

Personalising the Damn Small Linux to roll your own OS.



There's an old unused computer at home that doesn't have a great hardware profile—a Pentium II processor with 64 MB of RAM. I wanted to use this computer meaningfully, and hence came up with this idea of customising Damn Small Linux (DSL), a small GNU/Linux distribution that can work with minimal hardware.

The users of this system were my parents, who know very little about computers and merely wanted a simple interface to browse the Web and check e-mails. I decided to make the system as simple as possible so that anyone could switch it on and start using it.

Here are the requirements specifications:

- 1 Use low-profile hardware
- 2 Quick booting
- 3 Show my name during start up :-)
- 4 Should not require entering a user name and password to log in to the system
- 5 Show the Web browser with some URL, as default

### Setting up the environment

Let's get started with the actual work. I am using Debian Lenny (5.0.3) as my base system to carry out these customisations (you may require root permissions in order to carry out most of the tasks explained; I use *sudo* to achieve this). Download the DSL 4.4.10 ISO image for reference from [www.damnsmalllinux.org/download.html](http://www.damnsmalllinux.org/download.html). This is the ISO image you are about to customise for your needs. Create a new folder in your system with the following layout:

```
/dsl-sandbox
  iso-gen.sh
  dsl-4.4.10.iso
  /image
  /master
    /boot
    /KNOPPIX
  /source
    /KNOPPIX
```

You have an *iso-gen.sh* script—which you can use to generate your final ISO image—in the above folder layout. Its contents are as follows:

```
#!/bin/sh
# iso-gen.sh: Generate the final iso.

cd master
rm -f KNOPPIX/md5sums; find -type f -exec md5sum {} \; >> KNOPPIX/md5sums
mkisofs -pad -l -r -J -v -V "KNOPPIX" -no-emul-boot -boot-load-size 4 \
  -boot-info-table -b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat \
  -o ../styleisen.iso .
```

You need to mount the DSL ISO image that you have downloaded in order to have access to all the files that it contains. The steps are illustrated below. Note that you'll require the *cloop-utils* package installed in your Debian system—it offers the following *create\_compressed\_fs* and *extract\_compressed\_fs* tools.

```
$ cd dsl-sandbox
$ sudo losetup /dev/loop0 dsl-4.4.10.iso
$ sudo mount /dev/loop0 image
$ cd image; sudo find . -size -10000k -type f -exec cp -p --parents '{}' ../master/ \;
$ cd ..
$ sudo cp -pR image/boot/* master/boot/
$ umount image
$ sudo losetup -d /dev/loop0
```

There is a compressed KNOPPIX file in *image/KNOPPIX*, which you need to uncompress, and then extract the required files to customise as follows:

```
$ extract_compressed_fs image/KNOPPIX/KNOPPIX > /tmp/KNOPPIX_extracted
$ sudo mount -o loop /tmp/KNOPPIX_extracted image
$ sudo cp -Rp image/* source/KNOPPIX
$ sudo umount image
$ sudo losetup -d /dev/loop0
```

## Change boot image

Make a 640x400-pixel, 16-colour graphic and save it as a GIF (for example, *logo.16.gif*) using the GIMP. Feel free to create anything you like. All you have to take note of is to convert the image to a 16-colour graphic. Then we need to convert this GIF image to *lss16*. For this you require the *ppmtolss16* tool, which is part of the *syslinux* package in Debian:

```
$ giftopnm < logo.16.gif > logo.ppm
$ ppmtolss16 <logo.ppm > logo.16
```

Move the image *logo.16* to the master boot folder:

```
$ sudo mv logo.16 master/boot/isolinux/logo.16
```

Additionally, you can also take a look at the *master/boot/isolinux/boot.msg* file to change some text, if required.

