# IDOS kernel manual

2025--2026 by E. C. Masloch. Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument. DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

This document has been compiled on 2026-11-06.

# Contents


Section 1: Building the lDOS kernel . . . . . 4

Section 2: Starting the lDOS kernel . . . . . 6

2.1 Booting lDOS . . . . . 6

2.2 Installing boot sector loader using INSTSECT . . . . . 6

Section 3: Supported file systems . . . . . 8

Section 4: lDOS configuration . . . . . 9

4.1 Query patch . . . . . 9

4.1.1 Supported `patldos query` values . . . . . 9

4.1.2 Parsing of numbers in `patldos` . . . . . 9

4.2 lCFG block . . . . . 10

4.2.1 Supported `patldos progress` values . . . . . 10

4.2.2 Supported `patldos debug` values . . . . . 11

4.2.3 Supported `patldos verbose` values . . . . . 11

4.2.4 Supported `patldos scanmode` values . . . . . 11

4.2.5 Supported `patldos haltflag` values . . . . . 12

4.3 Kernel command line . . . . . 12

4.4 Configuration file . . . . . 13

Section 5: Additional usage conditions . . . . . 17

5.1 BriefLZ depacker usage conditions . . . . . 17

5.2 LZ4 depacker usage conditions . . . . . 17

5.3 Snappy depacker usage conditions . . . . . 17

5.4 Exomizer depacker usage conditions . . . . . 18

5.5 X compressor depacker usage conditions . . . . . 18

5.6 Heatshrink depacker usage conditions . . . . . 19

5.7 Lzd usage conditions . . . . . 19

5.8 LZO depacker usage conditions . . . . . 19
5.9 LZSA2 depacker usage conditions . . . . . 19
5.10 aPLib depacker usage conditions . . . . . 20
5.11 bzpack depacker usage conditions . . . . . 20
5.12 zerocomp depacker usage conditions . . . . . 21
5.13 mvcomp depacker usage conditions . . . . . 21
5.14 lzexedat depacker usage conditions . . . . . 21
Source Control Revision ID . . . . . 22

# Section 1: Building the lDOS kernel

To build only the kernel and sharer, the following tools are needed:

- bash shell
- GNU make, or compatible
- perl to process comload.lst listing file
- gcc to compile sumspace tool
- sumspace tool (source text shipped in repo)
- Netwide Assembler, a recent version needed (at least v2.15)
- dosemu2 to run DOS tools
- WarpLink linker (DOS tool, public domain)
- x2b2 tool (DOS tool, public domain)
- comtoexe tool (DOS tool, MIT license)
- lzexe tool (DOS tool, MIT license)

The DOS tools have to be found in the DOS path.

The following components are expected in sister directories to the repo:

- lmacros
- ldosboot
- scanptab
- kernwrap
- patchini (must be built separately using the mak.sh, with gcc-ia16, or provided in binary form)
- instsect (must be built separately using this repo's makinst.sh)
- inicomp (if to use compression support), also needs one of the compression tools:
    - brieflz (blzpack)
    - lz4
    - snappy (snzip)

- exomizer
- x-compressor
- heatshrink
- lzip
- lzop
- lzsa -- default choice, one of the fastest depackers
- apultra
- bzpack
- lzexedat

# Section 2: Starting the lDOS kernel

The lDOS current release build includes four different kernel files in the `bin/` subdirectory: `ldos.com`, `ldos.sys`, `ldospak.com`, and `ldospak.sys`. Any single one of these four files contains the complete kernel.

The `.com` files are called iniload variants, as they are wrapped in the lDOS iniload.asm (initial loader) stage. They are valid DOS MZ application executables and can be loaded as a number of different kernel formats.

The `.sys` files are called drload variants, from the stage named drload.asm (originally created for the lDOS flavour of the Enhanced DR-DOS kernel). They are only valid to be loaded as FreeDOS kernel (typically named `kernel.sys`) or as EDR-DOS kernel (typically named `drbio.sys`). The advantage is that they are smaller than the iniload variants.

