关于LabVIEW NXG和下一代智能测控软件的白皮书

简仪科技
JYTEK

2017年8月8日

目录

一：关于此白皮书 03
二：概述 04
三：LabVIEW NXG 的迁移之路 06
四：用户迁移到 LabVIEW NXG 的必要性和不必要性 08
五：不值得迁移到 LabVIEW NXG 的依据 10
六：酌情处理从现有 LabVIEW 到下一代测试软件的迁移 12
七：致学生、工程技术人员、老师、企业经理、研究院负责人 14
结语和关于简仪 15
免责声明 16
附录 1：LabVIEW 到 LabVIEW NXG 的兼容问题 17
附录 2：参考资料 22

一：关于此白皮书

2017年5月美国国家仪器有限公司(以下简称NI)同时发布了两个LabVIEW版本：LabVIEW 2017与LabVIEW NXG。同一家公司在同一个时间一起公布两个相近的产品(软件)，这是很不寻常的情况。鉴于LabVIEW在虚拟仪器测试测量领域的主导地位，NI的此举必将会对测试测量领域乃至与其该领域相关的工业、科研、教育等行业产生不小的连锁影响。

本白皮书旨在帮助相关行业的从业者和客户客观解读“两个版本LabVIEW”，给LabVIEW用户和非LabVIEW用户做决定时做一个参考，尤其是通过此白皮书可以帮助当前的LabVIEW用户在这个变革的时期，做出一个理性的、正确的下一步软件开发计划。

简仪科技提倡基于开源、C#、.NET等技术为基础的测试仪器软件。同时密切注视仪器软件的动态。简仪希望这份白皮书给业界人士带来中肯的建议，简仪也相信现今快速发展的软件技术终将给仪器软件带来翻天覆地的变化，并彻底打破多年来仪器软件被一家垄断的局面。

本文中的英文SeeSharp®是上海简仪科技的注册商标。LabVIEW, LabVIEW NXG, LabWindows/CVI, Measurement Studio及National Instruments均是美国National Instruments的注册商标；Visual Studio, .NET是美国微软的注册商标。

二：概述

通用仪器测试软件

当前测试测量领域使用的主要软件开发语言有C/C++、C#、VB、LabVIEW、Python等等，在虚拟仪器开发软件这一细分领域，在过去的30年中，LabVIEW以其直观的图形化编程特点，逐渐占据了该细分领域的主导地位，LabVIEW软件由美国国家仪器有限公司(以下简称NI)提出并常年推广。

由于测试测量行业的发展相对封闭和缓慢，相比当今的工程技术迅速发展，有很多优秀的技术并没有得到足够多的了解和重视，特别是在软件工程领域。现在的软件工程已经全面转向面向对象的编程思想，相比较测控界的软件工具和编程方法还停留在面向过程，LabVIEW就是面向过程软件的典型代表。面向过程能够快速解决小项目开发，但是在中型和大型复杂的定制化测试项目，面向过程的图形编程的各种弊病显现：可读性差，可维护性低，执行效率低，多线程实现难。相反，以C#为代表的微软主推的面向对象的语言快速发展，可以完美解决以上问题，其智能文本编程的编译环境易学易用，同时所有的编程逻辑和结构都可以进行面向对象的设计，使得代码重用，团队项目协作变得轻松自如。由于Visual Studio/C#/.NET是通用开发环境(IDE)，大量的应用并不在测试测量行业，又由于有一些测试测量行当专用的工具没有在此IDE中，这就造成了Visual Studio/C#/.NET IDE没有成为测试仪器主流软件。尽管如此，很多用户都已自发地发现了Visual Studio/C#/.NET的潜力，成功地开发了数不尽的测试测量应用。

简仪科技作为一个新成立的中国自主知识产权的测试公司，敏锐地察觉到Visual Studio/C#/.NET和开源社区的巨大潜力，在全球第一个创造性提出了直接用微软C#语言和.NET技术来直接搭建一个完全符合面向对象思维的测控软件平台：锐视开发软件—SeeSharp®。锐视开发软件SeeSharp®把成熟的Visual Studio/C#/.NET技术和开源社区的丰富内容进行系统的整理，有条不紊地引入到测控行业，这将彻底改变LabVIEW垄断通用测试软件的现状。

LabVIEW 和 LabVIEW NXG

LabVIEW 是NI 1986年推出的图形编程(G语言)软件，诞生的比Windows还早。最早期LabVIEW是用来控制仪器的，后来NI把LabVIEW的功能逐渐扩充到数据采集，PXI，工业控制，运动控制等多个领域。 在很大的程度上，LabVIEW就是虚拟仪器技术的别名，LabVIEW对仪器行业的贡献是巨大的，有目共睹的。NI在LabVIEW上巨大的投资加上NI早期在PXI的推动使得LabVIEW/PXI成为NI垄断PXI模块仪器市场的主要因素。LabVIEW也成为了最常用的通用测试软件。

随着LabVIEW垄断地位的逐渐形成，LabVIEW从9.0开始命名为LabVIEW 2009，此后版本以年份为代号频繁更新，一年一个版本，年中往往再增加一个.0.1版本或者SP1(例如LabVIEW 2009SP1)，实际上接近一年两个版本，很多用户喜欢用.0.1版本或者SP1，因为会解决掉一些软件使用问题。LabVIEW和其它商业软件一样，通过频繁的更新来促进围绕着LabVIEW的软硬件销售。

但是LabVIEW NXG不是LabVIEW 2017的寻常升级版本，否则它更可能的会被命名为LabVIEW 2018，而不是像现在这样，加一个NXG后缀说明 (即下一代：Next Generation)，还要从1.0做起。通过NI官网中NXG

与以往LabVIEW版本特性的比对表我们可以发现，不能想当然地认为当前的LabVIEW用户可以像以往那样升级，以往的代码从一个版本的LabVIEW转到下一个版本的LabVIEW，即所谓的升级，虽然不是完全顺畅，例如会有一些断线，新的工具包不能用，新的硬件驱动不支持等等问题。总的来说，用户已有的LabVIEW程序和应用还是能比较容易的升级到新的版本。 但这次NXG的出现和以往版本升级完全不一样，如果用户要继续保留以前多年的LabVIEW代码劳动成果，这次要大费周折，而且有些转换将是不可能的，例如RT、FPGA等产品线的相关应用。本白皮书将根据NI官方披露的信息来试图解读其前因后果。首先我们先来看一看软件技术发展的另外一支Visual Studio/C#/.NET。

