

把 UNIX 应用程序移植到 Linux

提示和技巧

IBM

2001 年 10 月 01 日

您知道 Linux? 现在已是世界第二的服务器操作系统吗? 今天, 您的应用程序在 Linux 上运行吗? 如果不是, 那么本文将给您些提示和技巧, 在您把应用程序移植到 Linux 的时候为您提供指导。本文依据的是软件工程师的经验, 这些工程师来自独立软件供应商 (Independent Software Vendors, ISV) 和全世界的 IBM 实验室, 他们参与了许多把应用程序移植到 Linux 的项目。

介绍

您知道 Linux 现在已是世界第二的服务器操作系统吗? 今天, 您的应用程序在 Linux 上运行吗? 如果不是, 那么本文将给您些提示和技巧, 在您把应用程序移植到 Linux 的时候为您提供指导。本文依据的是软件工程师的经验, 这些工程师来自独立软件供应商 (Independent Software Vendors, ISV) 和全世界的 IBM 实验室, 他们参与了许多把应用程序移植到 Linux 的项目。

首先我们描述 Linux 的特征和它在开发和管理应用程序中的具体价值。接着, 我们将回顾重要问题, 给出包含移植的主要步骤的移植指南。最后, 我们以 IBM ~ zSeries™ 为例, 为您演示从一个 Linux 平台移植到另一个平台是多么的容易。(zSeries 这个词指 S/390 和 zSeries。)

Linux 的特征和价值

开放性意味着应用程序灵活性: Linux 是真正的开放系统, 这使应用程序的可移植性成为现实。在很多情况下, 一次简单的重编译就能把应用程序带入 Linux。在优化您的开发和部署平台时, 这种快速移植给您真正的灵活性。

多种硬件支持意味着平台灵活性: Linux 几乎可在所有已知的处理器上运行, 无论是 RISC 还是 CISC, 32 位还是 64 位。它涵盖全部类型的计算机, 从大型机服务器 (IBM ~ zSeries) 到 PowerPC、Sun UltraSPARC、Alpha、PA-RISC 和 Intel 处理器, 往下还有个人数字助理和嵌入式系统。特别是所有的 IBM ~ 品牌服务器都支持 Linux。

源代码的可用性意味着自己动手的灵活性: 与多数以二进制形式发行的商业软件不同, Linux 内核的源代码是可以获得的。您可以拿到源代码, 然后修改和重编译它, 以满足您的特定要求。调试和解决问题或增加新的特性不再依赖任何供应商。您可以自己作出需要的更改, 只要它们符合 GNU General Public License 的规定。

严肃的设计意味着安全性：Linux 安全性是从内核上构建的，由于实现它的代码要接受成千上万的开发者的检验，所以弱点会被及早地发现。Linux 中建有許多路由和防火墙功能。

流行意味着存在技能：因为 Linux 在大学和学院中流行，所以多数新的毕业生有 Linux 经验。您几乎可以在任何城市的 Linux 用户组中找到该地区能够熟练应用 Linux 的人，而 Web 站点和因特网新闻组仍然是 Linux 使用指南信息和非正式技术咨询的一种宝贵而有效的来源。

开始前需考虑的重要问题

移植到 Linux 可能是容易的、简单的和直接的，特别是当您的 UNIX 应用程序的编写遵循公共的开放的标准。如果您认为移植到 Linux 是有吸引力的，那么您需要分析移植中涉及的潜在花费和风险及减小它们的方法。

迁移会涉及巨额初始投资吗？

成本 — 移植将涉及巨大的时间、人员和金钱的初期投入吗？该项目会冻结所有的其它新工作并消耗整个团队的资源吗？需要提前支付大量资金费用和再培训费用吗？这是不是冒着极大风险的提议：一旦开始，要么完成，要么完全失败？

移植到 Linux 是可管理的 — 本文的移植指南部分讲述怎样一步一步地移植而每个中间步骤本身是稳定的。这使您能根据您的优先级管理成本、人员和项目。这意味着您可以独立完成每一步，在完成每步后重新估计您的优先级和目标。这样的结果是风险更容易被控制而给您的公司带来的影响更小。