## Change the start-up script text

The start-up script text is the one that you see immediately after the boot screen. We can customise this text in order to suit your needs. This is how to access the script:

```
$ sudo gunzip master/boot/isolinux/minirt24.gz
$ sudo mount master/boot/isolinux/minirt24 image -t ext2 -o loop
```

Open the *image/linuxrc* file in a text editor and edit as required. Save it! Finally, repack it in as follows:

```
$ sudo umount image
$ sudo gzip -9 master/boot/isolinux/minirt24
```

## Modify the GRUB menu entry

Edit the file *source/KNOPPIX/boot/grub/menu.lst* to look like the following code:

```
default 0
timeout 2
title Styleesen OS
kernel /boot/linux24 root=/dev/hda1 quiet vga=normal noacpi noapm nodma
noscsi frugal
initrd /boot/minirt24.gz
```

Now you have all the files required for customising DSL in your *dsl-sandbox* folder. We *chroot* into the source directory and do the required customisation. We have 'vi'

to do all the editing tasks inside the *chroot* environment.

```
$ sudo chroot source/KNOPPIX/
bash-2.05b# mount -t proc /proc proc
bash-2.05b# rm etc/skel/dfmdesk/MyDSL etc/skel/dfmdesk/Apps etc/skel/dfmdesk/.linktohomedir -rf
```

## Desktop and browser changes

First, you need to remove all menu items from the default desktop—that is, JWM. In order to do this, edit the file named *opt/mydsl\_menu/jwm/menu\_template* and */etc/skel/jwmrc* inside the *chroot* environment so that its contents resemble the following code:

```
<?xml version="1.0"?>
<JWM>
</JWM>
```

One of the requirements is to start the OS and show Firefox as the default browser displaying some website. In order to achieve this, open the file */etc/skel/.xinitrc* and search for the word 'dillo'. When you are at this line, edit it to resemble the following code:

```
firefox &>/dev/null &
```

We may opt to remove the next line 'torsmo 2>/dev/null &' if we don't want to see system statistics on the desktop, by default.

Open the file */usr/local/firefox/defaults/pref/firefox-branding.js* and edit it to point to whatever URL you want, in order to provide a default homepage when Firefox starts up. Here is what my file looks like:

```
pref("startup.homepage_override_url","http://www.styleesen.org/");
pref("startup.homepage_welcome_url","http://www.styleesen.org/");
```

Have a look at */opt/backgrounds/readme.txt* in order to change the default desktop backgrounds.

After going through all the above steps, do not forget to unmount the */proc* filesystem and log out of the *chroot* environment, which is illustrated below:

```
bash-2.05b# cd /
bash-2.05b# umount proc
bash-2.05b# logout
```

## Build the OS

Now that we're done with all the customisations, let's start building the OS. It's very simple. If you have followed the folder structure that I have mentioned in the beginning, then issue the following commands from the *dsl-sandbox/* directory:

```
$ sudo rm -rf source/KNOPPIX/.rr_moved
$ sudo mkisofs -R -U -V "KNOPPIX.net filesystem" -publisher \
```