Visual Studio/C#/.NET

早期的Windows应用受DOS和x86体系的限制，开发工具有限，开发效率低，蓝屏现象普遍。开发者往往要从最底层的WinAPI着手开发应用。虽然Win/Intel系统是主流系统，但是Win/Intel系统的可靠性一直备受质疑。为了改善Win/Intel系统的开发环境，微软、HP、Intel于2000年就提出了.NET的体系结构并通过了工业标准的认证。简言之，.NET是管理Windows底层设备的独立中间运行层。微软希望所有的应用开发者通过调用.NET来开发Windows的应用程序。为了使调用.NET便于使用，微软还同时研制了类似于Java的编程语言C#。几乎是在同一时刻，C#也被纳入为工业标准之一。

经过十六年的发展，.NET已经非常成熟，C#作为微软推荐的程序语言也成为了微软Visual Studio IDE中的主要成员之一，基于Visual Studio/C#/.NET的资源也日益丰富，特别是开源社区的资源。为了使Visual Studio/C#/.NET成为软件工程的主流开发工具，顺应开源软件的潮流，微软不仅将大部分的.NET代码开源，更在2014年将Visual Studio免费，减少开发人员的开发成本。同时，微软还花了大力气并购了.NET在Linux和MacOS上的实现平台Mono，使得Visual Studio/C#/.NET成为了跨平台的最佳选择。

正是由于微软的巨大投资及开源社区的贡献，Visual Studio/C#/.NET已经成为了Windows平台下的无可争议的最佳开发环境。世界上没有任何一家其它软件公司可以有资源来竞争这一地位。NI当然也不例外。所以把Visual Studio/C#/.NET引入仪器测试行当成为主流的测试软件只是时间的问题，在技术上没有任何的悬念。简仪只是顺势而为，促进和加快了Visual Studio/C#/.NET在通用测试技术中的应用。

三：LabVIEW NXG的迁移之路

NI的LabVIEW PK 微软的Visual Studio/C#/.NET

微软的Visual Studio/C#/.NET策略对所有在.NET之前就开发的软件的维护，持续开发都是一个极大的挑战。使用.NET的构架就势必要重写很多程序，兼容性也是一个大的问题，不采纳.NET将失去很多.NET提供的优质工具，维护和持续开发的成本越来越大。如下图所示这是一个进退维谷的尴尬处境。

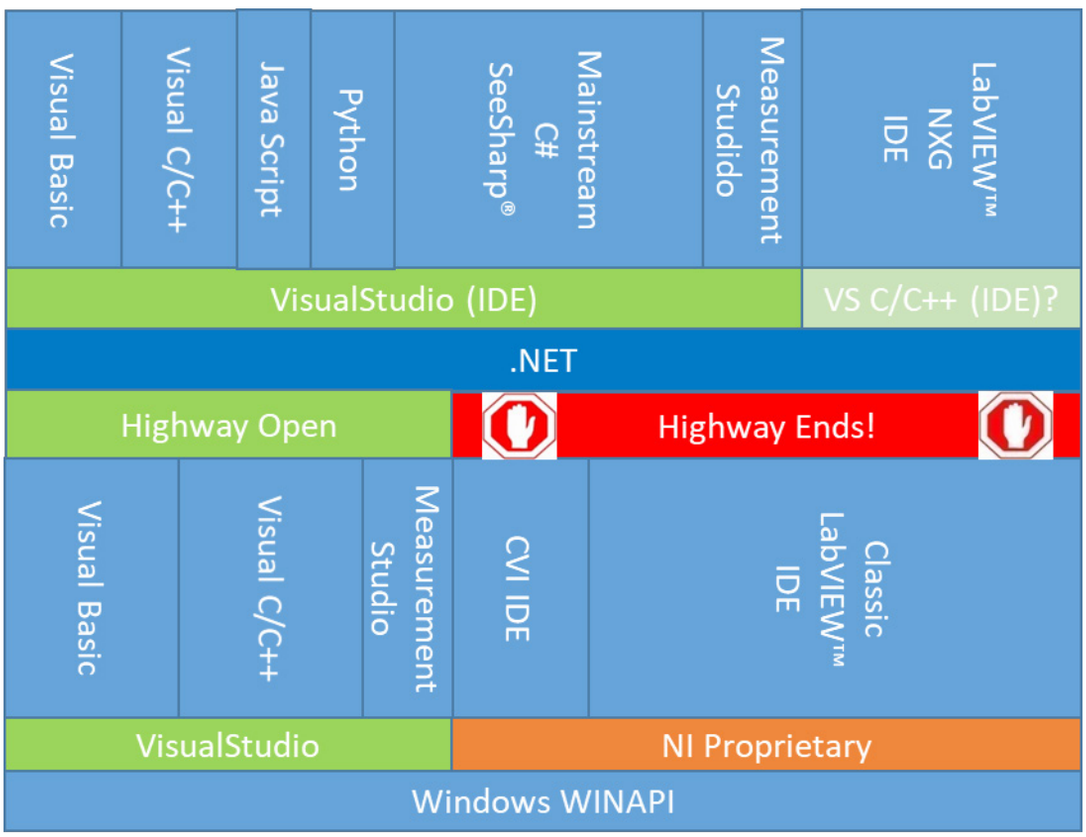


图1

值得额外提出的是：

(1) 原来NI的CVI 已经不再保持定期更新，一个可能的原因是微软Visual C 和CVI的功能高度重叠，Visual C的存在使得CVI的重要性大大下降。

(2) 老版本的非.NET LabVIEW底层源代码（原来基于Windows API编写）无法再在新的Windows环境下兼容运行。

(3) .NET和Visual Studio提供了大量高质量的制图功能，任何其它公司维护自己开发的制图软件都越来越不经济。所以LabVIEW NXG使用.NET也是情理中的事。综合以上几点，也就不难理解为何LabVIEW已经不得不升级到和.NET来兼容了。

(4) NI也提供一个Visual Studio环境下的附加收费控件，Measurement Studio。和锐视开发软件类似，Measurement Studio直接使用Visual Studio的编译。由于Measurement Studio和硬件的粘度低，长期以来不是NI市场的重点。