The `ldos` files are uncompressed. The advantage is faster load time and being able to patch the kernel image.

The `ldospak` files are compressed. The advantage is, again, the smaller file size.

## 2.1 Booting lDOS

The lDOS kernel can boot off diskette drives (fda, fdb, fdc, fdd) or FAT16 or FAT12 partitions on MBR-partitioned hard disk drives. Both primary and logical partitions are supported.

Loading off logical partitions may require loaders that support this at all and a correct (unit absolute) value for hidden sectors, either hardcoded into the boot sector loader or automatically detected at boot time. ldosmbr's altmbr, ldosboot boot.asm, and lDebug all do support this.

The default drive at boot time may be any drive that matches the boot partition, that is it may be different than A: or C:. If the exact partition cannot be identified from the load unit plus hidden sectors, then the first drive that is logged in on the same unit is selected as the default drive. If no such drive exists, then either drive A: or the first hard disk drive (C:, D:, or E:) is selected, depending on whether the load unit is < 128 or >= 128. The configuration files are searched for in the root directory of the default drive unless the kernel command line specifies pathnames that include a drive specification.

## 2.2 Installing boot sector loader using INSTSECT

The recommended choice is to install a boot sector loader using the instsect application. This application is found both in `instsect.com` and in the COMLOADER multi-executable tool which is embedded in `ldos.com`, using the command program name `instsect`.

The default filename for instsect's loaders is `ldos.com`. Note that the loaders use the lDOS load protocol. Trying to load `ldos.sys` or `ldospak.sys` using these loaders will fail with a check 'V'alue error (letter ‘V’ displayed at boot time).

The instsect online help can be accessed using the `/?` switch. Redirect to a file or to a tool like ‘`more`’ to read all of it.

# Section 3: Supported file systems

lDOS currently supports the following file systems:

- Diskette drives (up to 4)
- MBR-partitioned hard disks
- CHS or LBA access on int 13h units
- Primary partitions
- Logical partitions (in extended partitions of type 5, 0Fh, or 85h)
- FAT12 or FAT16 file systems
- Partition types 0Eh (FAT16 LBA) and 0Fh (Extended LBA) only supported if HDD is accessed using LBA extensions
- Cluster sizes that are a power of two between 1 and 64 sectors
- Cluster size of 128 sectors (MSW NT extension)
- Cluster size of 256 sectors (Enhanced DR-DOS extension)
- Int 13h unit must use a sector size of 512 Bytes.
- Kernel is LFN aware, that is it doesn't mistake LFNs for volume label entries. It doesn't support LFNs.
- FAT32 and FAT+ are not supported. (lDOS iniload can boot off a FAT32 drive but it won't be logged in and supported by the kernel after boot time.)

# Section 4: lDOS configuration

There are four inputs to the lDOS configuration:

- lDOS iniload query patch site, modified using `patldos query`
- lCFG block, modified using `patldos`
- lDOS kernel command line, can be passed using bootable lDebug to chainload lDOS
- Configuration file

## 4.1 Query patch

The query patch site determines whether the initial loader uses CHS or LBA access, and whether it queries the CHS geometry using int 13h function 08h. This is accessed using the `patldos` tool using the `query` keyword. (The `patldos` tool is included in the `ldos.com` COMLOADER. The `patldos` tool is available as an 8086 assembly language DOS program or as a portable C language program.)

The query patch site is not passed along to the lDOS kernel yet.

### 4.1.1 Supported `patldos query` values

```
Usage: patldos [QUERY [ldos.com [DISKETTE|HARDDISK [number]]]]

Supported query patch flags:
 01h: Force CHS
 02h: Force LBA
 04h: No query geometry
 08h: Force single sector reads
 40h: Do not re-check (TestWrit)
 80h: Apply to LDP (lDebug)
```

Default is 0 for both unit types. The flags are detailed more in the lDOS boot manual.