成本有多大？需要花多少时间？

花费的时间和金钱 — 根据和移植有关的因素，全面估计您的应用程序：

- 编译器的差异
- 您的代码中与硬件相关的构造（例如字的长度或字节顺序的相关性）
- 平台运行时服务
- 构建工具的相关性
- 数据库、联网和消息传递中间件的可用性
- 用户界面的可移植性
- 测试案例和测试环境

移植的工作量和复杂程度直接和依赖系统和环境的代码的多少成正比。如果您的应用程序只用标准的语言构造和标准的库，那么移植起来相对容易些。例如，Java 应用程序常常属于这一类。反之，如果您的应用程序是一个 C 程序，其中用到 Solaris 上不符合 POSIX 的服务或依赖在 Linux 上没有的第三方产品，那么移植难度将大大增加。通常，系统测试、配置的测试和软件的安装是重要步骤并且是移植的主要部分。

我的应用程序还能在原来的 UNIX 平台上运行吗？

缓解 — 即使把开发转移到 Linux 后，您仍可以保持原来的平台选择，以满足您的其它市场。移植到 Linux 通常意味着构建工具被 GNU 工具替代，平台相关的线程库被符合 POSIX 的替代。在其它 UNIX 平台上，GNU 工具和库都能用，使您能同时兼顾您原来的平台和 Linux。

C 和 C++ 应用程序的移植指南

以下技术讨论来自 Solaris to Linux porting guide (1)

(ibm.com/developerworks/linux/library/l-solar/), 这些讨论也可被用于其它主流 UNIX 系统。

Linux 操作系统、构建工具和多数语言的编译器可免费获得并可从因特网上下载。为了开始移植, 您将需要提供硬件和开发人员:

- 第 1 步 — 下载所需的开发工具和 Linux 分发版。
- 第 2 步 — 改用 Linux 开发工具但仍然运行 Solaris。这种方法使得转变更平缓。
- 第 3 步 — 熟悉 Linux 操作系统但仍旧在 Sun 硬件上运行。
- 第 4 步 — 把应用程序移到目标 Linux 平台。

第 1 步: 下载

- 以下列出的是所需的编译器、构建工具和相关的在线文档链接:
- GCC (GNU Compiler Collection) 工具套件包含一个 C 编译器和一个 C++ 编译器
gcc.gnu.org/releases.html
在线文档
gcc.gnu.org/onlinedocs/gcc-2.95.2/gcc_toc.html
- Solaris GNU 编译器和 make (gmake) 实用程序
www.sunfreeware.com/
在线文档
docs.sun.com/gnu.org/manual/make-3.79.1/html_mono/make.html
- Linux 可从以下 Linux 发行商那里获得。
Caldera www.caldera.com/
Red Hat www.redhat.com/
SuSE www.suse.com/
Turbolinux www.turbolinux.com/
— Linux 的 UltraSPARC 版本 Sun www.sun.com

第 2 步: 为 Solaris 上的 Linux 构建您的 C/C++ 应用程序

安装工具 — 先在 Solaris 上安装 GNU 工具。由于 GNU 编译器和 make 实用程序在 Solaris 和 Linux 上都能用, 所以使用这些工具来尝试初次构建您的 Linux 应用程序相对较容易。

转换 makefile — 构建您的应用程序时请用 GNU gmake 实用程序, 别用 Solaris make 实用程序。根据您的 makefile 中用到的构造, 您可能遇到 GNU gmake 实用程序产生的错误消息, 因为 GNU gmake 实用程序和 Solaris make 实用程序是有区别的。请用 make 工具文档来帮助找出 makefile 中有问题的地方, 然后对 makefile 作出所需的调整。

编译和调试 — 当您通过修改您的 makefile、使之能被 gmake 使用后, 把 makefile 中调用 C 编译器的名字 cc 改为 gcc, C++ 编译器的名字 CC 改为 g++。然后编译该应用程序。您在构建中所遇到的错误消息, 如果有的话, 可被分成两类: 命令行选项问题和代码问题。