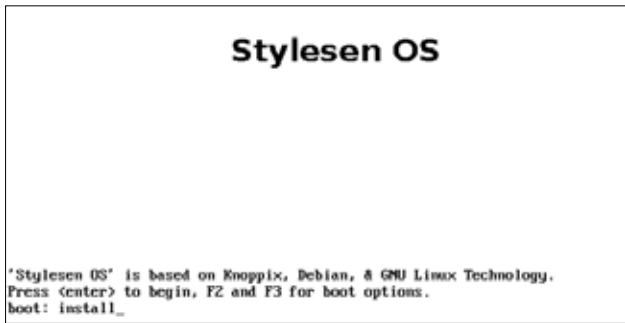


Figure 1: Boot from CD and install OS

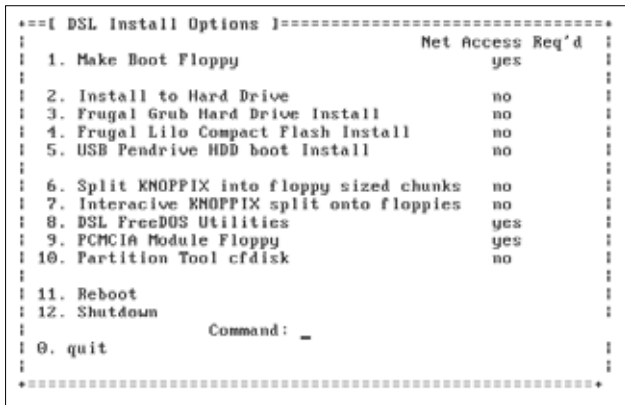


Figure 2: Installation Options Menu

```
"KNOPPIX www.knoppix.net" -hide-rr-moved -cache-inodes \
-no-bak -pad source/KNOPPIX | nice -5 create_compressed_fs \
-65536 > master/KNOPPIX/KNOPPIX
$ chmod +x iso-gen.sh
$ sudo ./iso-gen.sh
```

Once you have run these commands, you will get a new file called *stylessen.iso* (as specified in *iso-gen.sh* script) in your current folder.

Before burning this ISO image to disk, if you have a virtualisation solution at your disposal (say, VirtualBox), do test whether or not everything is in order. We can use this ISO image either as a live CD, or install it in our computer. The installation process is as follows:

- 1 Insert the CD where you burnt the *stylessen.iso* image.
- 2 A boot menu pops up; in the 'boot:' prompt, type 'install'.
- 3 You will get a menu that is self-explanatory and has features to partition your disk too.
- 4 Once the installation is over, reboot the system after removing the CD.
- 5 Now the system is ready and should ask you to set up the root password.
- 6 You will also be taken through the steps to configure your X server.
- 7 Once all the above steps are done, you'll have the new OS. Subsequent reboots will take you directly to the Firefox browser based upon how you have installed the system.

The following figures show a sample installation process and the final screens:

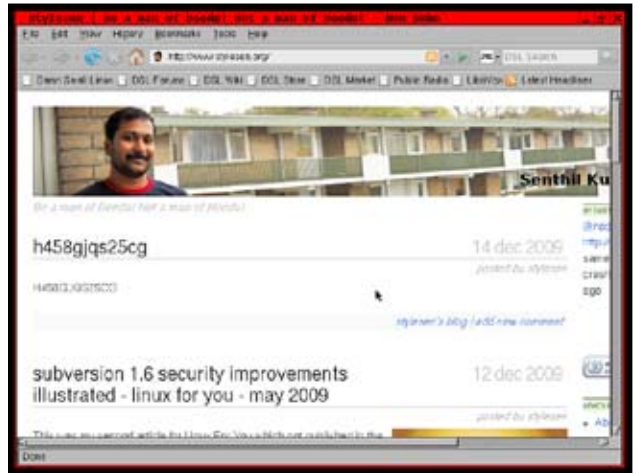



Figure 3: The system after booting



Figure 4: Shut down system after closing Firefox

This is how I made a customised GNU/Linux set up, which I run on a low-profile machine that serves my parents.

### A recap

This article explains the basics of setting up an environment to customise Damn Small Linux and walks you through some simple customisation steps. There is a lot more you can customise, which you can explore on your own. If you're running a browsing centre, you can consider this article to roll out your own customised OS to enable you to utilise cheap hardware. **END** 

#### References

- <http://www.damnsmalllinux.org/download.html>
- [http://www.knoppix.net/wiki/Knoppix\\_Remastering\\_Howto](http://www.knoppix.net/wiki/Knoppix_Remastering_Howto)
- <http://ryanlee.org/journal/view/5023/>

#### By: Senthil Kumaran S.

The author is currently employed by CollabNet, working for its Version Control Group. He is a full committer of the Subversion project and is a free software enthusiast. To know more, visit [www.stylessen.org](http://www.stylessen.org).