Error | U | - Undefined Label |
| L | - Label (syntax) | X | - Extra Opcode Invalid. |
| MP | - Macro Parameter | FW | - Forward Ref. Invalid. |

ASSEMBLER LANGUAGE DEFINITION.

The Assembler Language is the Z80 language and the programs are written as one or more Source Files which are assembled to machine code by the Assembler.



Each source file consists of source statements, one per line (screen row). The PYRADEV EDITOR is the ideal program to create and edit these source files. Each source file statement consists of an optional label, an opcode, optional operands and optional comments.

| LABEL | OPCODE | OPERANDS | COMMENTS |
|------------|--------|----------------|------------------|
| LABEL-1: | LD | HL,(VALUE) | ; load HL |
| | LD | (OLD-VALUE),HL | ; save HL |
| | JP | LABEL-X | ; jump... |
| ; | | | |
| VALUE: | DEFW | 00 | ; current values |
| OLD-VALUE: | DEFW | 00 | ; old value |

LABELS: These must start in column one and may be any length although it is usual to keep them down to less than 10 characters. The use of a colon after the label is optional.

OPCODES: These may be anywhere on a line and must be preceded by at least one space.

OPERANDS: An opcode must be followed by at least one space before the operand can be entered. The operand field must not contain embedded blanks.

COMMENTS: Operands must be followed by a least one space before a comment can be entered. The use of a semi-colon before the comment is optional except on RET statements. A semi-colon must be used if the comment starts at column one.

EXPRESSIONS.

An expression is an OPERAND which consists of one or more variables, labels and constants which the Assembler must evaluate into a 16 bit integer value. An expression is evaluated from LEFT to RIGHT and parenthesis may not be used. The following operators may be used...

+ - * / .MOD. .SHR. SHL. .AND. .OR. .XOR. .EQ. .GT. .LT. .UGT. .ULT.

They represent plus, minus, multiply, divide, shift-right, shift-left, AND, OR, exclusive-OR, Equates, greater-than, less-than, unsigned-greater, unsigned-less-than.

The dollar (\$) symbol may be used to represent the value of the program counter during assembly. For example JP\$+3 would generate a branch to the next instruction (a JP is 3-bytes).

The Assembler will accept numeric notation for binary, octal, decimal and hex-decimal expressions in the following formats...

| | | | | |
|----------|----------|----|----------|--|
| Binary | 1011100B | or | %1011100 | |
| Octal: | 134Q | or | @134 | (or 1340) |
| Decimal: | 134D | or | #5C | (must start numeric) (ie OFFH for 255D) |

ASSEMBLER DIRECTIVES.

These are written like instructions (opcodes and operands) but are commands to the Assembler and are 'executed' at assembly time. They control assembly listing options, code generation, and the construction of the binary (machine code) file.

DEFINING VARIABLES and STORAGE.

Bytes, Words and Character Strings may be defined using the directives shown in the following example.

| | | | |
|----------|-------|--------------------|------------------------------|
| | DBXON | | ; DB Expansion |
| LABEL1: | DB | "This is a string" | ; These are 16 byte strings. |
| LABEL2: | DEFB | "This is a string" | ; |
| BYTE: | DB | "B" | ; Single byte |
| MIXTURE: | DB | 1,"A",2,"b" | ; Mixed string |
| SSTRING: | DB | "SPECIAL"+80H | ; L has bit 7 on. |
| | | | ; |
| LABEL3: | DW | 256*2 | ; A word can hold a |
| LABEL4: | DEFW | 512 | ; 16 bit integer. |

```

;
LABEL5:  DB      255          ; Max value single byte
LABEL6:  DEFB    0FFH       ; is 255 decimal.
;
VALUE7:  EQU     1000       ; VALUE7 equals 1000
LABEL8:  DW      VALUE7    ; LABEL8 contains 1000
VALUE7:  DEFL    1001       ; VALUE7 redefined to 1001
LABEL9:  DEFW    VALUE7    ; LABEL9 contains 1001
        DBXOFF

```

The DBXON and DBXOFF cause the DB strings to be listed in long form or short form (1st four bytes) respectively.

The Source Files PROGRAM.001, PROGRAM.002 and PROGRAM.003 on the MASTER-DISC contain more examples of the assembler directives. The files can be viewed and/or printed using the PYRADEV Editor.

EJECT

This instructs the assembly print process to feed to next top of form. Normally a page width of 80 characters and form depth of 70 lines are assumed. This is standard A4 size. These parameters may be altered via the PRINTR directive, shown below.

END

This statement signals the END OF INPUT. The Assembler treats this as a hard end of file, even if source code follows the statement. Use of an END statement is optional.

ENDBIN

Ends binary code generation. Generally used at the end of a program before the DEFS or DS statements to keep a file size small, but can be used anywhere. Its opposite is the ORG statement which resumes code generation or the LOAD statement which 'pads' and resumes code generation.

```

; This coding example shows how an ; This coding example shows
; area is reserved at the front of ; how endbin is used at the
; of a program.                    ; end of a program.
;
        ORG      50H
        ENDBIN
;
        POP BC
        POP DE
        RET
;
BUFF-1:  DEFS    2048
BUFF-2:  DEFS    2048
;
TRUE-START: ORG      $          BUFF-1: DS 2048
          LD      (SAVE),SP  BUFF2: DS 2048
          CALL   INIT-PGM ;
          JP     GO-GAME ;
;
;                                     END
        ETC.....

```

EQU

This is the EQUATES directive. It equates a LABEL to an expression, for instance, MINUS1 EQU -1

EXEC

This directive defines the address to be used in the binary file header. When the assembled program is RUN, it will be loaded according to the LOAD (or ORG) statement. AMSDOS will then pass control to the EXEC address. If this directive is not supplied, the EXEC address is set to the first (true) ORG or LOAD address.

EXTRA

Enables the Assembler to process the additional NON-STANDARD Zilog Instructions. See page 14.

FREE

Enables the Assembler to process 'Free Format' expressions. Allows mixing of different types of storage expressions. See page 15.

LIST and NOLIST

Turn Selective Printing On and Off.

LOAD

This statement tells the Assembler to generate binary zeros code until the Program Counter reaches the LOAD value expression. Normal code generation is then resumed. Its purpose is to force sections of code to their proper positions in the binary file so that when the file is loaded, the code is at its correct address.

If it is not supplied, the LOAD address defaults to the first true ORG address. One of the two must be supplied (it is usually ORG) before any code generation can occur. The first occurrence of an ORG or LOAD directive is used in the binary file header (unless it is immediately followed by an ENDBIN statement). Subsequent ones are only used to control code generation and binary file structure.

ORG

This statement tells the Assembler to SET the Program Counter to the operand expression. Unlike the Load statement no filler code is generated. Use of the ORG statement allows sections of code which will be widely apart in memory to be squeezed together in the disc file being generated. It is the programmer's responsibility to ensure that such sections of code are moved to their correct locations before being executed. It is NORMAL practice for the first statement in a set of programs being assembled to be an ORG directive.

```
ORG 100H . This Section locates
code . correctly in memory
code . from location 100 hex
LOAD 200H . upwards.
code .
code .
code .
```



```

;
    ORG $+100      . This block will need
    code          . to be moved to its true
    code          . address as no filler
;                . code is generated.
    END

```

PAUSE

This statement causes the assembly process to 'Pause and Display' the message. This occurs when the END of the current input file is reached. Its use is to allow multiple discs to be used when assembling multiple source files. It should only be used on TWIN drive systems as the BINARY file must be written continuously to one disc. After changing the disc, press the space bar to continue.

PAUSE 'MOUNT NEXT SOURCE DISC IN DRIVE B' (assumes output drive A).

PRINTR

This statement can be used to define Paper-Width, Lines per Page (form-depth), Page-Pause and Line-Feed suppress. The Page-Pause option causes printing to pause at each top of form to allow paper adjustment. This is a REAL requirement on some friction feed printers which 'slew' the paper and make Assembler Listings difficult to produce. Line-Feed suppress stops line feeds being sent with each carriage return as some printers do this automatically.

```

PRINTR W80,D70      Width 80 chars, Depth 70 lines, P-Pause Off, L-Feeds.
PRINTR W132,D66,P  Width 132 chars. Depth 66 lines, Page-Pause On, LF's
PRINTR N           Use the defaults (W80,D70) and suppress line-feeds.
PRINTR P           Use the defaults (W80, D70) and use Page-Pause.

```

SELECT

This is a very powerful directive. When used, the named Source File is ADDED (not INCLUDED) to the END of the current input stream. Upto 32 files can be CHAINED in this manner, and the SELECT statements may appear anywhere. When assembling large programs, it is possible to start the assembly process with a small control file which does nothing more than SELECT files for the INPUT stream....

```

    ORG      100H      ; Program Origin
    SELECT  PROG1.ASM ;
    SELECT  PROG2.ASM ; Drive A Files
    SELECT  PROG3.ASM ;
    SELECT  B:PROG1.ASM ; ;
    SELECT  B:PROG1.ASM ; ; Drive B Files
    SELECT  B:PROG1.ASM ; ;

```

On large programming projects, the following information may be useful, assuming you are using twin disc drives.

| DRIVE-A | | DRIVE-B | |
|---|--|---|--|
| Disc 1 Side A | PYRADEV SYSTEM + Binary Files + Source Files. | Disc 2 Side A | SOURCE CODE – 178K, accessed by Editor and Assembly System |
| Disc 1 Side B | Spare, suggest you use PYRA-WORD to document project | Disc 2 Side B | SOURCE CODE – 178K, accessed by Editor and Assembly System |
| Disc 1 stays fixed, during development | | Disc 2 is reversed as / when required to edit / assemble source files. | |

TITLE

Change to the Top of Form Assembly Listing Header Message, example:-

TITLE MegaGame, Section 6.

CONDITIONAL ASSEMBLY DIRECTIVES.

The Assembler is able to include or exclude certain blocks of code during the assembly process through the use of flags and conditional directives. Combined with the SELECT system described above, the system becomes very flexible, as the first files may define flags which control the assembly of code in subsequent files.

The mechanism of conditional assembly is the classic IF something THEN do this ELSE do that.

The something is an Arithmetic Expression. If the expression is TRUE (non-zero) the first path (THEN) is taken. Otherwise the second path (ELSE) is taken. The second path is optional and the final directive must be an ENDIF statement.

The following examples show how the process can be used. The code on the left has the FLAG set to 1 (TRUE) so the THEN-CODE is assembled. The code on the right has the FLAG set to 0 (FALSE) so the ELSE-CODE is assembled.

| | |
|--|--|
| <pre> ; FLAG: EQU 1 ; IF FLAG THEN-CODE: LD A,(VALUE-1) ELSE ELSE-CODE: LD A,(VALUE-2) ENDIF ; END </pre> | <pre> ; FLAG: EQU 0 ; IF FLAG THEN-CODE: LD A,(VALUE-1) ELSE ELSE-CODE: LD A,(VALUE-2) ENDIF ; END </pre> |
|--|--|

Note the usage is always IF ... ELSE ...ENDIF. The THEN statement is implicit as the first branch after the IF statement. The ELSE section is optional. An IF directive must always have a corresponding ENDIF statement.

An alternate to using the IF..ELSE..ENDIF directives are the COND..ELSE..ENDC directives. They are both valid.

MACRO DEFINITIONS and USAGE

A MACRO is a short piece of code, defined in a file at the beginning of the assembly process. When its name is used, the previously defined piece of code is generated again.

A MACRO statement defines the start of the definition and it must have a label which is used as the Macro-Name. The name must be ALL alphabetic and may be up to SIX characters long. Imagine the Macro as a new Instruction for the Assembler. The two examples here show a macro without Parameter Usage on the left, and with Parameter Usage on the right.

```
SBCX:  MACRO
        XOR   A'
        SBC  HL,DE
        ENDM
;
        LD   HL,(VALUE1)
        LD   DE,(VALUE2)
SBCX
        LD   (VALUE3),HL
;

SWAP:  MACRO#P1,#P2
        PUSH AF
        LD   A,#P2      (LD A,C)
        LD   #P2,#P1    (LDC,D)
        LD   #P1,A      (LDD,A)
        POP  AF
        ENDM
;
        SWAP D,C
```

The example on the left shows how a small macro can be used as an additional instruction. In this case the SBCX macro is assembled as two instructions. The first clears the carry flag before executing the second SUBTRACT with CARRY instruction.

The example on the right shows parameter substitution. During Assembly usage of the SWAP macro causes a five byte routine to be generated which will cause the contents of the C and D registers to be exchanged. The SWAP macro can be used to exchange any two registers except for register A.

Macro Parameter Usage

As shown in the above example, parameter usage is positional and works by substitution. If parameters contain commas or quotes, they must be enclosed within single or double quotes as shown here:-

```
STRING:  MACRO      #P1,#P2
          DB         #P1      (DB 5)
          DB         #P2      (DB 21,22,23,24,'X')
          ENDM
;
          STRING      5,"21,22,23,24,'X'"
;
```

Macro Symbol Generator

If a macro definition contains labels, DUPLICATE LABEL errors will occur during the assembly process if the Macro is used more than once. In such cases the #SYM suffix must be added to the label. Each time the Macro is used, a 4-digit suffix is incremented and attached to the label.

The following example is a macro which tests HL and substitutes the hex-decimal constant 0FFFFH if HL is zero. Each time it is used, the JR TEST-#SYM instruction and TEST-#SYM labels are expanded with the next value. The first time thru the JR will be to the TEST-0001 label, then it will be TEST-0002 and so on...

```

;
;
;          MACRO tests HL, if ZERO replace with 0FFFFH.
;
;
TEST:     MACRO
          XOR          A           ; Clear Carry
          LD           DE,0       ; Set DE to zero.
          SBC         HL,DE      ; Subtract / TEST
          JR          NZ,TEST-#SYM ; if NZ JMP to TEST-000n
          LD          JL,0FFFFH  ; else set HL to 0FFFFH
TEST-#SYM: EQU          $         ; continue ... TEST-000n
;

```

Macro Listings

Normally only Macro Definitions are listed. To see the expanded code you must use the MLIST directive. To turn this facility OFF use the MNLIST directive.

EXTRA INSTRUCTIONS.

There are a number of Z80 instructions which are not normally shown in Z80 programming manuals because they do not always work! If you are writing software for other Z80 users, DO NOT USE THEM. If you must use them, the PYRADEV Assembler will accept them, but the Directive 'EXTRA' must be given first.

The first group of op-codes allow the general purpose 16 bit IX and IY registers to be used as four 8 bit registers by classifying them as LOW or HIGH order registers. We use operands LX, HX, LY, and HY to represent the Low and High bytes of IX and IY respectively. The alternate form XH, XL, YH and YL may also be used.

```

LD  HX,A LD  HX,B LD  HX,C LD  HX,D LD  HX,E LD  HX,n
LD  LX,A LD  LX,B LD  LX,C LD  LX,D LD  LX,E LD  LX,n

LD  HY,A LD  HY,B LD  HY,C LD  HY,D LD  HY,E LD  HY,n
LD  LY,A LD  LY,B LD  LY,C LD  LY,D LD  LY,E LD  LY,n

LD  A,HX LD  B,HX LD  C,HX LD  D,HX LD  E,HX
LD  A,LX LD  B,LX LD  C,LX LD  D,LX LD  E,LX

LD  A,HY LD  B,HY LD  C,HY LD  D,HY LD  E,HY
LD  A,LY LD  B,LY LD  C,LY LD  D,LY LD  E,LY

LD  HX,LX LD  LX,HX LD  HY,LY LD  LY,HY

INC  HX   INC  LX   INC  HY   INC  LY
DEC  HX   DEC  LX   DEC  HY   DEC  LY

```

| | | | |
|----------|----------|----------|----------|
| ADD A,HX | ADD A,LX | ADD A,HY | ADD A,LY |
| ADC A,HX | ADC A,LX | ADC A,HY | ADC A,LY |
| SBC A,HX | SBC A,LX | SBC A,HY | SBC A,LY |
| SUB HX | SUB LX | SUB HY | SUB LY |
| CP HX | CP LX | CP HY | CP LY |
| AND HX | AND LX | AND HY | AND LY |
| OR HX | OR LX | OR HY | OR LY |
| XOR HX | XOR LX | XOR HY | XOR LY |

A second group of codes provide additional SHIFT-LEFT-LOGICAL opcodes complementing the existing SRL instruction. They are:-

SLL A SLL B SLL C SLL D SLL E SLL H SLL L SLL (HL)

The operations is the same as the SLA instruction, however, a (1) bit is placed into bit position 0 instead of (0).

Please note: The above op-codes are not standard (since there is no formal definition for them) however, they do correspond with mnemonics used by a number of publications concerning Z80 programming. Where possible we have used common definitions.

FREE FORMAT

The use of the FREE directive enables the Assembler to process a type of Z80 expression useful for games writers. It is a mixed data and value expression system suitable for defining tables. Byte generation is always assumed unless a single exclamation symbol (!) precedes an expression in which case a word-value is generated. The use of DBXON is ideal when you first use free format to check that tables are being set up correctly. The following examples show how FREE format can be used...

```

;
;           FREE                ; FREE FORMAT.
;           DBXON              ; DB EXPANSION.
;
FLAG1:     EQU          0
FLAG2:     EQU          1
VALUE1:    EQU          2
;
ROUTINE1:  CALL          GET-CURSOR
;           INC           H
;           CALL          SET-CURSOR
;           RET
;
;           CONTROL      TABLE
;
1,2,3,!100,!200,!300,FLAG0,FLAG1,"STRINGA","ABC1","111",!ROUTINE1
4,5,6,!200,!400,!600,FLAG1,FLAG2,"STRINGB","ABC2","222",!ROUTINE2
7,8,9,!300,!600,!900,FLAG2,FLAG2,"STRINGC","ABC3","333",!ROUTINE3
;
;           END OF CONTROL TABLE
;
;           DBXOFF        ; DB EXPANSION OFF
;
;

```

Each of the three lines in the control table above will be assembled in a similar manner. Here we describe just the FIRST line and what binary code is generated....

