ecm's webbed site

This document has been compiled on 2024-10-29.

Contents

Section 1: News

2024-03-08 Mar Fri

    lDebug release 8

2024-02-16 Feb Fri

    lDebug release 7

2023-08-26 Aug Sat

    lDebug release 6

2023-08-20 Aug Sun

    Added Enhanced DR-DOS section and current release builds, as well as a repo.

    Added WarpLink current release build and repo, ported to NASM + WarpLink toolchain.

2023-03-08 Mar Wed

    lDebug release 5

2022-12-23 Dec Fri

    bootimg now allows to cut out boot loader parts from a given file, as well as fragmenting files preloaded into the image

2022-08-06 Aug Sat

    Added lCDebug build to the current releases that are created automatically. The new conditionally debuggable build option has also been covered by a blog post.

2022-07-20 Jul Wed

    Added Eric Auer's Terminal and the Insight debugger to the current releases that are created automatically.

2022-07-12 Jul Tue

    Updated the NASM revision that's used for building current releases. Now line numbers of included files are correct in listings as well as in the assembler's messages.

2022-07-10 Jul Sun

    lDebug symbolic branch merged into default branch, made a build option that is default off. (As announced on the blog.)

2022-04-24 Apr Sun

The new pushbx blog is officially online

2022-03-08 Mar Tue

lDebug release 4

2021-11-04 Nov Thu

lDebug current release of DDebug added

2021-11-02 Nov Tue

Other builds added (FreeDOS kernel and its SHARE for now)

2021-11-01 Nov Mon

KEEPHOOK got a fix to a bug that made for a few broken releases

2021-10-30 Oct Sat

RENUMBER added

AMITSRS listed

2021-10-26 Oct Tue

FDAPM added

2021-10-24 Oct Sun

TSRs changed to use Update IISP Header function as now provided by lDebug's AMIS
interface

lDebug current release file added

2021-10-19 Oct Tue

SHUFHOOK first release added

2021-10-13 Oct Wed

RxANSI listed

lClock listed

SEEKEXT listed

TSR example listed

TSR current releases added

2021-10-12 Oct Tue

KEEPHOOK first release added

2021-08-15 Aug Sun

lDebug release 3

2021-05-17 May Mon

    Moved from ulukai.org to pushbx.org domain

2021-05-05 May Wed

    lDebug release 2

2021-04-07 Apr Wed

    The lDebug manual had a News chapter added

2021-02-18 Feb Thu

    What is "pushbx"?

2021-02-15 Feb Mon

    lDebug release 1

Section 2: Projects

2.1 lDebug debugger

lDebug is a 86-DOS debugger based on the MS-DOS Debug clone FreeDOS Debug. It features DPMI client support for 32-bit and 16-bit segments, a 686-level assembler and disassembler, an expression evaluator, an InDOS and a bootloaded mode, script file reading, serial port I/O, permanent breakpoints, conditional tracing, buffered tracing, auto-repetition of some commands, and a number of extensions. There is also a symbolic debugging branch being developed.

The manual (text, PDF) is available online. The repo is hosted at https://hg.pushbx.org/ecm/ldebug/ and proper releases are available at https://pushbx.org/ecm/download/ldebug/ A current release is available at https://pushbx.org/ecm/download/ldebug.zip and a current release of the debuggable debugger is available at https://pushbx.org/ecm/download/lddebug.zip (You only need this one if you want to debug the debugger itself.) Further, a current release of the conditionally debuggable debugger is available at https://pushbx.org/ecm/download/lcdebug.zip (This one can switch debuggable mode off and on.)

2.2 TracList - Trace debugging output in listing files

A companion application to lDebug, connecting to the debugger's serial I/O. It traces execution through NASM format listing files. The manual (text, PDF) is available online. The repo is hosted at https://hg.pushbx.org/ecm/tractest/

2.3 RxDOS 7.2x kernel

RxDOS is an alternative DOS kernel. It is not yet ready for general use. (I am not working on RxDOS any longer.) Unlike the FreeDOS kernel, its source is entirely in assembly language. (As of RxDOS 7.20N the source has been ported from MASM to NASM.) Unlike MS-DOS and PC-DOS versions past v4.01 it is (copylefted) free software. It aims to support extended features such as FAT32, LFNs, and LBA. It already supports UMBs (for DOSDATA, DOSCODE) and the HMA (for DOSCODE). Support for additional extensions is planned, such as FAT+, cluster sizes exceeding 32 KiB, sector sizes up to 8 KiB, and moving many DOS data structures into the UMA and HMA. The repo is hosted at https://hg.pushbx.org/ecm/rxdos-7.2x/

2.4 Enhanced DR-DOS kernel and shell

Enhanced DR-DOS is the effort of The DR-DOS/OpenDOS Enhancement Project to update the kernel and shell of the 1997 Caldera OpenDOS 7.01 source release. The original, restrictive OpenDOS license has been superseded by the 2022-07-07 license agreement by DRDOS, Inc. which allows to use, distribute, and modify 'CP/M and its derivatives'. (I consider DR-DOS a derivative of CP/M. As it happens, the copyright notices of DR-DOS stretch back to 1976.)