### 4.1.2 Parsing of numbers in `patldos`

Numbers can be specified in three bases:

Default

: Decimal

`0x` prefix or `h` suffix

: Hexadecimal

`0b` prefix or `b` suffix

Binary

A number can be immediately preceded by a plus sign or minus sign. A plus sign indicates the number given is treated as an OR mask (set these bits). Likewise, a minus sign indicates that it should be treated as an inverted AND mask (clear these bits).

There may be multiple numbers specified wherever `patldos` parses a number. They need to be separated by blanks. Only the very first number may lack both a plus and minus sign. If such a number occurs, its value replaces the entire configuration choice. If the first number given does have a plus or minus, some bits of the prior choice read from the kernel file may be retained, depending on the given masks.

## 4.2 lCFG block

The lCFG block currently contains five configuration choices. All of them can be accessed using the `patldos` tool. (The `patldos` tool is included in the `ldos.com` COMLOADER. The `patldos` tool is available as an 8086 assembly language DOS program or as a portable C language program.) All numbers given as the last parameters to `patldos` are parsed as indicated in section 4.1.2.

The following configuration choices are used by the default lDOS build:

First, the inicomp depack progress display choice. This is only included in ldospak, ie the build in which the kernel image is compressed. This allows to choose none, dots, percentage, bar, or bar with percentage. This is accessed using the `patldos` tool using the `progress` keyword. Supported values are displayed by running without parameters after the `progress` keyword.

Second, the debugger detection choice. This indicates conditions for detecting a resident debugger in order to avoid overwriting its int 1 and int 3 handlers. This is accessed using the `patldos` tool using the `debug` keyword. Supported values are displayed by running without parameters after the `debug` keyword.

Third, the verbose flags choice. This specifies what informational messages to display during system init. This is accessed using the `patldos` tool using the `verbose` keyword. Supported values are displayed by running without parameters after the `verbose` keyword.

Fourth, the partition scan mode choice. This specifies several options on which partitions to log in and in what order. This is accessed using the `patldos` tool using the `scanmode` keyword. Supported values are displayed by running without parameters after the `scanmode` keyword.

Fifth, the halt flags choice. This specifies what conditions not to halt on. If a halt flag is set, then the condition causes a warning message but does not halt the system. If a halt flag is clear, the condition displays the message and halts. This is accessed using the `patldos` tool using the `haltflag` keyword. Supported values are displayed by running without parameters after the `haltflag` keyword.

### 4.2.1 Supported `patldos progress` values

```
Usage: patldos [PROGRESS [ldos.com [APP|DEV|BOOT [number]]]]

Supported progress choices:
     0: None
```

```
     1: Dots
     2: Percentage
     3: Bar
     4: Bar with percentage
```

Default is 1.

There is a special flag, with value 16. This is only supported for APP and DEVICE modes. If set, then the selected progress choice is written to stdout even when stdout is redirected to a file. If clear, using a redirected stdout forces the progress display disabled. (As the lDOS kernel inicomp payload only supports boot mode, this flag is never supported by the kernel.)

### 4.2.2 Supported `patldos debug` values

```
Usage: patldos [DEBUG [ldos.com [number]]]

Supported flags:
     1: Check
     2: Assume
     4: Check only valid
     8: Check only IISP
```

Default is 0.

### 4.2.3 Supported `patldos verbose` values

```
Usage: patldos [VERBOSE [ldos.com [number]]]

Supported flags:
     1: Intro line
     2: XMS use
     4: CDS writes
```

Default is 1.

