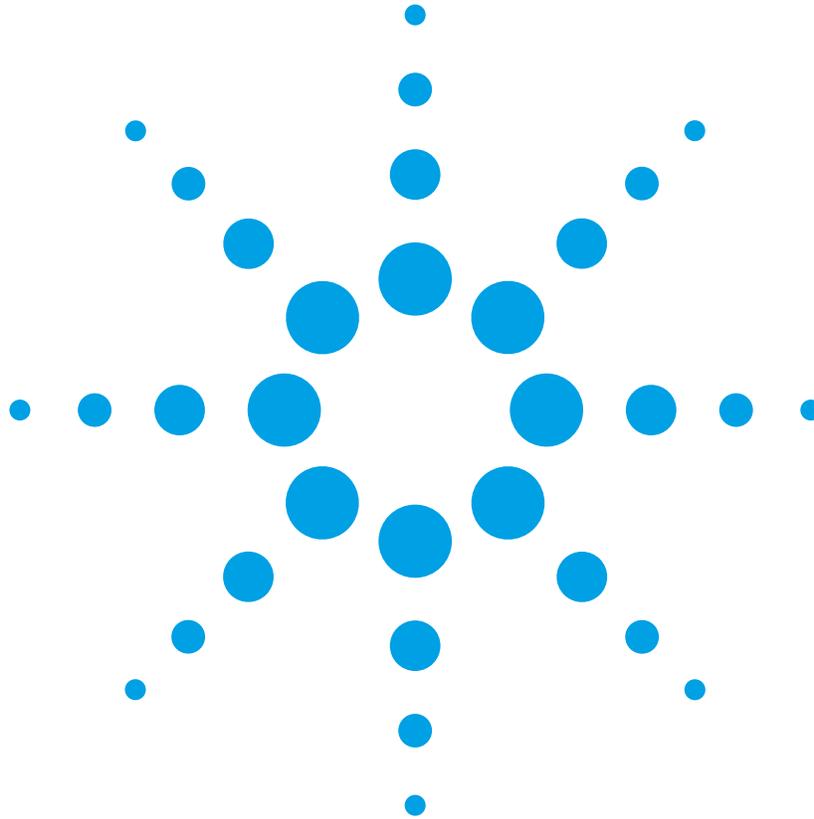


Using Linux in Your Test Systems: Linux Basics

Application Note 1465-27



The move to PC standard I/O interfaces is a key element of Agilent Open. You can take advantage of this strategy, especially if you are using Linux as the operating system for your test solution because the support for LAN and USB interfaces is built into the operating system. *Using Linux in Your Test Systems: Linux Basics* is the first in a series of application notes designed to explain how to control your test instruments under Linux.

Table of contents

Introduction—what is Linux?	2
Why some companies choose to use Linux for their test systems	2
Free software—the cornerstone for success of Linux	3
Languages and tools available for application development	4
Controlling instruments from Linux	5
Summary	5



Agilent Technologies

Introduction— what is Linux?



The history of Linux is closely linked to that of the GNU¹ project, which was founded in 1983 by Richard Stallman with the aim of developing a free (open-source) version of the UNIX operating system. While a lot of software was subsequently developed under the GNU flag, one key element was still missing: A free, open-source version of the operating system kernel (core).

This gap was filled with Linux. The Linux kernel was originally developed by Linus Torvalds in 1991 and made available under the GNU General Public License (GPL) in 1992, making it free software. Today's Linux distributions are sometimes called GNU/Linux because they are a combination of the Linux kernel and GNU utilities, libraries and tools (and additional software specific to the distribution).

Today, some Linux distributions (prepackaged sets of operating system and application software—see also figure 1) are commercial, i.e. there is a for-profit organization that manages the distribution, helps finance the development and typically offers for-pay support services. Other distributions rely solely on the contributions of a community of volunteers.

Since the 1990s, Linux has come a long way. It used to be an adventure playground for geeks: hard to install, configure and use. Today, it is a viable alternative to Microsoft® Windows® operating systems for both server and desktop applications, and it has also made its way into a fair number of embedded devices.