I based my own revisions on the last, 2011 release of EDR-DOS, mainly modifying parts of the kernel. The kernel of my EDR-DOS repo (lDOS flavour) now builds with a toolchain consisting of only NASM, WarpLink, x2b2, convlist.pl, and some tools (now all FLOSS with sources) shipping with the repo. Compressing the lDOS flavour kernel will make use of additional compression tools that correspond to the inicomp compression methods. Building the shell requires OpenWatcom and JWasm as well.

The EDR-DOS kernel, alike RxDOS and unlike FreeDOS, is written entirely in x86 assembly language. (The command.com shell utilises the C language in parts.) It fully supports FAT32 file systems and LBA int 13h disk access. Further, it supports FAT+ for file sizes up to 256 GiB minus 1 Byte on local FAT FS file systems. Consequently, a new function 7142h and an extension to the SFT entry layout are added to handle file seek offsets in terms of 64-bit offsets. Some LFN functions known from MSWindows 4 and doslfn are also supported in order to report FAT+ extended file sizes, though they only operate on SFNs. EDR-DOS introduced the file system choice of using 256 sectors per cluster, for 128 KiB per cluster at the normal sector size, now also supported by the FreeDOS kernel. Furthermore, the DR-DOS kernel is a highly compatible 8086 DOS beyond what is offered by FreeDOS.

The repo is hosted at https://hg.pushbx.org/ecm/edrdos/ and a current release is available at https://pushbx.org/ecm/download/edrdos.zip The original files released by The DR-DOS/OpenDOS Enhancement Project can be found in the edrdos subdirectory at https://pushbx.org/ecm/download/edrdos/

2.5 lDOS boot loaders

Advanced boot loading stages used by kernels or other bootloaded programs. The repos are hosted at https://hg.pushbx.org/ecm/ldosboot/ and https://hg.pushbx.org/ecm/ldosmbr/

A file documenting the lDOS boot load protocols is available online here.

2.6 86 Mode NASM macro collection

A macro collection featuring parts such as defaultable defines for conditional assembly, stack frame calculation, anonymous @@ labels, safer sectioning directives, and assisted flag bit accesses. The repo is hosted at https://hg.pushbx.org/ecm/lmacros/

2.7 NASM 2.05 based x86 Instruction Reference

An assembly language reference covering all general instructions up to a 686-level instruction set. Includes encoding schemes and some general English-language descriptions. It is available online here (text, PDF). The repo is hosted at https://hg.pushbx.org/ecm/insref/

2.8 ACEGALS - Assembly Comments Explained: Guide for Advanced Learning and Style

A reference guide for our assembly comments and formatting styles. It is available online here (text, PDF). The repo is hosted at https://hg.pushbx.org/ecm/acegals/

2.9 Boot image FAT FS formatting script

A NASM script that allows formatting an image file with FAT12, FAT16, or FAT32. The filesystem can be preloaded with files. Both diskette or harddisk images can be created. Optionally an MBR and a dosemu image header can be wrapped around the image. For FAT32 a

backup boot sector copy and an FSINFO sector can be initialised. Sector sizes between 32 Bytes and 8 KiB are supported. Likewise cluster sizes between 1 to 256 sectors. (Minimum cluster size is 32 Bytes, maximum is 2 MiB.) The repo is hosted at https://hg.pushbx.org/ecm/bootimg/

2.10 Terminate and Stay Resident (TSR) programs

The current releases are created automatically from the repos and are updated daily. Old files of the current releases are found at https://pushbx.org/ecm/download/old/

There is a manual (text, PDF) available online that lists the common switches supported by most of these TSR programs.

2.10.1 KEEPHOOK utility

A TSR which aids the uninstalling of other TSRs. The manual (text, PDF) is available online. The repo is hosted at https://hg.pushbx.org/ecm/keephook/ and a current release is available at https://pushbx.org/ecm/download/keephook.zip

2.10.2 RxANSI - ANSI and LESS sequence parser

RxANSI is a TSR to take over interrupt 29h and parse ANSI and LESS sequences. It only affects output - it does not do any input. The repo is hosted at https://hg.pushbx.org/ecm/rxansi/ and some releases are available at https://pushbx.org/ecm/download/ and a current release is available at https://pushbx.org/ecm/download/rxansi.zip

2.10.3 lClock - Clock display TSR

lClock is a TSR that continuously displays a clock in the upper right-side corner of the screen by writing to the text mode video buffer. It can be customised using build time options. The repo is hosted at https://hg.pushbx.org/ecm/lclock/ and a current release is available at https://pushbx.org/ecm/download/lclock.zip

2.10.4 SEEKEXT - Seek service extension

This TSR extends DOS with an interrupt 21h handler. It hooks the services 21.42 (short seek), 21.7142 (long seek), as well as 21.71A6 (file info). The short seek hook is to provide sensible results from positive or negative 32-bit seek offsets for file handles that implement long seek (21.7142). The long seek and file info hooks are to redirect calls to the special int 2F.1142, 2F.11C2, or 2F.11A6 redirector interfaces, which are used by recent dosemu2 to provide 64-bit seeking and file info on its redirected file systems. The repo is hosted at https://hg.pushbx.org/ecm/seekext/ and a current release is available at https://pushbx.org/ecm/download/seekext.zip