Three bytes containing the values 1, 2, and 3.

Three words containing the values 100, 200, and 300. Notice the !.

A byte containing the flag-0 value.

A byte containing the flag-1 value.

A string of bytes containing 'STRINGA'.

A string of bytes containing 'ABC1'.

A string of bytes containing '111'.

A single word with the address of 'ROUTINE1'. Notice the !.

MEMORY MANAGEMENT.

During the assembly process, memory allocation for the symbol table, source code storage, cross reference table and macro definitions varies according to available memory.

On the Amstrad 464 and 664 machines which both have 64K of memory, the symbol table is stored from the bottom of the memory upwards, whilst the optional cross-reference table is stored from the top downwards. In addition a 2K buffer is used to read source code. If you assemble very large programs, at some stage it will not be possible to obtain a cross-reference list as the 64K system will be fully utilised.

On the Amstrad 6128 (or a 464/664 using a RAM addon kit*) the following memory allocation scheme is used:-

- The source code is stored in the extra banks in pass 1, enabling pass 2 to read source from memory, speeding up the assembly process.
- The optional cross-reference table is held in 32K in the second bank of RAM if selected, allowing upto 6,500 entries.
This permits very large assemblies.

* The PYRADEV system has been extensively tested using a CPC464 with a DDI disc drive and CPC6128. Memory sizes from 64K upto a total of 384K have been used by using the DK Tronics 256K Ram Expansion Kit.

Additional RAM above 128K only provides a marginal improvement (about 10-15%) in overall assembly time.

ASSEMBLER STATISTICS.

At the end of pass 2 (object code generation) the following statistics are displayed in order that you can see how close you are to the Assembler's processing limits. The Free Symbol Memory is the critical value, and it must always be 'well above' zero!

Number of Errors nnnn
Number of Symbols nnnn
Symbol Table from nnnn to nnnn
Macro List from nnnn to nnnn
Number of X-Refs nnnn
X-Ref table from nnnn to nnnn
Free Symbol Memory nnnn
File Start: nnnn End: nnnn Length: nnnn

SUMMARY OF ASSEMBLER DIRECTIVES

| | | |
|--------|-------------|-----------------------------------|
| COND | «exp» | Conditional assembly. |
| DBXOFF | | DB expansion Off. |
| DBXON | | DB expansion On. |
| DEFB | «exp»,«exp» | Define bytes. |
| DEFL | «exp» | Redefine a label. |
| DEFM | «exp» | Define memory. |
| DEFS | «exp» | Define storage. |
| DEFW | «exp» | Define single word value. |
| EJECT | | Form Feed. |
| ELSE | | Part of conditional assembly. |
| END | | End of source input stream. |
| ENDBIN | | End binary code generation. |
| ENDC | | End of conditional assembly. |
| ENDIF | | End of conditional (IF) assembly. |
| ENDM | | End of MACRO. |
| EQU | «exp» | Equates a label. |
| EXEC | «exp» | Define EXEC address. |
| EXTRA | | Enable extra opcodes. |
| FREE | | Enable FREE format. |
| IF | «exp» | Conditional (IF) assembly. |
| LIST | | Turn on selective printing. |
| LOAD | «exp» | Generate zero bytes until «exp» |
| MACRO | «parm list» | Start macro definition. |
| MLIST | | Enable macro listing. |
| MNLIST | | Disable macro listing. |
| NOLIST | | Disable selective printing. |
| ORG | «exp» | Define object code address. |
| PRINTR | «parms» | Define printer options. |
| PAUSE | message | Pause Assembly. |
| SELECT | «filename» | Append another file. |
| TITLE | message | Change Listing Header. |

SCREEN DISPLAY.

The Monitor Screen is split into five different sections. The top left section contains trap information.

The middle-left section contains register contents (current and previous), cpu flag settings and the four bytes of memory pointed at by each of the address pair registers DE, HL, IX and IY.

The bottom section is a memory display. It alternates with a menu display.

The very top right of the screen shows the status of the lower and upper ROM sections.

The right hand section of the display is the main dis-assembly display.

RELOCATION.

The Monitor is fully relocatable, and will load itself initially into a high memory address (which it displays). It will then ask if relocation is required. If you reply (Y)es, you may then select the (S)tandard Monitor or (M)ini Monitor.

The Mini-Monitor is sub-set of the Standard-Monitor which can be used when memory space is restricted. The following functions are excluded from the Mini-Monitor: Write-File-to-Disc, Copy-Memory, Help-Menu, Ink-Changes, ROM-Display Screen-Printing and Dis-Assembly to Disc or Printer.

The Monitor will then ask whether you wish to specify a Low Boundary (start address) or High Boundary (end address).

Specifying a Low boundary instructs the Monitor to relocate itself such that its lowest address does not go below the Low-Boundary.

Specifying a High boundary instructs the Monitor to relocate itself such that its highest address does not exceed the High-Boundary.

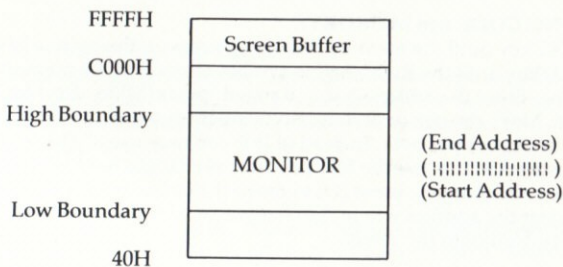


Figure 4.1

MENU DISPLAY.

When ready, the Monitor Menu can be displayed by pressing the ESC key. The menu alternates with the memory display section at the bottom of the screen.

| | | | | | |
|------------|---------------|----------|-----------|-------------|-----------|
| T....TRAP | G..GOTO-ADDR | L...LOAD | Q...QUERY | J...PAPER | C...CAT |
| Z.CLR-TRAP | G...STEP-1ST | W..WRITE | N.NEW-SCN | KK...PEN | T...TAPE |
| M.AMEMORY | S..SSTEP(f1) | O..OTHER | X....ROMS | D...DIS-ASM | «...PRINT |
| R.....REGS | S...DSTEP(f2) | U.UPDATE | Y....COPY | X...EXIT | |

Figure 4.2 Monitor Menu

DEVICE SELECTION.

Press (CTRL-T) to toggle TAPE or DISC operation. The selected device type is shown at the top of the screen and will be used to READ and WRITE files. Press (CTRL-C) to see the device CATALOGUE DISPLAY.

LOADING TEST CODE.

Press (L) to load a program file and enter the file-name. The file header will be examined in order to display the load-point and execution address will be displayed. You can alter the load point before pressing the enter key to read the file into memory if you wish.

DIS-ASSEMBLY DISPLAY.

A Right-Arrow symbol indicates where the address cursor is. Press the (O) key until the Right Arrow symbol is at the top right screen display and enter a dis-assembly address. The display can be scrolled with the arrow and shift arrow keys.

CHANGING CODE and MEMORY.

Use the ESC key until the memory display appears on the bottom of the screen. Use the (O) key until the Right-Arrow symbol appears at the memory display address line. Enter the address to be changed. press (M) to select MODIFY-MEMORY. Move the cursor and over-type memory as required by using two-digit hex-codes per byte. Instead of (M) you may use (CTRL-A) to modify memory using ASCII. Press the ESC key after all changes have been made. If you have made code changes, press (O) to move the cursor back to the top right display, enter the address where the changes were made and the new code will be dis-assembled onto the screen.

The display can be scrolled up and down by 1 byte, 16 bytes or 128 bytes by using the up and down arrow keys in normal, shifted and control states as required.

SETTING TRAPS.

Use the top right display to study code, then press (T) to set a trap. Enter the first byte address of the instruction you wish to trap. One of the five trap lines will be used to show which instruction the trap has replaced. Use the (G)oto function to start executing code. When the trap occurs, the relevant trap line will be displayed in reverse inks. There are two ways to continue from a trap:-

- S – Start single-step code execution from the trap address.
- G – Goto address, the default one continues execution.

When a trap occurs it automatically clears itself. If you want the trap to occur again, you should re-install the trap before continuing.

GOTO ADDRESS.

After traps are set, or have occurred, press (G) start or resume code execution. A resume address will be displayed and can be accepted by pressing the Enter key. To go to a specific execution point, overtype the address shown with one of your own choosing.

STEPPING THROUGH CODE.

You can step through code after a trap occurs, or by setting the first trap point with (CTRL-G).

Press (S) or (f1) to SINGLE-STEP through code. This form of stepping will follow the logical address path (control flow) as the code executes. If a CALL to ROM code occurs, the step operation will discontinue since the step mechanism cannot over-write ROM hardware in order to set the necessary traps.

Press (CTRL-S) for (f2) to DOUBLE-STEP. This form of stepping will set traps in ascending memory locations. This means that when single stepping, if a CALL occurs to a ROM routine, use a double step. This will set a trap on the next memory instruction so that stepping (single or double) can be resumed after the CALL operation is complete and control returns from the ROM routine. AMSTRAD programs normally use many ROM calls to effect system operation.

The MODE (0,1,2) your program uses can be altered using shift 0,1, or 2. When set to 2, it is the same as the MONITOR and so the display screen will NOT be cleared each time a trap or step is executed.

CHANGING REGISTER CONTENTS.

Press (R) to change register contents, then carefully overtype the existing values with the new ones. Press the ESCAPE key to exit from the change function. The new values will be used when code execution is resumed either by using (G)oto or the single (S)tep function.

NEW and UPDATE.

The current memory display or dis-assembly display can be updated by pressing (U). This will be required when monitoring memory which is changing. When code execution is over-painting the screen (making it difficult to read), the (N)ew command can be used to update the entire screen.

QUERY, SEARCH.

Pressing (Q) selects query mode and allows HEX or ASCII of upto 30 characters to be searched for. Wild cards (?) may be used in both ASCII and HEX formats. As each match is found, it is displayed on line two of the lower-screen memory display. The search can be confined between a low and high memory address and can be stopped or continued at each match.

EXAMINE ROMS.

Both the dis-assembly and memory displays read memory according to the LOWER and UPPER ROM selections. Pressing the (X) key allows the status of the ROM selections to be altered.

The display areas will change if they are displaying memory areas below 16K or above 48K as the lower and upper ROMS are switched.

DIS-ASSEMBLE TO DISC, VIDEO OR PRINTER.

Press CTRL-D and enter a START and END address to identify the area of memory to be dis-assembled. Enter a Workspace address where the symbol table can be built. The default address supplied defines the 1000 byte area and may be used in most cases. If the space is exhausted by the dis-assembly process (about 500 labels) an error message will be displayed and the dis-assembly will have to be re-run with an alternate and larger work-space.

Data areas may then be defined by pairs of Start-End addresses. Use a zero-pair to terminate the selections. The output of the dis-assembly process may be selected with (V), (D) or (P) denoting Video, Disc or Printer. Note: References to non-existent labels may occur if embedded data-areas are not defined before the dis-assembly.

If (D) is selected enter the file-name to be used. Dis-Assembly to Video or Print can be paused with the Escape key and resumed with any other key. If the (D) option is used, a single Escape key aborts the process.

Dis-assembly to disc file(s) will pause after each 30K of source code has been generated so that subsequent file names can be entered before the process resumes.

WRITE.

To write a section of memory as a binary file press (W). The load point, execution address and length (saved when the file was loaded) are re-displayed and can be overtyped if required. After entering a file-name the relevant section of memory is written to disc as a binary file.

COPY MEMORY.

A section of memory can be copied by pressing (Y). Define the block to be moved with a BEGIN and END address. Then specify the destination address. The block will be copied 'intelligently'. This means the destination may be anywhere, and may overlap the initial block area.

MEMORY BANK-BLOCK SELECTION.

Press (CTRL-B) to select an alternate block of memory to be mapped into the address range 4000H to 8000H. (128K and larger systems only).

SCREEN-PRINT.

Press the (.) key on the numeric (function) island. The contents of the screen will be decoded and printed. Press the ESC key twice to cancel printing.

WARNING: You should not modify disc sectors directly without first making a backup of the disc. You can easily lose your favourite game, weeks of source code development or the entire disc contents. You must be aware of the AMSTRAD disc structures and file header constructions before changing anything. Please note this warning!

DRIVE.

Press (D) and the prompt 'Drive A or B' will be displayed. Press (A) or (B) to select required drive and the relevant directory will be displayed.

You **MUST** do this if you change the disc(s) being examined!

TRACK and SECTOR.

Press (T) to enter a track address, press (S) to enter a sector address.

READ.

Press (R) to read the selected sector. After a sector is read, the following keys can be used. They provide the ability to follow file chains or read forwards or backwards at sector level.

| KEY: | Normal | Shift | Control |
|---------------|--|--|------------------------------------|
| LEFT RIGHT | ! Chain back 1/2 sec ! Chain frwd 1/2 sec | ! Chain back 1 sec. ! Chain frwd 1 sec. | ! Locate Beginning ! Locate End |
| UP DOWN | ! Previous 1/2 sector ! Next 1/2 sector | ! Previous sector ! Next sector | ! Not used. ! Not used. |

Figure 5.1

MODIFY SECTOR.

Press (M) to modify using HEX or (A) to modify using ASCII. Use the arrow keys to move the cursor to the required bytes and over-key the values as required. If you modify File-Header bytes, after making the changes press (CTRL-H) to re-calculate the checksum byte. When the sector changes are complete, press the ESC key. The changed sector can be written to disc with the (W) command...

WRITE.

Press (W) to write the displayed/modified sector back to disc. The write request must be confirmed with the (Y) key. Any other key aborts the Write Request.

WARNING: The sector will be written to the Track and Sector shown on the bottom right of the screen.

UN-ERASE.

This can be used to re-claim a file which has been accidentally deleted. Press (U) to select the un-erase function and enter a file-name. The DISC-NURSE will check the sector-allocation tables. If the file sectors have not been used, the file will be restored for normal use and will re-appear on subsequent directory displays.

FILE ACCESS.

To access the sectors belonging to a specific file, press (F) and enter a file-name. The first sector of the file will be read. See the (R)ead function above for a description of the scroll/browse key functions.

QUERY.

Pressing (Q) selects the query / search function. The search can be limited to a (S)ector, (F)ile or (D)isc and may be for an ASCII string or a HEX string. Wild Cards (?) are permitted.

The hex string may be entered as a continuous or broken string of hex numbers, ie NNNNNN or NN NN NN.

The search starts from the current track/sector position and proceeds to the last sector. Searching for an unlikely ASCII string is a good way of checking a disc.

When a match is found, the sector address, word offset and sector contents are displayed on the screen. The search can be continued by pressing the SPACE bar, or terminated with the ESC key.

CATALOGUE and EXTENDED DIRECTORY.

Pressing (C) will display the disc catalogue using the standard display format. For extended directory information and hard-copy facilities press (CTRL-C). If the output is to be printed, answer (Y) to the print question. A title line may be entered which will appear at the top of the listing.

SCREEN-PRINT.

Press the (.) key on the numeric (function) island. The contents of the screen will be decoded and printed. Press the ESC key twice to cancel printing.

PAPER and PEN COLOURS.

Use of the (J) and (K) keys will step the paper and pen colours through their next ink values. Experimenting with the inks will improve screen readability on both mono-chrome and colour monitors according to background lighting conditions. CTRL-J and CTRL-K can be used to step through the ink colours in a reverse fashion.

6.0 UTILITIES.

The Utilities Program provides general file management and copy facilities in a single and easy to use package. The following features are provided:-

- Directory Display of Drives (A) and (B).
- File Renaming, using a link to the |REN command.
- File Erasing, using a link to the |ERA command.
- File Transfer; any AMSDOS file type, Tape and Disc.

INITIAL PROMPT.

The following prompt is displayed....

Drive (A), (B), (D)elete, (R)ename, (C)opy or (X) to Exit.

Selection of (A) or (B) displays the appropriate directory. Selection of (D) or (R) provides file DELETE and RENAME functions and need no further explanation. (Wild cards may be used).

COPYING FILES.

The Copy Function is a general purpose copy routine which will copy ANY standard AMSDOS file to and from DISC or TAPE. When selected the following prompt appears:-

1: Disc-Disc 2: Disc-Tape 3: Tape-Disc

A valid reply must be given or control returns to the initial prompt. After a selection of 1, 2 or 3 an INPUT file name must be entered. A file-name is not required for option 3.

After the INPUT file is opened, the header information is displayed. The copy can then be continued by responding (Y)es to the COPY-? prompt.

Depending on the option chosen and file-type detected, the copy operation will go through a number of prompts. The options and valid responses are described on the next page.

NOTE:- The Destination Tape or Disc may be changed BEFORE the reply to 'PROTECTION ?' is entered.

COPY OPTION RESTRICTIONS.

| | Disc to Disc | Disc to Tape | Tape to Disc |
|---|---|--------------|--------------|
| ASCII | Compression? Reply Yes or No. 36K limit *** | | |
| BINARY | 36K limit *** | Note-1 | Note-1 |
| BASIC | 36K limit *** | | |
| CPM | Compression? Reply SPACE-BAR 36K limit *** | | |
| Protection can be INSERTED or REMOVED in all cases. | | | |

Figure 6.0 Copy Options.

NOTE-1: A BINARY file written to tape via the PYRADEV Utility System MUST be restored to disc with the same Utility in order to restore the correct LOAD point address in the File Header. Such files may NOT be RUN until this has been done. They can however be LOADED to a specific memory location from tape and CALLED. (Game Writers BEWARE).