升级到LabVIEW NXG 并不简单

LabVIEW在过去的版本中都提供平滑的版本升级：大部分代码可重用。但同样的产品设计和使用感受升级到LabVIEW NXG就不是一个简单的过程，它是一个到全新技术架构的迁移(基于.NET技术)：很多老版LabVIEW源代码会出现“断线”，原有项目代码如果不重写就很难被重用。任何软件工程师都有这样的体会，改别人写的软件难度并不亚于重新写新的程序。NI的LabVIEW虽然在测量仪器界名闻遐迩，但在软件工程的大环境里终究是一个较小的专业软件。无论是在软件的工程技术上还是在对软件的投资力度上和微软Visual Studio/C#/.NET相比终究是天壤之别。所以微软的Visual Studio/C#/.NET在各方面都胜于LabVIEW是不足为奇的。Visual Studio/C#/.NET之所以没有成为仪器软件的主流不是因为Visual Studio/C#/.NET 不胜任，而是因为Visual Studio/C#/.NET 没有被系统的引进到测试行业。

通往LabVIEW NXG的大迁移

基于各种考量，NI做出了整个LabVIEW软件移植到.NET技术架构的决策，称为LabVIEW NXG。 这一决定和微软从WinAPI的开发环境迁移到.NET的开发环境一样，从根本上就具有不兼容的一面: 从LabVIEW架构在非.NET的WinAPI上迁移到LabVIEW NXG架构在.NET上。如果说微软经过了16年之后才使得NI下定决心将LabVIEW迁移到.NET的构架下，简仪估计NI从LabVIEW迁移到LabVIEW NXG的路将也是漫长的，艰辛的。有些迁移是无奈的，有些迁移是必要的。在剩下来的章节及附件，简仪将针对这些问题进行更深一步的探讨。

四：用户迁移到LabVIEW NXG的必要性和不必要性

任何新工具和技术的出现都将培育出新的产品，淘汰过时的老产品。软件亦不例外。LabVIEW经过了三十年的发展，也一直处于变化的工程世界，其中软件技术的变更更是突飞猛进。三十年前给工业界带来耳目一新的革命性产品LabVIEW在崭新的OOP（面向对象）技术及开源技术的挑战下，逐渐显露出了苍老无力的一面。跟随时代的步伐，重新发挥LabVIEW的吸引力当然对NI至关重要。这些重要性主要是体现在以下三个方面：

(1) LabVIEW有极大的粘性。凡是使用NI LabVIEW的用户，都无法再跳出LabVIEW。这是因为LabVIEW独特的G语言和编程方式。世界上再无第二家类似的公司。

(2) LabVIEW软件和NI硬件的紧密捆绑实时形成了NI在某些特定行当的垄断，比如说PXI领域。按照NI公开发行的信息，NI在PXI模块行业占有超出80%的市场份额。

(3) LabVIEW除了在仪器控制行当是开放性的，在其它诸如PXI，DAQ等行当是相当封闭的。众多的厂商抱怨他们的硬件产品可以在C/C++的环境下与LabVIEW媲美，但在LabVIEW的环境下却无法达到设计指标。

以上三点决定了LabVIEW对NI商业成功的决定性作用。在很长的时期里，LabVIEW几乎就是NI的代名词。

为LabVIEW NXG迁移买单

为了维持LabVIEW在市场里的地位，LabVIEW NXG诞生了。LabVIEW NXG的终极目的当然是继续延续LabVIEW的品牌效益，保证上节所述的三点在新的平台下重新建立，继续保证LabVIEW品牌在通用测试软件行业中的主导地位。

由于这次LabVIEW NXG的迁移是在两个兼容性很差的系统中完成，再加上三十年里积累的大量LabVIEW程序，这些程序又面临要部分重写或者全部重写的尴尬地步。人员离职变动使得本来就不易读懂的LabVIEW程序给迁移带来了更大的挑战。 所以这次LabVIEW NXG迁移的成本是非常昂贵的。唯一能够使得迁移顺利进行的方式是把这些成本分摊到一个很长的时间段，减少在每一个时间段内的成本。这是通常的做法，无可厚非。但无论如何，LabVIEW NXG迁移的绝大部分成本都是让LabVIEW 用户来付出的。这是一笔巨大的费用。

LabVIEW NXG值得迁移吗？

既然每个用户都要付出昂贵的迁移成本，是否值得迁移就是每一个用户都必须扪心自问的了。简仪提供一下几个参考问题供读者考虑：

(1) 您是否愿意自己再掏一笔钱再把自己绑定在一家供应商上？

(2) 当有其他更好的技术替代LabVIEW, 而且长期来说这些技术更能发挥您的专长，降低您的应用成本，增强系统的可维护性时，您是否考虑干脆用新的技术来替代LabVIEW。

我们接下来的分析是假设您不愿意绑定在一家的厂商之下，并有足够的自信和兴趣来探讨用新的技术来代替LabVIEW。

简仪科技认为以坦诚开放的胸怀来认真对待技术的发展和变化是一个工程技术人员应有的素质，而且是一个良好的素质。简仪寄希望广大的国内工程师有这样的胸怀，携手简仪一起探讨、开发、完善基于开源技术的下一代通用软件技术，彻底改变LabVIEW的垄断局面。

简仪科技分析调研微软.NET技术之后，决定直接使用微软原生态C#来开发下一代测控软件：锐视开发软件。简仪的平台以面向对象(OOP)的编程为基础。我们将基于OOP的虚拟仪器技术定义为虚拟仪器3.0。全新一代虚拟仪器技术即将为行业翻开新的一页，其开源免费的特性也在第一时间吸引到了业界的肯定和欢迎，并已经有很多工程师加入了“锐视开源软件项目”。后面章节将有锐视开发软件SeeSharp®和LabVIEW NXG的详细对比。

五：不值得迁移到LabVIEW NXG的依据

LabVIEW NXG是NI下一代的测试测量软件。 它继续沿用NI的图形化编程语言的软件设计，前面板仪器界面，后面板有图形化的编程界面风格。但在技术架构上LabVIEW NXG和LabVIEW2017有很大差异。差异的其中之一是LabVIEW NXG必须在微软的.NET上运行，这和简仪科技的SeeSharp®使用完全相同的底层技术。为了大家更好的理解基于.NET技术如何开发测控软件，以下的对比图可以直观说明。