找出由于 GNU 和 Sun 编译器所接受的命令行选项有所不同而产生的消息。除了几个基本选项（例如“-c”和“-g”），两个编译器能接受的多数选项是不同的。请参阅编译器文档，修改您的 makefile，以适应编译器选项间的区别。

下一步，您可以处理剩下的与代码有关的错误消息和警告。请参阅编译器文档，理解并解决由于编译器之间的差别而产生的问题。

第 3 步：在 Linux 的 Sun UltraSPARC 版本上构建和测试您的应用程序。

如果所移植的应用程序依赖特定的 UltraSPARC 硬件，那么这是移植过程中的重要阶段。Solaris 开发者有机会熟悉 Linux 环境而不必放弃下层的 UltraSPARC 硬件，也不必修改应用程序中特定于硬件的部分。

安装 Linux 环境 — 在 Sun UltraSPARC 上，安装 Linux 和 GNU 实用程序。

编译并调试运行时 API — 因为您在构建您的应用程序时用的是 GNU 实用程序，所以您在这步看到的区别限于 Solaris 和 Linux 之间的运行时应用程序编程接口（application programming interfaces, API）的区别。如果您想了解这些区别的更多信息，请参阅运行时接口部分（第 7 页）。对应用程序作出所需的调整。

测试 — 当应用程序被完全地重新构建后，进行彻底的验证测试。

第 4 步：在目标 Linux 平台上构建和测试您的应用程序。

安装 Linux 环境 — 在目标硬件上安装 Linux 和 GNU 工具。

复制并重新构建应用程序 — 把您的源代码树和 makefile 复制到新的 Linux 机器上并重新构建应用程序。如果您没完成第 3 步，那么您可能需要修改应用程序，以适应 Solaris 和 Linux 之间运行时 API 的区别。最后，如果您的应用程序包含任何特定于 UltraSPARC 的代码，那么，请作出所需的修改，以解决那部分代码的问题。

修改安装例程，以适应目标 Linux 平台。各种 Linux 平台（分发版）的配置和启动文件的位置稍有不同。

测试 — 在结束移植前，对移植的应用程序运行彻底的验证测试套件。另外，在目标 Linux 平台上测试配置和安装。

警告！许可证和版权条款 — 当您移植到 Linux 并把您的应用程序和开放源代码的库链接时，请注意和这些库有关的许可证和版权。至少来说，您新移植的 C/C++ 应用程序将被链接到 GNU C/C++ 运行时库。这些库受 GNU Lesser General Public License 的保护。如果您移植的软件是专利性质的并且您想在用 GNU 工具构建该软件后出售它，那么谨慎的作法是完全理解 GNU General Public License 和 GNU Lesser General Public License（www.gnu.org/licenses/licenses.html）中列出的条款和条件。如果您的软件和受 GNU 许可证保护的库链接，那么 GNU 许可证中的某些条款和条件将对您的软件也适用。

Java 应用程序的移植指南

- 第 1 步 — 下载所需的开发工具。
- 第 2 步 — 在目标 Linux 平台上使用应用程序。

第 1 步：下载

把用 Java 写的应用程序的开发从 Solaris 移到 Linux 是很简单的。您移植任务所需的工具有：

- Java 2 Developer Kit 的 Linux 版包括重建的 Java 虚拟机 (Java Virtual Machine, JVM)，该 JVM 有改进的即时 (just-in-time) 编译功能 ibm.com/developerworks/java/jdk/linux130/
- GCC 工具套件包含一个 Java 编译器 (GCJ) gnu.org/software/gcc/java/

第 2 步：在目标 Linux 平台上使用应用程序

无论 JVM 运行在什么操作系统上，JVM 接受同样的字节码，所以您可以选择在某个操作系统中编译 Java 源文件，然后在另一操作系统中运行生成的类文件。例如，如果 Linux 是您的主要开发平台，那么您在 Linux 上创建的 Java 类文件能同样好地运行在 Solaris 或其它有兼容 JVM 的平台上。

运行时接口

尽管 Linux 和 Solaris 的绝大部分运行时接口是相同的，但有些地方还是有区别的。在您的应用程序中，每处用到 Linux 上没有或不完全相同的 Solaris 接口的地方都需要被修改，否则您的应用程序无法被正确地构建。以下列出的是有区别的地方。