7.0 TRAINING.

As you may have realised, the PYRADEV system, is small but powerful. There are many features and functions to explore and new commands to learn. For this reason the file 'PROGRAM.001' is supplied on the MASTER DISC for you to Assemble, Edit and generally play with.

The program contains a single routine called 'DEBUG'. It can be LOADED and CALLED from BASIC and will display register contents on row 25 of the screen. The program is very heavily commented with two types of text.

The (UPPER CASE TEXT) enclosed in brackets is all about the program code. Hopefully you will quickly see how the routine works, and perhaps adapt it as an additional debug routine for code that you will be writing.

The ;* Normal Text *; enclosed in semi-colons contains information about using the Source File Editor. You should read these comments and practice the functions. When you have finished 'playing' in the Editor, press CTRL-A to abort, then (Y) to confirm and exit.

The file PROGRAM.001 can be Assembled as it is. You should do this at an early stage with the Macro Assembler, and use the various OUTPUT options to see what happens. You are advised to use the default options to start with.

The file PROGRAM.002 is also supplied on disc. This is a complete example of ALL the Z80 instructions. If you have any problems with your code, check that you are using correct syntax, and code mnemonics by looking in this program. It also contains examples of the Assembler Directives described in section 3. It is a useful piece of reference code and can be Assembled and Listed as a reference chart.

The file PROGRAM.003 is a simple FILE-COPY program. It can be modified and used to transfer other ASCII file formats into the PYRADEV system by adding custom code to effect special changes during the transfer/copy operation.

Please ensure you understand how the system operates before using it on live project code, and make sure you always keep backups of development source code.

Thank you for buying PYRADEV. We hope that you will enjoy using it.

8.0 CONTROL KEY SUMMARY.

EDITOR (edit mode)

CTRL-A: Abort and Exit
CTRL-B: Begin Pointer
CTRL-D: Delete Block
CTRL-E: End Pointer
CTRL-F: Find Marker
Island(.): Print
ESCAPE: Help-Display

CTRL-G: Goto Line nnn
CTRL-L: Set Mark
CTRL-O: Open Up Text
CTRL-S: Search System
CTRL-T: Tab Settings
Island(0): KS memories
Small-Enter: Toggle KS Recording On and Off

CTRL-U: Undo Changes
CTRL-V: View Begin/End
CTRL-X: Save and Exit
CTRL-Y: Vary Cursor
CTRL-Z: Command Mode
Island(1-9): Replay KS

EDITOR (command mode)

| | | | |
|----------------|----------------|-----------------|-------------------|
| A: Set Drive A | D: Delete File | L: Load Options | J,CTRL-J: Paper |
| B: Set Drive B | R: Rename File | S: Save Options | K,CTRL-K: Pen Ink |
| C: Compression | Z: ZAP Backups | | |

ASSEMBLER (run time toggle keys)

| | | | |
|-----------------|------------------|--------------------|-------------------|
| V: Video Output | P: Print Listing | S: Selective Print | E: Error Printing |
|-----------------|------------------|--------------------|-------------------|

MONITOR (main menu)

| | | |
|------------------------|---------------------|-------------------------|
| T: Set Trap Point | S: Exec Single Step | Q: Query / Search |
| Z: Clear Trap Point | CTRL-S: Double Step | X: Examine all ROMs |
| R: Change Registers | L: Load Test File | Y: Copy Memory Area |
| M: Modify using HEX | W: Write Disc File | CTRL-D: Dis-Assembly |
| CTRL-A: Modify ASCII | O: Other Display | CTRL-X: Exit Monitor |
| G: Goto an Address | U: Update Screen | J,CTRL-J: Paper Ink |
| CTRL-G: Set Step Addr. | N: Refresh Screen | K,CTRL-K: Pen Ink |
| CTRL-T: TAPE/DISC | CTRL-C: Catalogue | Island(.): Screen Print |
| SHIFT 0,1,2: Set mode | CTRL-B: Bank Select | |

DISC-NURSE (main menu)

| | | |
|-------------------------|-------------------|-------------------------|
| D: Select Disc Drive | R: Read Sector | C: Catalogue Display |
| T: Set Track Value | W: Write Sector | CTRL-C: Extended Dir. |
| S: Set Sector Value | Q: Query / Search | M: Modify using HEX |
| F: Select File | J,CTRL-J: Paper | A: Modify using ASCII |
| U: Un-Erase File | K,CTRL-K: Pen Ink | (CTRL-H: Hddr Checksum) |
| Island(.): Screen Print | | |

UTILITIES (main menu)

| | | |
|--------------------|----------------|---------------------|
| A: Display Drive A | D: Delete File | X: Exit Program |
| B: Display Drive B | R: Rename File | C: Copy AMSDOS File |

UTILITIES (copy menu)

| | |
|-----------------|---------------------------------------|
| 1: Disc to Disc | (Optional ASCII Compression) |
| 2: Disc to Tape | (Protection Insert or Remove) |
| 3: Tape to Disc | (Always displays header information) |

P-Y-R-A-D-E-V

Manuel d'Utilisation

THE
COMPLETE
AMSTRAD
UTILITY

INCLUDES:
EDITOR
ASSEMBLER
MONITOR
DISC-NURSE
UTILITIES
SELF-TEACH

AMSTRAD CPC 464/664/6128

WHEN YOU HAVE MADE THE

DISCOVERY

NOTHING ELSE COMES CLOSE

Manuel d'Utilisation Section 1

PYRADEV

Table des Matieres

| | |
|--|---|
| ○ Introduction | 3 |
| ○ Préparation | 4 |
| ○ Nouveaux utilisateurs | 5 |
| ○ Disques de travail | 5 |
| ○ Restrictions imposées par le système | 5 |

Section 2

Editeur

| | |
|---|----|
| ○ Sélection des fichiers | 6 |
| ○ Affichage mode édition | 6 |
| ○ Page SOS-ESCAPE | 6 |
| ○ Session ABANDON Edition | 7 |
| ○ File SUSPENSION & SORTIE | 7 |
| ○ Mouvement du curseur | 8 |
| ○ Défilement de l'écran | 8 |
| ○ Insertion/effacement caractères | 8 |
| ○ Insertion/effacement lignes | 8 |
| ○ Copie blocs, effacement | 9 |
| ○ Taquets de tabulation | 9 |
| ○ Annexation de fichiers | 10 |
| ○ Recherche, Arrêt, Remplacement | 10 |
| ○ Mémoire de Frappes | 11 |
| ○ Impression | 12 |

| | |
|---|----|
| ○ Affichage répertoire | 12 |
| ○ Renommer fichiers effacés | 13 |
| ○ Mode compression | 13 |
| ○ Changement couleurs encres | 13 |
| ○ Fonctions sauvegarde : chargement | 13 |
| ○ fichiers ZAP | 13 |

Section 3

Assembleur

| | |
|--------------------------------------|----|
| ○ Options Début/Exécution | 13 |
| ○ Commande assemblage | 14 |
| ○ Définition Langage | 18 |
| ○ Expressions | 19 |
| ○ Directives | 19 |
| ○ Assemblage conditionnel | 23 |
| ○ Macro-Instructions | 24 |
| ○ Codes opérations LIBRES | 25 |
| ○ Gestion des mémoires | 28 |
| ○ Statistiques d'assemblage | 28 |
| ○ Récapitulatif des directives | 29 |

Section 4

Moniteur

| | |
|--|----|
| ○ Affichage de l'écran | 31 |
| ○ Translation | 31 |
| ○ Code Chargement test | 32 |
| ○ Changement code, mémoire | 32 |
| ○ Définition pièges | 33 |
| ○ GOTO adresse | 33 |
| ○ Etude pas à pas du code | 33 |
| ○ Code, pas double | 33 |
| ○ Changement registres | 33 |
| ○ Régénération de l'écran | 34 |
| ○ Mode Interrogation/recherche | 34 |
| ○ Mise en route et affichage ROM | 34 |
| ○ Ecriture binaire sur disque | 34 |
| ○ Désassemblage à source | 34 |
| ○ Copie mémoire | 35 |
| ○ Impression écran | 35 |
| ○ Couleurs papier et plume | 35 |
| ○ Affectation zones d'empilage | 35 |
| ○ Accès externe | 35 |

Section 5

Disc-Nurse

| | |
|--|----|
| ○ Prise en charge, disque | 35 |
| ○ Sélection Unité de disques | 36 |
| ○ Sélection piste | 36 |
| ○ Sélection secteur | 36 |
| ○ Lecture secteur | 36 |
| ○ Modification secteur | 37 |
| ○ Ecriture secteur | 37 |
| ○ Non-effacement fichier | 37 |
| ○ Sélection fichier | 37 |
| ○ Mode interrogation/reche-che | 37 |
| ○ Impression écran | 38 |
| ○ Couleurs Papier/plumes (fond/graphiques) | 38 |

Section 6

Programme Utilitaire

| | |
|-------------------------------|----|
| ○ Effacement fichiers | 38 |
| ○ Renommer fichiers | 38 |
| ○ Copie fichiers | 38 |
| ○ Restrictions de copie | 39 |

Section 7

Programme d'entraînement

| | |
|------------------------------------|----|
| ○ Programme de démonstration | 40 |
| ○ Programme Test Assemblage | 40 |
| ○ Informations générales | 40 |

Section 8

| | |
|---|----|
| ○ Récapitulatif des touches de commande | 41 |
|---|----|

1.0 INTRODUCTION

Le PYRADEV est un ensemble de cinq programmes qui, regroupés, fournissent un environnement sûr et confortable pour l'élaboration des logiciels AMSDOS et CPM sur les micro-ordinateurs AMSTRAD 464, 664 et 6128 équipés d'un système de disques DD1. Le système est basé sur l'AMSDOS et n'exige aucune mémoire ROM ou RAM supplémentaire. Il peut prendre en charge d'importants projets de programmation sur systèmes à une ou deux unités, et comporte les programmes suivants :

- Un EDITEUR DE FICHER SOURCE ultra-rapide, pleine page.
- Un MACRO ASSEMBLEUR/éditeur de liens Z80, multi-fichiers.
- Un puissant MONITEUR et DESASSEMBLEUR de mise au point.
- Un programme DISC-NURSE d'une grande simplicité d'utilisation.
- Un excellent programme utilitaire de GESTION DE FICHIERS.

L'EDITEUR utilise ses propres sous-programmes de lecture et d'écriture de disques spéciaux, ce qui permet d'extraire d'éditer rapidement d'importants fichiers (jusqu'à 32.300 octets). Le code peut être recherché, remplacé, entré, supprimé, déplacé, copié et fusionné (entre les fichiers). En outre, on peut effacer, rebaptiser, supprimer, etc les fichiers. Tous les programmes utilisent la compression des données afin de garantir un emploi optimal de la mémoire AMSTRAD et de la capacité des disques.

L'ASSEMBLEUR permet de sélectionner jusqu'à 32 fichiers dans un seul assemblage. Ces fichiers peuvent se trouver sur des disques différents (si l'on utilise deux unités), ce qui permet de traiter en un seul fichier binaire jusqu'à 992K de code source (5 disques environ). On peut commuter sur une sortie vidéo ou sur imprimante selon les besoins, et produire des tables de correspondance. L'assembleur est extrêmement rapide et peut traiter, sur les systèmes plus grands, la mémoire RAM supplémentaire est utilisée au maximum afin d'augmenter les performances.

Le MONITEUR auto-translatable peut charger, interrompre, piéger, fonctionner en pas à pas, ou en deux pas, modifier et ré-écrire le code sur disque. Par ailleurs, on peut sélectionner et analyser des mémoires ROM. En mode de mise au point, on peut modifier les registres et obtenir à tout moment des impressions sur écran. Le désassemblage du code peut être sorti sur imprimante ou mis en fichiers ASCII.

Le programme DISC-NURSE traite les disques système, données et style IBM. On peut rechercher, visualiser, imprimer, modifier et ré-écrire des secteurs de fichiers. La recherche peut également, si besoin est, s'étendre sur la totalité du disque. Chaque fois que le système lit un secteur, l'article principal du fichier est affiché. On peut récupérer les fichiers précédemment effacés si l'affectation ultérieure des secteurs n'a pas dégradé données. Une fonction de répertoire étendu permet de faire imprimer le répertoire de disques avec un message d'en-tête, dans un format détaillé qui convient aux listes de gestion interne des disques.

Le programme UTILITAIRE (gestion de fichiers) offre des fonctions de visualisation, de suppression, de changement de nom et de copie sur disques et bandes et à partir de disques et bandes des répertoires, le tout regroupé en un programme unique et extrêmement simple. Les en-têtes de fichiers d'entrée sont toujours affichés et on peut introduire ou éliminer la protection dans toutes les opérations de copiage. Ceci constitue donc la manière la plus facile de gérer TOUS les fichiers AMSDOS standards.

1.1 PREPARATION

Le PYRADEV est fourni sur un DISQUE D'exploitation sur lequel on ne doit jamais écrire. Il ne devrait être utilisé que pour réaliser des copies de TRAVAIL à l'aide du programme utilitaire CPM DISCOPY ou DISCKIT3. Consulter le manuel approprié et utiliser le PYRADEV MASTER DISC comme disque émetteur pour créer un disque de TRAVAIL ou disque récepteur.

Après avoir fini de copier, retirer le PYRADEV MASTER DISC de l'unité et le ranger dans un endroit sûr.

1.2 MISE EN ROUTE

Placer le disque de TRAVAIL dans l'unité de disque A et vider la machine en appuyant sur CTRL-SHIFT-ESCAPE. Entrer RUN PYRADEV et appuyer sur la grande touche ENTER. Le MENU DU SYSTEME apparaît. La quantité de mémoire disponible affichée en bas et à droite de l'écran.

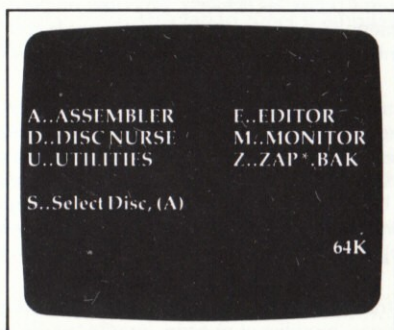


Figure 1.0 menu du système

- | | |
|--------------------------------|---------------|
| A. = ASSEMBLEUR | E. = EDITEUR |
| D. = DISC-NURSE (GESTION) | M. = MONITEUR |
| U. = PROGRAMMES UTILITAIRES | Z. = ZAP*.BAK |
| S. = SELECTION DISQUE, (A) | |

Les programmes du système se sélectionnent par appui sur la touche unique appropriée «A, E, D, M, U». Ils reviennent tous au menu ci-dessus lorsqu'on sort d'un programme.

L'option «Z» efface tous les fichiers de sauvegarde (*.BAK) sur le disque implicite en cours d'emploi. N'utiliser cette option qu'avec grand soin.

L'option «S» vous permet de modifier le réglage de l'unité de disques implicite sur (A) ou (B) pour les fichiers données/source. Le disque PYRADEV doit toujours se trouver dans l'unité de disques (A) car il s'attend à y trouver ses programmes, quel que soit le disque implicite. NE PAS utiliser l'option «S» dans les systèmes mono-unité.

1.3 NOUVEAUX UTILISATEURS

Lorsque le MENU DU SYSTEME apparaît, appuyer sur «E» pour sélectionner l'Editeur. Entrer le nom PROGRAM.001 comme fichier d'entrée et appuyer deux fois sur la barre d'espace. Le fichier est alors lu en mémoire. Parcourir le fichier à l'aide des touches flèches UP et DOWN (VERS LE HAUT et VERS LE BAS). Le fichier est un petit programme qui contient des quantités de commentaires et d'exemples qui vous expliquent comment utiliser les programmes Editeur et Assembleur.

1.4 NOUVEAUX DISQUES

Quand on crée un DISQUE DE TRAVAIL, ce disque peut contenir des fichiers de système superflus. Appuyer sur «U» pour sélectionner le programme Utilitaire, puis appuyer sur «A» ou «B» pour visualiser le répertoire du disque. On peut utiliser la commande «D» (delete) pour éliminer des fichiers. Cette liste décrit les fichiers du système.

| | |
|-------------|--|
| PYRADEV.BIN | MENU DU SYSTEME, nécessaire sur -TOUS- les disques de Travail. |
| PYRAMED.BIN | Editeur page pleine, pour édition du Programme Source. |
| PYRAMAS.BIN | Macro Assembleur multi-fichiers Z80. |
| PYRAMON.BIN | Moniteur de Texte et Désassembleur. |
| PYRADSC.BIN | "Disc-Nurse" pour modification des Secteurs. |
| PYRAUTL.BIN | Utilitaires fichier pour Copie d'ensemble. |
| PROGRAM.001 | Programme Source de Démonstration pour l'entraînement. |
| PROGRAM.002 | Exemples de Code Assembleur et de Fichier de Test. |
| PROGRAM.003 | Simple programme de copie de fichiers ASCII. |

Après avoir supprimé les fichiers non-essentiels, le nouveau disque de Travail est prêt à l'emploi. La suppression de fichiers est également possible dans le programme Editeur.

Restrictions Imposées Par Le Système

Le PYRADEV utilise les points d'implantation 40 hexa à 4F hexa pour faire passer des variables dans ses programmes. Veuillez éviter d'utiliser ces zones durant l'élaboration et l'essai d'un programme.

Une fois lancé, le PYRADEV initialise toutes les mémoires ROM de fond en appelant le sous-programme KL.ROM.WALK. Ensuite, le système définit sa propre valeur HIMEM basée sur la valeur HL fournie. Idéalement, seule la DISC.ROM devrait être connectée. Cependant, l'adjonction d'autres mémoires additionnelles (ADD.ONS) ne devrait entraîner de problèmes que si elles utilisent d'importantes zones de la mémoire RAM. La zone de travail de l'Editeur occupe normalement 32.300 octets environ.

