

1. NedoOS - Multitasking operating system for ZX Spectrum

1.1. Features

- Works with TR-DOS floppies, SD-card, IDE HDD (FAT12/16 or FAT32 with long filenames support). Device letters:
 - "E".."H" - IDE HDD (Master),
 - "I".."L" - IDE HDD (Slave),
 - "M" - SD-card (Z-controller),
 - "N" - SD-card (NeoGS),
 - "O" - USB flash,
 - "A".."D" - TR-DOS floppies. IDE controller (ATM or Nemo IDE) depends on the batch file you run. Includes support for segmented TR-DOS files of any size (according to TR-DOS sequential access files standard).
- Up to 16 tasks running at the same time. Tasks may be active (one of them has the focus, so it can read input devices and print on visible terminal screen) or frozen. Tasks may give away their time slot to the system using YIELD, but don't have to.
- NedoOS can open up to 8 files on FAT, up to 8 files on TR-DOS, and up to 8 pipes between tasks at the same time.
- User program can access the whole memory between 0x0100..0xffff, any 16K window can be switched via OS calls (see below). File operations and BDOS data transfer can be done at any address in the userspace.
- User programs can modify the interrupt handler (for example, to restore stack data) and move the stack pointer.
- Gfx editor Scratch, music players NedoPlayer and modplay, text editor texted, disk image mounter dmm, snapshot runner/switcher nmisvc, compiler NedoLang, assembler NedoAsm, basic interpreter NedoBasic, picture viewer NedoView, decompressors for *.zip, *.gz,

*.rar, *.tar and archivers for *.rar and *.tar, network utilities (NedoBrowser, dmirc, dmftp, netterm etc.), games...

1.2. System requirements

- ATM Turbo 2 or ATM3 compatible computer (depends on the main executable)
- Kempston mouse with a wheel recommended
- SD-card recommended (Z-Controller with shadow ports or NeoGS) or harddrive (NemoIDE or ATM IDE)
- DDp's palette scheme (4+4+4) recommended
- Real-time clock (Mr.Gluk's schematics) recommended
- ZXNETUSB network interface card recommended
- General Sound or NeoGS sound card recommended
- TurboSound FM sound card recommended

1.3. Keyboard shortcuts

NedoOS handles keys and mouse events and forwards them to the task that is in focus. The key combinations Ext+letter (Tab+letter) correspond to ASCII control codes 1..26. The Ext+Number key combinations (Tab+Number, CapsShift+SymbolShift+Number) are equivalent to function keys F1..F10 on a PS/2 keyboard.

Keyboard shortcuts used by the kernel (PS/2 shortcuts are shown in brackets):

- Caps Shift (Left Shift) - hold down to stop scrolling on the screen
- Symbol Shift + Enter (Right Shift + Enter) - switches visual tasks (that is, those that called CMD_SETGFX), while the task to which they were switched receives the key_redraw key code (upon learning this, it can redraw the screen)
- C+M+D (press simultaneously) - execute cmd.com if there are no active tasks. (polled in the idle task, so it works only when there

are no active tasks). To spawn a new terminal, use the command `term`.

- Caps Shift + 2 (Caps Lock) - switch case
- Caps Shift + 1 (Alt + Shift) - switch between keyboard layouts (Russian/English). In the Russian mode (it also supports the Ukrainian language), the ШВЕПТЫ layout is enabled. In this case, you can type some letters by two presses (ja=я, jo=ё, ju=ю, je=е, ji=ї, jy=й, cg=_, cc=ц, hh=э, jj=ъ). Because letters “й”, “ц”, “ш”, “э” are used as a prefix of combinations, you can input them separately by pressing any other key after them that is not included in their combination. The key pressed afterward is not ignored, but also processed. In the literary text, the combinations “йа”, “йо”, “йу”, “йе”, “чч”, “хх”, “йи”, “йы”, “йй”, “цг” are practically excluded (sometimes in foreign words). To enter such combinations use cursor movement. All punctuation marks available in Latin mode are also available in Russian. The Russian encoding is CP866 and CP1125 is used as Ukrainian encoding.
- Ext+Enter (Tab+Enter, Caps Shift+Symbol Shift+Enter, use Tab+Right Shift+Enter on incompatible keyboards) - switch on/off the pseudo-graphic input mode
- In case of TR-DOS errors (the red border is shown), the R (Retry), I (Ignore sector), A (Abort) keys work.