系统调用、C 库和 C++ 库

API 差别 — Solaris 内核提供逻辑卷支持、文件的访问控制表 (access control list, ACL) 的管理和系统审计日志功能。Linux API 中的这些功能 (如果有的话) 与 Solaris 上的并不相同。还有，在缺省情况下，Linux 不支持 STREAMS。在网络中用到 STREAMS 的应用程序需要改用 POSIX 套接字。另外，不少 Linux 上也有的 Solaris API 使用不同的返回和参数类型，或在不同的头文件中声明。但是，这些差别往往不大，通常不需要在应用程序中修改代码。

C++ 库 — Solaris Forte 6 C++ Compiler 包含三个类库：复数 (-compat=4)、经典 iostream 和 Standard C++ Library (它包含标准的复数和 iostream)。GCC 只包含 Standard C++ Library。如果所移植的应用程序使用 -compat=4 复数库或经典 iostream 库，那么应用程序需改用包含在 Standard C++ Library 中的那些库的标准版本。作这种修改的额外好处是提高了应用程序的可移植性。

桌面：CDE 对 GNOME/KDE

图形用户界面 (graphical user interface, GUI) — Solaris 的缺省桌面 Common Desktop Environment (CDE) 并不包含在常见的 Linux 分发版中。对于要求 CDE 的应用程序或偏爱这种特别的 GUI 环境的用户，CDE 的 Linux 版本可从 Xi Graphics (www.xig.com) 以他们的 DeXtop 产品的形式获得。在 Linux 上运行 DeXtop 使 Solaris 用户置身于熟悉的 GUI 环境从而使他们很容易地适应转移到 Linux 所带来的变化。

DeXtop 要求使用也是由 Xi Graphics 提供的 Accelerated-X Linux X-Server。尽管 XFree86 Linux X-Server 可以和 DeXtop 一起运行，但是 Xi Graphics 并不正式支持在 XFree86 上运行 DeXtop。

线程和进程管理

线程 / 轻量级进程 (Light Weight Process, LWP) 支持 — Solaris 支持 POSIX 线程和特定于 Solaris 的线程模型。如果所移植的应用程序使用特定于 Solaris 的线程 API, 那么, 最好修改应用程序, 使其改用 POSIX 线程。POSIX 线程的可移植性更好而且 Linux 也支持它。

进程管理: /proc 文件系统 — /proc 伪文件系统提供了访问内核数据结构的方便的方法。在 Solaris 上, /proc 包括关于活动的进程和线程的信息, /proc 还提供控制这些进程和线程的接口。Solaris 上的 /proc 控制接口通常被调试器用来跟踪程序的执行。在 Linux 上, /proc 并不提供进程控制接口。在 Linux 上调试时, 为了控制进程, 请用 ptrace() API 而不是 /proc。

运行时要考虑的因素

系统管理: 一般地说, 管理 Linux 和管理任何其它 UNIX 操作系统非常相似。但在某些命令和任务的细节上是有差别的。IBM 和 UNIXGuide (unixguide.net/unixguide.cgi) 已出版了比较图表, 该图表不仅总结了 Solaris 和 Linux 之间的一些差别, 还总结了 AIX 和其它版本的 UNIX 之间的差别。有经验的 Solaris 系统管理员不会遇到多大困难就能得心应手地用 Linux。

其它第三方工具、实用程序和库: 许多 Solaris 上常见的流行的第三方工具、实用程序和库在 Linux 上也有, 例如:

- Rational Rose 和其它 Rational 产品在 Solaris 和 Linux 上都能用。
- Rogue Wave SourcePro C++ 库在 Forte/Solaris 上能用, 在 GCC/Solaris 和 GCC/Linux 上也能用。
- Trolltech Qt C++ GUI 框架在 Solaris 和 Linux 上都能用。