Why some companies choose to use Linux for their test systems

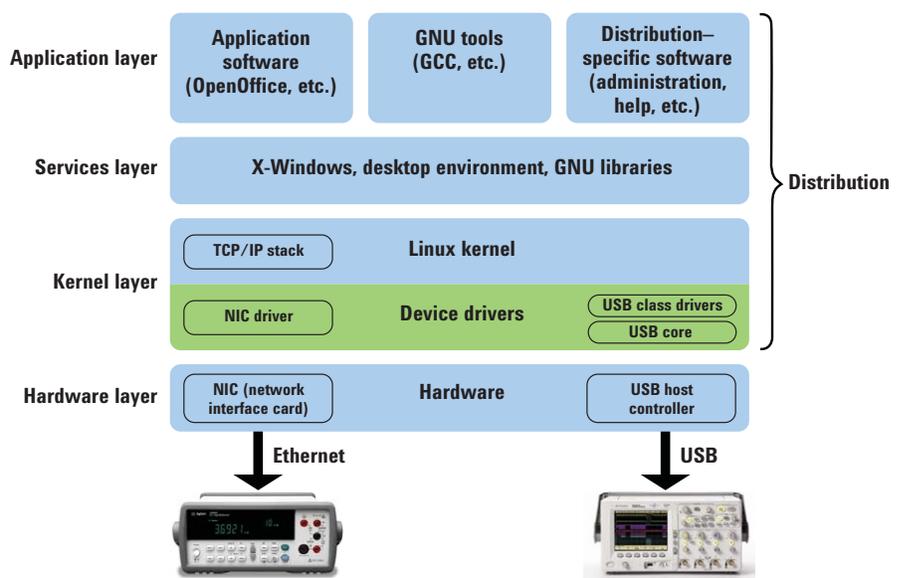
Companies or government agencies that own hundreds or thousands of PCs may choose Linux to reduce the cost of software licenses. Others adopt it for its high security standard. In test and measurement applications, other factors are often seen as more important.

- **Threat of software obsolescence.** The Linux operating system and its accompanying development tools are licensed under the GNU GPL (see below)—unlike a commercial product, they cannot be made obsolete. It is safe to assume that as long as the tools are being used somewhere, the community will continue to maintain them. This is a concern in industries such as aerospace, defense and automotive where long-term support is vital.
- **Flexibility/performance.** The operating system, and especially its kernel, can be modified and tailored for individual applications. For example, a stripped-down version of the system can yield exceptional performance even on a small, lightweight processor.

Also, the scheduler can be configured to optimize response time and soft real-time performance instead of CPU efficiency. These traits are very important in physical control applications and for testing electronic control units (ECUs). They are also the reason for the popularity of Linux in embedded applications.

- **Stability.** Linux and UNIX in general have always had the reputation of being relatively stable environments (admittedly, Windows seems to have caught up in recent years). Stability is a concern where tests run for extended periods of time and where downtime or loss of data is not an option (e.g. destructible testing, process monitoring).
- **Leveraging know-how and software.** Some companies, universities and research labs used UNIX in the past and developed competence and application software. Linux is a modern, open-source alternative for new applications, including those in test and measurement that builds on those existing competencies and simplifies reuse of existing UNIX software.

Figure 1. Structure of a Linux distribution and system



Free software—the cornerstone for success of Linux

The success of Linux is largely related to the fact that it is free software. In this context, free means freedom for the end user to use, inspect, modify and copy the source code. Most free software these days, including the Linux kernel, GNU software and third-party software, is licensed under the GNU General Public License (GNU GPL)² created by Richard Stallman and other members of the GNU project. Understanding the license terms is critical if you want to distribute your work, and especially if you need to protect your IP/source code.

The GPL mandates that software be made available in source code format and that everybody who accepts the license conditions be allowed to modify and redistribute the software. However, redistribution is only allowed if you retain the GPL and, consequently, the modified or enhanced source code (the derivative work) needs to be included in the redistribution. System integrators, take notice: unlike a lot of public domain software, the GPL doesn't rule out commercial use – and it is acceptable to charge a distribution or support fee for your derivative software.

It is not always obvious what should be considered a piece of derivative work. There has long been disagreement, for example, on whether application software that merely links to a GPL library is already derivative work. The Free Software Foundation (FSF)³ takes the view (considered aggressive by some) that any software that links to a library (whether statically or dynamically) is indeed a derivative. Many believe this position is a stretch and that it would not be upheld by courts of law.

The original GPL fits application software much better than libraries. Consequently, in 1991, an additional license, the GNU Library GPL (GNU LGPL)—later called the Lesser GPL—was introduced for libraries. In contrast to the original GPL, it explicitly allows commercial (non-GPL) software to link to an LGPL-based library while retaining its own proprietary license.

