



Frenk_Danielle Kaufmann, Fotolia

Linux distributions now come with polished interfaces and all kinds of eye candy. Compared with other distributions, Puppy Linux [1] looks dated and less attractive (Figure 1). Puppy Linux might not win any beauty contests, but it's what is inside that matters. If you are willing to look beyond its appearance, you'll discover a gem of a Linux distribution.

Created by Barry Kauler, a retired Australian professor, Puppy Linux was designed to be a small, efficient, and user-friendly Live CD Linux distribution. A few other well-known distributions fall into this category, including Damn Small Linux (DSL), SLAX, and SAM Linux, but Puppy Linux stands out in the pack:

- Built from scratch, Puppy Linux is lean and resource efficient.
- On boot, the entire distro is loaded into RAM and runs without accessing the CD, which makes Puppy Linux lightning fast.
- Puppy Linux offers the ability to save session data in a separate file, even when you are running the distro from a CD-RW.
- Puppy Linux installs on virtually any type of storage device: a USB stick, an external hard drive, or a memory card (Figure 2). An installer walks you through the installation process and takes care of the technical side.
- Easy-to-use configuration tools allow you to tweak the system's settings.
- Puppy Linux includes a set of lightweight applications for the Internet, office productivity, graphics, video, and sound, and there are even a few games to keep you entertained.
- Puppy Linux includes its own package manager, making the task of installing additional applications a breeze.

A little Linux with more bite than bark

PUPPY LINUX

We'll play with Puppy Linux, a well-trained little Live CD Linux that requires minimal housebreaking. **BY DMITRI POPOV**

All this makes Puppy Linux an excellent distribution for use on old machines, as well as a handy tool to have on hand when you are on the move.

As with any Live CD Linux distribution, the first thing to do is download an ISO image of the latest Puppy Linux release and burn it onto a CD. Before booting your machine from the freshly baked CD, you have to make sure that the machine's optical drive is set as the first boot device, which is done in BIOS.

Similar to other Live CD distros, Puppy Linux supports boot parameters. For example, the *puppy prefix = ram* pa-

rameter makes Puppy Linux load in RAM by passing saved sessions, whereas the *puppy prefix = purge* parameter forces a radical file cleanup, which can be useful for fixing a broken system. For a full list of available boot parameters and their descriptions, check the *Boot Parameters* page at the Puppy Linux wiki [2].

During the boot process, you are prompted to choose the correct keyboard layout and an X graphics server from two options: Xorg or Xvesa. Xorg supports many of the advanced features of modern hardware, but it might not work well with older machines. Xvesa has a

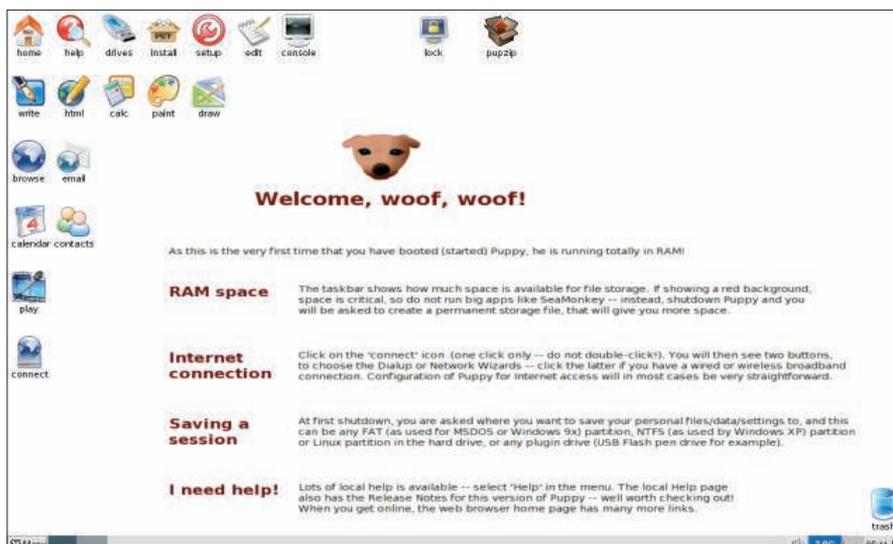


Figure 1: Puppy Linux proves that looks are not everything.