由于 Linux 越来越受到欢迎, 支持 Linux 的第三方供应商的数量在不断增加。

尾数格式 — Sun SPARC 和 UltraSPARC 处理器用大尾数法格式存储整数。如果您想把应用程序移植到使用小尾数法的 Linux 版本, 例如 Linux/Intel, 那么您必须解决应用程序中所有的尾数格式的相关性, 以使应用程序能正确运行。另一种方法是把应用程序移植到 Linux/PowerPC、zSeries 上的 Linux 或 Linux/UltraSPARC (它们都是大尾数法平台) 而不是 Linux/Intel。

从 Linux 移植到 zSeries 上的 Linux

现在移植到一个平台上 (比如说 Intel) 的 Linux 已完成, 那么移植到其它硬件平台上的 Linux 怎样呢? 简短的回答是: 这很容易。

由于以下两个原因, 我们选择以 zSeries 为例。

1. zSeries 是容易地从一种 Linux 移植到另一种 Linux 的好例子, 因为 zSeries 体系结构和 Intel 体系结构完全不同。许多 ISV 先把他们的 UNIX 应用程序移植到 Intel 上的 Linux, 这当然是自然的选择。接着再把应用程序移植到 zSeries 上的 Linux 就不需要太多的努力。
2. 由于 zSeries 在虚拟化领域的独特功能及其硬件可靠性和可用性, 所以, 对您来说, zSeries 可能是有吸引力的平台。

为什么要移植到 zSeries 上的 Linux?

zSeries 上的 Linux 是纯粹的 Linux：它不是现有的 zSeries 操作系统上特别的 Linux，也不是为 zSeries 体系结构而修改的 Linux 的特别版本。zSeries 上的 Linux 的特征与其它平台上的 Linux 的特征相同，例如它是纯粹的 ASCII 环境。对于所有的体系结构，绝大部分的 Linux 结构是相同的。和 zSeries 相关的修改使 Linux 能和 zSeries 内存、zSeries 磁盘及通信硬件进行通信。Linux 和应用程序的接口及 Linux 和用户的界面并不受到影响。

应用程序的优点 — 您在 zSeries 上的 Linux 中的 Linux/UNIX 应用程序使您能访问存储在 zSeries 环境中的企业数据（后端集成）。这样提高了响应速度，减少了不必要的数据复制。zSeries 出众的容量、可伸缩性、可靠性、可用性和安全性使之成为企业服务器应用程序的完美部署平台。另外，在商务环境中，zSeries 上的 Linux 减少了服务器的数量，从而简化操作并降低成本。

zSeries 上的 Linux 移植提示和技巧

这些提示和技巧是由一个 IBM 技术小组总结的，该小组负责把应用程序移植到 zSeries 上的 Linux。该文档的全文可从 ibm.com/servers/esdd/articles/linux_s390/index.html 获得。

小尾数法到大尾数法

S/390 是大尾数法系统。如果代码要处理由小尾数法系统上产生的面向字节的数据，那么这些代码可能需要某种字节对换。这些数据可能需要被重新生成，如果这不可能（例如，共享的文件），那么，应用程序需被修改，以适合处理小尾数法数据。如果原来的平台是大尾数法平台（例如 UltraSPARC、IBM ~ pSeries™……），那么就不存在这个问题。

汇编代码

汇编代码需用 S/390 汇编器重写。所有的操作码需改成 S/390 操作码。如果应用程序代码使用汇编头文件，那么就需要该头文件的 S/390 版本。S/390 汇编代码的 Linux 版本使用 S/390 操作码但遵守 GNU 汇编器的语法约定。您可以从 www.gnu.org/manual/gas-2.9.1/as.html 下载 GNU 汇编器手册。

绝对地址和高阶位

出于各种目的，一些应用程序使用硬编码地址。一个可能的目的是在内存中定义固定的页，用于分配或用 mmap() 来内存映射。mmap() 调用的一种形式允许页固定。在这种模式中，mmap() 试图使用在所请求的地址的存储器来内存映射。在 Intel 平台上，这个地址可能被指定在 zSeries 上的 Linux 无法使用的地方。

