Atitit 应用程序二进制接口（ABI-Application Binary Interface）

应用程序二进制接口（ABI-Application Binary Interface）定义了一组在PowerPC系统软件上编译应用程序所需要遵循的一套规则。主要包括基本数据类型，通用寄存器的使用，参数的传递规则，以及堆栈的使用等等。

ABI涵盖了各种细节：如数据类型、大小和对齐;[调用约定](http://zh.wikipedia.org/w/index.php?title=%E8%B0%83%E7%94%A8%E7%BA%A6%E5%AE%9A&action=edit&redlink=1" \o "调用约定（页面不存在）)（控制着函数的参数如何传送以及如何接受返回值）；[系统调用](http://zh.wikipedia.org/wiki/%E7%B3%BB%E7%BB%9F%E8%B0%83%E7%94%A8" \o "系统调用)的编码和一个应用如何向操作系统进行系统调用；以及在一个完整的操作系统ABI中，[目标文件](http://zh.wikipedia.org/wiki/%E7%9B%AE%E6%A0%87%E6%96%87%E4%BB%B6" \o "目标文件)的二进制格式、程序库等等。一个完整的ABI，像[Intel二进制兼容标准](http://zh.wikipedia.org/w/index.php?title=Intel%E4%BA%8C%E8%BF%9B%E5%88%B6%E5%85%BC%E5%AE%B9%E6%A0%87%E5%87%86&action=edit&redlink=1" \o "Intel二进制兼容标准（页面不存在）) (iBCS)[[1]](http://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E4%BA%8C%E8%BF%9B%E5%88%B6%E6%8E%A5%E5%8F%A3" \l "cite_note-0) ，允许支持它的操作系统上的程序不经修改在其他支持此ABI的操作体统上运行。

而 ABI 是另一种形式的接口，[二进制接口](http://en.wikipedia.org/wiki/Application_binary_interface" \t "http://blog.csdn.net/xinghun_4/article/details/_blank)。除非你直接使用汇编语言，这种接口一般是不能直接拿来用的。比如，内核系统调用用哪些寄存器或者干脆用堆栈来传递参数，返回值又是通过哪个寄存器传递回去，内核里面定义的某个结构体的某个字段偏移是多少等等，这些都是二进制层面上的接口。这些接口是直接给**编译好的**二进制用的。换句话说，如果 ABI 保持稳定的话，你在之前版本上编译好的二进制应用程序、内核模块，完全可以无须重新编译直接在新版本上运行。另一种比较特殊的 ABI 是像 /proc，/sys 目录下面导出的文件，它们虽然不是直接的二进制形式，但也会影响编译出来的二进制，如果它里面使用到它们的话，因此这些“接口”也是一种 ABI。

而规定 ABI 的标准就不多，而且也没那么强势，Linux 上面的 ABI 标准似乎只有 Linux Foundation 提供的[一些标准](http://refspecs.freestandards.org/lsb.shtml" \t "http://blog.csdn.net/xinghun_4/article/details/_blank)

**，其实保持一个稳定的 ABI 要比保持稳定的 API 要难得多。比如，在内核中 int register\_netdevice(struct net\_device \*dev) 这个内核函数原型基本上是不会变的，所以保持这个 API 稳定是很简单的，但它的 ABI 就未必了，就算是这个函数定义本身没变，即 API 没变，而 struct net\_device 的定义变了，里面多了或者少了某一个字段，它的 ABI 就变了，你之前编译好的二进制模块就很可能会出错了，必须重新编译才行。**

你可能会感到意外，上游的 Linux

## Linux standard Base（LSB）

噢，这到底是巧合呢？还是什么：所有Linux 程序都是在所有机器上运行。我也带个着这个问题去找一些资料。   
想信大家听说到POSIX规范，Linux Standard Base(LSB)规范。起初，我以为LSB是API规范而非ABI规范。当我去深入分析时，发现它是一个ABI规范，它解决两方面的问题：

* 在发行版Linux A上运行的程序，可以运行在发行版本Linux B上。
* 应用程序在现在的机器上运行，也能在未来新系统和机器上运行。

最近Linux基金会发起LSB项目就是希望Linux发行商能做到相互兼容，并且也能向前兼容。解决Linux应用的可移植性和兼容性。

LSB实际是是一组ABI接口的定义，它规范了运行环境所需要的类型，宏，变量和函数的二进制接口。

那么，哪些东西受到LSB的制约呢？也即LSB规范约束的范围多广。刚开始，我还以为只有kernel, glibc等一些关键的组件。事实上，打开LSB官网说明会发现它包罗万象，从命令，ELF，glibc接口，到桌面，脚本语言等等都有相关的规定。

API、ABI区别 - xinghun\_4的专栏 - 博客频道 - CSDN.NET.html