ShredOS x86_64

For all Intel and compatible 64 bit processors

For the 32 bit version of ShredOS that will run on both 32bit and 64bit processors, see ShredOS i686

Total downloads x86_64 all releases 14k Total downloads i686 all releases 2.2k

Download .img and .iso files for burning to USB flash drives and CD-R/DVD-R

ShredOS Version	Nwipe Version	Number of Downloads	.img for USB Flash	.iso for CD-R/DVD-R
Latest x86_64 64 bit version	v0.31	downloads@latest 5.1k	.img 64bit	.iso 64bit legacy boot
Latest i686 32 bit version	v0.30.001	downloads@latest 955	.img 32bit	Not available yet

bios/UEFI version of the .iso is in development and will be released shortly.

Demo video showing ShredOS having booted straight into Nwipe where you can then select one or more drives to be erased.

What is ShredOS?

ShredOS is a USB bootable (BIOS or UEFI) small linux distribution with the sole purpose of securely erasing the entire contents of your disks using the program nwipe. If you are familiar with dwipe from DBAN then you will feel right at home with ShredOS and nwipe. What are the advantages of nwipe over dwipe/DBAN? Well as everybody probably knows, DBAN development stopped in 2015 which means it has not received any further bug fixes or support for new hardware since that date. Nwipe originally was a fork of dwipe but has continued to have improvements and bug fixes and is now available in many Linux distros. ShredOS hopefully will always provide the latest nwipe on a up to date Linux kernel so it will support modern hardware.

ShredOS supports either 32bit or 64bit processors. You will need to download the appropriate 64bit or 32bit .img or .iso file, depending upon your target processor and whether you want to burn ShredOS to a USB memory stick, in which case you would download the .img file. Alternatively, if you wanted to burn ShredOS to CD/DVD, then you would download the .iso file.

Shredos includes the latest Nwipe official release, but in addition includes other disc related utilities such as Smartmontools, hdparm and a hexeditor hexedit. Nwipe automatically starts it's GUI in the first virtual terminal (ALT-F1), hdparm, smartmontools and hexeditor can be run in the second virtual terminal, (ALT-F2). Nwipe will erase drives using a user selectable choice of seven methods. hdparm, amongst many of its options can be used for wiping a drive by using the drives internal firmware. The program loadkeys can be used for setting the keyboard type. i.e. loadkeys uk, loadkeys fr etc.

ShredOS boots very quickly and depending upon the host system can boot in as little as 2 seconds (typically 4 to 6 seconds) on modern hardware, while on an old Pentium4 may take 40+ seconds. Nwipe automatically starts in GUI mode and will list the disks present on the host system. In fact, Nwipe can launch so fast that the USB devices have not yet initialised so the first time nwipe appears it may not show any USB drives. If you then use Control-C to exit and restart nwipe, you should now see any attached USB devices. You can then select the methods by which you want to securely erase the disk/s. Nwipe is able to simultanuosly wipe multiple disks using a threaded software architecture. I have simultaneously wiped 28 loop devices in tests and know of instances where it's been used to wipe upwards of 10 drives on a system.

The vanilla version of ShredOS boots into nwipe's GUI and shows the available discs that can then be selected for wiping. It does not autonuke your discs at launch, however it is capable of doing that, if you edit the grub.cfg file and specify the appropriate nwipe command line option. Details of configuring nwipe's launch behaviour is shown below How to run nwipe so you can specify nwipe command line options

What do I do after I've erased everything on my disk? What is actually erased?

This paragraph is for those that are not familiar with wiping discs. if you know what you are doing skip to the next section. So you have erased your disc with ShredOS/nwipe and nwipe reported zero errors and the disc was erased. In it's erased state and depending upon the method you used every block on the drive contains either zero's or meaningless random data. In this state the disc won't be recognised by your operating system except at a very low level or by specialised programs. You won't be able to write files to the disc because nwipe has removed everything, absolutely everything, the operating system is gone, all your data is gone, the partition table is gone, the file system gone, the MBR and all the files have been erased without a trace and will never ever be recovered from the disk. The only thing left is a whole load of zeros or random data. To make the disc usable again you will either need to format the disk, which creates a partition table and directory structure or install a new operating system such as Linux or Windows. Of course, if you are just disposing of or reselling the disk then you don't need to do anything else. So if you are reasonably happy that you know what you are doing and you understand that you will need to format the disc then I hope this software does it's job and is useful to you. Before you press that 'S' key to start the wipe, pause and double check you have selected the correct drive/s, something I always do !

