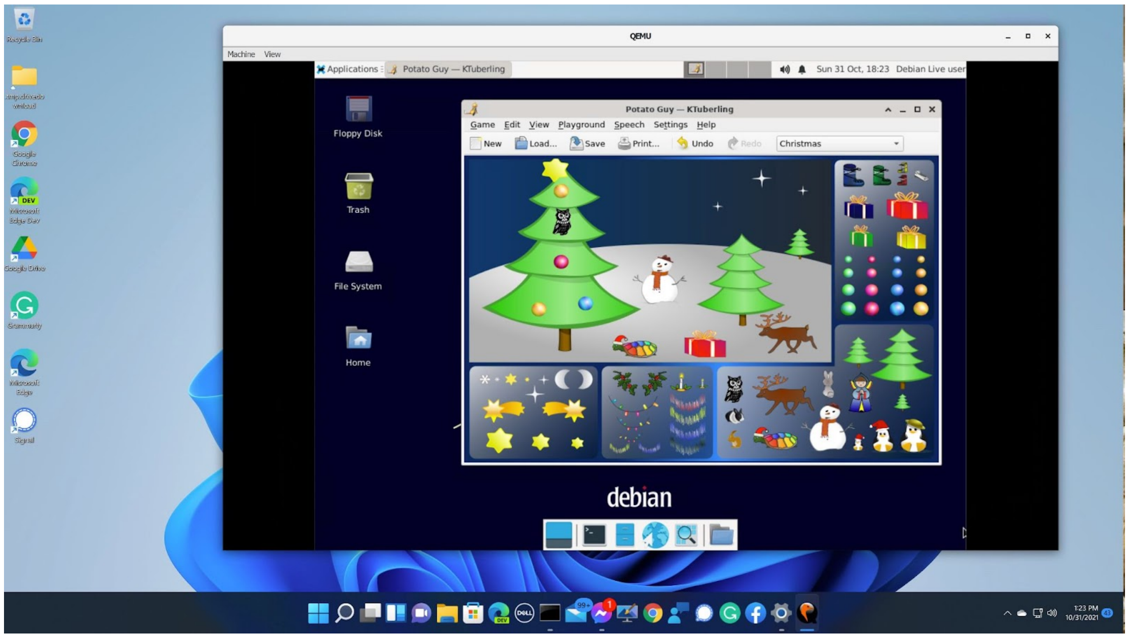
Linux as a Screensaver for Windows: The Gift of Open Source Games and SBOMs for the Holidays

By **Linux.com Editorial Staff** - December 7, 2021

Abstract: Construct and package a Linux® Live DVD to install using the standard Microsoft® Windows® install process and operate as a classic Windows screensaver.


QEMU
Machine View
Applications
Potato Guy — KTuberling
Sun 31 Oct, 18:23 Debian Live user
Game Edit View Playground Speech Settings Help
New Load... Save Print... Undo Redo
Christmas
Floppy Disk
Trash
File System
Home
debian

Introduction

- Back in 2005, IBM wanted to promote Linux, so developerWorks was offering $1000 per article to IBMers who wrote articles for the Linux Zone. The 2005 article is no longer online from IBM but is available on ResearchGate https://www.researchgate.net/publication/272094609_Linux_screensaver_for_Windows for the interested reader.
- This software still works and is still fun to use and to decorate your Windows desktop.
- Since 2005, there have been improvements and changes. Debian is now used instead of the original KNOPPIX. Additionally, full mouse integration now works between Windows and the screensaver due to kernel contributions.
- Future possibilities probably lie with the integration of hardware virtualization acceleration.
- Like all software of significant size, many components need tracking. The modern standard for this is SPDX and SBOM; as this screensaver is built fully from public source code, it makes a cool demo for SPDX and SBOM, which anyone may use.
- Though putting Linux on screen saver is a very interesting idea, there is a bit of a downside: power consumption. Screen savers initially proposed to protect the screen by providing moving pixels (by activating different pixels to avoid pixels burnin) when the user is not using their screen. If the power/energy option is not set properly it may draw more power/energy [1]. Basically, the Linux system (power governers) would prevent the OS from entering the deep power state where there are lots of opportunities to save energy when the system is idle.