每个平台很可能选择不同的程序堆栈位置、系统库、堆等等，所以在移植到其它系统的时候，硬编码地址往往是糟糕的选择。zSeries 寻址方式还忽略高阶位，所以硬编码地址 0x80000000（高阶位是 1）将被转换成 0x00000000，而这是 zSeries 上的保留地址。在生成地址时，对地址的算术运算也能使高阶位变为 1。

如果应用程序因使用绝对地址而产生段违例或其它错误，那么必须修改它。/proc/<pid>/map 显示一个活动的进程是怎样使用它的寻址空间内的存储范围的。如果代码必须使用绝对地址，那么这个 map 可被用来查找还未被保留的地址范围。

某些地址字段的高阶位可能要用“AND”来变为 0（31 位模式位）。如果该地址被用于算术运算，那么您必须这样做。

应用程序开发工具

部署应用程序要求在目标服务器上有一套健壮的工具。在 ibm.com/servers/eserver/zseries/os/linux/ldt/ 的 Web 站点描述了将部署到 zSeries 上的 Linux 的 C/C++ 应用程序的开发或移植的几个阶段并给出在每个阶段中有用的工具列表。

杂项

ptrace 和返回结构 — ptrace 和返回结构的使用与体系结构相关。

Configuration/build/Makefile 脚本或文件 — 很可能需要增加对 S/390 平台的支持。

proc 文件系统 — 有些区别，例如：

```
/proc/cpuinfo #####  
/proc/interrupts #####  
/proc/stat ### INTR ##
```

char 是无符号的 — 在 zSeries 上，字符类型 char 被缺省地解释为无符号的 char，而在其它平台上，char 被解释为有符号的 char。

Va_args — 在 Linux 的 zSeries 版本中，va_list 的定义与其它平台的不同。如果您的应用程序简单地使用一个 C 赋值语句来复制 va_list，那么在 Linux 的 zSeries 版本中，您必须换用 _va_copy 宏来复制。

总结

从任何 UNIX 移植到 Linux 都可以是容易的、简单的而又直接的，特别是当您的 UNIX 应用程序的编写符合常见的开放的标准时，更是如此。本文讲述了您的应用程序在这个新环境中所能得到的好处，帮您考虑在移植到 Linux 的过程中将遇到的风险。把您的应用程序移植到 zSeries 上的 Linux 还有其它的好处，那就是使您能容易地访问存储在 zSeries 环境中的企业数据和应用程序。在总结时，我们想强调以下几点：

- 从任何 UNIX 移植到 Linux 的项目可被合理安排，从而不影响您现有的任何业务。不同的应用程序的移植复杂性是不同的，且移植复杂性由所用的编程语言来决定。
- 所要求的技能是应用程序开发者的常用技能，所以您应该具备开始移植所需的所有技能。如果您需要帮助和支持，IBM 在全世界提供技术支持小组和移植中心来帮您移植应用程序。
- 详细的移植指南包含宝贵的提示和技巧，帮您判断您所移植的应用程序的复杂性。这是制定一致的移植计划的基础，使您的项目成功完成。

Linux 巨大的市场成功告诉我们，您的应用程序应该也能在 Linux 上运行。为什么错过这个机会？

移植经验 — 其他人怎么说

许多印刷的和因特网上的文章为那些考虑向 Linux 移植的人提供更多的深刻见解。当然，把应用程序移植到 Linux 的工作量会因应用程序的复杂程度、所用的编程语言等因素而有所不同。

ISV Sendmail (www.sendmail.com)：“最近，Sendmail, Inc. 把我们的商业产品移植到 IBM 的基于 Linux 的 z900。我们用了六天时间就完成了最初的移植，性能要比预想的好。Sendmail, Inc. 服务小组与见多识广的 IBM 员工紧密合作，这样大大简化了该项目。”

ISV BMC (www.bmc.com) : “当我们拿了些基于 UNIX 的代码并把它移植到 zSeries 上的 Linux 时, 我们发现这不是真正的移植, ”BMC 的 Linux 计划的战略主任 Fred Johannessen 这样说。“我们只重新编译了一下, 它就能运行了。我们对自己说, ‘不可能这么简单’, 但实际上就是这么简单。”