Nwipe's erasure methods

For an upto date list of supported wipe methods see the nwipe page.

- Quick erase - Fills the device with zeros, one round only.
- RCMP TSSIT OPS-II - Royal Candian Mounted Police Technical Security Standard, OPS-II
- DoD Short - The American Department of Defense 5220.22-M short 3 pass wipe. 1,2,& 7.
- DoD 5220.22M - The American Department of Defense 5220.22-M full 7 pass wipe. 1-7
- Gutmann Wipe - Peter Gutmann's method. (Secure Deletion of Data from Magnetic and Solid-State Memory)
- PRNG Stream - Fills the device with a stream from the PRNG.
- Verify only - This method only reads the device and checks that it is all zero.
- HMG IS5 enhanced - Secure Sanitisation of Protectively Marked Information or Sensitive Information

Nwipe also includes the following pseudo random number generators:

- Mersenne Twister (mt19937ar-cok)
- ISAAC (rand.c 20010626)

Obtaining and writing shredos to a USB flash drive, the easy way !

You can of course compile shredos from source but that can take a long time and you can run into all sorts of problems if your not familiar with compiling an operating system. So if you just want to get started with using shredos and nwipe then just download the shredos image file and write it to a USB flash drive. Please note this will over write the existing contents of your USB flash drive.

Download shredos for 64 bit processors from here

Download shredos for 32 bit processors (also runs on 64 bit processors) from here

Linux (and MAC) users

Check it's not corrupt by running the following command and comparing with the checksum shown in the release notes:

```
$ sha1sum shredos.img.tar.gz (shasum instead of sha1sum if your using a MAC)
(example) sha1 db37ea8526a17898b0fb34a2ec4d254744ef08a1 shredos.img.tar.gz
```

If the image file has a .img.tar.gz extension then use the following commands to extract the .img file. If the file extension simply ends with .img and there is no tar.gz then skip this step.

```
$ gunzip shredos.img.tar.gz
$ tar xvf shredos.img.tar
```

If you are using linux or a MAC write the shredos.img file (also sometimes called shredos-2020MMDD.img i.e. shredos-20200418.img etc) to your USB flash drive using the following command. (/dev/sdx is the device name of your USB drive, this can be obtained from the results of sudo fdisk -l)

```
dd if=shredos.img of=/dev/sdx
```

Windows users:

If you are a windows user, use a program such as Rufus or etcher to write the image file to a USB stick, remembering that the entire contents of the USB flash drive will be overwritten. Winzip be used to extract the shredos.img file from the compressed shredos.img.tar.gz file that you downloaded. hashtab can be downloaded and used to confirm the sha1 checksum.

Virtual Terminals

ShredOS has three tty terminals, ALT-F1 (Where nwipe is initially launched), ALT-F2 (A virtual terminal), ALT-F3 (console log, login required which is root with no password).

How to run nwipe so you can specify nwipe command line options

The version of nwipe that runs in the default terminal will automatically restart when you exit it, either at the end of a wipe or using CONTROL-C to abort. So if you want to run nwipe in the traditional way, along with any command line options you require, then use the second terminal ALT-F2, as an example, you could then use the command `nwipe --nousb --logfile=nwipe.log` etc. If you do use ALT-F2 to run a second copy of nwipe, please remember that if you already have one copy of nwipe wiping, the second copy of nwipe will hang on starting. Therefore nwipe in the default terminal should be left at the drive selection screen to prevent the second occurence of nwipe from hanging. Alternatively, a second occurrence of nwipe could be started by specifying the drive on the command line as long as that drive is not already being wiped by the first instance of nwipe, i.e. `nwipe /dev/sdc` etc.

How to change the default nwipe options so the change persists between reboots

To change the default settings of nwipe you will need to place the nwipe options required on the kernel command line in /boot/grub/grub.cfg and /EFI/BOOT/grub.cfg

Example of default grub.cfg

```
set default="0"
set timeout="0"

menuentry "shredos" {
        linux /boot/shredos console=tty3 loglevel=3
}
```

Adding nwipe_options="..." to grub.cfg to make the default nwipe start up with zero method, no verification, no blanking, ignore USB devices and automatically power off the computer at the end of the wipe.

```
set default="0"
set timeout="0"

menuentry "shredos" {
        linux /boot/shredos console=tty3 loglevel=3 nwipe_options="--
method=zero --verify=off --noblank --nousb --autopoweroff"
}
```