Answering the most common concern about open source software, this article shows that, yes, Linux will run under Windows.

So why should you read this article? Why, indeed, should I write it? My motive is to help remove two obstacles to the wider adoption of free and open source software.

Those obstacles are:

- The perceived difficulty and disruptive effects of installing Linux
- The uncertainty of hardware support for Linux

Most computer users are familiar with a Microsoft Windows environment and the variety of screensavers available to prevent unauthorized access to the data on the computer when unattended. The good news is that there is plenty of free and open source software available nowadays to enable Linux to install and run as a Windows screensaver. This article shows you how to construct an appropriate package, and in doing so, demonstrates that the “free” and “non-free” sides of the software Grand Canyon are not so far apart after all.

Running Linux Under Windows As A Screensaver App

But which Linux? Without knowing what a client intends to do, it would be irresponsible to make a blanket recommendation. However, on December 25, 2021, the demand for games will be great, and the delivery capability will be sufficient. And if you configure it as a screensaver, even the possibility of pressing the wrong key to start it is eliminated.

Making It Work: Nuts, Bolts, And Screws

Getting the ISO to run under another operating system requires an open source PC emulator, including an open source BIOS and an open source virtual graphics adapter (such as QEMU version 6.1.0). The emulator enables you to set up a virtual PC within a real one. To construct a screensaver, the best way is to configure it with a virtual DVD drive, keyboard, screen, and mouse, but without any virtual disks. This all runs using the magic of software emulation, but modern PC hardware is sufficiently fast for the task (which we originally designed in 2005). Some corporate environments would require the virtual PC not to have a network adapter — you can run Firefox in the screensavers here. This package has a network adapter, but it is simple to change this if required since all source code is supplied.

Here are the steps to make this work.

QEMU

You can build QEMU from source available here https://www.qemu.org/download/ , but there is a suitable prebuilt QEMU for Windows available at https://qemu.weilnetz.de/ . This example was built and tested with QEMU 6.1.0 .

It is necessary to write a small stub program to go into the **C:\WINDOWS\SYSTEM32** directory as an SCR file, which runs QEMU with appropriate parameters.
https://github.com/tjcw/screensavers/blob/master/packaging/crunqemu-usb.c is sufficient for this; it runs QEMU with 1024 MB of memory, one processor, and the mouse connected as if it were a USB tablet.

This stub can be built with mingw64, from the Cygwin open source package, or presumably (though untested) with a commercial Windows C compiler.

Disabling the network adapter in the virtual PC can be done with parameter "**-nic none**" on the QEMU command line.

Inno Setup

Inno Setup is an open-source packaging/installation tool for Windows available here https://jrsoftware.org/isinfo.php . I used version 6.0 for this example. Packaging with Inno Setup results in a warning from Microsoft Defender when installing the screensaver; this warning can be overridden with 2 mouse clicks. A future version of this blog will explain how to package with Microsoft-licenced (non-open-source) tooling to eliminate this warning.

Prebuilt screensaver distribution

The screensavers are available here on this torrent feed:

https://linuxtracker.org/index.php?page=downloadcheck&id=1185c790b15b92b039d616ed742e873ae57db6ce

You will need a torrent client, such as Transmission, to download it. It is especially important to check the sha256sum values as this channel is not under the control of Linux Foundation.

After downloading, you should check the 'sha256sum' of the files. This validates that you have indeed got the files the author intends. For Windows there is a no-charge 'Hash Tool' in the Microsoft app store which will do the job; for Linux you use the command line.

```
$ sha256sum *

b483ed3250fbfdb91c3bace04f46ad9ad0b507a9890e3a58185c3342e6711441  QemuSaverO

95f3a8d6217f2ff93932ab5ac6d8a2a30a4d0ea09afe3096f148f5be17961428  QemuSaverO
```

Extract the two zip files using the built-in Windows extract feature, and run the installer .exe files. Then go to the Windows screensaver selection screen and select either '**fr2**' or '**gk2**' as appropriate.

There will be a 4-minute hiatus in the middle of startup while the X server initializes — be patient.