### 4.2.4 Supported `patldos scanmode` values

```
Usage: patldos [SCANMODE [ldos.com [number]]]

Supported flags:
     1|2: Ext5 setting
     4|8: Ext15 setting
    16|32: Ext85h setting
   40h=   64: Ignore subsequent nested
   80h=  128: Ignore subsequent logical
  100h=  256: Ignore subsequent primary
  200h=  512: Traditional order logical
  400h= 1024: Active priority
  800h= 2048: Subsequent primary after
 2000h= 8192: Ignore FAT32
 4000h=16384: Ignore LBA
 8000h=32768: DLA sort disable
Supported ext settings:
```

```
 0 = follow all
 1 = ignore all
 2 = follow first of any ext
 3 = follow first of this type
```

Default is 0.

### 4.2.5 Supported `patldos haltflag` values

```
Usage: patldos [HALTFLAG [ldos.com [number]]]

Supported flags:
     1: Don't halt on TAGCRASH
     2: Don't halt on MCB corrupted
     4: Don't halt on HMCB alloc corrupted
     8: Don't halt on HMCB free corrupted
```

Default is 0Ch.

## 4.3 Kernel command line

The kernel command line is a text string of up to 255 bytes that can be passed to the kernel from the prior loader. Bootable lDebug allows to pass a command line when loading a kernel using the RxDOS.2, RxDOS.3, lDOS, FreeDOS, or EDR-DOS load protocols. (The latter two need to enable this using a `CMDLINE=1` parameter.)

The commands of the command line are separated by semicolons. A semicolon immediately following another semicolon is parsed as an escaped semicolon. The following command names are recognised:

CONFIG

Highest priority configuration file override. This file is tried first. Default is empty.

ALTCONFIG

Second configuration file override. This file is tried second. Default is '`ldos.ini`'.

OLDCONFIG

Third configuration file override. This file is tried last. Default is '`config.sys`'.

PREPEND

Prepend the following line to the configuration file contents.

APPEND

Append the following line to the configuration file contents.

CHECKDEBUGGER

Modify check debugger value from lCFG block. Refer to section 4.2.2 for supported values. Numbers may be prepended by '`+`' plus sign to OR number into the variable, or by '`-`' minus sign to AND the bitwise NOT of the number. Otherwise the number will rewrite the entire variable to the number. Numbers default to decimal format, unless leading '`0x`' or

trailing ‘h’ is specified. Multiple numbers may be specified separated by commas.

VERBOSE

Modify verbose value from lCFG block. Refer to section 4.2.3 for supported values. Numbers may be prepended by ‘+’ plus sign to OR number into the variable, or by ‘-’ minus sign to AND the bitwise NOT of the number. Otherwise the number will rewrite the entire variable to the number. Numbers default to decimal format, unless leading ‘0x’ or trailing ‘h’ is specified. Multiple numbers may be specified separated by commas.

SCANMODE

Modify scan mode value from lCFG block. Refer to section 4.2.4 for supported values. Numbers may be prepended by ‘+’ plus sign to OR number into the variable, or by ‘-’ minus sign to AND the bitwise NOT of the number. Otherwise the number will rewrite the entire variable to the number. Numbers default to decimal format, unless leading ‘0x’ or trailing ‘h’ is specified. Multiple numbers may be specified separated by commas.

HALTFLAG

Modify halt flag value from lCFG block. Refer to section 4.2.5 for supported values. Numbers may be prepended by ‘+’ plus sign to OR number into the variable, or by ‘-’ minus sign to AND the bitwise NOT of the number. Otherwise the number will rewrite the entire variable to the number. Numbers default to decimal format, unless leading ‘0x’ or trailing ‘h’ is specified. Multiple numbers may be specified separated by commas.

‘.’ (dot)

Command ignored, possibly used for future optional extensions.

## 4.4 Configuration file

If a configuration file is found, its length is checked. The length should not exceed 56 KiB. Then a buffer is allocated and the entire file is read. Too long file contents may be truncated. The command line PREPEND and APPEND lines are also stored in the buffer.