1.4. cmd - Command interpreter

It is an interactive prompt (command line) with scrolling to the left and right. Cursor key Up will recall the previous command.

Following file types are executed (the first character of the extension is checked):

- .com (the extension can be omitted) - an executable program, a new parallel process is created. Command-line parameters are passed to the process to the `COMMANDLINE` address (see the “Programming” section).

- `.bat` is a sequence of commands that `cmd` understands, including starting `.com` programs (commands are executed sequentially and not concurrently, except when the `start` command is used). Each command is shown on the screen before it is executed.

Programs located in the `bin/` directory of the system disk can be executed in the command line from any directory of any disk (the current directory has priority for calling programs before the `bin/` directory). The current directory does not change when programs are executed. The following commands (and their aliases) are supported:

- `exit` - quit `cmd`
- `a: ... o:` - change the current disk
- `dir (ls)` - list the content of the current directory. You can redirect output to a file: `dir> filename`
- `cd <path>` - change the current directory to `.` can contain a drive name.
- `cd ..` - change directory to the parent directory.
- `md (mkdir) <path/dirname>` - create a new directory with the specified name in the current or the specified path
- `del (rm) <path/filename>` - delete a file or an empty directory in the current directory
- `copy (cp) <path/filename> <newpath/newfilename>` - copy file
- `ren <path/filename> <newpath/newfilename>` - rename or move file
- `mem (free)` - print the number of free memory pages
- `proc (ps)` - list current processes and their state ("`+`" or "`-`" - activity, "`g`" for graphical tasks)
- `drop (kill) <ID>` - terminate the process with the given ID
- `date` - print current date and time
- `rem` - do nothing (needed for comments in `.bat` files)
- `start <path/filename>` - execute the program in the background (by default, the executed program blocks `cmd`)
- `copydir <dir1> <dir2>` - recursive copy of the `dir1` directory with all files to the `dir2` directory. Full paths must be specified, not rela-

tive!

- `pause` - wait for a keypress (used in `.bat` files)
- `echo <message>` - print a message (needed for `.bat` files)
- `type <path/filename>` - print file content

You can also use the complex syntax:

```
dir > filename.txt
dir | more.com
more < filename.txt
```

The command-line parameters of the `.bat` file are available through macros `%0` to `%9` (`%0` contains the filename of the script).

The *idle* system task launches the `term.com` terminal, which creates `stdin` and `stdout` pipes and launches the shell to execute `autoexec.bat`. When there is no running process, the *idle* enters into an infinite loop and it is waiting for C+M+D key combination (see above).

When launched, the shell looks at its command line and executes it. If the command line is `cmd.com autoexec.bat`, the shell enters interactive mode after `autoexec.bat` is executed, otherwise exits. You can start another independent terminal with `term` command or a network terminal (`netterm`) accessible via Telnet (TCP/IP port 2323).