'**QemuSaverOpen-1-6.zip**' is the required base package with the educational screensaver named **fr2**, and '**QemuSaverOpenGames-1-4.zip**' is an optional extension package with the games screensaver called **gk2**.

The source code for all components is available on the public Internet, and these links will lead you to it.

- **Debian 11** http://mirrorservice.org/sites/cdimage.debian.org/debian-cd/current/source/iso-dvd/
- **Qemu** https://gitlab.com/qemu-project/qemu
- **Qemu Windows build** https://qemu.weilnetz.de/w64/
- **Inno setup** https://jrsoftware.org/isinfo.php
- **Mingw64/cygwin** https://www.cygwin.com/
- **My GitHub repository** https://github.com/tjcw/screensavers

The screensavers can be uninstalled with the standard Windows uninstall tool.

File structure for the extracted zip file

The following file structure is used for the live DVD filesystem:

- An **exe** file is the installer.
- Files in **/qemu** are the installable QEMU files, which will be copied to **C:\Program Files\qemusaver.**
- Files in **/extras** are the screensaver and the built Live Linux ISO
- Files in **/screensavers** are a clone of my git repository. They are not used by the installed screensaver but are provided for the convenience of anyone who wants to explore how it works.

Creating the ISO image

The live-build package does the 'hard work' of building the ISO in Debian Testing (There is currently a bug in the Debian 11 version of **live-build**). You will need to install a (real or virtual) machine with the Debian Testing image available here:

https://www.debian.org/devel/debian-installer/

A script https://github.com/tjcw/screensavers/blob/master/bin/do_oi wraps this to provide a simple interface; see https://github.com/tjcw/screensavers/blob/master/README.md for a short guide on how to use it.

The ISO is bootable, so it is also possible to write this to a USB key and boot your system from there. Rufus https://rufus.ie/en/ is a suitable open-source tool if you want to do this under Windows. You will need a USB key of 16GB or larger to try this option.

That's really all it takes to install Linux from a zip file to run as a screensaver on a Windows machine.

Future Directions

The screensaver could usefully be enhanced to exploit hardware virtualization acceleration. This is done with HAXM on an Intel processor or WHPX on an AMD processor. It requires changing a BIOS setting and some configuration in the internals of Windows, so it is not currently suitable for use in a simple screensaver application.

As Linux and Windows march forward, it may be necessary to rebuild the screensaver package from time to time, mainly to pick up new certificates for web browsing.

Software Bill Of Materials (SBOM) For The Live DVD

In furthering the desire to improve education around open source software and increase awareness of how to minimize security vulnerabilities and exposure in the software supply chain, we wanted to update this article with a short tutorial on generating a Software Bill of Materials (SBOM) using the SPDX toolset.

This is how it is done.

The first is the script that needs to be injected into the screensaver build process:

```
#!/bin/bash -x

cp -pr live-build/config/content/. .

cd /var/cache/apt/archives && (

dpkg --version >/tmp/dpkg.version

COLUMNS=100 dpkg -l >/tmp/dpkg.dependencies
awk '{ print $2 }' </tmp/dpkg.dependencies >/tmp/dpkg.inslist

for p in $(</tmp/dpkg.inslist)

do

  dpkg --info $p*|grep Depends

done >/tmp/dpkg.deplist

for p in $(</tmp/dpkg.inslist)

do

  dpkg -p $p

done >/tmp/dpkg.depdetail

) </dev/null
```

This results in 5 files that need to be fed to the SPDX/SBOM tool. This script is in place in the 'screensavers' repository above and results in the files being placed in **/tmp** in the screensaver, also available as **chroot/tmp** on the screensaver build system.

Then it is a simple matter to run the SPDX/SBOM tool, and the ISO standards dependency list is generated.

[1] https://www.environment.admin.cam.ac.uk/resources/mythbusters-facts-top-tips/screens

Author: Chris Ward, Sr. Programmer, IBM
Co-authors: Nirav Patel, Vice President and Chief Architect, Linux Foundation and Eun Kyung Lee, Manager Hybrid Cloud Infrastructure Software Research, IBM

Linux.com Editorial Staff

The editorial team for Linux.com.