图2

作为编译环境，LabVIEW NXG是在C/C++/.NET之上再封装了一个全新的编译环境IDE（IDE：编译环境的英文简称）。同样是利用.NET技术，简仪科技的SeeSharp®直接是对原生态微软C#调用，效率远优于多叠加了一层图形化IDE的LabVIEW NXG。

我们进一步对基于微软C#/.NET的SeeSharp®和NI LabVIEW的编译环境主要性能指标进行对比。

	微软 C#/.NET 之上的SeeSharp®	LabVIEW™ NXG
原生微软编译环境	是 (Visual Studio)	否 (LabVIEW G)
需要.NET技术支持	是	是
对于.NET技术的全面调用	可全面调用	LabVIEW™ NXG 自身决定 (NI有选择性的调用)
对于第三方硬件的开放性	完全透明架构	NI 决定
使用难易度	容易使用	小系统使用简单, 使用难度随着系统的复杂程度上升而上升
效率	高	比原生C#/.NET IDE 低(因为额外的编译层)
原生支持Python语言	支持 (通过微软Visual Studio环境)	不支持
扩平台支持(Linux, MAC OS)	相对简单	不确定
对于仪器的更多扩展支持	开发中	初期比较好用
费用成本	免费	收费
开源	是	否
对前序产品的支持	是	不兼容（根据目前NI官网信息）

微软Visual Studio编译器 对 NI Graphic编译器(G编译器)

微软Visual Studio编译器是微软平台下最主流和最重要的编译器。基本可以这么说只要在微软的平台下，它是最有效率的编译器了。VS支持C#, C/C++, VB, Python等多种语言。值得一提的是C#是微软首推的应用编程语言， 在C/C++, VB之前。C#的使用在全球排名第四，远在LabVIEW之前。LabVIEW NXG的G语言是用C/C++/C#再封装出来的IDE。仅从两点之间直线距离最短的原理出发，LabVIEW NXG的编译效率低于Visual Studio是一目了然的了。

Visual Studio的易用性已经不低于LabVIEW

使用LabVIEW NXG G语言的一个理由是：虽然又进行了一次封装，但封装后的LabVIEW NXG更加合适测试测量行当，所以编译效率的降低可以由使用效率补回来。这个论点在十多年前可能还站得住脚，但在今天已经站不住脚了。简仪科技提出了基于OOP的MACOs体系结构。在这个结构上的锐视开发软件SeeSharp®和LabVIEW NXG一样是专门为测试测量行业设计的。它引入了最新的软件工程技术，易用性、可维护性、注释性能都远在LabVIEW NXG之上。

Visual Studio的透明程度，和任何第三家硬件的连接都远远高于LabVIEW NXG

由于Visual Studio是为了所有Windows开发者和用户设计的，微软就必须保持Visual Studio的透明程度使得所有愿意使用微软产品的开发者都有相同的待遇。LabVIEW NXG则恰恰相反。从NI现在公布的信息来看，LabVIEW NXG控制第三方的接入程度，不是所有的.NET功能都在LabVIEW NXG可以使用。这也使得很多基于.NET的硬件无法顺利接入LabVIEW NXG系统，更无法和LabVIEW NXG无缝捆绑。相比而言， NI硬件和LabVIEW NXG的无缝连接使其在LabVIEW NXG环境下比任何其它厂商的硬件都占据无法逾越的优势。

开源软件的重要性

简仪科技提出的基于开源测试软件的概念对国内客户具有更重要意义。简仪科技对所有真诚愿意一起开发培养开源测试软件的单位和个人都可以提供源程序。简仪希望通过大家的共同的努力，以中国为起点，建立一个由众多家厂商和个人共同维护的免费、正版和开源测控软件。这就是锐视开发软件SeeSharp®的努力方向。在这一点上，基于开源平台的C#/.NET测试软件比LabVIEW NXG有更大的吸引力，也是符合中国国情的。

费用的考量

LabVIEW NXG不仅是收费软件，由于LabVIEW NXG占用的资源多，完成同样的处理需要更高配置的硬件，所以使用LabVIEW NXG的系统费用将远远高于基于原生态C#/.NET的锐视软件系统。

软件工程师职场发展的考量

Visual Studio/C#/.NET是Windows下最通用的IDE。工程师选择这个平台必定是最好的工具。无论是否在测试测量行当，使用微软的平台保证在可预见的将来里，VS/C#/.NET不会荒废。相反，可能会更加实用。对任何一位工程师，选择个人投资在微软或NI应该是一个比较简单的选择。简仪选择微软。

通过以上的对比，简仪阐述了不选择迁移至LabVIEW NXG的理由。仁者见仁，智者见智，希望读者能以平静的心态来看待简仪的阐述。

六：酌情处理从现有LabVIEW到下一代测试软件的迁移

作为正在使用LabVIEW或者有测控软件需求的您应该怎么做？

请先根据事实情况做您的尽调。重新确认您的需求和选择，得出您自己的结论。如果您认同简仪的市场策略，您可以选择和简仪一起创造最佳的基于开源的测控软件；您当然也可以选择迁移到LabVIEW NXG。本章向读者提供更多有关LabVIEW的信息，供读者在做决定是参考。

LabVIEW的生命周期

下图是NI官方网站披露LabVIEW生命周期:(可参考http://www.ni.com/zh-cn/shop/labview/upgrade.html)。一个LabVIEW软件从发布到主流支持会持续4年时间，基本上有最长7年的从发布到版本停止的时间。在主流支持期间内，LabVIEW会支持NI的新硬件。超出主流支持的长期支持只是针对极少数付了超长期支持费用的客户。对绝大数客户，过了主流支持期限的LabVIEW就需要更新。否则无法使用新的硬件，无法获得除网站公开资料外的技术支持了。