You are not only limited to nwipe options, you can also specify devices along with those options. As would be the case when using nwipe from the command line, the devices to be wiped come after the options, as shown in the example below.

```
set default="0"
set timeout="0"

menuentry "shredos" {
        linux /boot/shredos console=tty3 loglevel=3 nwipe_options="--
method=zero --verify=off --noblank --nousb --autopoweroff /dev/sdd /dev/sde"
}
```

For reference and as of nwipe version 0.30, listed below are all the options that you can use with nwipe and can place on the kernel command line in grub.cfg as described in the examples above.

```
Usage: nwipe [options] [device1] [device2] ...
Options:
  -V, --version           Prints the version number

  -v, --verbose           Prints more messages to the log

  -h, --help              Prints this help

      --autonuke          If no devices have been specified on the command
line,
                          starts wiping all devices immediately. If devices
have
                          been specified, starts wiping only those specified
                          devices immediately.

      --autopoweroff      Power off system on completion of wipe delayed for
                          for one minute. During this one minute delay you can
                          abort the shutdown by typing sudo shutdown -c

      --sync=NUM          Will perform a sync after NUM writes (default:
100000)
                          0 - fdatasync after the disk is completely written
                              fdatasync errors not detected until completion.
                              0 is not recommended as disk errors may cause
nwipe
                              to appear to hang
                          1 - fdatasync after every write
                              Warning: Lower values will reduce wipe speeds.
                          1000000 - fdatasync after 1000000 writes etc.)

      --verify=TYPE       Whether to perform verification of erasure
                          (default: last)
                          off   - Do not verify
                          last  - Verify after the last pass
                          all   - Verify every pass

  -m, --method=METHOD     The wiping method. See man page for more details.
                          (default: dodshort)
                          dod522022m / dod       - 7 pass DOD 5220.22-M method
                          dodshort / dod3pass    - 3 pass DOD method
                          gutmann                - Peter Gutmann's Algorithm
                          ops2                   - RCMP TSSIT OPS-II
                          random / prng / stream - PRNG Stream
                          zero / quick           - Overwrite with zeros
                          verify                 - Verifies disk is zero
filled

  -l, --logfile=FILE      Filename to log to. Default is STDOUT

  -p, --prng=METHOD       PRNG option (mersenne|twister|isaac)

  -r, --rounds=NUM        Number of times to wipe the device using the
selected
                          method (default: 1)

      --noblank           Do NOT blank disk after wipe
                          (default is to complete a final blank pass)
      --nowait            Do NOT wait for a key before exiting
                          (default is to wait)

      --nosignals         Do NOT allow signals to interrupt a wipe
                          (default is to allow)

      --nogui             Do NOT show the GUI interface. Automatically invokes
                          the nowait option. Must be used with the --autonuke
                          option. Send SIGUSR1 to log current stats

      --nousb             Do NOT show or wipe any USB devices whether in GUI
                          mode, --nogui or --autonuke modes.

  -e, --exclude=DEVICES   Up to ten comma separated devices to be excluded
                          --exclude=/dev/sdc
                          --exclude=/dev/sdc,/dev/sdd
                          --exclude=/dev/sdc,/dev/sdd,/dev/mapper/cryptswap1
```

How to set the keyboard map using the loadkeys command (see here for persistent change between reboots)

You can set the type of keyboard that you are using by typing, `loadkeys uk`, `loadkeys us`, `loadkeys fr`, `loadkeys cf`, `loadkeys by`, `loadkeys cf`, `loadkeys cz` etc. See /usr/share/keymaps/i386/ for full list of keymaps. However you will need to add an entry to `loadkeys=uk` etc to grub.cfg for a persistent change between reboots.

Examples are: (azerty:) azerty, be-latin1, fr-latin1, fr-latin9, fr-pc, fr, wangbe, wangbe2

(bepo:) fr-bepo-latin9, fr-bepo

(carpalx:) carpalx-full, carpalx

(colemak:) en-latin9

(dvorak:) ANSI-dvorak, dvorak-ca-fr, dvorak-es, dvorak-fr, dvorak-l, dvorak-la, dvorak-programmer, dvorak-r, dvorak-ru, dvorak-sv-a1, dvorak-sv-a5, dvorak-uk, dvorak, no

(fgGIod:) tr_f-latin5, trf

(include:) applkey, backspace, ctrl, euro, euro1, euro2, keypad, unicode, windowkeys

(olpc:) es, pt

(qwerty:) bashkir, bg-cp1251, bg-cp855, bg_bds-cp1251, bg_bds-utf8, bg_pho-cp1251, ... by, cf, cz, dk, es, et, fi, gr, il, it, jp106, kazakh, la-latin1, lt, lv, mk, nl, nl2, no, pc110, pl, ro, ru, sk-qwerty, sr-cy, sv-latin1, ua, uk, us (for the full list see /usr/share/keymaps/i386/qwerty)

- hdparm is also available for those that want to do a firmware supported wipe. A firmware wipe is a planned enhancement to nwipe.