Quand vous utilisez le MONITEUR, n'oubliez pas que toutes les ROM de fond raccordées à l'AMSTRAD ont été initialisées. En initialisant deux fois une ROM, on risque d'obtenir des effets indésirables.

2.0 EDITEUR DE FICHIER SOURCE

L'Editeur a été conçu pour offrir une élaboration et une édition de programmes extrêmement rapides. On le sélectionne dans le Menu du Système en appuyant sur (E). Une fois chargé, il affiche le répertoire de disques implicite, puis attend que vous entriez les noms de fichiers d'Entrée et de Sortie. L'Editeur revient toujours à l'Ecran Inactif suite à une session d'édition.

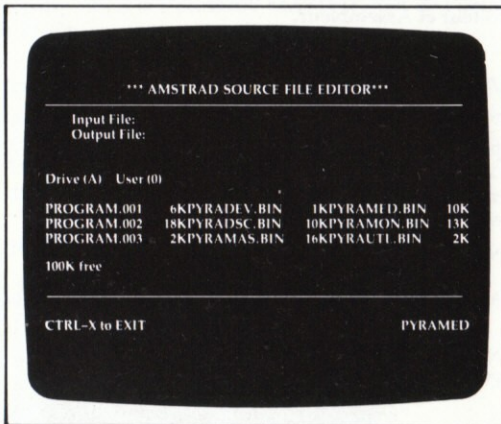


Figure 2.0 Fichier Inactif
(Voir légende page 1a)

- Pour VISUALISER UN fichier : Entrer simplement un nom de FICHIER D'ENTREE
- Pour CREER un fichier : Entrer simplement un nom de FICHIER DE SORTIE
- Pour MODIFIER un fichier : Entrer les noms des fichiers d'ENTREE et de SORTIE
- Pour VISUALISER le répertoire : Appuyer ESPACE ESPACE (c'est-à-dire sans noms).
On peut donner au nom du fichier d'entre le préfixe A: ou B: pour déterminer une nouvelle unité de disques implicite.

Figure 2.0

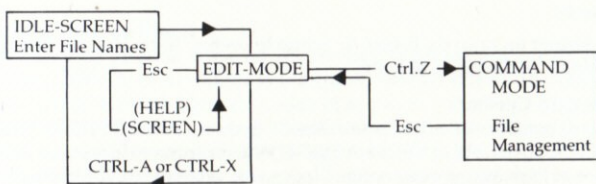
- | | |
|------------------------------|-----------------------|
| 1. EDITEUR DE FICHIER SOURCE | 5. Utilisateur |
| 2. Fichier d'entrée | 6. Libres |
| 3. Fichier de sortie | 7. CTRL-X POUR SORTIR |
| 4. Unité de disques | |

2.1 MODE D'EDITION

Après avoir entré les noms de fichiers, le système lit le fichier d'ENTREE et affiche les 24 premiers enregistrements. Si vous avez sélectionné le mode CREATE (CREER), l'écran demeure vide.

L'emploi de la touche ESCAPE ou de la touche CONTROL-Z fait alterner la Session d'Édition entre le Mode d'Édition, la Page "SOS" (Help-Page) et le Mode de Commande.

1. **ECRAN INACTIF** — IDLE-SCREEN
Entrer noms fichiers — Enter File Names
2. **Changement** — Esc
3. **MODE EDITION** — EDIT-MODE
4. **MODE COMMANDE**—COMMAND MODE
5. **Gestion de fichiers** — File Management
6. **PAGE "SOS"** — (HELP SCREEN)



```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
261
```

Save & Exit (Sauvegarde & Sortie)

La sortie normale se fait par la touche (CTRL-X). Le nom du fichier de sortie est alors affiché et peut être recouvert selon les besoins. Le fichier sera écrit sur disque et le contrôle repasse alors à l'Écran Inactif. En cas d'échec d'écriture de fichier, le message **** FILE SAVE FAILED **** apparaît et le message sélectionne le Mode de Commande, ce qui vous permet de sélectionner le disque (c'est-à-dire d'effacer les fichiers superflus) ou d'utiliser un autre disque. Suite à une sauvegarde réussie, l'Éditeur repasse à l'Écran Inactif.

On peut donner au nom du fichier de Sortie le préfixe A: ou B: pour définir une nouvelle unité de disques implicite.

Mouvement du Curseur

Le curseur se contrôle à l'aide des touches de flèches VERS LE HAUT, VERS LE BAS, VERS LA GAUCHE et VERS LA DROITE ou en entrant un texte. Le défilement vers l'Avant et l'Arrière est automatique lorsque le système atteint la rangée-1 ou la rangée-24. Le texte est TOUJOURS recouvert par la frappe à l'endroit où se trouve le curseur. Des enregistrements sont ajoutés en tête ou en queue du fichier si l'on déplace le curseur au-delà des premier ou dernier enregistrements à l'aide des flèches vers le haut et vers le bas. La touch ENTER vous envoie toujours à la ligne suivante, colonne un. Les touches TAB et CTRL-TAB déplacement respectivement le curseur jusqu'à la tabulation suivante ou précédente.

Vitesse du Curseur

On peut faire déplacer le curseur alternativement rapidement et lentement, à l'aide de la touche (CTRL-Y). Chaque fois que l'on change de vitesse, la demande (Flash?) apparaît. Répondez (Y) si vous désirez un curseur clignotant ou (N) pour un curseur à éclairage constant.

Défilement de l'Écran

La flèche de DECALAGE VERS LE BAS et la flèche de DECALAGE VERS LE HAUT font défiler respectivement l'écran d'une ligne vers le haut ou vers le bas, tout en conservant la position due curseur. Les flèches CTRL-VERS-LE-BAS et CTRL-VERS-LE-HAUT font défiler l'écran et vers l'arrière, 24 lignes à la fois. Ces quatre fonctions ne font pas défiler l'écran au-delà du début ou de la fin du fichier.

Character-Insert-Delete (Insertion/Effacement de Caractères)

Les flèches de DECALAGE VERS LA DROITE et de DECALAGE VERS LA GAUCHE (SHIFT-RIGHT/SHIFT-LEFT) assurent ces fonctions. Nous soulignons que la fonction d'insertion place un espace vierge unique dans le texte, espace sur lequel on peut ensuite taper. Les actions d'Insertion et d'Effacement surviennent toutes deux à l'endroit où se trouve le curseur. Les touches (CLR) et (DEL) permettent également d'effacer des caractères ou d'effacer/inverser (comme en BASIC).

Line-Insert-Delete (Insertion/Effacement de Lignes)

Les flèches CTRL-DROITE et CTRL-GAUCHE (CTRL-RIGHT/CTRL-LEFT) assurent ces fonctions; elles agissent toutes deux sur la ligne en cours, ligne qu'indique le curseur. L'insertion de ligne introduit une ligne vierge dans le texte, ligne sur laquelle on peut taper. Lorsqu'on tape le texte de la fin de la ligne, le système insère une autre ligne vierge. La même chose se produit si vous appuyez sur la touche ENTER. Pour annuler le Mode d'Insertion de Lignes, utilisez la touche CTRL-LEFT-ARROW (pour effacer une ligne vierge nouvellement créée) ou faites défiler au-delà de la rangée 24.

Open Space (Ouverture d'un Espace)

Vous pouvez utiliser la touche (CTRL-O) pour ouvrir ou diviser une ligne à l'endroit où se trouve le curseur, et constituer ainsi deux nouvelles lignes. La première contiendra le texte qui figurait jusqu'à la position du curseur, et la seconde, le texte qui figurait à partir de la position du curseur. Cette touche est utile lorsque l'on veut insérer un nouveau code sur une ligne qui contient déjà un label.

Block-Copy (Copie de Blocs)

Pour copier un bloc de code, amenez le curseur à la première ligne du code et appuyez sur (CTRL-B) pour définir le marqueur BEGIN (DEBUT). Amenez le curseur jusqu'à la dernière ligne du code et appuyez sur (CTRL-E) pour définir le marqueur END (FIN). Le message 'Block Saved' apparaît. Placez le curseur à l'endroit où vous voulez copier le bloc et appuyez sur la touche (COPY). Le bloc sera COPIE-INSERE sur la ligne suivante. Le bloc sauvegardé demeurera en mémoire jusqu'à ce que l'on marque un autre bloc. On peut le copier à plusieurs reprises dans n'importe quel point du fichier, sur un autre fichier ou sur un autre disque.

Block-Delete (Effacement de Blocs)

Marquer le Bloc à l'aide des touches (CTRL-B) et (CTRL-E) comme pour la fonction de Copie de Blocs. Lorsque le message 'Block Saved' apparaît, appuyez sur la touche (CTRL-D). Ceci fait afficher le bloc en ENCREES inverses. Appuyez sur (Y) pour autoriser l'opération d'effacement. Précisons que, suite à un effacement, le bloc demeure sauvegardé et peut être recopié, si nécessaire, au moyen de la touche (COPY).

Block-Move (Mouvement de Blocs)

Constitué par la fonction EFFACEMENT DE BLOCS, suivi de l'appui sur la touche (COPY), comme ci-dessus.

Tab-Stops (Taquets de Tabulation)

Un appui sur (CTRL-T) fait afficher les taquets de tabulation en cours d'utilisation sous forme de petits triangles sur la ligne 24. Les tabulateurs sont posés ou supprimés en modifiant la position du curseur et en appuyant sur la touche (TAB) jusqu'à ce que le symbole triangulaire apparaisse ou disparaisse. Un appui sur «CTRL-TAB» supprime tous les taquets de tabulation. Lorsque vos nouveaux taquets sont prêts, appuyez sur la grande touche ENTER pour les enregistrer. Les tabulations peuvent être sauvegardées pour emploi ultérieur – voir Mode de Commande.

Mark-Find-Line (Marquage-Recherche de Lignes)

On peut marquer une seule LIGNE en utilisant la touche CTRL-L. On peut ensuite la ramener sur l'écran à l'aide de la touche CTRL-F (FIND). Ces deux opérations sont utiles en ce qu'elles permettent de revenir rapidement au code entré après avoir étudié un code qui se trouve dans une autre partie du fichier.

View-Begin-End (Debut-Fin-Visualisation)

Appuyez sur CTRL-V pour sélectionner 'View' puis appuyez sur (B) pour voir le début du fichier (enregistrement-1) ou (E) pour voir la fin du fichier (dernier enregistrement).

Go To Record (Passage à un Enregistrement)

Un appui sur CTRL-G permet d'entre un numéro d'enregistrement. L'Editeur passe immédiatement à cet enregistrement. L'emploi d'un zéro ou d'un chiffre supérieur à celui du dernier enregistrement font respectivement afficher le Début ou la Fin du fichier. La fonction CTRL-G est utile pour retrouver un code programme dans le Listage de l'Assembleur.

New-Input (Nouvelle Entree)

L'emploi de la fonction CTRL-N permet d'ajouter un autre fichier INPUT (ENTREE) au fichier qui se trouve actuellement dans la zone de travail de la mémoire. Il est ainsi possible d'élaborer de nouveaux programmes ou documents à partir d'un code ou d'un texte précédent. Après adjonction du nouveau fichier dans la mémoire, les fonctions Block Delete (Effacement Bloc) et Move (Mouvement) peuvent rapidement amener le code ou le texte à l'endroit nécessaire.

Cette fonction ne peut s'utiliser que si l'on dispose encore au minimum de 1028 octets.

Search-System (Systeme de Recherche)

CTRL-S sélectionne le système de recherche et offre deux types d'opérations.

Search and Stop (Recherche et Arrêt)

Entrez une chaîne de recherche entre apostrophes ou guillemets, puis appuyez deux fois sur la touche ENTER, en laissant vide la chaîne de remplacement. La recherche débute. Chaque fois que la chaîne est retrouvée, l'enregistrement qui la contient est affiché, accompagné des enregistrements précédent et suivant, dans sa propre 'fenêtre'. Vous pouvez ensuite utiliser les options suivantes:

- Touche Enter... Ignorer cette correspondance et continuer à chercher.
- Touche Escape... Terminer et revenir au texte initial.
- Touche (G)..... Passer à l'enregistrements dans lesquels la concordance a été trouvée.
- Up/Down
(Vers le haut/bas). Faire défiler les enregistrements dans lesquels la concordance a été trouvée.

La fonction Recherche et Arrêt permet d'obtenir rapidement la position du code programme à l'aide de labels de code source qui sont (généralement) uniques.

Search and Replace (Recherche et Remplacement)

Entrez une chaîne de recherche entre apostrophes ou guillemets. Entrez ensuite une chaîne de remplacement également entre apostrophes ou guillemets. N'oubliez pas d'utiliser selon les besoins, les majuscules ou les minuscules selon la chaîne de remplacement. Après avoir entré la chaîne de remplacement, répondez à la question concernant la sélection d'un remplacement Automatique ou Conditionnel: la recherche débue.

Si vous avez choisi l'option de Remplacement Automatique, l'opération se poursuit jusqu'à la fin sans aucune autre intervention. Si vous avez utilisé l'option Conditionnelle, chaque fois que la chaîne est trouvée, elle sera affichée, accompagnée des enregistrements précédent et suivant. Appuyez sur (R) pour effectuer le REMPLACEMENT, ou sur (C) pour CONTINUER, sans autre intervention.

Toutes les fonctions de recherche fonctionnent à partir de la ligne en cours jusqu'à la fin de fichier. Utilisez CTRL-V B pour sélectionner l'enregistrement 1 si vous désirez examiner le fichier complet. Une bonne séquence de frappe à utiliser avant l'emploi de la fonction de recherche est CTRL-L CTRL-V B. Ceci repère la ligne en cours et assure le passage au début du fichier. Après l'emploi de la fonction de recherche, la frappe CTRL-F renvoie à la ligne repérée ou marquée.

Vous pouvez annuler n'importe quelle recherche avec la touche ESCAPE.

Note: Il surviendra une troncature si une chaîne de remplacement est plus longue que la chaîne source et si les caractères sont forcés au-delà de la colonne 80.

Keystroke-Memories (Mémoires de Frappes)

L'Éditeur contient neuf mémoires de frappes qui peuvent chacune enregistrer 32 frappes. Vous pouvez les utiliser pour enregistrer des touches de commande ou de texte. Vous pouvez ensuite les "relire" selon les besoins.

Visualisation

Pour visualiser le contenu des mémoires de frappes, appuyez sur la touche 0 (f0) du bloc numérique. Les mémoires seront affichées. Appuyez sur la touche ESCAPE pour revenir en Mode d'Édition.

Record (Enregistrement)

Appuyez sur la touch SMALL-ENTER et sélectionnez une mémoire à l'aide des touches 1 à 9 du bloc numérique. Après sélection, le message 'Keystroke Recording ON' sera affiché. Pour terminer l'enregistrement, appuyez à nouveau sur la petite touche Enter. Si un enregistrement exède la limite des 32 touches, la mémoire suivante sera raccordée et utilisée. Si vous atteignez la fin de la mémoire de frappes neuf, l'enregistrement cesse et le message KS-sequence-saved (séquence mémoire de frappes sauvegardée) est affiché.

Replay (Relecture)

Pour relire une séquence enregistrée, appuyez sur la touche du bloc numérique que vous avez utilisée pour faire l'enregistrement (1 à 9). Les touches sont reprises et utilisées comme si vous veniez de les entrer au clavier. Pour faire cesser la relecture, appuyez sur la touche ESCAPE.

Erase (Effacement)

Appuyez sur CTRL-SMALL-ENTER et sélectionnez une mémoire à l'aide des touches 1 à 9 du bloc numérique. La mémoire sera vidée et vous allez pouvoir la réutiliser.

Note 1: Les fonctions de mémoires ne peuvent être enregistrées ou initiées pour lecture qu'en Mode d'Édition normal. Les séquences enregistrées peuvent, par exemple, sélectionner le système de Recherche et lancer une recherche de chaîne de caractères, puis revenir en Mode Edition. Il est cependant impossible de lancer un enregistrement ou une relecture de mémoire de frappes si l'on a déjà sélectionné manuellement le système de Recherche.

Note 2: Vous pouvez sauvegarder les mémoires de frappes. Voir Mode de Commande.

Printing (Impression)

En appuyant sur la touche du point décimal (virgule en Français) du bloc numérique, on fait commencer l'impression à partir de la ligne en cours. En appuyant deux fois sur cette touche, on fait cesser les changements de lignes déterminés par les retours de curseur. L'impression va continuer jusqu'à la fin du fichier. Un appui sur la touche ESCAPE fait cesser l'impression, que vous pouvez ensuite poursuivre avec n'importe quelle touche. Deux appuis sur la touche ESCAPE annulent l'opération d'impression.

On peut inclure des codes d'impression spéciaux dans le texte en cours d'impression, afin de contrôler l'imprimante de la manière suivante...

Pendant l'impression, le symbole flèche vers le haut (qui se trouve sur la touche du signe livre Sterling) signale que le caractère suivant est un symbole de contrôle. La valeur 33 est soustraite à la valeur ASCII du caractère. Si elle est négative, le caractère est ignoré sans quoi il est envoyé à l'imprimante. Un symbole flèche vers le haut à la fin d'une ligne fait sauter le caractère CR invisible, ce qui permet d'obtenir une longue ligne d'impression à partir de deux lignes sur l'écran.

Exemples:

⊠«E représente ESCAPE-E,
⊠0 représente code 15,

⊠«S représente ESCAPE-S,
⊠/ représente code 14.