2.10.5 FDAPM - A tool for APM power management and energy saving

The power savings utility for FreeDOS, originally by Eric Auer. Extended to use IISP headers, an AMIS multiplexer, and to provide the UNLOAD command, with the advanced deinstallation method. The repo is hosted at https://hg.pushbx.org/ecm/fdapm/ and a current release is available at https://pushbx.org/ecm/download/fdapm.zip

2.10.6 TSR example

This project is an example of a TSR. It was originally based on RxANSI, and was in turn the base for lClock, SEEKEXT, and KEEPHOOK. It implements an AMIS multiplexer and

features optimal installation and the advanced deinstallation method. The repo is hosted at https://hg.pushbx.org/ecm/tsr/ (No releases are going to be provided because this is useful only for developers.)

2.10.7 Tools to work with TSR programs

2.10.7.1 SHUFHOOK utility

This tool allows to re-order interrupt handlers. The manual (text, PDF) is available online. The repo is hosted at https://hg.pushbx.org/ecm/shufhook/ and a current release is available at https://pushbx.org/ecm/download/shufhook.zip

2.10.7.2 RENUMBER

This tool allows to change the AMIS multiplex numbers of resident multiplexers. The repo is hosted at https://hg.pushbx.org/ecm/renumber/ and a current release is available at https://pushbx.org/ecm/download/renumber.zip

2.10.7.3 AMITSRS - List resident AMIS multiplexers

This tool lists AMIS multiplexers that are resident. It is based on the tool of the same name that was released to the Public Domain by Ralf Brown in 1992. The verbose mode has been extended to list the multiplex number and interrupt list of each resident multiplexer. It has also been updated to compile with the ia16 gcc and libi86. The repo is hosted at https://hg.pushbx.org/ecm/amitsrs/ and a current release is available at https://pushbx.org/ecm/download/amitsrs.zip

The AMITSRS tool has been ported to run as an Extension for lDebug (ELD) that can be loaded from the debugger's command line. It has been extended a little to allow specifying different keywords on the ELD's command line. This update wasn't backported to the original tool, which treats any arguments as indicating "VERBOSE" output.

2.11 Other builds

The current release scripts got updated to build a few projects that are kept in git repos. These are https://pushbx.org/ecm/download/fdkernel.zip (from repo at https://github.com/fdos/kernel.git) and https://pushbx.org/ecm/download/fdshare.zip (from repo at https://github.com/fdos/share.git) and https://pushbx.org/ecm/download/freecom.zip (from repo at https://github.com/fdos/freecom.git) (While these contain some contributions by me, I do not host current repos of them anywhere.)

Note: fdkernel builds after the 2021-11-23 release and before the 2022-04-06 release did not build correctly.

Section 3: Links

ecm's files

    https://pushbx.org/ecm/

ecm's hgweb

    https://hg.pushbx.org/ecm/

ecm's github

    https://github.com/ecm-pushbx/

ecm's stackoverflow

    https://stackoverflow.com/users/738287/ecm?tab=profile

ecm's NASM bug reports

    https://bugzilla.nasm.us/buglist.cgi?f1=reporter&list_id=1912&o1=equals&order=changeddate%20DES

ecm's twitter

    https://twitter.com/pushbx/

BTTR Software

    https://www.bttr-software.de/

DOS ain't dead

    https://www.bttr-software.de/forum/

3.1 The pushbx blog

Longer form comments or explanations can now go to the pushbx blog which runs on a dokuwiki instance on this server. It is expected that content that would be shared privately prior to this may land on there. That may include descriptions that do not fit into commit messages or are written later on with more hindsight. As of 2022 April this was experimental and may be used more or less frequently. As of 2024 October I regularly post to the blog, usually once a week. Comments are generally enabled on individual blog pages. Posting a comment requires signing up for a gratis account with the dokuwiki that runs the blog. All comments are screened manually.

Section 4: What is "pushbx"?

The username 'pushbx' originates in Eric Auer's callver (public domain). A line there reads the instruction 'push bx' but has a comment that *also* reads 'push bx'. The reason? There's a line further down which has the instruction 'pop ax' with the comment 'but pop ax'. This hints at the fact that, to a beginner, it is unusual to push then pop with two different registers as the source and destination. I've been using pushbx as a username ever since at the latest 2012 March. **NEW:** The little used twitter account indicates that I have used the username pushbx as early as 2009-11-27 in the first tweet.

Source format of this page

This page has been created using the libre Halibut documentation preparation system.

Source Control Revision ID

hg ddbcb6ac5602, from commit on at 2024-10-29 19:08:08 +0100

If this is in ecm's repository, you can find it at https://hg.pushbx.org/ecm/webecm/rev/ddbcb6ac5602