1.5. **nv, nvfast - Nedovigator, dual-pane file manager**

Keyboard short cuts:

- cursor keys (up, down) - navigate through files
- Home (Symbol Shift+Q) - jump to the first file
- End (Symbol Shift+E) - jump to the last file
- Tab (Caps Shift+Symbol Shift) - change the current panel

- Space - tag file
- * - invert tagging
- BackSpace (Caps Shift+0) - switch to the parent directory
- Enter - launch the file under cursor in the blocking mode (.com and .\$.c are launched directly, other types may use the external launchers, see below)
- Enter - execute command line in the blocking mode, then nv will release the focus and will wait till the end of the called program or till OS_HIDEFROMPARENT in that program. Text output of the finished program can be displayed with the mouse wheel (in nv) or Esc (in nvfast).
- Caps Shift+Enter or F9 - insert the file name on the command line
- 3 - text files viewer
 - use cursor keys, PageUp (Caps Shift+3), PageDown (Caps Shift+4),
 - s - switch encoding,
 - w - switch line breaks,
 - Break (Esc, Caps Shift + Space) - exit,
 - Tab (Caps Shift + Symbol Shift) - switch to hex viewer/editor:
 - * use cursor for moving control, PgUp, PgDn,
 - * input numbers and A-F
 - * Caps Shift+Enter or F2 - save,
 - * Break (Esc, Caps Shift + Space) - exit,
 - * Tab (Caps Shift + Symbol Shift) - go to the normal text viewer
- 4 - open the text editor *texted* to modify the file under the cursor
- 5 - copy marked files or the file under the cursor to the opposite panel
- 6 - rename the file/directory (do not use "/" and "" in the name)

- 7 - create a directory in the current panel (Esc [CS+Space] - cancel)
- 8 - delete tagged files (only empty directories are deleted)
- 1, 2 - select a drive in the left or right panel (cursor, Esc [CS+Space] to cancel, Enter - confirm selection)
Note: Instead of numbers, you can press F1..F10 on the PS/2 keyboard.
- Symbol Shift+1..5 - select sorting mode (by name, extension, size, date, and no sorting). The sorting order is reversed if the same mode is invoked a second time. The change to a different mode will use the ascending order.
- Break (Esc, Caps Shift + Space) - exit (Esc - cancel, Enter - confirm)

The remaining keys are used to edit the command line (numbers are not entered when the command line is empty).

The text output of the launched program from the command line is saved (but not a program launched in another terminal). See above, how to display it.

The configuration file `nv.ext` describes the association which launcher is used for the given file extension, for example:

```

bmp, scr: scratch.com
bat: cmd.com

```

1.6. `texted` - Text editor

Provide `texted` with the name of the file for editing as the command line parameter. The file size is only limited by the amount of available free memory.

Shortcuts:

- cursor keys, PageUp (Caps Shift+3), PageDown (Caps Shift+4) - cursor movement
- Home (Symbol Shift+Q) - jump to the beginning of the line

- End (Symbol Shift+E) - jump to the end of the line
- BackSpace (Caps Shift+0) - delete a character before the cursor
- Del (Caps Shift + 9) - delete a character to the right of the cursor
- Caps Shift+Enter or F2 - save
- Break (Esc, Caps Shift + Space) - exit

Other keys are used to input the text.

1.7. basic - NedoBasic, BASIC interpreter

Files with *.bas extension can be called as command line parameter.

Data types:

- integers (32bit signed), can be used for boolean values (0=false, -1=true). If used as an index for a cycle, they consume more memory,
- strings (up to 255 bytes + zero terminator), also used as an array of unsigned bytes,
- one-dimensional arrays of integers (32bit signed).

Variable names must consist of only one alphabetical character, for example:

- i (number),
- a\$ (string),
- a(10) (10th element of the array, counting from 0),
- a\$(10) (10th character of the string, counting from zero).

The following operations are used in expressions:

- lowest priority: =, <, >, <=, >=, <>
- medium priority: +, -
- high priority: * and /
- highest priority: unary -, expression brackets ()

The function \$rnd generates random numbers in the range 0..65535, at least one space must be used after the function name!