Command Mode (Mode de Commande)

Vous sélectionnez le Mode de Commande avec la touche CTRL-Z quand vous êtes en Mode d'Édition. La Session d'Édition est provisoirement suspendue, tandis que les sous-programmes de Gestion de Fichiers et d'Options deviennent disponibles. Pour sortir du Mode Commande, utilisez la touche Escape.

Un petit menu est affiché en bas de l'écran; vous sélectionnez des fonctions en appuyant sur les touches appropriées, comme suit:-

| | |
|---------------------------|-----------------------------|
| A,B: Selection Disque | C: Compression/condensation |
| D,R: Effacer, Rebaptiser | Z: Effacement fichiers |
| L,S: Options de charge | sauvegarde Zap |
| J,K: Enres (papier/plume) | S: Sauvegarde Options |

A,B: Sélection Disque A ou B

Un appui sur (A) ou sur (B) fait commuter l'unité de disques implicite et affiche le répertoire, les noms de fichiers et l'espace libre du nouveau disque.

C: Compression (Condensation) Operante/Inoperante

Si vous êtes en train d'éditer et d'Assembler un code dans le système PYRADEV, vous devez maintenir cette option OPERANTE. L'Éditeur et Assembleur vont en effet utiliser la compression (ou condensation) des données de manière à ce que vos fichiers demeurent aussi courts que possible. Si vous employez l'Éditeur pour éditer des fichiers en BASIC avant d'utiliser CTRL-X pour ré-écrire le fichier sur le disque, vous devez rendre la compression INOPERANTE. Le fichier sera alors ramené à sa longueur normale et peut ensuite être utilisé par le système BASIC.

D: Delete File (Effacement Fichier)

Cette fonction est analogue à la directive 'ERA' BASIC/CPM. Elle peut s'utiliser pour effacer un seul fichier ou un groupe de fichiers. On peut sélectionner une unité de disques spécifiée en utilisant le préfixe A: ou B:.

Exemples:

| | |
|--------------|---|
| TEMPFILE.001 | - effacer le fichier |
| TEMP.* | - effacer tous les fichiers TEMP. |
| TEMP????.* | effacer les noms de fichiers débutant par TEMP. |
| B:TEMP.FIL | - effacer TEMP.FIL de l'unité B. |

R: Rename File (Rebaptiser Fichier)

On peut remplacer le nom d'un ANCIEN fichier par celui d'un NOUVEAU fichier si le nouveau fichier n'existe pas encore.

J: Jump Paper Ink (saut de couleur de fond)

Chaque fois que vous appuyez sur la touche (J), l'Encre à Papier (= le fond) passe à sa valeur suivante. Un appui sur (CTRL-J) exerce l'effet contraire.

K: Jump Pen Ink (saut de couleur écriture/graphismes)

Chaque fois que vous appuyez sur la touche (K), l'Encre (= couleur écrit.graph.) passe à sa valeur suivante. Un appui sur (CTRL-K) exerce l'effet contraire.

S: Save User Options (Sauvegarde des Options Utilisateur)

Un appui sur (S) fait écrire les Tabulations, les valeurs Papier et Encre et les Mémoires de Frappes en cours d'utilisation dans un fichier sur disque pour emploi ultérieur. On peut entrer un nom de fichier quelconque. L'extension de fichier (suffixe) suggérée est -OPT. On peut utiliser plusieurs Fichiers d'Options différents pour définir divers 'environnements'.

L: Load User Options (Charger Options Utilisateur)

L'emploi de la touche (L) permet de créer un fichier d'"Options" avec la commande (S)ave pour remise en oeuvre dans le cadre de la Session d'Edition en cours. Après lecture du fichier, les nouveaux Tabulateurs, ainsi que les nouvelles Encres et Mémoires de Frappes seront opérants.

*Z: Zap *.BAK Files (fichiers Zap *.BAK)*

Efface tous les fichiers de sauvegarde du répertoire du disque implicite. Cette fonction est identique à l'emploi de la fonction (D)elete mais avec le nom de fichier *.BAK.

3.0 MACRO ASSEMBLER (MACRO ASSEMBLEUR)

Le Macro Assembleur PYRADEV accepte le langage de programmation ZILOG Z80 (avec quelques modifications et extensions) et peut traiter un ou plusieurs fichiers de programme source pour créer un fichier de chargement binaire, directement sur un disque. On le sélectionne à partir du MENU DU SYSTEME en appuyant sur (A). Après chargement, il affiche le répertoire du disque implicite et attend que les diverses RUN OPTIONS (OPTIONS D'EXECUTION) soient entrées....

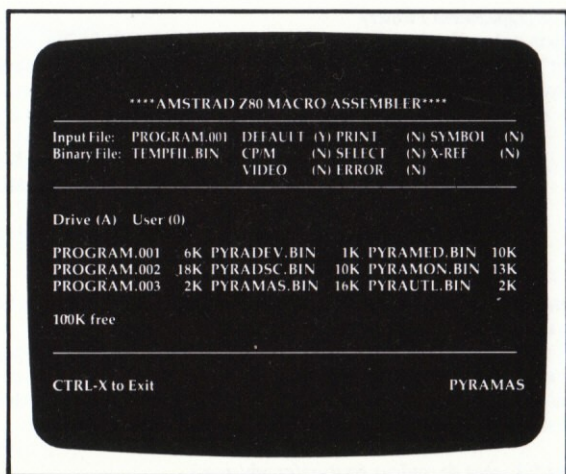


Figure 3.0 Options d'Exécution de l'Assembleur

INPUT FILE = FICHIER D'ENTREE
BINARY FILE = FICHIER BINAIRE
DEFAULT = IMPLICITE
PRINT = IMPRESSION

DRIVE = UNIT DE DISQUES
USER = UTILISATEUR
100 FREE = 100 LIBRES

Input-File (Fichier d'Entrée)

Nomme le premier fichier source de la chaîne d'entrée. Il peut s'agir d'un programme complet ou du premier d'un grand nombre de fichiers qui vont être regroupés pour constituer le programme de chargement binaire. On peut utiliser des instructions SELECT intégrées dans les fichiers de code source pour ajouter d'autres fichiers au train d'entrée. L'un des emplois du système SELECT consiste à n'entrer dans le premier fichier que des instructions SELECT. Un bon suffixe à donner à ce fichier est .CTL car ceci représente un fichier d'ORDRES (CONTROL file). Un nom de fichier binaire implicite avec suffixe .BIN est créé à partir du nom de fichier d'entrée. L'inclusion d'une déclaration END (FIN) dans le train d'entrée est optionnelle mais, s'il en existe une, elle terminera immédiatement le train d'entrée, même s'il existe d'autres fichiers.

Binary-File (Fichier Binaire)

Ceci est le nom qui doit être utilisé lorsqu'on catalogue sur disque le fichier à code binaire. Ce fichier est établi à partir du nom du fichier d'entrée (par adjonction d'un suffixe .BIN) et peut être recouvert. Un appui sur la touche ENTRE le laisse tel quel. Si vous n'avez pas besoin d'un fichier binaire, remplissez cette zone de blancs.

Defaults (Valeurs Implicites)

Si vous appuyez sur la touche Enter ou sur (Y), l'Assembleur va utiliser les valeurs implicites affichées et lancer le processus d'assemblage. Si vous désirez modifier l'une quelconque des options d'exécution, appuyez sur (N) pour cette première option.

CP/M-Output (Sortie CP/M)

Si cette option est validée (Y), le fichier binaire est écrit sur le disque sans en-tête de fichier et convient à l'emploi comme programme CP/M transitoire. Aucun autre contrôle n'est effectué pour garantir que le programme est conforme aux normes CP/M.

Video-Display (Affichage Video)

Si validé (Y), l'enregistrement source en cours de traitement sera affiché pendant le passage 1. Durant le passage 2, c'est le code objet et l'enregistrement source qui seront affichés.

Print-Output (Sortie Impression)

Lorsqu'elle est validée (Y), cette option ordonne à l'Assembleur de produire une liste d'assemblage durant la génération de code objet du passage 2. Cette option fonctionne en conjonction avec l'option SELECT...

Selective Print (Impression Selective)

On peut répondre (Y) (OUI) ou (N) (NON) à l'option SELECT, qui ordonne à l'Assembleur de respecter (YES) ou d'ignorer (NO) les directives LIST et NOLIST contenues dans le code source.

List Errors (Erreurs de Listage)

Lorsqu'elle est validée (Y), cette option force les messages d'ERREURS à être listés sur l'imprimante pour emploi ultérieur lors de la correction des erreurs de codage à l'aide de l'EDITEUR. Elle fonctionne indépendamment des options PRINT et SELECT décrites ci-avant, mais les erreurs sont toujours listées si on a répondu (Y) à l'option d'impression (PRINT).

Recapitulatif des Options d'Impression

| PRINT | SELECTIVE | ERRORS | Printed Code | Printed Errors |
|-------|-----------|--------|--------------|----------------|
| NO | NO | NO | NONE | NONE |
| NO | NO | YES | NONE | ALL |
| NO | YES | NO | NONE | NONE |
| NO | YES | YES | NONE | ALL |
| YES | NO | NO | ALL | ALL |
| YES | NO | YES | ALL | ALL |
| YES | YES | NO | SELECTIVE | ALL |
| YES | YES | YES | SELECTIVE | ALL |

Figure 3.1 Options d'impression

PRINT = IMPRESSION

SELECTIVE = SELECTIVE

ERRORS = ERREURS

Printed Code = Code Imprimé

Printed Errors = Erreurs Imprimées

NONE = NEANT

ALL = TOUS/TOUTES

Avertissement: L'emploi des options VIDEO ou PRINT ralentit APPRECIABLEMENT le processus d'Assemblage. Pour obtenir les résultats les plus rapides, répondez (Y) aux options implicites. L'exception: faites lister les ERREURS, ce que vous trouverez très utile durant les premiers Assemblages de grands programmes.

Symbol-Table (Table des Symboles)

Si elle est validée (Y), cette option produit une liste des symboles (labels) utilisés dans le ou les programme(s) source(s) accompagnés de leurs valeurs. La sortie se fera sur ECRAN ou IMPRIMANTE en fonction des options sélectionnées ci-avant. On peut faire précéder chaque symbole d'un (U), d'un (D) ou d'un (N), qui signifient, respectivement, non défini, Double Définition ou Sans Référence.

Cross-Reference (Correspondances)

Si cette option est validée (Y), une table de correspondances sera sortie sur Ecran ou Imprimante une fois le processus d'assemblage terminé. Le listage affichera tous les fichiers sources lus, et les numérotera comme fichiers, 1,2,3, etc. La table des symboles est imprimée, avec tous les numéros de fichiers et les numéros de lignes des instructions qui les désignent.

Vous êtes prié de noter que, si elles sont utilisées, les options SYMBOL ou X-REF attendront que les options VIDEO ou PRINT (ECRAN ou IMPRIMANTE) aient été validées avant que le processus d'assemblage puisse sortir les informations demandées.

Touches de Durée d'Exécution de l'Assembleur

Une fois lancé, le processus d'Assemblage peut se contrôler par commande au clavier des OPTIONS D'EXECUTION décrites ci-avant. Pour ce faire, il vous suffit d'appuyer sur des touches uniques. L'affichage des options d'exécution change lorsque vous utilisez les touches.

Vous utiliserez les touches suivantes:

- (P) Inversion de l'option IMPRESSION.
- (S) Inversion de l'option SELECTION.
- (E) Inversion de l'option ERREURS.
- (V) Inversion de l'option VIDEO (ECRAN).

On peut également utiliser la touche Escape pour INTERROMPRE MOMENTANEMENT le programme d'Assemblage, qui est utile pour examiner le code objet affiché sur l'écran. Appuyez sur n'importe quelle touche pour reprendre le processus d'assemblage.

Un appui sur la touche CTRL-A met un terme au processus d'assemblage. Tous les fichiers sont abandonnés et aucun changement de disque ne survient. L'Assembleur vous demande d'appuyer sur n'importe quelle touche afin qu'il puisse repasser la commande au Menu du Système...

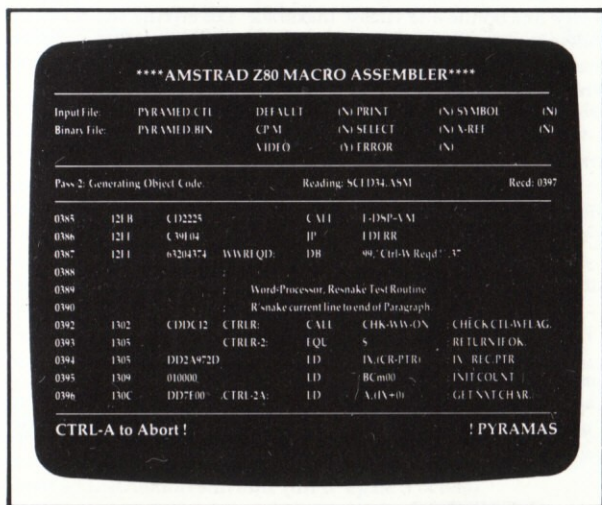


Figure 3.1 Durée d'exécution (VIDEO-ON)

Input File = Fichier d'Entrée

Binary File = Fichier Binaire

Pass 2 : Generating Object Code = Passage 2 : Génération du Code Objet

Reading = Lecture

Recd = Enreg.

Word-Processor, Resnake Text Routine = TdT, Déplacer Sous-Programme Texte

Resnake current line to end of Paragraph = Déplacer ligne courante à la fin du Para.

CTRL-A to Abort! = CTRL-A : Abandon

Traitement des Erreurs d'Assemblage

La détection des erreurs par le processus d'assemblage sera traitée de trois manières différentes, en fonction des options de sortie définies plus haut.

a) Si l'on a validé (Y)es l'une quelconque des options PRINT ou ERROR, l'Assembleur suppose que vous avez connecté une imprimante. Le code d'erreur et la description seront imprimés. Le processus d'assemblage ne sera PAS interrompu.

b) Si vous avez validé (Y)es l'option VIDEO, le code d'erreur et la description seront affichés. Le processeur d'assemblage cesse momentanément, mais vous pouvez le reprendre en appuyant sur une touche quelconque.

c) Si vous avez inhibé (N)o les options PRINT, ERROR et VIDEO, le processus d'assemblage fonctionne à sa vitesse maximale. Les erreurs seront envoyées sur l'Ecran Vidéo, mais le processus d'assemblage ne s'interrompra qu'après affichage de dix messages d'erreurs. En ce cas, appuyez sur une touche quelconque pour faire poursuivre l'assemblage.

Codes d'erreurs

| | |
|--------------------------------|--------------------------------------|
| B - Erreur de Branchement/Saut | N - Expression Numérique |
| C - Erreur Conditionnelle | OE - Erreur d'Opérande(s) |
| D - Double Définition | OM - Opérande Manquant |
| F - Erreur de Fichier | GF - Trop de SELECTIONS |
| G - Erreur d'Usage Drg | U - Label Non-défini |
| L - Label (syntaxe) | X - Code Op. supplémentaire invalide |
| MP - Macro Paramètre | FW - Référence aval invalide |

Definition du Langage Assembleur

Le Langage Assembleur est le langage Z80 et les programmes sont écrits sous forme d'un ou de plusieurs Fichiers Sources qui sont assemblés en code machine par l'Assembleur.



Chaque fichier source se compose d'instructions de source, une par ligne (rangée sur l'écran). L'EDITEUR PYRADEV est le programme idéal pour la création et l'édition de ces fichiers sources. Chaque instruction de fichier source se compose d'un label optionnel, d'un code opération, d'opérandes optionnels et de commentaires optionnels.

| LABEL | CODE OP | OPERANDES | COMMENTAIRES |
|-----------|---------|----------------|-----------------|
| LABEL-1 | LD | HL,(VALUE) | ; load HL |
| | LD | (OLD-VALUE),HL | ; save HL |
| | JP | LABEL-X | ; jump ... |
| ; | | | |
| VALUE: | DEFW | 00 | ; current value |
| OLD-VALUE | DEFW | 00 | ; old value |

load = charger **save** = sauvegarder **jump** = sauter
current value = valeur en cours **old value** = valeur ancienne

LABELS: Ce labels doivent débiter dans la colonne un et peuvent avoir n'importe quelle longueur; on les maintient généralement à une longueur inférieure à 10 caractères. L'utilisation des deux points (:) après le label est optionnelle.

CODES OP : Ces codes peuvent se trouver à n'importe quel endroit sur une ligne, et doivent être précédés d'un espace au minimum.

OPERANDES : Un code opération doit être suivi d'un espace au minimum avant que l'on puisse entrer l'opérande. la zone de l'opérande ne doit pas contenir d'espaces vierges imbriqués.

COMMENTAIRES : Les opérands doivent être suivis d'un espace au minimum avant qu'on puisse entrer un commentaire. L'emploi d'un point virgule (;) avant le commentaire est optionnel, sauf sur les instructions RET. utiliser un point virgule si le commentaire débute à la colonne un.

Expressions

Une expression est un OPERANDE qui se compose d'une ou de plusieurs variables, de labels et de constantes que l'Assembleur doit évaluer pour produire une valeur entière de 16 bits. Une expression s'évalue de GAUCHE à DROITE et il est interdit d'utiliser les parenthèses. On peut utiliser les opérateurs suivants...

+ * / .MOD. .SHR. .SHL. .AND. .OR. .XOR. .EQ. .GT. .LT. .UGT. .ULT.

Ils représentent respectivement : plus, moins, multiplier, diviser, décalage à droite, décalage à gauche, ET, OU, OU-exclusif, Egale, plus grand que, plus petit que, plus grand-sans signe, plus petit-sans signe.,

On peut utiliser le symbole du dollar (\$) pour représenter la valeur du compteur de programme pendant l'assemblage. Par exemple: JP \$+3 génèrerait un branchement à l'instruction suivante (un JP égale 3 octets).