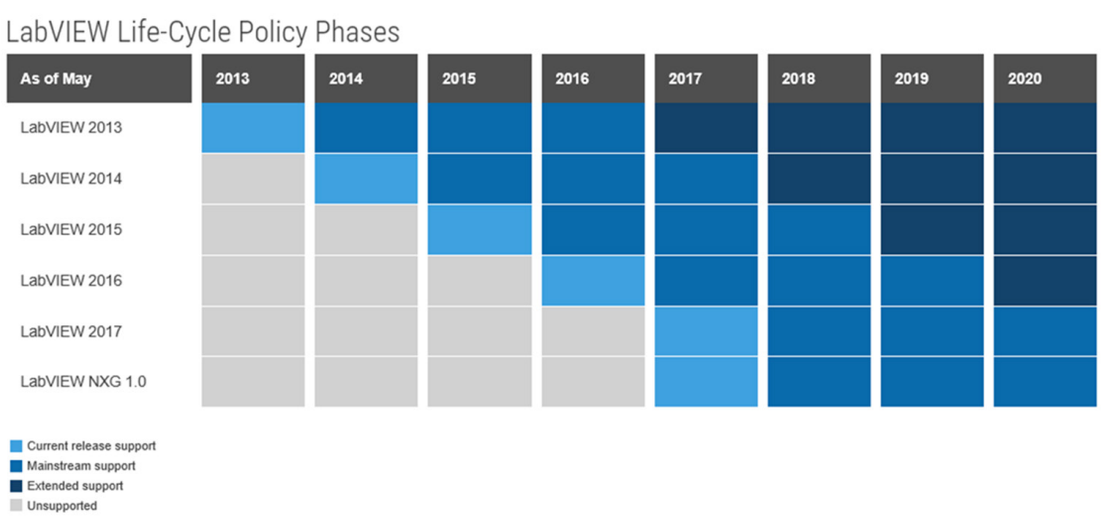


图3

根据事实情况做您的尽调

在本文的附录，我们附上了从NI官网获得的新版软件和硬件的兼容性信息。我们列出了LabVIEW NXG 1.0 或者 2.0 Beta里没有的而LabVIEW 2017有的功能特性。其中包括超过40项的硬件无法兼容和超过80项的软件无法兼容。读者应该仔细审视这些不兼容性对您的影响。

四个可能的选择

(1) 保持使用现有的LabVIEW不做任何改变

(2) 使用现有LabVIEW但开始升级到下一代LabVIEW

(3) 使用现有的LabVIEW但在新项目中开始用SeeSharp®锐视开发软件

(4) 立刻迁移到SeeSharp®开发平台上

选择1：继续使用现有版本的LabVIEW

这个选择适用情况：在未来的4至7年内您现有的LabVIEW软件、LabVIEW代码、LabVIEW支持的硬件和现有操作系统可以满足你的要求。这是一个很大的假设，技术发展一日千里，操作系统的变更，元器件造成的新老产品替代都可能影响以上假设。但您确认这是可以的假设或者您在7年内不再从事测试项目工作或准备退休，不做任何变动也不失为选择之一。但对绝大部分中轻年技术人员，这不可能是一个合理的选择。

选择2：迁移到LabVIEW NXG

如前所述，迁移到LabVIEW NXG很大可能会需要重写你现在LabVIEW 程序。这是一笔很大的开支。无论从技术角度和经济的角度，迁移到LabVIEW NXG都是不划算的。但是有很多时候，一个决定不单纯是由技术和经济来做的。比如以下的因素可能导致用户继续地使用LabVIEW NXG：对LabVIEW 单一品牌的青睐，担心在短期内由于使用新技术而对现有工作带来的风险和影响，无法说服只懂LabVIEW的领导，对学习新技术的抵触，担心新技术冲击已经获得的地位和利益等。这些因素也都是人之常情，无可厚非。

选择3：结束现有LabVIEW软件项目，在新项目上开始使用基于开源的锐视开发软件平台开发新的测试项目。

虽然您完全认可简仪的方法，但现实的工作由于项目时间紧迫，或者过去大量的LabVIEW代码使得您无法下决心立即全面切换到锐视平台。这是完全可以理解的。如果如此，您可以在完成结束现有LabVIEW项目的同时，在新项目上采纳建议全开放透明的C#/.NET原生态系统。简仪告知读者，您以前在NI硬件上的投资并不会浪费。一个全开放的系统当然可以支持也应该支持NI的硬件。如果您对这个选择有兴趣，简仪及简仪的合作伙伴可帮助您设计如何来完成从LabVIEW到C#/.NET的转换。请联系简仪科技获得更多细节。

选择4：立刻完全转向锐视开发软件平台

如果以下应用场景适用于您现在的情况，您应该立刻开始转换平台的工作。这个转换比您想象的要简单。在很多情况下，转换后您可以继续使用NI的硬件。一旦完成，您可以更好地掌握自己的项目“命运”。不再受一家厂商的制约。已经有很多客户完成了这个转变，简仪科技会帮助您完成这个转变。

七：致学生、工程技术人员、老师、企业经理、研究院负责人

作为学生，你的首要工作就是完成学业，掌握知识和工作的工具。工具在很大程度上决定了你今后的工作。简仪认为每一位学生应该尽量使用最通用的工具，而不是掌握把自己拴在某家厂商里的专有工具。现在的科技发展使得通用工具和很多专有工具一样的强大，无论你是否在测试行当工作，使用C#/.NET可以使你有更多的时间来考虑你的未来。你的选择不会因为你专业的改变而浪费。如果你有兴趣，可以浏览一下简仪的网站（www.jytek.com）或Wikipedia。读一下C#，Mono，.NET的发展史。你就知道学会C#使得你很容易在其它编程环境里上手。这是一举多得的！

作为工程技术人员，接受新的技术或者学一门新的技术是一个很大的挑战。任何一门新的技术都有从小到大的过程。简仪是一家小公司，我们推荐的SeeSharp®又想要做一件大事，难度当然很大。我们希望此文能使您见微知著，把握科技的动态。我们也相信会在你们中间找到很多知音，一起来完成这一工作。

老师曾经是一个受人尊敬的职位。一日为师，终身为父是对教师职业尊敬的写照。 一个老师能给学生最好的礼物就是教会学生学习的方法，接受新生事物的胸怀。技术在飞进，老师又有繁琐的日常工作，所以要求老师能了解最先进的科技也未免强人所难。简仪愿意和所有的老师一起认真探讨此文对大学教学带来的影响。简仪最新的教育产品将大幅度降低学生学习测量技术的成本。简仪也希望坐在领导位置上的老师能够支持年轻教师学习使用新的技术，不被老的技术所制约。