Commands (it is possible to join several commands on one line separated by a colon):

- `run` - start the program
- `list` - view the listing of the program
- `quit` - exit NedoBasic
- `edit <expression>` - call to edit the line with the specified number
- `clear` - clear variables
- `new` - erase the program
- `let <variable>=<expression>`
- `print <expression>; <expression>` - if the last character is a semi-colon, then no line feed is printed at the end)
- `cls` - clear the screen in black
- `goto <expression>` - jump to the specified line number. When the number does not exist the program will continue from the next line after the used one.
- `if <expression> then <command>` - if expression is not false, execute commands to the end of the line
- `dim <variable> (<expression>)` - create an array of specified size
- `for <variable>=<initial value expression> to <final value expression> step <step expression>` - the beginning of the cycle. Step can be positive or negative, but not 0.
- `rem <text>` - use for text comments
- `next <variable>` - the end of for-cycle
- `gfx 0` - enable graphic mode 320x200x16 colors
- `gfx 6` - enable text mode (when the program exits, it will automatically enable text mode back)
- `pause` - waiting for the keypress
- `plot <x-expr>, <y-expr>, <color-expr>` - draw a point in the graphic mode
- `line <x2-expr>, <y2-expr>, <color-expr>` - draw a line in the graphic mode (from the previous point or the end of the previous line)
- `save <filename>` - save the program. The filename can contain

the path, for example, "m:/path/file.bas". The filename can be also a string variable.

- `load <filename>` - load a program. The filename can contain the path, for example, "m:/path/file.bas". The filename can be also a string variable.
- `system <command_line>` - launch a command via cmd, it is waiting for completion. Also, a string variable can be used.

Press Esc (Break, CS+Space) to break the execution of the program or code scrolling.

1.8. **tp - Turbo Pascal 3.0 by Borland**

Short cuts are displayed on the screen. Source code can be compiled to the memory or file. English documentation is located in:

<http://www.retroarchive.org/docs/software/turbodoc.html>

See examples - `t.pas` ("Hello" in a cycle), `mc.pas` (spreadsheet).

1.9. **cc, cc2, clink, c.ccc, deff.crl, deff2.crl - BDS C compiler**

BDS C compiler was developed by Brain Damage Software. Example: `cc.bat ex` (compile `ex.c`, link and run the code).

```
#include <stdio.h>
```

```
main(argc, argv)
```

```
char **argv;
```

```
{
```

```
    int i;
```

```
    printf("Hello world!\n");
```

```
    for (i = 1; i < argc; i++) printf("Arg #%d = %s\n",i,argv[i]);
```

```

    getchar();
}

```

Compiler collection examples and usage:

- `cc filename.c`
- `cc2` - second compilation is invoked automatically
- Object files and libraries have the `.crl` extension (`deff2.crl` is automatically connected by the linker). Examples of libraries are `deff2a.csm`, `deffgfx.csm`.
- Link object file to executable:
 - `clink filename`
 - `clink filename deffgfx` (see `cc.bat` and example `ex.c`)
- How to build the concatenation tool (`concat outfile infile1 infile2`):
 - `cc concat.c`
 - `clink concat`

1.10. player - NedoPlayer

The simple music player for `*.pt2`, `*.pt3` (with TurboSound support) and `*.tfc` formats. Use the command-line parameter to specify the filename for playing. Prints the filename when the screen is refreshed. To exit use Break (Esc, Caps Shift+Space).

1.11. modplay

The simplest MOD music player for General Sound compatible sound cards. Use the command-line parameter to specify the filename for playing. Launching without parameters is going to stop playback.

1.12. pkunzip

Unpacker for `*.zip` and `*.gz` archives. It unpacks the entire archive to the current directory.

1.13. tar

Unpacker for *.tar archives. It unpacks the entire archive to the current directory. If the parameter is not a .tar archive, then this is a file from which the archive will be created (if a directory is specified, then it will all be packed into an archive with all content, files, and directories). The archive name corresponds to the file name, with the extension replaced by .tar.

1.14. unrar - Unpack *.rar (2.x) archives

It reads the name of the archive via the command line, then it works in the interactive mode.