Figure 2: You can install Puppy Linux on virtually any storage device.

rather limited feature set, but it works with pretty much anything. As a rule of thumb, you might want to try Xorg first, and if it doesn't play well with your display, you can fall back on Xvesa.

Once Puppy Linux is booted, you are prompted to select the correct screen resolution. Usually, Puppy Linux does a good job of guessing the appropriate resolution, so in most cases, you only have to press the *Test* button to ensure that the selected settings are correct, and then confirm them by pressing *Okay*. Alternatively, you can select the appropriate resolution manually. Once Puppy Linux is up and running, take a look at the default wallpaper, which contains a few tips that can help you get started with Puppy Linux, including information about RAM space, configuring an Internet connection, and saving your settings and data.

Installing Puppy Linux

Although Puppy Linux will happily run off a CD, you can also install it onto virtually any storage device. Puppy Linux comes with its own installer, which you can launch by choosing *Menu | Setup | Puppy universal installer*. Because the installer provides clear descriptions of every step of the installation process, all you have to do is read them carefully and choose the appropriate options; however, a few settings require particular attention. For example, to boot Puppy Linux from a USB stick, which uses the FAT32 file system, you need to install a

bootloader, select the *mbr.bin* option when the installer displays a dialog window with a list of available bootloaders. If you're installing Puppy Linux onto a new USB stick, chances are that it is not formatted as a bootable device. In this case, the installer prompts you to set its boot flag with the GParted tool. To do this, launch GParted, right-click on the USB stick's partition, and choose *Manage Flags*. Next, select *Boot*, press *OK* to close the window, and then press the *Apply* button to commit the changes. Finally, close GParted and the installer does the rest.

Installing Puppy Linux on a hard disk is equally easy, and you can choose between a frugal and full installation. The frugal installation simply copies the Puppy Linux files (*vmlinuz*, *initrd.gz*,

pup_301.sfs, and *zdrv_301.sfs*) from the CD to a partition on the hard disk, which allows you to run Puppy Linux as a Live CD distro from the hard disk and save the settings in a separate file. This approach makes it possible to install Puppy Linux alongside another Linux distro on the same partition, but the downside is that you have to edit the GRUB bootloader manually. The full installation allows you to install Puppy Linux as a full-blown Linux distribution, which occupies the entire partition you set for it.

Running Puppy Linux with QEMU

Puppy Linux installed on a USB stick makes a great portable computing environment. Instead of lugging a laptop, you can carry Puppy Linux in your pocket and run it on any computer. Some situations, however, do not permit you to reboot a Windows machine into Puppy Linux. The QEMU Manager tool [3] – essentially the QEMU emulator wrapped into an easy-to-use interface – provides a nifty solution to this problem (Figure 3). More importantly, QEMU Manager is a portable application, so you can install it on a USB stick with Puppy Linux.

To create a QEMU-based Puppy Linux virtual machine (VM), download the portable version of QEMU Manager and the latest ISO image of Puppy Linux. Unzip the QEMU Manager archive and move the resulting folder onto your USB stick. Move the ISP image into the images directory inside the QEMU folder, then launch *QemuManager.exe*. Pressing



Figure 3: Puppy Linux running in QEMU Manager on Windows.