L'Assembleur accepte la notation numérique pour les expressions binaires, octales, décimales et hexadécimales, dans les formats suivants...

| | | | | |
|-----------|----------|----|----------|--|
| Binaire : | 1011100B | ou | %1011100 | |
| Octal : | 134O | ou | @134 | (ou 1340) |
| Décimal : | 134D | ou | 134 | |
| Hexa : | 5CH | ou | #5C | (doit débiter numériquement) (c-à-d 0FFH pour 255D) |

Directives de l'Assembleur

Ces directives sont écrites comme des instructions (codes opérations et opérands) mais sont des commandes adressées à l'Assembleur et sont "exécutées" au moment de l'assemblage. Elles régissent les options de listage d'assemblage, la génération de code, et la construction du fichier binaire (en code machine).

Définition des Variables et Stockage

On peut définir des Chaînes d'Octets, de Mot et de Caractères, à l'aide des directives précisées dans l'exemple suivant.

| | | | |
|----------|-------|--------------------|---------------------------------|
| | DBXON | | ; Expansion DB. |
| LABEL1: | DB | "This is a string" | ; Chaînes de 16 octets. |
| LABEL2: | DEFB | "This is a string" | ; |
| BYTE: | DB | "B" | ; Un seul octet |
| MIXTURE: | DB | 1,"A",2,"b" | ; Chaîne mixte |
| SSTRING: | DB | "SPECIAL"+80H | ; Le bit 7 de l'est en fonction |

Equ

La directive EQUATES (EGALE) : associe un LABEL à une expression, par exemple, MINUS1 EQU-1

Exec

Cette directive définit l'adresse à utiliser dans l'en-tête du fichier binaire. Lorsque le programme assemblé est EXECUTE, il va être chargé selon l'instruction LOAD (ou ORG). AMSDOS fait ensuite passer la commande à l'adresse EXEC. Si cette directive n'a pas été fournie, l'adresse EXEC est réglée sur la première (vraie) ORG ou LOAD.

Extra

Permet à l'Assembleur de traiter les instructions Zilog NON-STANDARDS additionnelles. Voir page 14.

Free (Libre)

Permet à l'Assembleur de traiter des expressions en "Format Libre". Autorise le mélange de différents types d'expressions de stockage. Voir page 15.

List et Nolist

Met l'Impression Sélective EN FONCTION ou HORS FONCTION.

Load (Charger)

Cette instruction ordonne à l'Assembleur de générer un code de zéros binaires jusqu'à ce que le Compteur du Programme atteigne l'expression de valeur LOAD, après quoi le système reprend la génération de code normale. Cette instruction a pour but de forcer des sections de code à leurs positions appropriées dans le fichier binaire, afin que, lorsqu'on charge ce fichier, le code se trouve à son adresse correcte.

Si l'adresse n'est pas fournie, l'adresse LOAD choisit implicitement la première adresse ORG vraie. Il faut donc fournir l'une de ces adresse (le plus souvent ORG) avant d'obtenir une génération de code. La première occurrence d'une directive ORG ou LOAD s'utilise dans l'en-tête de fichier binaire (sauf si elle est immédiatement suivie d'une instruction ENDBIN). Les autres ne s'utilisent que pour contrôler la génération du code et la structure du fichier binaire.

Org

Cette instruction ordonne à l'Assembleur de POSITIONNER le Compteur de Programme sur l'expression opérande. Au contraire de l'instruction Load (Charger), aucun code de remplissage n'est généré. L'emploi de l'instruction ORG permet de resserrer ensemble dans le fichier de disque en cours de génération des section de code qui, dans la mémoire, seront très écartées. Il incombe au programmeur de veiller à ce que ces sections de code soient ramenées à leur position correcte avant leur exécution. Il est NORMAL que l'instruction d'un jeu de programme en course d'assemblage soit une directive ORG.

```
ORG 100H      . Cette section plante
code          . correctement en mémoire,
code          . à partir du point 100 hexa.
LOAD 200H    .
code          .
code          .
code          .
;
ORG           $+100      . Il faudra ramener ce bloc
code          . à son adresse vraie, étant
code          . donné qu'aucun code de
;              . remplissage n'a été généré.

END
```

Pause

Cette instruction fait "pauser et afficher" le message. Ceci survient lorsque la FIN du fichier d'entrée en course est atteinte. Elle a pour objet de permettre l'emploi de plusieurs disques quand on assemble des fichiers sources multiples. Elle ne doit être utilisée qu'avec les systèmes à DEUX unités étant donné que le fichier BINAIRE doit être écrit continuellement sur un seul disque. Après changement de disque, appuyez sur la barre d'espace pour pouvoir continuer.

PAUSE 'MOUNT NEXT SOURCE DISC IN DRIVE B' (avec assumption que la sortie se fait sur l'unité A).

Print

Cette instruction peut s'utiliser pour définir la largeur de Papier, les Lignes par page (hauteur de format), les Pauses de page et la suppression d'Avance de ligne. L'option de Pause de page fait pauser l'impression en haut de chaque page pour permettre de positionner le papier. C'est un besoin très REEL lorsqu'on et rendent difficile la production des Listages d'Assembleur. L'Avance de ligne élimine l'envoi des avances de ligne correspondant à chaque retour de curseur, étant donné que certaines imprimantes le font automatiquement.

PRINTR W80,D70 Largeur 80 car., Hauteur 70 lignes, Pause-P Inopérante, Avance de ligne.
PRINTR W132,D66,P Largeur 132 car, Hauteur 66 lignes, Pause-P Opérante, Avance de Ligne.
PRINTR N Utilise les options implicites (W80,D70) et élimine les Avances de ligne.
PRINTR P Utilise les options implicites (W80,D70) et utilise les Pauses de page.

Select

Cette directive est très puissante. Quand on l'utilise, le Fichier Source nommé est AJOUTE (et non INCLUS) à la FIN du train d'entrée en cours. On peut CHAINER de cette manière jusqu'à 32 fichiers et les instructions SELECT peuvent apparaître n'importe où. Lorsqu'on assemble de grands programmes, il est possible de lancer le processus d'assemblage avec un petit fichier de commande qui ne fait rien d'autre que SELECTIONNER des fichiers pour le train d'ENTREE....

```
ORG            100H            ; Origine du programme
SELECT        PROG1.ASM       ;
SELECT        PROG2.ASM       ; Fichiers de l'unité de disques A
SELECT        PROG3.ASM       ;
SELECT        B:PROG1.ASM      ; ;
SELECT        B:PROG1.ASM      ; ; Fichiers de l'unité de disques B
SELECT        B:PROG1.ASM      ; ;
```

Dans les grands projets de programmation, les informations suivantes s'avèreront utiles pourvu que vous utilisiez deux unités de disques.

UNITE A

Disque 1
Face A

PYRADEV SYSTEM
+ Fichiers binaires
+ Fichiers sources

Disque 1
Face B

Réserve, suggérons
emploi PYRA-WORD pour
documenter le projet.

Le Disque 1 demeure
fix durant
la réalisation

UNITE B

Disque 2
Face A

CODE SOURCE - 178K,
accès par Editeur et
Système d'Assemblage.

Disque 2
Face B

CODE SOURCE - 178K,
accès par Editeur et
Système d'Assemblage.

Le Disque 2 est inversé
selon les besoins pour éditer/
assembler des fichiers sources.

Title (Titre)

Modifie le Message d'En-tête des Listages d'Assemblage, en Haut des Pages.

Exemple: TITLE MegaGame, Section 6.

Directives d'Assemblage Conditionnelles

L'Assembleur est capable d'inclure ou d'exclure certains blocs de code durant le processus d'assemblage, grâce à l'emploi de drapeaux et de directives conditionnelles. Associé au système SELECT décrit ci-avant, ce système devient extrêmement souple étant donné que les premiers fichiers peuvent définir des drapeaux qui, dans les fichiers ultérieurs, vont contrôler l'assemblage du code. Le mécanisme d'assemblage conditionnel est le classique SI quelque chose ALORS faites ceci OU ALORS faites cela.

Le quelque chose est une expression arithmétique. Si l'expression est VRAIE (c'est-à-dire NON-ZERO), le premier choix (THEN/ALORS) est sélectionné. Sans quoi, le système sélectionnera le second choix (ELSE/OU ALORS). Le second est optionnel et la directive final doit être une instruction ENDIF.

Les exemples suivants montrent comment on peut utiliser ce système. Le DRAPEAU du code de gauche est positionnée sur 1 (VRAI) et le THEN-CODE est assemblé. Dans la code de droite, le DRAPEAU est positionné sur 0 (FAUX) : c'est le ELSE-CODE qui est assemblé.

```

;
FLAG:      EQ      1
;
;          IF      FLAG
THEN-CODE: LD      A,(VALUE-1)
;          ELSE
ELSE-CODE: LD      A,(VALUE-2)
;          ENDIF
;
;          END

```

```

;
FLAG:      EQU     0
;
;          IF      FLAG
THEN-CODE: LD      A,(VALUE-1)
;          ELSE
ELSE-CODE: LD      A,(VALUE-2)
;          ENDIF
;
;          END

```

Vous noterez que l'emploi est toujours l'ordre IF... ELSE... ENDIF. L'instruction THEN est en effet implicite puisqu'elle constitue le premier branchement suivant l'instruction IF. La section ELSE est optionnelle. Une directive IF doit toujours être accompagnée d'une déclaration ENDIF correspondante.

Au lieu d'utiliser les directives IF..ELSE..ENDIF on peut utiliser les directives COND..ELSE..ENDC. Toutes deux sont valides.

Definitions et Utilisation des Macro-Instructions

Une MACRO-INSTRUCTION est un petit élément de code, défini dans un fichier au début du processus d'assemblage. Lorsque son nom est utilisé, l'élément de code précédemment défini est à nouveau généré.

Une MACRO-INSTRUCTION détermine le début de la définition et doit avoir un label qui sert de Macro-Nom. Ce nom doit être ENTIEREMENT alphabétique et peut posséder jusqu'à SIX caractères. Imaginez la MACRO-INSTRUCTION comme constituant une nouvelle instruction pour l'Assembleur. Les deux exemples qui suivent représentent une MACRO-INSTRUCTION sans Utilisation de Paramètre à gauche, et avec une Utilisation de Paramètre à droite.

```
SBCX:  MACRO
        XOR  A
        SBC  HL,DE
        ENDM
;
        LD  HL,(VALUE1)
        LD  DE,(VALUE2)
        SBCX
        LD  (VALUE3),HL

SWAP;  MACRO#P1,#P2
        PUSH AF
        LD  A,#P2      (LD A,C)
        LD  #P2,#P1   (LD C,D)
        LD  #P1,A     (LD D,A)
        PDP  AF
        ENDM
;
        SWAP  D,C
```

L'exemple de gauche explique comment on peut utiliser une petite macro-instruction comme instruction additionnelle; En ce cas, la "macro" SBCX est assemblé sous forme de deux instructions distinctes. La première supprime le drapeau de report avant exécution de la seconde SOUSTRACTION avec instruction de REPORT.'

L'exemple de droite représente une substitution de paramètre. Durant l'Assemblage, l'utilisation de la macro SWAP fait générer un sous-programme de cinq multiplats qui va faire interchanger le contenu des registres C et D. La macro SWAP peut s'utiliser pour interchanger deux registres quelconques, sauf le registre A.

Utilisation des Macro Paramètres

Comme explicité dans l'exemple ci-avant, l'emploi des paramètres est positionnel et fonctionne par substitution. Si les paramètres contiennent des virgules ou des guillemets, ils doivent être renfermés entre des apostrophes ou des guillemets, comme représenté ci-dessous:

```
STRING:  MACRO      #P1,#P2
          DB         #P1      (DB 5)
          DB         #P2      (DB21,22,23,24,'X')
          ENDM
;
          STRING    5,"21,22,23,24,'X'"
;
```

Générateur de Symboles Macro

Si une macro définition contient des labels, il surviendra des erreurs de DUPLICATION DE LABELS durant l'assemblage si l'on utilise la Macro-instruction plus d'une fois. En ces cas, il faut ajouter le suffixe #SYM au label. Chaque fois que l'on utilise la Macro, un suffixe à 4 chiffres est incrément et rattaché au label.

L'exemple suivant est une macro-instruction qui éprouve HL et substitue la constante hex-décimale 0FFFFH si HL est égal à zéro. Chaque fois qu'elle est utilisée, l'instruction JR TEST-#SYM et les labels TEST-#SYM sont portés à la valeur suivante. Au premier passage, le JR sera au label TEST-0001 puis il passera à TEST-0002 etc...

```
;  
;  
;          MACRO tests HL, if ZERO replace with 0FFFFH.  
;          (MACRO teste HL, si ZERO, remplacer par 0FFFFH.)  
;  
TEST:     MACRO  
          XDR          A          ; Clear Carry.          1.  
          LD           DE,0       ; Set DE to zero.       2.  
          SBC          HL,DE      ; Subtract / TEST      3.  
          JR           NZ,TEST-#SYM ; if NZ JMP to TEST 000n 4.  
          LD           HL,0FFFFH  ; else set HL to 0FFFFH 5.  
TEST-#SYM: EQ          $          ; continue ... TEST-000n 6.  
;
```

1. Supprimer report.
2. Set DE à zéro
3. Soustraire / TEST
4. si NZ SAUT à TEST-000n
5. sans quoi pos. HL sur 0FFFFH
6. poursuivre ... TEST-000n

Listages Macro

En règle générale, seules les Macro Définitions sont listées. Pour visualiser le code complet, vous devez utiliser la directive MLIST. Pour SORTIR de cette facilité, utilisez la directive MNLIST.

Instructions Supplémentaires

Il existe plusieurs instructions Z80 qui ne figurent pas normalement dans les manuels de programmation Z80 parce qu'elles ne fonctionnent pas toujours ! Si vous écrivez des logiciels pour d'autres utilisateurs de Z80, NE LES UTILISEZ PAS. S'il vous est essentiel de les employer, l'Assembleur PYRADEV les accepte mais vous devez tout d'abord donner la Directive 'EXTRA'.

Le premier groupe de codes opérations permet d'utiliser les registres universels à 16 bits IX et IY comme quatre registres à 8 bits en les catégorisant comme registres de POIDS FORT ou FAIBLE. Nous utilisons les opérandes LX, HX, LY et HY pour représenter respectivement les octets de poids Faible et de poids Fort de IX et IY. On peut également utiliser la variante XH, XL, YH et HL.

```

LD HX,A LD HX,B LD HX,C LD HX,D LD HX,E LD HX,n
LD LX,A LD LX,B LD LX,C LD LX,D LD LX,E LD LX,n
LD HY,A LD HY,B LD HY,C LD HY,D LD HY,E LD HY,n
LD LY,A LD LY,B LD LY,C LD LY,D LD LY,E LD LY,n
LD A,HX LD B,HX LD C,HX LD D,HX LD E,HX
LD A,LX LD B,LX LD C,LX LD D,LX LD E,LX
LD A,HY LD B,HY LD C,HY LD D,HY LD E,HY
LD A,LY LD B,LY LD C,LY LD D,LY LD E,LY
LD HX,LX LD LX,HX LD HY,LY LD LY,HY
INC HX INC LX INC HY INC LY
DEC HX DEC LX DEC HY DEC LY
ADD A,HX ADD A,LX ADD A,HY ADD A,LY
ADC A,HX ADC A,LX ADC A,HY ADC A,LY
SBC A,HX SBC A,LX SBC A,HY SBC A,LY
SUB HX SUB LX SUB HY SUB LY
CP HX CP LX CP HY CP LY
AND HX AND LX AND HY AND LY
OR HX OR LX OR HY OR LY
XOR HX XOR LX XOR HY XOR LY

```

Un second groupe de codes fournit d'autres codes opérations SHIFT-LEFT-LOGICAL (DECALAGE CIRCULAIRE A G.) qui complètent l'instruction SRL existante. Ce sont:

```
SLL A SLL B SLL C SLL D SLL E SLL H SLL L SLL (HL)
```

Les opérations sont les mêmes que pour l'instruction SLA, mais un (1) bit est placé dans la position de bit 0, où il remplace un (0).

Note: Les codes opérations qui précèdent ne sont pas standards (puisque'ils ne possèdent aucune définition formelle); néanmoins, ils correspondent bien aux codes mnémoniques utilisés dans plusieurs publications relatives à la programmation du Z80. Chaque fois que possible, nous avons utilisé les définitions communes.

Format Libre

L'emploi de la directive FREE (LIBRE) permet à l'Assembleur de traiter un type d'expression Z80 qui est très utile à ceux qui écrivent des jeux. C'est un système d'expressions mélangées données et valeurs, qui convient à la définition de tables. La génération des octets est toujours implicite sauf lorsqu'un seul point d'exclamation (!) précède une expression, auquel cas le système génère une valeur de mot. L'emploi de DBXON est idéal lorsque vous commencez à travailler en format libre, car elle vous permet de vérifier si les tables sont correctement définies. Les exemples suivants expliquent comment utiliser le format LIBRE....


```

;
FREE                                ; FREE FORMAT.
DBXON                               ; DB EXPANSION.
;
FLAG1:    EQU            0
FLAG2:    EQU            1
VALUE1:   EQU            2
;
ROUTINE1: CALL          GET-CURSOR
          INC           H
          CALL          SET-CURSOR
          RET
;
CONTROL   TABLE
;
;
1,2,3,!100,!200,!300,FLAG0,FLAG1,"STRINGA","ABC1","111",!ROUTINE1
4,5,6,!200,!400,!600,FLAG1,FLAG2,"STRINGB","ABC2","222",!ROUTINE2
7,8,9,!300,!600,!900,FLAG2,FLAG2,"STRINGC","ABC3","333",!ROUTINE3
;
;
END OF CONTROL TABLE
;
;
DBXOFF                                ; DB EXPANSION OFF
;

```

FREE FORMAT = FORMAT LIBRE DB EXPANSION EXPANSION DB
 CONTROL TABLE = TABLE DE COMMANDE
 END OF CONTROL TABLE = FIN TABLE DE COMMANDE
 DB EXPANSION OFF = SORTIE EXPANSION DB

Chacune des trois lignes de la table de commande ci-avant sera assemblée de manière analogue. Nous ne décrivons ici que la PREMIERE ligne, et le code binaire généré...