ISV Pentaprise (www.pentaprise.de) : “我们把我们的整个 ERP 系统从 Linux Intel 移植到所有的 IBM Linux 硬件平台, 平均每个 IBM 平台在一天内就完成了, 包括对 DB2 的支持。我们从 IBM 得到的组织上的、技术上的和市场支持是出色的。”— Reinhard Herrmann, CEO

SAP (www.sap.com/linux/), “自从 1999 年圣诞节以来, 您可以在 Linux 上运行 mySAP Technology, 用于关键任务环境中的生产, 这使我们很自豪。Linux 正以难以置信的速度发展, 它绝对满足 SAP 的质量标准。由于 SAP 体系结构是与平台无关的, SAP 内核的移植相当容易。没有一个 SAP 业务应用程序需被修改。”

ISV TRUSTIX (www.trustix.com) 是挪威的 ISV, 已成功地把一个叫 Trustix Xplore 的 Linux 系统管理 (Linux Systems Management) 框架移植到 Linux 的 S/390 版本。“移植到 Linux 的 S/390 版本比我们预想的要容易得多”, 在不到一天的时间里就完成移植的工程师这样评论。这次移植完成后, 除了 IBM ~ iSeries™, TRUSTIX 应用程序现在可在所有的 IBM 服务器上使用。

ISV SICS (www.sics.se) 把 Prolog 编译器从 Linux 移植到 Linux 的 S/390 版本。“先对配置脚本作修改, 使其能够识别正确的机器类型, 再安装所要求的库和软件的正确版本后, 实际移植就是简单的重编译并且在不到一天的时间里就完成了”, 进行移植的工程师这样评论。

更多信息

- IBM 的 Solaris to Linux porting guide ibm.com/developerworks/linux/library/l-solar/
- zSeries 上的 Linux 移植提示和技巧
ibm.com/servers/esdd/articles/linux_s390/index.html
- IBM 解决方案合作伙伴中心 (Solution Partnership Center, SPC) 移植信息
developer.ibm.com/spc/portinfo.html
- S/390 ISV 技术支持 etpgw02.dfw.ibm.com
- 如果您想获得目前提供支持 Linux 的 zSeries 版本的应用程序的 ISV 的列表, 请访问以下站点。
ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html
- ISV Announcements
ibm.com/servers/eserver/zseries/solutions/s390da/applications/appnews.html
- PartnerWorld for Developers developer.ibm.com/
- 请到 IBM ~ 开发者园地了解更多关于 IBM xSeries、pSeries 和 zSeries 上的 Linux 的信息。
ibm.com/servers/esdd/
- Linux Application Tools ibm.com/servers/eserver/zseries/os/linux/ldt/ > ibm.com/servers/eserver/zseries/os/linux/ldt/
- Developer Products for Linux for zSeries and S/390
ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html
- 请阅读 Linux 专区中的 developerWorks 的双周刊“LTC bulletin”, 了解 IBM 正在 Linux 方面做什么 ibm.com/developerworks/linux/, 还有 Linux at IBM Web 站点 ibm.com/linux
- 浏览 developerWorks 上的更多 Linux 参考资料 ibm.com/developerworks/linux/?article=lr
- 浏览 developerWorks 上的更多开放源代码的参考资料 ibm.com/developerworks/oss/
- 阅读 Linuxcare 的一篇文章, 了解对 Linux 2.4 的技术改进