- v - view the contents of the archive
- e - unpack necessary files from the archive
- m - enter a mask for files (the first characters of the file name in the archive, including the internal path)

1.15. zxrar - Packer of (2.x) *.rar archives

It reads the name of file via the command line and creates archive mynewrar.rar or adds to it.

1.16. browser - NedoBrowser, web browser

NedoBrowser is a text-based web browser. Invoke from the command line:

- browser file://m:/girl.jpg (file:// is optional)
- browser http://alonecoder.nedopc.com/ (http:// and the trailing slash is optional in the URL)
- HTTPS protocol (https://) is supported with a proxy.

The status bar is displayed at the bottom of the screen, it contains:

- full path to the current file
- number of busy pages
- rendering time
- errors (conn.err - connection error, load err - loading error)

Following formats are supported:

- html (does not support all tags yet and displays only windows-1251 and UTF-8 encodings [by default]),
- jpeg (so far only plain scan),
- gif (so far only normal scan, animations are supported),
- png (only normal scan for now),
- bmp (so far only normal line order, 24bit),
- svg (no fill, coordinates in a limited range).
- For large pictures use cursor keys, Z - changes the scale.

Short cuts:

- cursor keys, PageUp (Caps Shift+3), PageDown (Caps Shift+4) - movement
- Enter - follow the link
- S - save the current file (it will be saved as `download.fil`, the first letter is increased with each file)
- L - download the file from the hyperlink (calls the `wget` program)
- 5 - reload the file
- E - edit url (arrows left, right, Enter, BackSpace [Caps Shift+0])
- U - change encoding UTF-8 / windows-1251
- BackSpace (Caps Shift + 0) - back in the browsing history
- Break (Esc, Caps Shift + Space) - exit browser

1.17. wget

The non-interactive HTTP downloader. It with play music files and display `*.src` images automatically.

1.18. dmm

The tool for mounting of TRD, SCL, FDI, TAP images via Evo Service. Can also mount TRD images using xBIOS ROM at ATM2.

1.19. time

The tool for network time protocol (NTP). Command-line options:

- -H help
- -T set time(-T17:59:38)
- -D set date(-D21-06-2019)
- -N ntp-server default: -N2.ru.pool.ntp.org
- -Z time-zone default: -Z3
- -i get date time from the internet

1.20. dmirc - IRC client

1.21. dmftp - FTP client

1.22. ping

Send ICMP ECHO_REQUEST to network hosts. Example: ping 1.2.3.4.

1.23. telnet - Telnet client

The user interface to the TELNET protocol. The TCP port can be specified with: telnet url:1234, the default port is 23.

1.24. 3ws - Web server

The web server supports the sharing of the system disk. See 3ws.txt for more details. You can use your own page design (files in the subdirectory of the same name).

1.25. wizcfg

Provides basic network setup of ZXNETUSB. It uses configuration saved in `net.ini`.

1.26. nmisvc

Launches and translates snapshot. Runs a `.SNA` snapshots (48K or 128K) specified on the command line. Exit the snapshot to OS by pressing the NMI button. You can save the memory state to a new snapshot or continue the execution. At the same time, you can manually switch to another task in the OS. OS allows you to between snapshots and the system.

It can also run BASICs from TR-DOS disc.

1.27. view - NedoView, image viewer

NedoView supports the following graphics formats:

- `scr` (6144 and 6912)
- `fnt` (linear and screen format, 768 and 2048)
- `img` (dual screens with interlace)
- `3` (AGA editors, 8col)
- `888` (8col editor)
- `+ or -` (MultiStudio editor)
- `Y` (packed 8-color picture for the ManyColor+/XColor)
- `plc` (Laser Compact 5, BGE)
- `mc` (multicolor)
- `mlt` (multicolor from ZX Paintbrush editor)
- `mcx` (multicolor with interlace)
- `grf` (hardware multicolor ATM/Profi)
- `ch$` (large pictures with attributes, with or without interlace)
- `mg1, mg2, mg4, mg8` (MultiArtist editor)
- `rm` (R-Mode)
- `16c` (32K memory image + 32 bytes palette)

1.28. hddfdisk

Utility for partitioning and formatting IDE partitions.