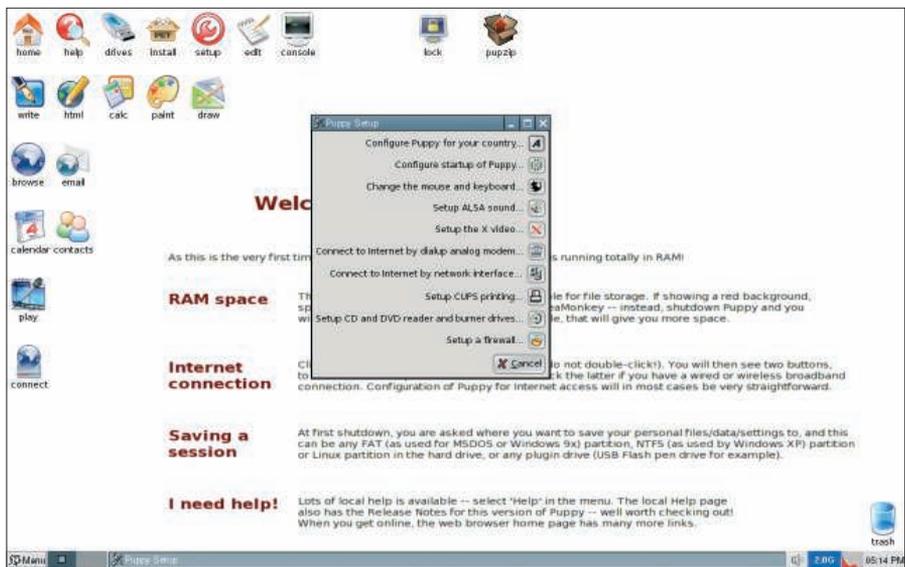


Figure 4: Puppy Linux comes with a few easy-to-use configuration wizards.

the *Create New Virtual Machine* button launches a wizard that helps you set up a new VM. All the options in the wizard are pretty self-explanatory, so you shouldn't have any difficulties configuring the basic settings.

Once you've walked through all the steps of creating the VM, make sure that the *View Advanced Configuration Options After Saving* box is selected and press the *Save Virtual Machine* button, which saves the VM and opens the Virtual Machine Configuration window. Next, switch to the *Disk Configuration* tab. In the CD-ROM section, click on the browser button, select the downloaded Puppy Linux ISO image, and select *Boot From CD-ROM*.

Pressing *Save* saves the settings, then you can close the window. Now you can launch the Puppy Linux VM by selecting it in the main window and then *Launch Selected Virtual Machine*.

Tweaking the Puppy Linux Settings

Regardless of which installation route you choose, you must configure the system's settings once it's up and running. Puppy Linux provides a control panel that allows you to configure and tweak all the settings using wizards, which is one reason Puppy Linux is a good distribution for users new to Linux.

To access the control panel, choose *Menu | Setup | Wizard Wizard*. The control panel includes the wizards that can help you configure virtually every aspect of Puppy Linux, including local settings,

ALSA sound, X video, dial-up and network connections, printing, and firewall (Figure 4).

Although Puppy Linux usually does a decent job of detecting and configuring the hardware, network settings almost always require additional tweaking by hand (which is also true for other Linux distros). To tackle this problem, Puppy Linux has the *Puppy Network Wizard* (Figure 5), an indispensable tool for setting up wired and wireless network connections. Configuring the latter can be rather tricky, but the wizard offers a few clever features that can help you deal with the task.

If Puppy Linux fails to correctly identify and configure the wireless card, you can use the wizard to load the correct driver manually. To do this, select *Load*

module, then select the appropriate module from a comprehensive list of wireless drivers and press *Load*. If the driver for your wireless card is not on the list, you have the option of installing a Windows driver with the *NDISwrapper*. Switch to the *More* section, select *NDISwrapper*, point to the correct driver file, and press *OK*. Once the correct driver is loaded, you have to create a new profile. Select the wireless interface by pressing the appropriate button, then select *Wireless*, create a new profile, and fill out the required fields. The wizard supports multiple profiles, which can come in handy if you often switch between different wireless networks.

Customizing the default Puppy Linux appearance is easy (Figure 6). As a minimum, you can change the default wallpaper with the built-in utility: *Menu | Desktop | Puppbackground desktop image*. Also, you can remove the application icons you don't need from the Desktop by right-clicking on the icon you want to remove and selecting *Remove*. You can replace the existing icons with, for example, icons from the Tango icon library [4]; right-click on the default icon, select *File | Set Icon*, and drag a replacement icon onto the window.