- linuxcare.com/about-us/collateral/whitepapers/linux24-enterprise.epl。
- 如果您想了解 64 位 Linux, 请访问 IA-64 Linux Project Web 站点 linuxia64.org/
 - 如果您想了解应用程序开发和部署生命周期 (Application Development and Deployment Life Cycle) 和 Linux 的 zSeries 版本上的可用的工具的列表, 请访问 ibm.com/servers/eserver/zseries/os/linux/ldt/。
 - GNU General Public License gnu.org/copyleft/gpl.html、GNU Lesser General Public License www.gnu.org/copyleft/lesser.html和 gnu.org/licenses/licenses.html让任何人查看和改进 Linux 源代码。
 - GCC (GNU Compile Collection) gcc.gnu.org/releases.html和在线文档 gcc.gnu.org/onlinedocs/gcc-2.95.2/gcc_toc.html
 - GCC Suite 还包括 Java 编译器 (GCJ) gnu.org/software/gcc/java/
 - GNU 汇编器手册在 gnu.org/manual/gas-2.9.1/as.html
 - Solaris GNU 编译器和 make (gmake) 实用程序可从 sunfreeware.com/ 获得, SUN 在线产品文档在 docs.sun.com 和 gnu.org/manual/make-3.79.1/html_mono/make.html
 - 开发者如果要访问高端 Linux 系统, 可到 Open Source Development Lab www.osdlab.org/ 和 IBM 的 Linux Community Development System ibm.com/servers/eserver/zseries/os/linux/lcds/。
 - 支持 Linux 的第三方运行时库的包括 Rogue Wave SourcePro C++ 库 rougewave.com/ 和 Trolltech Qt C++ GUI 框架 www.trolltech.com/。
 - Java 工具包括 Sun 的 Java 2 Platform 标准版 java.sun.com/j2se/1.3/和 Forte Java 社区版 sun.com/forte/ffj/ IDE, 还有 IBM Developer Kit 的 Linux 版本 ibm.com/developerworks/java/jdk/linux130/
 - Red Hat 的 Solaris-to-Linux Porting Guide redhat.com/devnet/whitepapers/solaris_port/book1.html。
 - 请看由 UNIX guide 站点发布的系统管理任务的图表比较 unixguide.net/unixguide.cgi。
 - 源代码管理系统包括 GNU CSSC cssc.sourceforge.net/和 CVS cvshome.org/。
 - Xi Graphics 开发了 CDE 的 Linux 版本 xig.com/。
 - 通过研究在重要系统中使用 Linux 的公司来衡量 Linux 的商业可用性。 Aaxnet.com/design/linux2.html

了解更多关于 Linux at IBM 的信息 — ibm.com//linux/

更多关于 zSeries 上的 Linux 的详细信息 — ibm.com/zSeries/linux

Copyright IBM Corporation 2002

Integrated Marketing Communications, Server Group

Route 100

Somers, NY 10589

U.S.A.

Produced in the United States of America

01/02

All Rights Reserved

IBM、IBM 徽标、e-business 徽标、AIX、DB2、iSeries、PowerPC、pSeries、S/390 和 zSeries 是国际商业机器公司在美国和 / 或其他国家或地区的商标或注册商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Micorosoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和 / 或其他国家或地区的注册商标。

Intel 是 Intel Corporation 在美国和 / 或其他国家或地区的商标。

Linux 是 Linus Torvalds 的注册商标。

其他公司、产品和服务名称可能是其他公司的商标或服务标记。

涉及非 IBM 产品的信息是从这些产品的供应商或其出版声明中获取。有关非 IBM 产品性能的问题应该与这些产品的供应商联系。IBM 硬件产品在制造时使用的是新零件或新的和有用的旧零件。无论怎样，我们的保证条款都适用。

IBM 可能不在其他国家或地区提供本文中讨论的产品、服务或功能部件并且这里的信息可能在没有事先声明的情况下有所变动。关于现有的产品和服务信息，请咨询当地的 IBM 公司业务联系代表。所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

性能用内部吞吐量率（Internal Throughput Rate, ITR）来表示，ITR 是基于在受控的环境中使用标准的 IBM 基准程序所取得的测量值和预测值。每个用户遇到的实际吞吐量将根据用户的作业流中的多道程序设计（multiprogramming）的量、I/O 配置、存储配置、所处理的工作量等情况而有所不同。所以，不能保证个人用户能获得等同于在这里声明的性能比的吞吐量提升。

所有引用的客户示例代表某些客户是怎样使用 IBM 产品及他们可能已取得的结果。实际环境成本和性能特征将根据每个客户的配置和条件而有所不同。

1 Solaris to Linux porting guide 是 IBM 文档，写于 Toronto Development Lab。

关于作者

IBM

IBM has authored this article

© 版权所有 IBM 公司 2001

(www.ibm.com/legal/copytrade.shtml)

商标

(www.ibm.com/developerworks/cn/ibm/trademarks/)