Table 1 summarizes the license situation and typical use models.

Table 1: The GPL and typical use models

License	When applied to...	Typical use model
GPL	Application	<p>Use: Classic free software application. Whatever is distributed to the end user must in turn be based on the GPL. The end users' right to copy the software or make changes themselves cannot be restricted.</p> <p>Commercial aspects: The distributor of the software can charge a one-time distribution fee and offer additional for-pay services such as a support contract.</p>
	Library	<p>Use: Classic free library approach. If the library is distributed as part of a bigger application or system (especially when statically linking), the whole application might be considered derivative work and, to be on the safe side, you should consider adopting the GPL. At the same time, many people take the view that an application with a proprietary license is allowed to dynamically link to a GPL library.</p> <p>Commercial aspects: There is some ambiguity (and associated risk), but the prevailing opinion is that application software can have a commercial license (non-free software with arbitrary conditions and funding scheme) as long as no GPL elements are included statically. Of course, the use of LGPL-based libraries is clearly preferable in order to avoid legal ambiguities.</p>
LGPL	Library	<p>Use: This license has become popular with libraries because it explicitly allows proprietary software to link (even statically) to the library. This fact and the unambiguous license situation in general make the library appealing for commercial use.</p> <p>Commercial aspects: This license is the safest and most flexible option for commercial software (with proprietary license).</p>

Languages and tools available for application development

Many open-source development tools are available under Linux. The following list is a starting point only. These tools are widely used in all kinds of applications and are available for a large number of distributions. Some will be available through your distribution's standard installation media.

- **Compilers.** The GNU Compiler Collection (GCC) is undoubtedly the most popular compiler suite used under Linux. It is an important element of the GNU project and a vital part of most Linux systems, since it is typically used to compile (generate) the system kernel. GCC is a modular compiler with front-ends existing for C, C++, Java, Pascal and a number of other languages. It also comes with standard libraries for those languages. See <http://gcc.gnu.org> for more information.

- **Debugger.** The GNU Debugger (GDB), another very popular tool, is a powerful command-line debugger to go along with GCC. See <http://www.gnu.org/software/gdb> for details.

- **Program editors.** Emacs (a.k.a. GNU Emacs) is a flexible text-based editor. It can be extended and customized with macros written in Lisp. For more information, see <http://www.gnu.org/software/emacs>. Many programmers simply use the standard graphical editor that comes with their chosen desktop environment, such as gedit (written for GNOME) or kedit (written for KDE).

- **Development of graphical user interfaces (GUIs).**

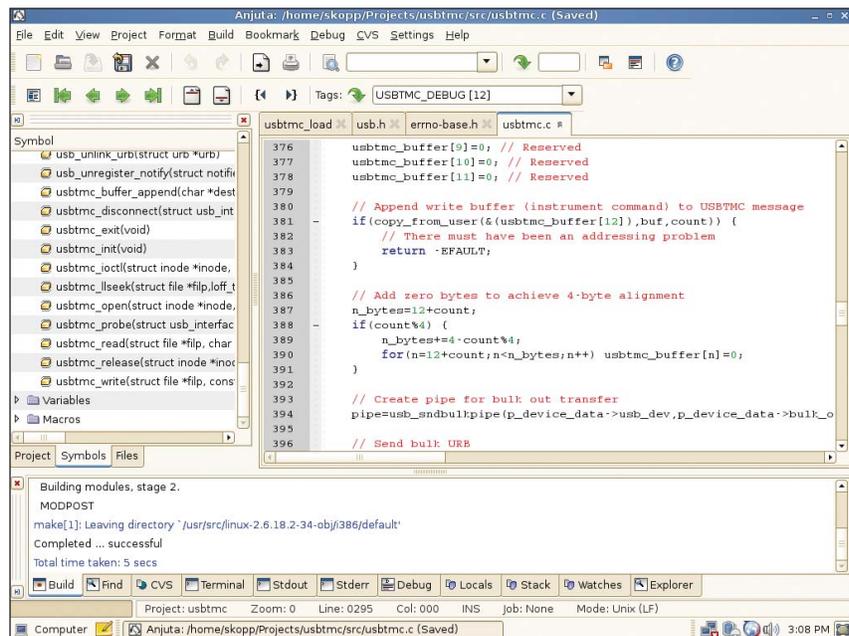
- **The GIMP Toolkit (GTK+)** is part of the GNU project and licensed under the LGPL. It supports multiple platforms (including Windows). The GNOME desktop environment is based on GTK+. For more details, see <http://gtk.org>.
- **Trolltech's Qt Toolkit** is available in a GPL version (for the development of non-commercial software) and in a commercial version (for proprietary software). It also supports several platforms, including Windows. KDE is based on Qt. See <http://www.trolltech.com>.
- **The Fast Light Toolkit (FLTK)** is a lean toolkit, licensed under the LGPL. Its small size (and its