Aside all the usual MS-DOS v4 level configuration directives, some additional directives are supported:

DEVICEHIGH

Like DEVICE but attempt to load into the UMA

INSTALLHIGH

Like INSTALL but attempt to load into the UMA

INSTALLLAST, INSTALLLASTHIGH

Like INSTALL, INSTALLHIGH but after final DOSDATA and DOSCODE relocation

INSTALLMID, INSTALLMIDHIGH

Like INSTALL, INSTALLHIGH but before final DOSDATA relocation, after any DEVICE directive has been processed. Same as default INSTALL (absent COMPAT settings to change INSTALL order).

INSTALLFIRST, INSTALLFIRSTHIGH

Like INSTALL, INSTALLHIGH but during the pass that processes DEVICE directives.

SHELLHIGH

Like SHELL but attempt to load into the UMA

COMPAT

List comma-separated compatibility options. The following options are recognised:

(NO)ALTAH

Control allocation of a compatible ALTAH (CON device driver look-ahead buffer) that should be found at the segmented address 70h:61Bh.

(NO)DOSDATAFIRST

Control relocation of DOSDATA segment first, before loading any device drivers. This means the segment will be allocated in the Low Memory Area.

(NO)DOSDATAEARLY

Control relocation of DOSDATA segment early, after any device driver. The segment is relocated after the first device driver load that provides DOS with a sufficiently large (ca 6 KiB) Upper Memory Block. If no such UMB is provided at all then the DOSDATA segment is eventually relocated into the Low Memory Area at the default end relocate time.

(NO)BOUNCEHMACHS

Control bouncing CHS disk accesses involving the HMA through the Low Memory Area (DOSENTRY) disk sector buffer. This slows down the access but may be required if the access won't work in the HMA.

(NO)BOUNCEHMALBA

Control bouncing LBA disk accesses involving the HMA through the Low Memory Area (DOSENTRY) disk sector buffer. This slows down the access but may be required if the access won't work in the HMA. (Appears to be needed for qemu.)

AUTOBOUNCEHMACHS

Enable automatic determination of bouncing CHS disk access involving the HMA. This is done by reading all DOS-internal CHS accessed nonremovable drives' boot sectors to the HMA shortly after it is allocated, but before DOSCODE is installed into it. If any of them are written to an A20-wrapped address into the disk sector in the DOSENTRY section which is in the Low Memory Area, bouncing is enabled for all CHS accesses. A message indicating the first drive that needs this is then displayed. Specifying either BOUNCEHMACHS or NOBOUNCEHMACHS disables this automatic detection.

AUTOBOUNCEHMALBA

As above but for LBA accessed nonremovable drives.

(NO)BOUNCEUMACHS

Control bouncing CHS disk accesses involving the UMA, or HMA. If enabled, all accesses to the UMA (typically starting at segment A000h) and also the HMA will be bounced through the LMA (DOSENTRY) disk sector buffer. If disabled, UMA accesses are not bounced but HMA accesses still may be.

(NO)BOUNCEANYCHS

If enabled, bounce any CHS disk access through the DOSENTRY disk sector buffer. If disabled, LMA accesses are not bounced but UMA and HMA accesses still may be.

(NO)BOUNCEUMALBA

(NO)BOUNCEANYLBA

As above but for LBA accessed drives.

(NO)HIDEDOSENTRY

Controls hiding of the first MCB, which is for the DOSENTRY segment. This MCB typically is located at segment 60h. Some programs such as FreeDOS MEM with its /D switch may not expect the entry segment and its device driver headers to appear in an MCB. If this is enabled, the first MCB in the DOS list of lists is changed to point past the DOSENTRY MCB.

(NO)HIDEDOSDATAFIRST

Controls hiding of the DOSDATA MCB. This only takes effect if DOSDATAFIRST is enabled. The DOS list of lists first MCB field is changed to point past the final DOSDATA MCB after the relocation, then. This implies that the DOSENTRY MCB is hidden as well. The ALTAH MCB may also be hidden due to this, if the ALTAH option is enabled.