如果您是企业经理或研究所负责人，您一定了解测试测量对企业产品质量的重要性，也一定了解选定合适的测试平台对企业的重要性。如果您曾经是LabVIEW的用户，很有可能是LabVIEW帮助您逐渐走上了领导岗位。您可能已经没有时间再来负责日常的技术工作。对于新的技术平台，您可能不再愿意承担风险。这也是人之常情。简仪建议您和简仪联系，从LabVIEW转到C#/.NET的风险可能远远低于您的估计，回报可能远远超出您的想象。希望您积极支持您部门技术人员的创新工作。

开源，自主可控是国家的大政。过去，我们可能没有选择，所以不得不使用诸如LabVIEW的专有软件。现在我们有了选择，我们应该一起把这个选择做成世界上最好的通用测试软件平台。就像Linux操作系统一样，我们加起来的聪明智慧不是一家商业公司可以媲美的。如果能做到这一点，虽然软件是免费的，简仪科技也一定会是一家成功的商业公司。

结语和关于简仪

技术的变革在给测试测量领域带来了不确定性也同时带来了激动人心的机遇。 简仪科技在这个时刻帮助工业界更好的理解这个变化并且帮助客户尽快利用最好的技术。本文通过NI官方发布的信息，简仪对虚拟仪器技术的积攒，发表了对LabVIEW NXG迁移的看法和对用户的推荐。简仪认为，封装在Visual Studio, .NET之上的LabVIEW NXG已不再对用户带来实质上的好处，相反，它将把大多数客户进一步地约束在LabVIEW NXG之中。简仪建议所有LabVIEW的用户立即展开尽职调研，开始规划由LabVIEW向微软VisualStuidio/C#/.NET原生态IDE的迁移。一旦完成迁移，客户将可以在最公开透明的IDE系统里使用包括简仪，NI的硬件。

关于简仪科技：

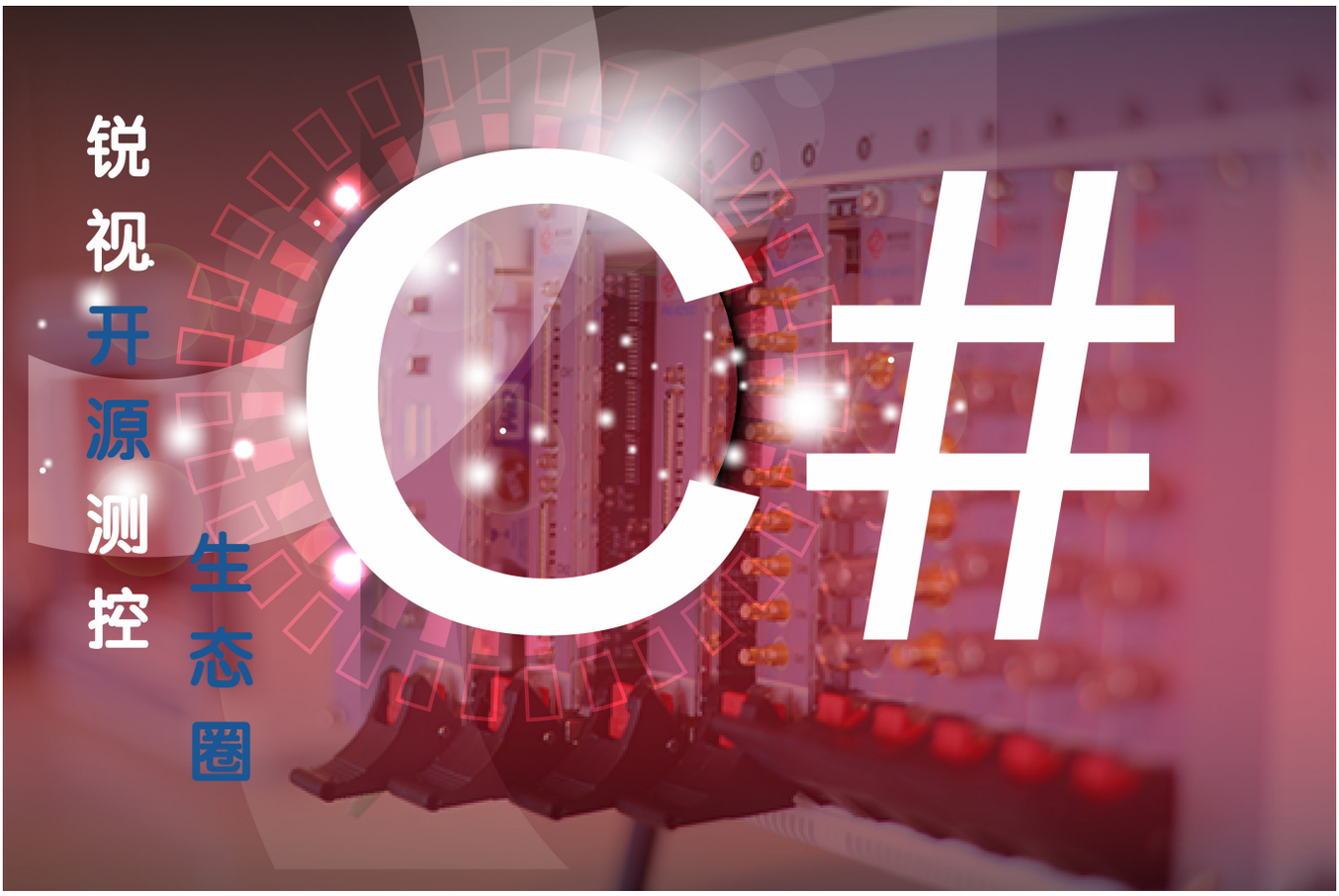
简仪科技是由曾参与过LabVIEW 1.0的开发工作的部分原NI大中华区资深员工，业界资深测试测量工程师和凌华科技（中国）测试测量产品事业部重组成立的本土测试测量公司。简仪科技的很多员工过去从LabVIEW获益匪浅。但在科技日新月异的今天，因循守旧，墨守成规已经不能满足简仪员工的志向。我们钦佩LabVIEW当年对通用仪器软件做出的巨大贡献，我们也看到了日益封闭的LabVIEW给我们行业带来的新机遇。我们提出了全球第一个开源测试平台：锐视测控平台®，整合凌华科技高质量的测试测量硬件产品，包括PXI模块化仪器和DAQ数据采集产品，总线形式覆盖PXI/PXIe、PCI/PCIe和USB，采样频率从低频到中高频不等。我们着重打造适合中国国情和工程师诉求的开源、高效的软件平台，同时也是国内唯一提供软硬件无缝结合的解决方案的测控公司。简仪科技真正打造了一个本土的高科技测试测量品牌，全面提升了中国的测试测量技术等级，为国家的发展做出力所能及的贡献。