Trois octets contenant les valeurs 1, 2, et 3.

Trois mots contenant les valeurs 100, 200, et 300. Remarquez le !.

Un octet contenant la valeur drapeau-0.

Un octet contenant la valeur drapeau-1.

Une chaîne d'octets contenant 'STRINGA'.

Une chaîne d'octets contenant 'ABC1'.

Une chaîne d'octets contenant '111'.

Un mot unique avec l'adresse 'ROUTINE1'. Notez bien le !.

Gestion des Mémoires

Durant le processus d'assemblage, l'affectation réserve en mémoire à la table des symboles, à la mémorisation du code source, à la table des correspondances, et aux macro-définitions, varie en fonction de la mémoire disponible.

Dans les Amstrad 464 et 664, qui possèdent tous deux 64K mémoire, la table des symboles est mémorisée à partir du bas de la mémoire vers le haut, tandis que la table des correspondances (optionnelles) est mémorisée du haut vers le bas. En outre, le système utilise une mémoire-tampon de 2K pour la lecture du code source. Si vous assemblez de très grands programmes, vous parviendrez à un point où il sera impossible d'obtenir une liste des correspondances, étant donné que le système 64K aura été utilisé au maximum.

Dans l'Amstrad 6128 (ou avec un 464/664 avec mémoire RAM supplémentaire*), les espaces mémoires sont affectés comme suit:

- Le code source est mémorisé dans les blocs supplémentaires durant le premier passage ; ainsi, durant le second passage, la source est lue à partir de la mémoire, ce qui accélère le processus d'assemblage.
- La table des correspondances optionnelle est conservée dans 32K du second bloc de RAM si sélectionné, ce qui autorise au maximum 6.500 entrées. Ceci permet d'exécuter très importants assemblages.

*Le système PYRADEV a fait l'objet d'essais très poussés avec un CPC464, une unité de disques DDI et un CPC6128. On a pu utiliser des mémoires de 64K jusqu'à 384K au total, en travaillant avec le Kit d'Expansion de RAM 256K de Dk'TRONICS.

En poussant la capacité RAM au-dessus de 128K, on n'obtient qu'une amélioration marginale (10 à 15% environ) de la durée d'assemblage globale.

Statistiques de l'Assembleur

A la fin du deuxième passage (génération du code objet), les statistiques suivantes sont affichées afin que vous puissiez voir où vous vous trouvez par rapport aux limites de traitement de l'Assembleur. La Mémoire à Symboles Libres est le paramètre critique, et doit toujours demeurer bien AU-DESSUS du zéro !

Nombre d'Erreurs nnnn

Nombre de Symboles nnnn

| | |
|-----------------------------------|-----------------|
| Table des Symboles de | nnnn à nnnn |
| Liste des Macros de | nnnn à nnnn |
| Nombre de Correspondances | nnnn |
| Table des Correspondances de | nnnn |
| Mémoire des Symboles Libres .. | nnnn |
| Fichier débute : nnnn | Finit : nnnn |
| | Longueur : nnnn |

Recapitulatif des Directives de l'Assembleur

| | | |
|--------|--------------|--------------------------------------|
| COND | «exp» | Assemblage conditionnel. |
| DBXOFF | | Expansion DB inopérante. |
| DBXON | | Expansion DB opérante. |
| DEFB | «exp», «exp» | Définition d'octets. |
| DEFL | «exp» | Redéfinition d'un label. |
| DEFM | «exp» | Définition mémoire. |
| DEFS | «exp» | Définition stockage. |
| DEFW | «exp» | Définition valeur mot unique. |
| EJECT | | Avance papier. |
| ELSE | | Partie de l'assemblage conditionnel. |
| END | | Fin du train d'entrée source. |
| ENDBIN | | Fin du généré du code binaire. |
| ENDC | | Fin d'assemblage conditionnel. |
| ENDIF | | Fin d'assemblage conditionnel (IF). |
| ENDM | | Fin de MACRO. |
| EQU | «exp» | Egale un label. |
| EXEC | «exp» | Définition adresse EXEC. |
| EXTRA | | Activer codes op. supplémentaires. |
| FREE | | Activer format LIBRE. |
| IF | «exp» | Assemblage conditionnel (IF). |
| LIST | | Activer impression sélective. |
| LOAD | «exp» | Générer octets zéro jusqu'à «exp». |
| MACRO | «parm list» | Lancer macro définition. |
| MLIST | | Activer macro listage. |
| MNLIST | | Désactiver macro listage. |
| NOLIST | | Désactiver impression sélective. |
| ORG | «exp» | Définir adresse code objet. |
| PRINTR | «parms» | Définir options imprimante. |
| PAUSE | message | Suspendre assemblage. |
| SELECT | «filename» | Ajouter un autre fichier. |
| TITLE | message | Changer l'en-tête de listage. |
| DB | comme DEFB | |
| DL | comme DEFL | |
| DM | comme DEFM | |
| DS | comme DEFS | |
| DW | comme DEFW | |

Visualisation

L'Écran du Moniteur est divisé en cinq sections différentes. La section supérieure gauche contient l'information d'intersection/piégeage.

La section central gauche contient le contenu des registres (actuels et précédents) et les réglages de drapeaux, ainsi que les quatre octets de mémoire correspondant à chacun des quatre registres de paires d'adresses DE, HL, IX et IY.

La section inférieure est une visualisation de mémoires. Elle alterne avec l'affichage d'un menu.

Tout en haut et à droite de l'écran, vous voyez l'état des sélections de mémoires ROM inférieure et supérieure.

La droite de l'écran affiche le désassemblage principal.

Translation

Le Moniteur est parfaitement translatable et, dans un premier temps, se charge dans une adresse élevée de mémoire (qu'il affiche). Il demande ensuite si la translation est nécessaire. Si vous répondez Oui (Y), vous pouvez ensuite sélectionner le Moniteur (S)standard ou (M)ini.

Le Mini-Moniteur est un sous-ensemble du Moniteur-Standard, que vous pouvez utiliser lorsque l'espace disponible en mémoire est limité. Les fonctions qui suivent sont exclues du Mini-Moniteur : Ecriture-Fichier-à-Disque, Copie-Mémoire, Menu-Help, Changements-d'Encres, Sérigraphie Affichage-ROM et Désassemblage sur Disque ou Imprimante.

Le Moniteur demande ensuite si vous voulez spécifier une borne inférieure (adresse de départ) ou une borne supérieure (adresse de fin).

Si vous spécifiez une borne Inférieure, vous demandez au Moniteur de se traduire de manière à ce que son adresse la plus basse ne descende pas au-delà de la borne Inférieure.

Si vous spécifiez une borne Supérieure, vous ordonnez au Moniteur de se traduire de manière à ce que son adresse la plus élevée n'excède pas la borne Supérieure.

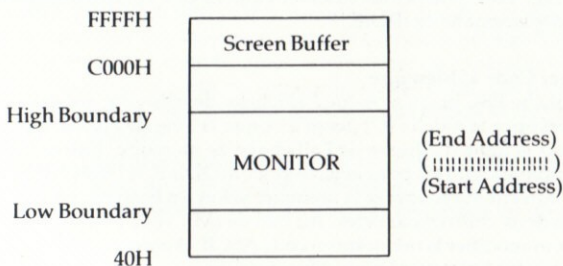


Figure 4.1

1. Tampon d'écran
2. Borne supérieure
3. Borne inférieure
4. Adresse de fin
5. Adresse de lancement

Affichage du Menu

Une fois prêt, vous pouvez faire afficher le Menu du Moniteur, en appuyant sur la touche ESC. Le menu alterne avec la section d'affichage de la mémoire en bas de l'écran.

| | | | | | |
|------------|--------------|----------|-----------|-------------|------------|
| T....TRAP | G..GOTO-ADDR | L...LOAD | Q...QUERY | J◊..PAPER | C◊...CAT |
| Z.CLR-TRAP | G◊..STEP-1ST | W..WRITE | N.NEW-SCN | KK◊....PEN | T◊A...TAPE |
| M.A◊MEMORY | S..STEP(f1) | O..OTHER | X...ROMS | D◊..DIS-ASM | ◊...PRINT |
| R....REGS | S◊.DSTEP(f2) | U.UPDATE | Y...COPY | X◊....EXIT | |

Figure 4.2 Menu du Moniteur

Selection de Périphérique

Appuyez sur (CTRL-T) pour commuter sur BANDE ou DISQUE. Le périphérique sélectionné est indiqué en haut de l'écran, et sera utilisé pour LIRE et ÉCRIRE des fichiers. Appuyez sur (CTRL-C) pour consulter la visualisation du CATALOGUE périphérique.

Code de Test de Chargement

Appuyez sur (L) pour charger un fichier de programme, et entrez le nom du fichier. L'en-tête de fichier sera examiné par le système afin qu'il puisse afficher le point de chargement et l'adresse d'exécution. Vous pouvez modifier le point de chargement avant d'appuyer sur la touche ENTER pour lire le fichier en mémoire si vous le désirez.

Affichage de Désassemblage

Un symbole Tête de Flèche vers la Droite vous signale où le curseur d'adressage se trouve. Appuyez sur la touche (O) jusqu'à ce que la Tête de Flèche vers la Droite se trouve en haut et à droite de l'écran, puis entrez une adresse de désassemblage. Vous pouvez faire défiler l'affichage à l'aide de la flèche et des touches de déplacement de flèche.

Pour Changer Code et Mémoire

Utilisez la touche ESC jusqu'à ce que l'affichage de mémoire apparaisse en bas de l'écran. Frappez la touche (O) jusqu'à ce que la Tête de Flèche vers la Droite apparaisse sur la ligne d'adresse de l'affichage de mémoire. Entrez l'adresse à modifier. Appuyez sur (M) pour sélectionner MODIFICATION-MEMOIRE. Déplacez le curseur et recouvrez la mémoire selon les besoins, en utilisant des codes hexa à deux chiffres par octet. Au lieu de (M), vous pouvez utiliser (CTRL-A) pour modifier la mémoire en code ASCII. Appuyez sur la touche ESC lorsque vous aurez fini toutes les modifications. Si vous avez modifié des codes, appuyez sur (O) pour ramener le curseur en haut et à droite de l'écran, entrez l'adresse où les modifications ont été faites : le nouveau code sera désassemblé sur l'écran.

Vous pouvez faire défiler l'affichage vers le haut et le bas d'un octet, de 16 octets ou de 128 octets, en utilisant les touches avec flèche vers le bas et le haut, dans les états normal, décalé et de commande, selon les besoins.

Positionnement des Pièges (Traps)

Utilisez la partie supérieure droite de l'écran pour étudier le code, puis appuyez sur (T) pour positionner un piège. Entrez la première adresse d'octet de l'instruction que vous voulez piéger. L'une des cinq lignes de piégeage sera utilisée pour signaler l'instruction qui a été remplacée par le piège. Utilisez la fonction (G) pour Goto) pour lancer l'exécution du code. Lorsque le piège survient, la ligne de piégeage appropriée sera affichée avec couleurs inversées. A partir d'un piège, vous pouvez continuer de deux manières.

S – Lancer l'exécution de code pas-à-pas à partir de l'adresse du piège.

G – Goto (allez à) l'adresse : l'adresse implicite poursuit l'exécution.

Lorsqu'un piège survient, il s'efface automatiquement. Si vous voulez qu'il survienne à nouveau, vous devez le repositionner avant de poursuivre.

Intruction "Goto Address"

Après avoir positionné les pièges, ou s'ils sont survenus, appuyez sur (G) pour lancer ou relancer l'exécution du code. Une adresse de relancement sera affichée et vous pouvez l'accepter en appuyant sur la touche Enter. Pour passer à un point d'exécution spécifique, recouvrez l'adresse affichée avec une adresse de votre choix.

Etude Pas à Pas du Code

Après un piège ou en définissant le premier point de piégeage avec (CTRL-G), vous pouvez étudier un code pas à pas.

Appuyez sur (S) ou sur (f1) pour faire une étude PAS A PAS. Cette forme de progression suit un parcours d'adressage logique (débit de commande) au fur et à mesure de l'exécution du code. S'il survient un code "CALL to ROM" (APPEL MEMOIRE ROM), l'opération pas à pas est suspendue étant donné que le mécanisme d'avance pas à pas ne peut pas recouvrir la mémoire ROM pour établir les pièges nécessaires.

Appuyez sur (CTRL-S) ou (f2) pour fonctionner en DOUBLE PAS. Cette forme d'avance va définir des pièges dans des emplacements de mémoire, en ordre croissant. Ainsi, lorsque l'on travaille pas à pas et qu'il survient un "CALL to ROM" (APPEL DE MEMOIRE ROM), on utilise un double pas. Ceci définit un piège sur l'instruction de mémoire suivant, et on peut ainsi reprendre l'avance pas à pas (pas unique ou double) après achèvement de l'opération CALL et retour de la commande à partir du sous-programme ROM. Les programmes AMSTRAD utilisent en règle générale de nombreux appels ROM pour l'exploitation du système.

Le MODE (0,1,2) qu'utilise votre programme peut être modifié à l'aide de shift 0,1, ou 2. Lorsqu'il est sur 2, le mode est identique au MONITEUR et, par conséquent, la visualisation ne sera PAS effacée chaque fois qu'on exécute un piège ou un pas.

Changement du Contenu d'un Registre

Appuyez sur (R) pour changer le contenu d'un registre, puis recouvrez soigneusement les valeurs existantes par les nouvelles valeurs. Appuyez sur la touche ESCAPE pour sortir de la fonction de changement. Les nouvelles valeurs seront alors utilisées quand on reprend l'exécution du code, soit par emploi de la touche (G)oto soit en fonction (S)tep (pas unique).

Nouveau et Actualisation

Vous pouvez actualiser l'affichage de mémoire ou de désassemblage en cours, en appuyant sur (U). Vous devez faire cette opération lorsque vous surveillez une mémoire en cours de changement. Lorsque l'exécution du code recouvre l'écran, (ce qui le rend difficile à lire), vous pouvez utiliser la commande (N)ew (nouveau) pour actualiser l'écran complet.

Interrogation, Recherche

Un appui sur (Q) sélectionne le mode interrogation (query) et permet de faire rechercher un HEXA ou un ASCII de 30 caractères au maximum. On peut utiliser des caractères de remplacement (?) en format ASCII aussi bien que HEXA. Chaque fois que le système trouve une correspondance, il l'affiche sur la ligne deux de la limitée entre une adresse de mémoire inférieure et supérieure, et peut être suspendue ou poursuivie après chaque correspondance.

Examen des Mémoires ROM

Les affichages de désassemblage de mémoire lisent tous deux la mémoire en fonction des sélections de LOWER et UPPER ROM (ROM INFÉRIEURE ET SUPÉRIEURE). En appuyant sur la touche (X), vous pouvez modifier l'état des sélections ROM.

Les zones d'affichage seront modifiées si elles affichent des zones de mémoire qui se trouvent au-dessous de 16K ou au-dessus de 48K. lorsqu'on commute les ROM inférieure et supérieure.

Désassemblage sur Disque, Ecran ou Imprimante

Appuyez sur CTRL-D et entrez une adresse START et END pour identifier la zone de mémoire à désassembler. Entrez une adresse de Zone de Travail dans laquelle pourra être définie la table des symboles. L'adresse implicite fournie définit une zone de 1000 octets et peut s'utiliser dans la plupart des cas. Si l'espace est entièrement occupé par le processus de désassemblage (500 labels environ), un message d'erreur sera affiché et il faudra reprendre le désassemblage avec une zone de travail différente plus largement dimensionnée.

On peut définir des zones de données par des paires d'adresses Début-Fin. Utilisez une paire de zéros pour terminer vos sélections. On peut sélectionner la sortie du processus de désassemblage à l'aide de (V), (D) ou (P) qui dénotent respectivement l'Ecran, le Disque ou l'Imprimante. Note: Il peut survenir des références à des labels inexistant si les zones de données imbriquées ne sont pas définies avant le désassemblage.'

Si vous avez sélectionné (D), entrez le nom de fichier à utiliser. Le désassemblage sur Ecran ou Imprimante peut être suspendu à l'aide de la touche Escape et repris avec n'importe quelle autre touche. Si vous avez choisi l'option (D), un seul appui sur la touche Escape fait abandonner le processus.

Le désassemblage sur fichier(s) de disque pause après génération de chaque groupe de 30K de code source, ce qui permet d'entre les noms de fichiers suivants avant la reprise du processus.

Ecriture

Pour écrire une section de mémoire sous forme d'un fichier binaire, appuyez sur (W). Le point de chargement, l'adresse d'exécution et la longueur (sauvegardée au moment du chargement du fichier) sont réaffichés et peuvent être recouverts le cas échéant. Après avoir entré un nom de fichier, la section de mémoire appropriée est écrite sur disque sous forme d'un fichier binaire.

Copie de Mémoire

On peut faire copier une section de mémoire en appuyant sur (Y). Définissez le bloc à déplacer avec une adresse DEBUT et FIN (BEGIN et END). Spécifiez ensuite l'adresse de destination. Le bloc va être copié "intelligemment". Ceci signifie que la destination peut se trouver n'importe où et peut chevaucher la zone du bloc initial.