How to make a persistent change to keyboard maps

The default grub.cfg looks like this

```
set default="0"
set timeout="0"

menuentry "shredos" {
        linux /boot/shredos console=tty3 loglevel=3
}
```

Add the following options to the kernel command line, i.e. `loadkeys=uk`, `loadkeys=fr` etc

```
set default="0"
set timeout="0"

menuentry "shredos" {
        linux /boot/shredos console=tty3 loglevel=3 loadkeys=uk
}
```

Reading and saving nwipes log files

The nwipe that is automatically launched in the first virtual terminal ALT-F1, creates a log file that contains the details of the wipe/s and a summary table that shows successfull erasure or failure. The file is time stamped within it's name. A new timestamped log file is created each time nwipe is started. The files can be found in the / directory. A example being nwipe_log_20200418-084910.txt. As currently, shredos does not have persistent storage, if you want to keep these files between reboots of shredos, you will need to manually copy them to the USB stick as follows:

1. Locate the device name of your USB stick from it's model & size.

For Linux: If the | character isn't displayed properly use loadkeys fr etc to select the correct keyboard if not US qwerty prior to running this pipe command.

```
fdisk -l | more
```

For MACS:

```
diskutil list
```

2. Create a directory that we will mount the USB flash drive on

```
mkdir /store
```

3. Mount the USB flash drive, replacing sdx with the device name of your USB flash drive found in step 1

```
mount /dev/sdx1 /store
```

4. Copy the log files to the USB flash drive

```
cp /nwipe_log* /store/
```

5. Unmount the USB flash drive

```
cd /;umount store
```

ShredOS includes the following related programs

smartmontools

Nwipes ability to detect serial numbers on USB devices now works on USB bridges who's chipset supports that functionality.Smartmontools provides nwipe with that capability. Smartmontools can be used in the second or third virtual terminal. ALT-F2 and ALT-F3.

hexedit

Use hexedit to examine and modify the contents of a hard disk. Hexedit can be used in the second or third virtual terminal. ALT-F2 and ALT-F3.

hdparm

hdparm has many uses and is a powerfull tool. Although Nwipe will be adding ATA secure erase capability, i.e using the hard disk own firmware to initiate an erase, nwipe currently wipes drives using the traditional method of writing to every block. If you want to initiate a ATA secure erase using the drives firmware then hdparm will be of use.

Compiling shredos and burning to USB stick, the harder way !

The ShredOS system is built using buildroot. The image (.img) file is 47.4MiB and can be burnt onto a USB memory stick with a tool such as dd or Etcher.

You can build the image by doing:

```
$ git clone https://github.com/PartialVolume/shredos.x86_64.git (or
shredos.i686.git for 32bit)
$ cd shredos
$ make shredos_defconfig
$ make
$ ls output/images/shredos*.img
$ cd output/images
$ dd if=shredos-20200412.img of=/dev/sdx (20200412 will be the day you
compiled, sdx is the USB flash drive)
```

Shredos is based on buildroot

Buildroot is a simple, efficient and easy-to-use tool to generate embedded Linux systems through cross-compilation.

The documentation can be found in docs/manual. You can generate a text document with 'make manual-text' and read output/docs/manual/manual.text. Online documentation can be found at http://buildroot.org/docs.html

To build and use the buildroot stuff, do the following:

1. run 'make menuconfig'
2. select the target architecture and the packages you wish to compile
3. run 'make'
4. wait while it compiles
5. find the kernel, bootloader, root filesystem, etc. in output/images

You do not need to be root to build or run buildroot. Have fun!

Buildroot comes with a basic configuration for a number of boards. Run 'make list-defconfigs' to view the list of provided configurations.

Please feed suggestions, bug reports, insults, and bribes back to the buildroot mailing list: buildroot@buildroot.org You can also find us on #buildroot on Freenode IRC.

If you would like to contribute patches, please read https://buildroot.org/manual.html#submitting-patches