1.29. term

Terminal emulator for programs that use `stdio.asm`.

- Transmits ANSI codes (VT-100+) with mouse events.
- Scrolling with the mouse wheel.
- Mouse click in the top left corner will save terminal text to `pasta.txt`
- Mouse click in the bottom left corner will insert 80 characters from `pasta.txt`.

1.30. netterm - TELNET server

The network terminal server listens on port 2323. Supports programs that use `stdio.asm`. Set the following configuration in your telnet client (for example PuTTY):

- VT-100
- Local echo off
- Local line editing off
- Backspace = Control-H

1.31. pt - Pro Tracker 3.x

Pro Tracker 3.x provides a larger window and supports hard disk and General Sound (see the manual in a separate file).

1.32. playtap - Player for .tap files.

TAP files are played on your physical tape output. The file to play must be named `tilt.tap`.

1.33. Programming (for details see `api_base.txt`)

User programs are compiled with the header file `../_sdk/sys_h.asm`, which includes the file `sysdefs.asm` with constants.

Programs are loaded and run with `PROGSTART (0x0100)`, with the command line in `COMMANDLINE (0x0080)` and with its length stored at `COMMANDLINE_sz (0x0080)`. It is highly recommended to use these constants rather than numeric values (the same goes for key codes, call numbers, etc.). The stack initially grows from the top of `0x0000`, it can be manually rearranged to any place above `0x3b00`. File operations and data transfers in `BDOS` can be done at any address in the userspace.

It is not recommended to read keyboard ports manually, use `OS_GETKEYMATRIX` – it returns the pressed keys only to the task that is currently in focus.

It is not recommended to use the `syssets.asm` dependency in user programs.

Application developers use the symbolic key names defined in `sysdefs.asm` under “Usable key codes”.

The system is currently unable to assemble itself. To reach this goal, we will follow a number of guidelines for programming in assembly language:

- it is not recommended to use external utilities (`*.exe`) for building programs, except for the assembler and the `NedoLang` package.
- it is recommended to format hexadecimal numbers in the `0xffff` format, it is not recommended to use binary numbers (in extreme cases, write in the `0b0101` format).
- it is recommended to write arithmetic expressions so that they are executed correctly even in the absence of priority of operations. If this requires starting an expression with a parenthesis, write a `+` sign before the parenthesis.
- it is recommended to write arithmetic expressions so that they are executed correctly in unsigned multiplication and division.
- it is not recommended to use `ifn a == b`, use `if a != b`.
- it is not recommended to use `ORG` other than the initial one. Use `ds addr-$`.

- it is not recommended to use `DUP` . . `EDUP`, you can use `include` for large blocks, and `expand` small ones.
- it is not recommended to use `EQU` and `STRUCT`, use the “=” sign.
- it is not recommended to use digital labels and transitions of type `1b` (especially transitions of type `1f!`).
- it is not recommended to write several commands in a line and several sets of parameters for one command.
- it is not recommended to use UTF-8 encoding, use `Windows-1251` or `CP866` instead.

As the native build system will be improved, these restrictions will be relaxed.