license) allows static linking with proprietary applications. FLTK is available for several platforms, again including Windows. See <http://www.fltk.org> for details.

- **Integrated development environments (IDEs).** A number of open source IDEs are available, including Anjuta (written for GNOME, see <http://anjuta.sourceforge.net>), KDevelop (written for KDE, see <http://www.kdevelop.org>) and Eclipse (see <http://www.eclipse.org>). See Figure 2.

If you want to go look for other free software tools, take a look at the Free Software Directory at <http://directory.fsf.org>. This categorized directory is run by the Free Software Foundation (FSF) and supported by UNESCO.

Figure 2. Integrated development environments like Anjuta allow more efficient development and debugging



Controlling instruments from Linux

All of the popular instrument driver standards, including VXI*plug&play*, IVI-C and the latest IVI-COM drivers are designed for Windows—they do not work under Linux. The good news is, in most situations you do not need an instrument driver.

Most Agilent instruments are message-based: they are controlled through text commands, typically following the SCPI⁴ (Standard Commands for Programmable Instruments) standard, published in 1990. The beauty of SCPI lies in the fact that the standard is both vendor- and interface-independent. For example, many Agilent instruments feature several options for I/O (e.g. GPIB, USB and LAN)—and the exact same SCPI commands work through all of these interfaces.

For many years, Agilent has been advocating the use of standard PC interfaces, especially LAN and USB, for test instruments. Linux users benefit from this strategy because there is no need to install proprietary interface drivers—which may not be available for your chosen distribution and version. Support for LAN and USB is available in the operating system kernel no matter what flavor of Linux you are using. And, you can freely choose the distribution that best fits your requirements and taste.

LXI⁵ instruments are controlled through the VXI-11⁶ protocol (based on RPC) or direct TCP socket communication. Both options are straightforward and easy to use, but TCP is exceptionally elegant, because it is based on a message/stream approach itself. The next application notes in the series include AN1465-28 *Using Linux to Control LXI Instruments Through VXI-11* and AN1465-29 *Using Linux to Control LXI Instruments Through TCP* and will describe both options (VXI-11 and TCP, respectively) in detail.

Agilent's USB instruments are typically implemented according to the USBTMC⁷ standard (USB Test and Measurement Class). This standard was developed by Agilent and others in close cooperation with the USB Implementers Forum (USB-IF). Application note AN1465-30 *Using Linux to Control USB Instruments* will describe how to control USB instruments using the USB support built into the Linux kernel.

Summary

With the large number of open-source development tools available, Linux brings a lot of flexibility to the table. Agilent Open and the move to PC standard interfaces extend that flexibility into the test and measurement area. You can choose your favorite distribution and version and rely on the built-in support for LAN and USB for controlling your instruments.

- 1 See <http://www.gnu.org> for details.
- 2 See <http://www.gnu.org/licenses> for further details and the exact terms of the different GNU licenses.
- 3 The Free Software Foundation is a non-profit organization founded by Richard Stallman with the goal to promote free software. See <http://www.fsf.org> for details.
- 4 See <http://www.scpiconsortium.org> for details.
- 5 See <http://www.agilent.com/find/lxi> for details.
- 6 The VXI-11 specification is available for download at <http://www.vxi.org/freepdfdownloads>
- 7 The USBTMC specification is available for download at http://www.usb.org/developers/devclass_docs#approved

Related Agilent literature

The 1465 series of application notes provides a wealth of information about the creation of test systems, the successful use of LAN, WLAN and USB in those systems, and the optimization and enhancement of RF/microwave test systems. All of the individual notes listed below are also available in a compilation:

- *Test System Development Guide: A Comprehensive Handbook for Test Engineers* (pub no. 5989-5367EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-5367EN.pdf>