If you've chosen the frugal install or you're running Puppy Linux from a USB stick or another external storage device, all your settings and data are saved in a separate *pup_save.2fs* file. When you shut down the system for the first time, Puppy Linux prompts you to save the data in a file, and if you choose to do so, it runs a simple wizard that walks you

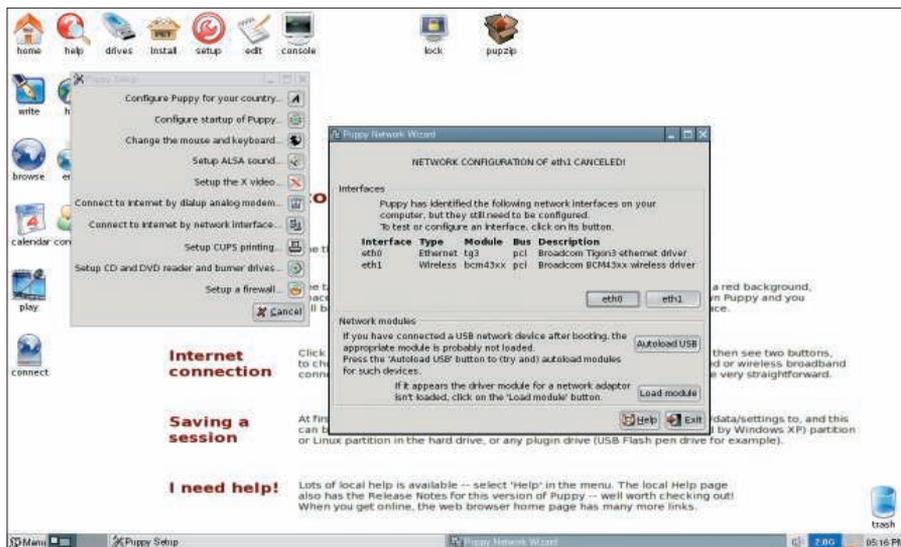


Figure 5: Puppy Network Wizard has everything you need to configure a network connection.

though the process. When you next boot Puppy Linux, it automatically loads the created *pup_save.2fs* file.

Installing Software

Puppy Linux comes with its own package manager, which you can use to install additional applications easily from the official repository (see Figure 7). Puppy Linux uses its own package format called PET, and although the list of available PET packages is rather short, it includes many essential applications such as Mozilla Firefox, OpenOffice.org, GIMP, and others. Installing PET packages with the Puppy package manager is as easy as selecting the applications you want and pressing *Okay*. Then, the manager downloads the selected packages, checks for dependencies, and installs the software.

Besides PET packages, Puppy Linux can handle *.deb* packages, which allow you to tap into Debian's immense software archive. To enable this feature, you have to install two PET packages with the Puppy package manager: the Dillo web browser and the *pb_debianinstaller* tool. Next, download the *.deb* package you want to install from the Debian repository [5].

Launch the terminal with *Menu | Utility | Rxtv Terminal Emulator* and run the *pb_debianinstaller* command. This launches the web-based package installer in the Dillo browser. Then, press the *Choose file* button and select the downloaded *.deb* package, press the *Check dependencies* button, and install the required packages, if necessary. After