(NO)COLLECTALLMCBS

Controls coalescing of all free MCBs during memory allocation. Default is enabled. When disabled, only free candidate MCBs are grown within the Allocate MCB main loop, avoiding the separate loop.

(NO)COLLECTFREEMCB

Controls coalescing of free MCBs behind the freed MCB during memory deallocation. Default is disabled. EDR-DOS is known to act like COLLECTFREEMCB.

(NO)INSTALLASLAST

If enabled, INSTALL acts like INSTALLLAST. Default is disabled (so that INSTALL acts like INSTALLFIRST or INSTALLMID).

(NO)INSTALLASFIRST

If enabled, and INSTALLASLAST is not enabled, INSTALL acts like INSTALLFIRST. Default is disabled (so that INSTALL acts like INSTALLLAST or

INSTALLMID).

# Section 5: Additional usage conditions

The kernel executables can be compressed with a choice of different compressors. (This is done in the inicomp stage.) The files then contain a decompression stub. Some of these stubs have their own usage conditions. The following stub usage conditions apply, if one of these stubs is used.

## 5.1 BriefLZ depacker usage conditions

BriefLZ - small fast Lempel-Ziv

8086 Assembly lDOS iniload payload BriefLZ depacker

Based on: BriefLZ C safe depacker

Copyright (c) 2002-2016 Joergen Ibsen

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

## 5.2 LZ4 depacker usage conditions

8086 Assembly lDOS iniload payload LZ4 depacker

by E. C. Masloch, 2018

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.3 Snappy depacker usage conditions

8086 Assembly lDOS iniload payload Snappy depacker

by E. C. Masloch, 2018

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.4 Exomizer depacker usage conditions

8086 Assembly lDOS iniload payload exomizer raw depacker

by E. C. Masloch, 2020

Copyright (c) 2005-2017 Magnus Lind.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented * you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any distribution.

## 5.5 X compressor depacker usage conditions

MIT License

Copyright (c) 2020 David Barina

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 5.6 Heatshrink depacker usage conditions

8086 Assembly lDOS iniload payload heatshrink depacker

by E. C. Masloch, 2020

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.7 Lzd usage conditions

Lzd - Educational decompressor for the lzip format

Copyright (C) 2013-2019 Antonio Diaz Diaz.

This program is free software. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## 5.8 LZO depacker usage conditions

8086 Assembly lDOS iniload payload LZO depacker

by E. C. Masloch, 2020

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.9 LZSA2 depacker usage conditions

8086 Assembly lDOS iniload payload LZSA2 depacker

by E. C. Masloch, 2021

based on:

decompress_small.S - space-efficient decompressor implementation for 8088

Copyright (C) 2019 Emmanuel Marty

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

## 5.10 aPLib depacker usage conditions

8086 Assembly lDOS iniload payload aPLib depacker

by E. C. Masloch, 2021

based on:

aplib_8088_small.S - size-optimized aPLib decompressor for 8088 - 145 bytes

Copyright (C) 2019 Emmanuel Marty

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

## 5.11 bzpack depacker usage conditions

8086 Assembly lDOS iniload payload bzpack depacker

by E. C. Masloch, 2021

BSD 2-Clause License

Copyright (c) 2021, Milos Bazelides

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 5.12 zerocomp depacker usage conditions

8086 Assembly lDOS iniload payload zerocomp depacker

by E. C. Masloch, 2024

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.13 mvcomp depacker usage conditions

8086 Assembly lDOS iniload payload mvcomp depacker

by E. C. Masloch, 2024

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

## 5.14 lzexedat depacker usage conditions

8086 Assembly lDOS iniload payload LZEXE depacker

by E. C. Masloch, 2020--2025

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

# Source Control Revision ID

hg c661c6316db2, from commit on at 2026-11-06 22:06:08 +0100

If this is in ecm's repository, you can find it at https://hg.pushbx.org/ecm/msdos4/rev/c661c6316db2