关于锐视开源生态圈和锐视开源测控项目：

鉴于简仪科技提出的锐视开发软件SeeSharp®是一个完全基于开源的软件，我们进一步提出了锐视开源测控项目，简仪科技将打造全球第一个基于开源软件技术的测控软件项目，而且希望相关领域的工程师都来贡献基于锐视软件开发标准和规范的开源代码，一起打造属于测控工程师的“Linux”。具体项目详情请浏览www.jytek.com。相信在大家的支持和帮助下，锐视开发软件作为一个开源的测试测量平台不但可以变成主流而且会变成最好的软件产品。

关于此白皮书的PDF和简要版PPT，可以在简仪网站下载
http://www.jytek.com/whitepaper20170808_labview_nxg

或关注微信公众号:shjianyi 获取更多内容。



免责声明

本白皮书发布自简仪科技的调查研究之后。简仪科技中国对于信息的准确性负责。

本白皮书引用的NI官方2017年7月8日的信息。简仪科技建议和结论是严格按照以下得出：所有推论是根据NI披露的信息做出的合理解释和描述。此外简仪也通过自己对LabVIEW的了解及咨询其它一些资深的LabVIEW用户来旁证简仪的结论。简仪不排除由于NI对LABVIEW NXG计划会做出调整而得出不同于本文的结论。

附录 1：LabVIEW到LabVIEW NXG的兼容问题

所有信息直接来源于NI的官方网站 http://www.ni.com/zh-cn/shop/labview/compare-labview-nxg-and-labview.html

除了NI官方公布的不兼容细节外，简仪科技也在第一时间做了简单的兼容性测试并找到了LabVIEW 2017中的例子不能顺利升级兼容到LabVIEW NXG版本。NI公布的清单和简仪自己的尽职调研使得我们得出结论：从LabVIEW 2017升级到LabVIEW NXG需要付出非常巨大的工作，其中就包括重写您现有的LabVIEW老代码。

LabVIEW NXG和LabVIEW不兼容的硬件清单：

AS OF July 5, 2017. Source: http://www.ni.com/zh-cn/shop/labview/compare-labview-nxg-and-labview.html	LabVIEW 2017	LabVIEW NXG 1.0	LabVIEW NXG 2.0 (Beta)
Explore the product	Download Evaluation	Download Evaluation	Visit Software Technology Preview
Hardware Support Summary			
Deploying to NI real-time hardware (LabVIEW Real-Time Module)	YES	NO	NO
Deploying to NI FPGA-based hardware (LabVIEW FPGA Module)	YES	NO	NO
CompactRIO platform	YES	NO	NO
Software defined radios	YES	LabVIEW Communications	Partial
Engineering education hardware	YES	Partial	Partial
PXI platform	YES	Partial	Partial
CompactDAQ platform	YES	Partial	Partial
NI data acquisition hardware	YES	Partial	Partial
Data Acquisition and Control Hardware Detailed Support List			
PXI Multifunction Reconfigurable I/O Modules and Multifunction Reconfigurable I/O Devices (R Series)	YES	NO	NO
Ethernet CompactDAQ Chassis (cDAQ-9185/89)	YES	NO	YES
CompactDAQ Controllers	YES	NO	NO
CompactRIO Controllers	YES	NO	NO
CompactRIO Single-Board Controllers	YES	NO	NO
C Series Modules (NI-RIO driver support)	YES	NO	NO
Vehicle Communication Buses—CAN/LIN/FlexRay (PXI/PXIe)	YES	NO	YES
PXI PROFIBUS Interface Module	YES	NO	NO
PXI CANOpen Interface Module and PXI DeviceNet Interface Module	YES	NO	NO
I2C/SPI Interface Device	YES	NO	NO
Industrial Controllers	YES	NO	NO
Smart Cameras	YES	NO	NO

Compact Vision Systems	YES	NO	NO
PXI Frame Grabber Module	YES	NO	YES
Frame Grabber Reconfigurable Devices (1473R/8273R)	YES	NO	NO
PXI Motion Control Module	YES	NO	NO
Electronic Test and Instrumentation Hardware Detailed Support List			
PXI Programmable Resistor Modules (PXI-272x)	YES	NO	NO
Digital Waveform Instruments	YES	NO	NO
Deploying to NI real-time hardware (LabVIEW Real-Time Module)	YES	NO	NO
Virtual Bench	YES	NO	NO
Wireless Design and Test Hardware Detailed Support List			
PXI Vector Signal Analyzers	YES	NO	YES
PXI Vector Signal Generators	YES	NO	YES
PXI RF Analog Signal Generators	YES	NO	YES
PXI Vector Signal Transceivers (NI-RFSA/RFSG support)	YES	NO	YES
PXI Vector Signal Transceivers (FPGA extensions)	YES	NO	NO
RF Power Meters	YES	NO	YES
PXI RF Amplifiers/Attenuators	YES	NO	YES
PXI Vector Network Analyzers	YES	NO	YES
mmWave Transceiver System	YES	NO	NO
Ettus Research USRP2	YES	NO	NO
Ettus Research N200/210	YES	NO	
Ettus Research X300/310	YES	NO	NO
All other NI software defined radios	YES	LabVIEW Communications	YES
ATCA-3671	YES	NO	NO
Engineering Education Hardware Detailed Support List			
myRIO	YES	NO	NO
NI ELVIS	YES	NO	NO
Analog Discovery	YES	NO	NO