Test System Development

- *Test System Development Guide: Application Notes 1465-1 through 1465-8* (pub no. 5989-2178EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-2178EN.pdf>
- *Using LAN in Test Systems: The Basics* AN 1465-9 (pub no. 5989-1412EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1412EN.pdf>
- *Using LAN in Test Systems: Network Configuration* AN 1465-10 (pub no. 5989-1413EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1413EN.pdf>
- *Using LAN in Test Systems: PC Configuration* AN 1465-11 (pub no. 5989-1415EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1415EN.pdf>
- *Using USB in the Test and Measurement Environment* AN 1465-12 (pub no. 5989-1417EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1417EN.pdf>
- *Using SCPI and Direct I/O vs. Drivers* AN 1465-13 (pub no. 5989-1414EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1414EN.pdf>
- *Using LAN in Test Systems: Applications* AN 1465-14 (pub no. 5989-1416EN)
<http://cp.literature.agilent.com/litweb/pdf/5989-1416EN.pdf>
- *Using LAN in Test Systems: Setting Up System I/O* AN 1465-15 (pub no. 5989-2409)
<http://cp.literature.agilent.com/litweb/pdf/5989-2409EN.pdf>

Related Agilent literature, continued

- *Next-Generation Test Systems: Advancing the Vision with LXI*
AN 1465-16 (pub no. 5989-2802)
<http://cp.literature.agilent.com/litweb/pdf/5989-2802EN.pdf>

RF and Microwave Test Systems

- *Optimizing the Elements of an RF/Microwave Test System*
AN 1465-17 (pub no. 5989-3321)
<http://cp.literature.agilent.com/litweb/pdf/5989-3321EN.pdf>
- *6 Hints for Enhancing Measurement Integrity in RF/Microwave Test Systems*
AN 1465-18 (pub no. 5989-3322)
<http://cp.literature.agilent.com/litweb/pdf/5989-3322EN.pdf>
- *Calibrating Signal Paths in RF/Microwave Test Systems*
AN 1465-19 (pub no. 5989-3323)
<http://cp.literature.agilent.com/litweb/pdf/5989-3323EN.pdf>

LAN eXtensions for Instrumentation (LXI)

- *LXI: Going Beyond GPIB, PXI and VXI*
AN 1465-20 (pub no. 5989-4371)
<http://cp.literature.agilent.com/litweb/pdf/5989-4371EN.pdf>
- *10 Good Reasons to Switch to LXI*
AN 1465-21 (pub no. 5989-4372)
<http://cp.literature.agilent.com/litweb/pdf/5989-4372EN.pdf>
- *Transitioning from GPIB to LXI*
AN 1465-22 (pub no. 5989-4373)
<http://cp.literature.agilent.com/litweb/pdf/5989-4373EN.pdf>
- *Creating hybrid systems with PXI, VXI and LXI*
AN 1465-23 (pub no. 5989-4374)
<http://cp.literature.agilent.com/litweb/pdf/5989-4374EN.pdf>
- *Using Synthetic Instruments in Your Test System*
AN 1465-24 (pub no. 5989-4375)
<http://cp.literature.agilent.com/litweb/pdf/5989-4375EN.pdf>

Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to

www.agilent.com/find/removealldoubt



Agilent Email Updates

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.



Agilent Direct

www.agilent.com/find/agilentdirect
Quickly choose and use your test equipment solutions with confidence.



Agilent Open

www.agilent.com/find/open
Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.



LXI

www.lxistandard.org
LXI is the LAN-based successor to GPIB, providing faster, more efficient connectivity. Agilent is a founding member of the LXI consortium.

www.agilent.com

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at: www.agilent.com/find/contactus

Phone or Fax

Americas

Canada	877 894 4414
Latin America	305 269 7500
United States	800 829 4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	81 426 56 7832
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Thailand	1 800 226 008

Europe

Austria	0820 87 44 11
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700
Germany	01805 24 6333* *0.14€/minute
Ireland	1890 924 204
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
Switzerland (French)	44 (21) 8113811 (Opt 2)
Switzerland (German)	0800 80 53 53 (Opt 1)
United Kingdom	44 (0) 7004 666666

Other European Countries:

www.agilent.com/find/contactus

Revised: March 23, 2007

Tux logo created by Larry Ewing using GIMP.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2007
Printed in USA, May 7, 2007
5989-6715EN



Agilent Technologies