Restarts in the kernel (it is highly recommended to use them through macros since it is planned to free `0x0000` for the user, make `CALLBDOS` a restart, and `SETPG` . . . calls for speed):

- `QUIT` (`0x0000`) - close the current task and free its memory
- `CALLBDOS` (`0x0005`) - calling `BDOS` (see functions in `sys_h.asm`, function number in `C`). You should not call this macro directly, for each command there is a separate macro `OS_ . . .`. Registers are not saved!
- `GET_KEY` (`0x0008`) - read a key (`HA`=code with language, `BC`=code without language, key codes are specified in `sys_h.asm`) and read the mouse at the same time (`de`=mouse position (`y`, `x`), `l`=mouse buttons (bits 0, 1, 2: 0=preserved)), `nz`=the program is not in focus, the buttons are not displayed, the mouse position must be ignored (`=0`)
- `PRCHAR` (`0x0010`) - print character `A` (registers are not saved!)
- `SETPG16K` (`0x0018`) - enable page `A` at `0x4000` (corrupts the `BC` register). The page number is stored in (`CURPG16K`)
- `SETPG32KLOW` (`0x0020`) - enable page `A` at `0x8000` (corrupts the `BC` register). The page number is stored in (`CURPG32KLOW`)
- `SETPG32KHIGH` (`0x0028`) - enable page `A` at `0xc000` (corrupts the `BC` register). The page number is stored in (`CURPG32KHIGH`) (switch the page at `0x0000` via `OS_SETMAINPAGE`, while the page must have a kernel!)

- 0x0030 - far call is planned
- 0x0038 - interrupt handler

BDOS functions: see `sys_h.asm`.

The entry to the standard interrupt handler looks like this:

```
push af
push bc
push de
```

How to capture an interrupt handler in your program:

- replace 3 bytes at 0x0038 with `jp <address of your handler>`, and take the `intjp` address from 0x0038+3 and copy to yourself.

For example:

```
swapimer                                ;the first call will turn on your
handler,                                  ;the second call will return the
standard handler
di
ld hl, (0x0038 + 3) ;address intjp
ld (intjpaddr), hl
ld de, 0x0038
ld hl, oldimer
ld bc, 3
swapimer0
ld a, (de)
ldi                                     ;[oldimer] -> [0x0038]
dec hl
ld (hl), a                               ;[0x0038] -> [oldimer]
inc hl
jp pe, swapimer0
ei
ret
```

```

oldimer
    jp on_int          ;will be replaced with the code from
0x0038
    jp 0x0038 + 3

```

- your interrupt handler should call oldimer. For example:

```

on_int
    ex de, hl          ;de = "hl", hl = "de"
    ex (sp), hl        ;hl = exit address, de = "hl", on the
stack "de"
    ld (on_int_jp), hl
    ld (on_int_sp), SP
    ld sp, INTSTACK    ;in order not to mess up the stack
    push af
    push bc
    push de             ;"hl"
    ...
    call oldimer        ;ei
    ...
    pop de              ;"hl"
    pop bc
    pop af
on_int_sp = $ + 1
    ld sp , 0
; de = "hl", on the stack "de"
    pop de
;

```

- do not use YIELD, use HALT instead. Otherwise, the interrupt will go to another task, and there is another interrupt handler.
- Call NedoOS functions (except for page switching) either in the interrupt handler or immediately after HALT.
- if you just install your music player, use OS_SETMUSIC, while you can ignore the previous two points (music will work anyway).

1.34. Developers

- Project manager, code, documentation – Dmitry Mikhailovich Bystrov (Alone Coder/Conscience).
- Networking, patches to the disk subsystem, utilities, testing - DimkaM.
- A bit of code and documentation – Nikolay Aleksandrovich Grivin.
- NedoBasic was written with the participation of Kirill Lovyagin as part of the assembly language training.
- Further development of NedoBasic and Nedovigator, build scripts for Linux, utilities – demige.
- Sorting files, fixing build scripts for Linux, aynet_psg utility, sjasm and UnrealSpeccy fixes – Lord Vader.
- rdtrd, wrtrd – Konstantin Kosarev.
- ZX Battle City game – Slip, music – nq, testing – Videogames Sematary, porting – Alone Coder.
- Porting the game Eric and the Floaters – Rasmer, coloring – Alone Coder, Sashapont.
- Porting the game Black Raven – Alone Coder, coloring – Alone Coder, Sashapont, Kitty, Louisa.
- Logos – Louisa, Sashapont, Wizard.