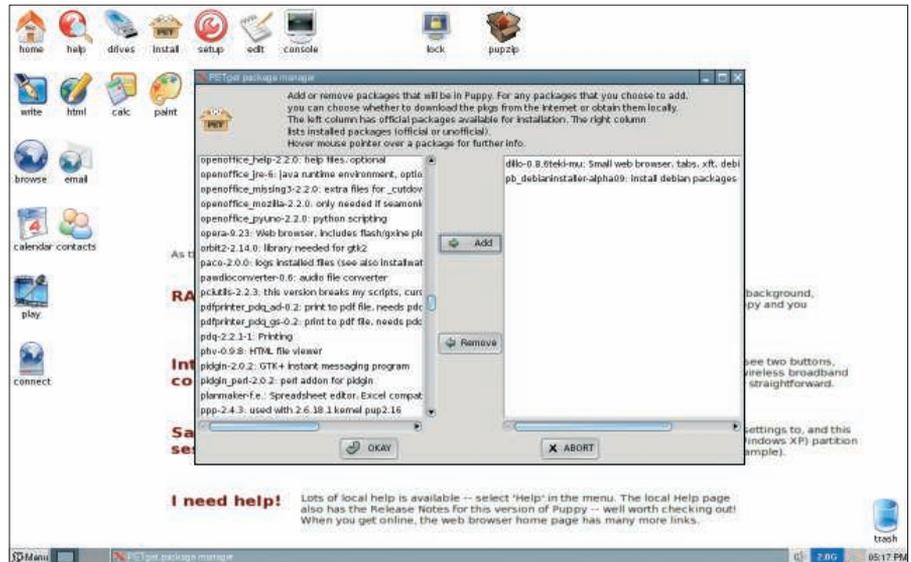


Figure 7: Puppy package manager makes it easy to install additional software.

pressing the *Install now* button and *Finish*, you are done.

Now you can launch the installed application in the Terminal. To remove the installed Debian packages, you can use the Puppy package manager.

When installing Debian packages, keep in mind that the *pb-debianinstaller* tool is still experimental and can make your system unstable or even crash it – so use it with caution, and don't forget to back up your data.

Remastering Puppy Linux

After you've spent hours tweaking and customizing Puppy Linux and installing additional applications, you might want to turn the final result into your own Linux distro. A remastering tool included with Puppy Linux (*Menu | Setup | Re-*

master Puppy Live-CD) allows you to do so with just a few clicks. The tool simply integrates all custom settings into the *pup_301.sfs* file (where 301 refers to the current Puppy Linux version), creates an ISO image, and burns it onto a CD or DVD. All you have to do is choose the partition or disk that the remastering tool uses to create the ISO image and pop a blank CD or DVD into the burner when prompted.

Final Word

With all the attention and hype that surround mainstream distributions like Ubuntu, openSUSE, and Mandriva, it's easy to overlook small wonders like Puppy Linux. This distro runs fast, even on older hardware; you can install it onto virtually any kind of storage device; and even a newbie can configure it with the aid of the wizards. Add to this Puppy's package manager, the ability to use Debian software, and the clever remastering tool, and you have one fantastic Linux distribution. ■

INFO

- [1] Puppy Linux: <http://www.puppylinux.com>
- [2] Puppy Linux boot parameters: <http://www.puppylinux.org/wikka/BootParms>
- [3] QEMU Manager: <http://www.davereyn.co.uk>
- [4] Tango icon library: http://tango.freedesktop.org/Tango_Icon_Library
- [5] Debian packages: <http://www.debian.org/distrib/packages>

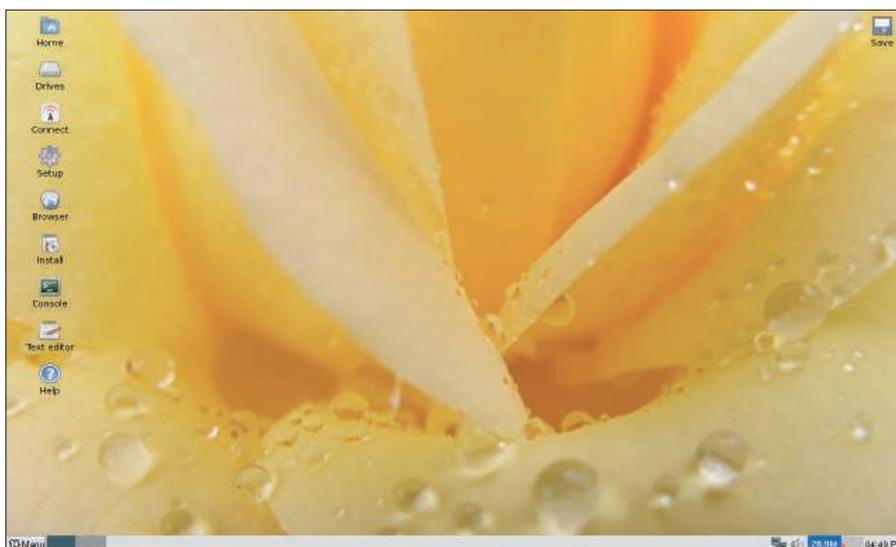


Figure 6: With a bit of tweaking, you can make your Puppy Linux look like this.