Sélection de Bloc-Zone Mémoire

Appuyez sur (CTRL-B) pour sélectionner un autre bloc de mémoire à implanter dans la gamme d'adresses 4000H à 8000H. (Systèmes 128K et plus grands uniquement).

Impression de l'Ecran

Appuyez sur la touche (.) du bloc numérique (fonctions). Le contenu de l'écran sera décodé et imprimé. Appuyez deux fois sur la touche ESC pour annuler l'impression.

Couleurs, Papier et Plume

L'emploi des touches (J) et (K) fait passer les couleurs de papier et de plume à leurs valeurs d'encres suivantes. En expérimentant avec les encres, vous améliorez la lisibilité de l'écran sur les moniteurs monochrome aussi bien que couleurs, en fonction des conditions d'éclairage de fond. Vous pouvez utiliser CTRL-J et CTRL-K pour faire défiler les couleurs d'encres en sens inverse.

Affectation de Zones d'Empilage

Si le Moniteur est chargé au-dessus de 4100H, il affecte une zone d'empilage (stack area) à l'aide de la valeur HIMEN courante. Si le Moniteur est chargé au-dessus de 4100H, la zone d'empilage est affectée just au-dessous du point de chargement utilisé.

La fonction catalogue (CTRL-C) utilise un tampon de 2K qui se trouve à 2048 + 200 octets au-dessous du "stack".

Acces Externe

La première instruction du moniteur est JP nnnn. Vous pouvez l'utiliser pour "sauter" dans le moniteur à partir de votre programme si besoin est.

5.0 LE "DISC-NURSE"

On sélectionne DISC-NURSE dans le menu du système en appuyant sur (D). Ce sous-programme attend alors que vous ayez chargé un disque. Après le chargement du disque, appuyez à nouveau sur la touche (D) sans quoi vous ne pourrez pas utiliser le DISC-NURSE. Il vous offre plusieurs fonctions regroupées dans un menu très facile à utiliser. Ces fonctions vous permettent d'explorer votre disque ou vos disques, et si nécessaire, d'y apporter des modifications.

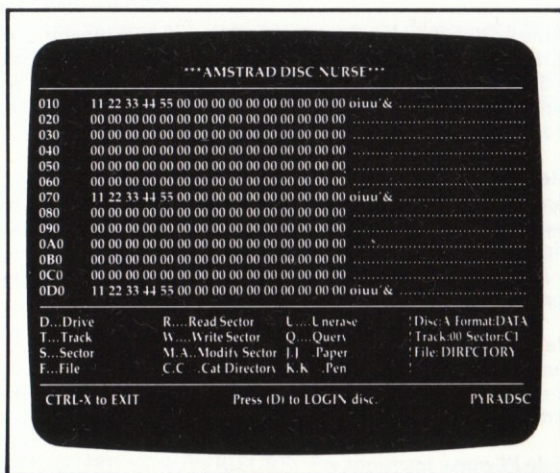


Figure 5.0 Disc-Nurse

- | | | |
|-----------------------------|-------------------------------|--------------------------|
| D = Unité de disques | T = Piste | S = Secteur |
| F = Fichier | R = Secteur lecture | W = Secteur écriture |
| M.A. = Secteur modification | C.C.⊙. = Catalogue/répertoire | U = Annuler l'effacement |
| Q = Interrogation | J.J.⊙. = Papier | K.K⊙. = Plume |

CTRL-X to EXIT = Appui sur CTRL pour SORTIER

Press (D) to LOGIN disc. = Appui sur (D) pour faire prendre le disque en charge.

Avertissement : Vous ne devez pas modifier directement des secteurs de disque sans avoir au préalable effectué une version de sauvegarde du disque, sans quoi vous pouvez aisément perdre votre jeu favori, des semaines de développement de code source ou même le contenu total du disque. Vous devez être conscient des structures de disque AMSTRAD et des constructions d'en-tête de fichier avant de changer quoi que ce soit. Prenez bien note de cet avertissement !

Unité de Disques

Appuyez sur (D). Le système affiche la demande 'Drive A or B'. Appuyez sur (A) ou (B) pour sélectionner l'unité de disques requise : le répertoire approprié est affiché.

Il est IMPERATIF d'effectuer cette opération si vous changez le(s) disque(s) en cours d'examen !

Piste et Secteur

Appuyez sur (T) pour entrer une adresse de piste, et sur (S) pour entrer une adresse de secteur.

Lecture

Appuyez sur (R) pour lire le secteur sélectionné. Après lecture d'un secteur, vous pouvez utiliser les touches suivantes. Elles vous permettent de suivre des chaînes de fichiers ou de lire progressivement et régressivement au niveau du secteur.

| TOUCHE | Normal | Décalage | Commande |
|-----------------------------|---|---|------------------------------|
| GAUCHE DROITE | Recul chaîne 1/2 sec Avance chaîne 1/2 sec | Recul chaîne 1 sec Avance chaîne 1 sec | Trouver début Trouver fin |
| VERS LE HAUT VERS LE BAS | 1/2 secteur précédent 1/2 secteur suivant | Secteur précédent Secteur suivant | Inutilisée Inutilisée |

Figure 5.1

Modification d'un Secteur

Appuyez sur (M) pour modifier, en utilisant HEX ou (A) si vous modifiez en ASCII. Utilisez les touches de curseur (flèches) pour amener le curseur sur les octets que vous voulez, et surfrappez les valeurs selon les besoins. Si vous modifiez les octets d'en-tête de fichiers, appuyez sur (CTRL-H) après la modification pour recalculer l'octet du total de contrôle. Une fois les changements de secteur achevés, appuyez sur la touche ESC. Le secteur modifié peut être écrit sur disque à l'aide de la commande (W).

Ecriture

Appuyez sur (W) pour ré-écrire le secteur affiché/modifié sur disque. La demande d'écriture doit être confirmée avec la touche (Y). Toute autre touche fera abandonner la demande d'écriture.

Avertissement : Le secteur sera écrit sur la Piste et le Secteur indiqué en bas et à droite de l'écran.

Annulation d'Effacement

Vous pouvez utiliser cette fonction pour reprendre un fichier ayant été accidentellement effacé. Appuyez sur (U) pour sélectionner la fonction d'annulation d'effacement, et entrez un nom de fichier. Le DISC-NURSE va vérifier les tables d'affectation de secteurs. Si les secteurs de fichier n'ont pas été utilisés, le fichier sera rétabli alors à l'emploi normal et réapparaîtra sur les affichages de répertoire ultérieurs.

Accès aux Fichiers

Pour accéder aux secteurs appartenant à un fichier spécifique, appuyez sur (F) et entrez un nom de fichier. Le premier secteur du fichier sera lu. Voyez la fonction (R)ead (Lecture) ci-avant qui contient une définition des fonctions de la touche défilement/parcours.

Interrogation

En appuyant sur (Q), on sélectionne la fonction interrogation/recherche. La recherche peut se limiter à un (S) pour secteur, (F) pour fichier, ou (D) pour disque, et peut s'appliquer à une chaîne ASCII ou à une chaîne HEXA. Les caractères de remplacement (7) sont autorisés.

Vous pouvez entrer la chaîne hexa sous forme d'une chaîne continue ou interrompue de chiffres hexa, c'est-à-dire NNNNNN ou NN NN NN.

La chaîne débute à partir de la position courante de piste/secteur, et se poursuit jusqu'au dernier secteur. La recherche d'une chaîne ASCII peu probable constitue une bonne manière de vérifier un disque.

Lorsque vous trouvez une correspondance, l'adresse du secteur, le décalage du mot et le contenu du secteur sont affichés sur l'écran. Pour poursuivre la recherche, appuyez sur la BARR D'ESPACEMENT (SPACE) ou terminez avec la touche ESC.

Catalogue et Répertoire Entendus

Appuyez (C) pour faire afficher le catalogue du disque en format de visualisation standard. Pour obtenir des informations de répertoire plus complètes, ainsi que des fonctions de reprographie, appuyez sur (CTRL-C). Si la sortie doit être imprimée, répondez (Y) à la question sur l'impression. On peut entrer une ligne de titre qui apparaîtra en haut de la liste.

Impression de l'Écran

Appuyez sur la touche (.) du bloc numérique (fonctions). Le contenu de l'écran sera décodé et imprimé. Appuyez deux fois sur la touche ESC pour annuler l'impression.

Couleurs, Paper et Plume

L'emploi des touches (J) et (K) fait passer les couleurs de papier et de plume à leurs valeurs d'encres suivantes. En expérimentant avec les encres, vous améliorez la lisibilité de l'écran sur les moniteurs monochrome aussi bien que couleurs, en fonction des conditions d'éclairage de fond. Vous pouvez utiliser CTRL-J et CTRL-K pour faire défiler les couleurs d'encres en sens inverse.

6.0 PROGRAMME UTILITAIRE

Le Programme Utilitaire offre des fonctions générales de gestion de fichiers et de copie regroupées dans un logiciel unique et d'emploi aisé. Les fonctions sont les suivantes:

- Visualisation des répertoires d'unités de disques (A) et (B).
- Changement de noms de fichiers, avec liaison à la commande !REN.
- Effacement de fichiers, avec à la commande !ERA.
- Transfert de fichiers ; n'importe quel des fichiers AMSDOS, bande et disque.

Message Initial

L'écran affiche le message suivant

Drive (A), (B), (D)elete, (R)ename, (C)opy, ou (X) to Exit.

En sélectionnant (A) ou (B), vous faites afficher le répertoire approprié. La sélection de (D) ou (R) vous donne les fonctions de suppression et de changement de nom de fichier, et s'explique d'elle-même. (Vous pouvez utiliser des caractères de remplacement).

Copie de Fichiers

La fonction de Copie est un programme de copie universel, qui peut copier N'IMPORTE QUEL fichier AMSDOS standard sur DISQUE ou BANDE ou à partir d'un DISQUE ou d'une BANDE. Si vous la sélectionnez, le système affiche:

1: Disc-Disc 2: Disc-Tape 3: Tape-Disc

Vous devez donner une réponse valide sans quoi la commande repasse au message initial. Après avoir sélectionné 1, 2 ou 3 vous devez entrer un nom de fichier d'ENTREE. L'option 3 ne requiert pas de nom de fichier.

Après ouverture du fichier d'ENTREE, le système affiche l'information d'en-tête. Vous pouvez ensuite poursuivre la copie en répondant (Y) à la question COPY-?.

Selon l'option choisie et le type de fichier détecté, l'opération de copie produit plusieurs messages. Les options et réponses valides sont décrites sur la page suivante.

NOTE: Vous pouvez modifier la Bande ou le Disque Destinataire AVANT d'entrer la réponse à 'PROTECTION?'.

Restrictions de l'Option de Copie

| | Disque/Disque | Disque/Bande | Bande/Disque |
|---------|--|--------------|--------------|
| ASCII | Compression? Répondez Y ou N Limite 36K *** | | |
| BINAIRE | Limite 36K *** | Note-1 | Note-1 |
| BASIC | Limite 36K *** | | |
| CPM | Compression? Répondez SPACE-BAR Limite 36K *** | | |

Vous pouvez INTRODUIRE ou RETIRER la protection dans tous les cas.

Figure 6.0 Options de Copie

Note-1: Un fichier BINAIRE écrit sur bande par l'intermédiaire du système Utilitaire PYRADEV DOIT être ré-enregistré sur disque avec le même programme Utilitaire de manière à obtenir l'adresse de point de CHARGEMENT correcte dans l'en-tête de fichier. Ces fichiers NE DOIVANT PAS être EXECUTES avant cette opération. Vous pouvez cependant les CHARGER dans un emplacement de mémoire spécifique, à partir d'une bande et après avoir utilisé la touche CALL. (Si vous écrivez des jeux : ATTENTION !).

7.0 ENTRAÎNEMENT

Comme vous l'aurez réalisé, le système PYRADEV est petit mais puissant. Il possède de nombreuses caractéristiques et fonctions qu'il vous faudra découvrir et de nouvelles commandes que vous devrez apprendre. C'est pourquoi le fichier 'PROGRAM.001' existe sur le DISQUE SYSTEME (MASTER DISQUE) afin de vous permettre d'apprendre à Assembler, à Editer et en fait à jouer.

Le programme contient un seul sous-programme dénommé 'DEBUG'. Il peut être CHARGE (LOAD) et APPELE (CALL) à partir du BASIC et affichera le contenu des registres sur la rangée 25 de l'écran. Ce programme est accompagné de nombreux commentaires en deux types de texte.

Le (TEXTE EN MAJUSCULES) qui est entre parenthèses concerne entièrement le code du programme. Nous espérons que vous apprendrez rapidement comment le programme fonctionne, et que vous pourrez peut-être l'adapter comme sous-programme de mise au point additionnel au code que vous allez écrire.

Le texte normal (;*Normal Text*) renfermé entre des points virgules, contient des informations sur l'emploi de l'Editeur de Fichier Source. Vous devrez lire ces commentaires et vous exercer aux fonctions. Lorsque vous aurez fini de "jouer" avec l'Editeur, appuyez sur CTRL-A pour l'abandonner, puis (Y) pour confirmer et sortir.

Le fichier PROGRAM.001 peut être Assemblé tel quel. Faites-le dès le début avec le Macro Assembleur, et utilisez les diverses options OUTPUT (SORTIE) pour voir ce qui arrive. Nous vous recommandons d'utiliser tout d'abord les options implicites.

Le fichier PROGRAM.002 est aussi fourni sur disque. Il constitue un exemple complet de TOUTES les instructions Z80. Si vous avez des programmes de code, vérifiez si vous avez utilisé la syntaxe correcte, et les mnémoniques de code en examinant ce programme. Il contient également des exemples des directives d'Assemblage décrites dans la section 3. Il constitue un élément de code de référence utile et peut être Assemblé et Listé comme tableau de référence.

Le fichier PROGRAM.003 est un simple programme FILE-COPY (FICHER/COPY). Vous pouvez le modifier et l'utiliser pour transférer d'autres formats de fichiers ASCII dans le système PYRADEV, en éditant un code spécialisé qui effectuera des modifications spéciales durant l'opération de transfert/copie.

Soyez sûr que vous comprenez à fond le fonctionnement du système avant de l'utiliser pour un code de projet réel, et veillez à toujours conserver des copies de sauvegarde de votre code source d'élaboration.

Merci d'avoir acheté PYRADEV. Nous espérons qu'il vous plaira.

8.0 RECAPITULATIF DES TOUCHES DE COMMANDE

Editeur (mode d'édition)

CTRL-A: Abandon et Sortie
CTRL-B: Début Pointeur
CTRL-D: Effacement Bloc
CTRL-E: Fin Pointeur
CTRL-F: Trouver Marqueur
Bloc num.(.): Impression
ESCAPE: SOS-Affichage
CTRL-G: Goto Ligne nnn
CTRL-L: Posit. Marqueur
CTRL-O: Ouvrir Text
CTRL-S: Recherche Système
CTRL-T: Poser les Tabs

Editeur (mode commande)

A: Unité AS Opérante
B: Unité B Opérante
C: Compression
D: Effacement Fichier
R: Changement Nom Fichier
Z: Sauvegardes ZAP

Assembleur (touches de commutation de temps d'exécution)

V: Sortie Vidéo
P: Listage Impression
S: Impression Sélective
E: Impression Erreurs

Moniteur (menu principal)

T: Définition valeur piste
Z: Définition valeur secteur
R; Changement Registres
M: Modif. avec HEXA
CTRL-A: Modif. ASCII
G: Goto une Adresse
CTRL-G: Défini. adresse de pas
CTRL-T: BANDE/DISQUE
SHIFT 0,1,2: Déf. Mode
S: Exéc. Pas Unique
CTRL-S: Double Pas
L: Charge. Fichier Test
W: Ecrire Fichier Disque

Disc-Nurse (menu principal)

D: Sélection Unité Disques
T: Définition valeur piste
S: Définition valeur secteur
F: Sélect. Fichier
U: Annul. effacement fichier
Bloc num.(.): Impression écran
R: Lecture Secteur
W: Ecriture Secteur

Programme Utilitaire (menu principal)

A: Affichage Unité A
B: Affichage Unité B
D: Effacement Fichier

Programme Utilitaire (menu copie)

1: Disque/Disque (Compression facultative ASCII)
2: Disque/Bande (Introduction ou Extraction de Protection)
3: Bande/Disque (Affiche toujours l'Information d'entête)

Bloc num. (0): Mémoires KS
Petit-Enter: Commute KS
CTRL-U: Annuler Changements
CTRL-V: Visual. Début/Fin
CTRL-X: Sauvegarde et Sortie
CTRL-Y: Varier Curseur
CTRL-Z: Mode Commande
Bloc num(1-9): Relecture KS
Marche/Arrêt enregistrement

L: Options Chargement
S: Options Sauvegarde
J,CTRL-L: Papier
K,CTRL-K: Encre Plume

O: Autre Affichage
U: Actualisation Ecran
N: Régénération écran
CTRL-C: Catalogue
CTRL-B: Sélection Bloc
Q: Interrogation / Recherche
X: Examiner toutes ROM
Y: Copie Zone Mémoire
CTRL-D: Désassemblage
CTRL-X: Sortie Moniteur
J,CTRL-J: Encre Papier
K,CTRL-K: Encre Plume
Bloc num.(.): Impression Ecran

Q: Interrogation/Recherche
J,CTRL-J: Papier
K,CTRL-K: Encre Plume
C: Affichage Catalogue
CTRL-X: Répertoire Etendu
M: Modifier avec HEXA
A: Modifier avec ASCII
(CTRL-H: Total contrôle En-têtes)

R: Changement nom Fichier
X: Sortie Programme
C: Copier Fichier AMSDOS