The disk subsystem is based on the FatFS library with drivers from Savelij13 and DimkaM and the iofast library from the NedoLang suite. The operating system idea was brought in 2007 when the first version of the kernel was written (but never tested). The main portion of the kernel code, cmd and nv, and part of Scratch graphic editor was written in 2018.

1.35. License

Free distribution of the program and its source code is allowed. Please ask for the approval from project manager if you want to port the code or its parts to a different platform.

1.36. System setup

Clock usage according to Mr. Gluk's schematics for ATM2 is tested in UnrealSpeccy 0.37.1.

Disk images (the HDD image can be mounted through WinImage, and the SD-card image can be mounted in Windows). Do not forget to unmount it once done:

<http://alonedcoder.nedopc.com/sd.zip>

<http://alonedcoder.nedopc.com/hdd.zip>

Installing the system on a real HDD:

1. Run mkatm2.bat (or mkatm3.bat for ATM3).
2. Using nv, copy all the files from the *.trd received to e:/bin/.
3. Run mkatm2hd.bat (or mkatm3hd.bat for ATM3), copy the resulting *. $\$c$ to drive 1.

Then you can start the system (*. $\$c$) from the HDD.

Installation of the system on a real SD-card: copy the bin/ directory and the required $\$c$ to the root of a SD-card.

Settings for the UnrealSpeccy emulator:

```
[ZC]           ; Z-Controller settings
; sdcard image
SDCARD="sd.vhd"
```

```
[HDD]
Scheme = NEMO-DIVIDE (for ATM3 version) or ATM (for ATM2 version)
```

```
Image0 = hdd.ima
CHS0=609/16/63 ; max size, accessible through CHS. not used
for real drive
LBA0=614400    ; max size, accessible through LBA. not used
for real drive
HD0R0=0       ; read only flag
```

CD0=0 ; 1 if image is cdrom

eVHDattach.bat (you can mount also by clicking on *.vhd):

```
d:
cd zx\us035\
@echo off
if exist VHDattach.txt (
    @echo on
    echo VHDattach.txt exist
    @echo off
) else (
    @echo on
    echo create VHDattach.txt
    @echo off
    echo select vdisk file="%cd%\sd.vhd" > VHDattach.txt
    echo attach vdisk >> VHDattach.txt
    rem echo select vdisk file="%cd%\sd.vhd" >> VHDattach.txt
    rem echo select part 1 >> VHDattach.txt
    rem echo assign letter=K >> VHDattach.txt
)
if exist sd.vhd (
    @echo on
    echo sd.vhd attach
    diskpart /s VHDattach.txt
    @echo off
) else (
    @echo on
    echo create VHDcreate.txt
    @echo off
    echo create vdisk file="%cd%\sd.vhd" MAXIMUM=300 TYPE=FIXED
>
VHDcreate.txt
    echo select vdisk file="%cd%\sd.vhd" >> VHDcreate.txt
    echo attach vdisk >> VHDcreate.txt
```

```

echo create part primary >> VHDcreate.txt
echo select part 1 >> VHDcreate.txt
echo format label="ZX" quick fs=FAT32 >> VHDcreate.txt
echo assign >> VHDcreate.txt
@echo on
diskpart /s VHDcreate.txt
del VHDcreate.txt
@echo off
)
@echo on
pause

```

eVHDdetach.bat:

```

d:
cd zx\us035\
@echo off
if exist VHDdetach.txt (
    @echo on
    echo VHDdetach.txt exist
    @echo off
) else (
    @echo on
    echo create VHDdetach.txt
    @echo off
    echo select vdisk file="%cd%\sd.vhd" > VHDdetach.txt
    echo detach vdisk >> VHDdetach.txt
)
@echo on
diskpart /s VHDdetach.txt
pause

```

On Windows 10, you can use right-click to mount and unmount.