LabVIEW NXG和LabVIEW不兼容的软件清单：

AS OF July 5, 2017. From NI: http://www.ni.com/zh-cn/shop/labview/compare-labview-nxg-and-labview.html	LabVIEW 2017	LabVIEW NXG 1.0	LabVIEW NXG 2.0 (Beta)
Explore the product	Download Evaluation	Download Evaluation	Visit Software Technology Preview
LabVIEW Module and Toolkit Summary			
LabVIEW Real-Time Module	YES	NO	NO
LabVIEW FPGA Module	YES	NO	NO
Vision Development Module	YES	NO	YES
LabVIEW Datalogging and Supervisory Control Module	YES	NO	NO
LabVIEW Report Generation Toolkit	YES	NO	NO
LabVIEW Database Connectivity Toolkit	YES	NO	NO
LabVIEW Application Builder Module	YES	NO	YES
LabVIEW Advanced Signal Processing Toolkit	YES	NO	NO
LabVIEW SoftMotion Module	YES	NO	NO
LabVIEW Control Design and Simulation Module	YES	NO	NO
LabVIEW Sound and Vibration Toolkit	YES	NO	YES
Programming Environment Details			
Object-oriented programming	YES	NO	YES
Configuration-based VIs (Express VI)	YES	NO	NO
Dynamic control of UI elements, VIs, and the environment (VI server)	YES	NO	YES
Custom editing of UI control appearance	YES	NO	NO
Custom editing of UI control behavior (Xcontrols)	YES	NO	NO
Ability to control programs remotely using remote front panels	YES	NO	NO
NI Software Interoperability			
TestStand integration	YES	NO	YES
Switch Executive API (Basic)	YES	NO	NO
Switch Executive API (Configuration)	YES	NO	NO
Requirements Gateway	YES	NO	NO
VeriStand integration	YES	NO	NO
Vision Builder for Automated Inspection integration	YES	NO	NO
Third-Party Software Interoperability and Code Integration			
Call .NET assemblies	YES	NO	NO
Software Engineering Tools			
Graphical diff and merge	YES	NO	NO
Dynamic code analysis (LabVIEW Desktop Execution Trace Toolkit)	YES	NO	NO
Static code analysis (LabVIEW VI Analyzer Toolkit)	YES	NO	NO

LabVIEW Unit Test Framework Toolkit	YES	NO	NO
Desktop UI Controls			
Basic table	YES	NO	YES
ActiveX and .NET containers	YES	NO	NO
3D graph and picture	YES	NO	NO
Mixed signal graphs	YES	NO	NO
List box	YES	NO	YES
Picture ring	YES	NO	NO
Tree	YES	NO	NO
Subpanel	YES	NO	NO
Picture control	YES	NO	NO
Industrial automation controls (LabVIEW DSC Module)	YES	NO	NO
Code Deployment and Distribution			
Ettus Research X300/310	YES	NO	NO
All other NI software defined radios	YES	LabVIEW Communications	YES
ATCA-3671	YES	NO	NO
Engineering Education Hardware Detailed Support List			
Build application (.exe)	YES	NO	YES
Build shared libraries (.llb, .lvlib,.lvlibp equivalents)	YES	NO	YES
Build DLL	YES	NO	NO
Build .NET assembly	YES	NO	NO
Build distributions	YES	NO	YES
Signal Processing and Control			
Fuzzy logic (LabVIEW PID and Fuzzy Logic Toolkit)	YES	NO	NO
Wavelet analysis (LabVIEW Advanced Signal Processing Toolkit)	YES	NO	NO
Time series analysis (LabVIEW Advanced Signal Processing Toolkit)	YES	NO	NO
Digital filter design	YES	NO	NO
Vision analysis (Vision Development Module)	YES	NO	YES
Motion control libraries (LabVIEW SoftMotion Module)	YES	NO	NO
Model-based control and simulation (LabVIEW Control Design and Simulation Module)	YES	NO	NO
LabVIEW Sound and Vibration Toolkit	YES	NO	YES
Modulation Toolkit	YES	NO	YES
NI-RFmx	YES	NO	YES
WLAN Test Toolkit	YES	NO	NO
Bluetooth Test Toolkit	YES	NO	NO
GNSS Test Toolkit	YES	NO	NO
FM/RDS Test Toolkit	YES	NO	NO

Read, Write, Share Data			
Visualize and analyze image files (Vision Development Module)	YES	NO	YES
LabVIEW Report Generation Toolkit	YES	NO	NO
LabVIEW Database Connectivity Toolkit	YES	NO	NO
LabVIEW DataFinder Toolkit	YES	NO	NO
FTP API	YES	NO	NO
Data Communication			
Static Shared Variable migration	YES	NO	NO
Creating LabVIEW-based web services	YES	NO	NO
HTTP API	YES	NO	YES
OPC UA (LabVIEW Real-Time or DSC module)	YES	NO	NO
Modbus (LabVIEW Real-Time or DSC module)	YES	NO	NO
EtherNet/IP	YES	NO	NO
EtherCAT	YES	NO	NO
Power utility protocols (IEC 61850, DNP3, IEC 60870-5)	YES	NO	NO
Automotive Diagnostic Command Set Toolkit	YES	NO	NO
ECU Measurement and Calibration Toolkit	YES	NO	NO

附录 2：参考资料

(1) 常用编程语言调查。C#排名第四，MatLab排名18，LabVIEW榜上无名。
最常用的编程工具：Visual Studio排名第一。
https://insights.stackoverflow.com/survey/2017#technology-programming-languages

(2) .NET主页：https://www.microsoft.com/net/

(3) Visual Studio主页：https://www.visualstudio.com/zh-hans/

(4) .NET发展历史：https://en.wikipedia.org/wiki/.NET_Framework#History

(5) .NET跨平台的应用，Mono, Xamarin：https://en.wikipedia.org/wiki/.NET_Framework#History

(6) 开源软件的励志故事

世界顶级的操作系统Unix和Linux是建立在这三项伟大的开源工作之上：GNU, Berkeley BSD, LINUX。除了Windows以外，MacOS, Android等都是在这些开源程序发展出来的。以Richard Stallman, Bill Joy, William Jolitz, Lynne Jolytiz, Linux Torlvads为代表的数以万计的软件工程师，致力于开源Unix/Linux的开发，把源于AT&T贝尔实验室的商用Unix转变成了今日的Unix/Linux。

https://en.wikipedia.org/wiki/FreeBSD#History

https://en.wikipedia.org/wiki/GNU_Project

https://en.wikipedia.org/wiki/Linux