Atitit 可移植性之道

[1. .3.2 移植的策略 2](#_Toc2787)

[1.1. 3.2.1 完全移植策略几乎总是可以找到功能类似的产品 2](#_Toc13565)

[1.2. 3.2.2 部分移植策略 2](#_Toc11840)

[2. 可移植性标准 GB/T29833 2](#_Toc2423)

[3. 俩大可移植 3](#_Toc22976)

[3.1. 源码可移植性 3](#_Toc24141)

[3.2. 二进制可移植性 3](#_Toc24283)

[4. 移植运行环境 3](#_Toc10050)

[4.1. 跨语言可移植 3](#_Toc20940)

[4.2. 跨语言体系可移植 锁定板os相当于 3](#_Toc24859)

[4.3. 跨os可移植 3](#_Toc30678)

[4.4. 软件的平台移植 3](#_Toc20832)

[5. 可移植性目标 4](#_Toc26754)

[5.1. 语法可移植性 4](#_Toc265)

[5.2. Api可移植 4](#_Toc538)

[5.3. 类库可移植 4](#_Toc12713)

[6. Atitit.源码可移植性 与兼容 规范java php c# js 4](#_Toc4777)

[7. 可移植性最佳实践 4](#_Toc18642)

[7.1. .分层设计策略，隔离平台相关的代码。 4](#_Toc25158)

[7.2. 使用h5做界面ui 4](#_Toc19407)

[7.3. 使用cmd命令行 api 代替 普通api 4](#_Toc28793)

[7.4. 尽可能使用第三方工具 5](#_Toc5598)

[7.5. 避免平台环境特点api 5](#_Toc1524)

[7.6. 分离界面表现与内部逻辑也是非常有用的 5](#_Toc1427)

[7.7. 最好清楚不同平台的资源限制 5](#_Toc1279)

[7.8. 标准的sdk接口。正是这些标准的接口使得应用开发和硬件以及底层相分离。 5](#_Toc531)

[7.9. 要做到无关就得定义统一的接口规范。让应用程序调用接口规范规定的接口，底层软件根据不同的平台封装不同的程序来实现其接口功能。 5](#_Toc30271)

[8. 硬件 传感器api移植 5](#_Toc32568)

[9. 第4章 操作系统环境的移植 6](#_Toc31939)

[10. 第5章 文件系统移植 开发中容易碰上的FS差异主要有如下几个： 6](#_Toc8173)

[10.1. 目录分隔符的差异；大小写敏感的差异；路径中禁用字符的差异。 6](#_Toc29570)

[10.2. 3、代码中涉及FS操作，尽量使用现成的库 6](#_Toc20536)

[10.3. ★文本文件的回车CR/换行LF 6](#_Toc18343)

[10.4. ★文件搜索路径（包括搜索可执行文件和动态库） 7](#_Toc15569)

[10.5. ★环境变量 7](#_Toc32353)

[10.6. ★动态库 7](#_Toc31799)

[11. 第7章 网络服务移植 dhcp ftp http等 主要讨论了WINS、DNS、DHCP等典型网络服务的移植方案 7](#_Toc2578)

[12. 第11章 Web服务移植 7](#_Toc7582)

[13. 第12章 数据库移植 8](#_Toc29198)

[14. 其他 8](#_Toc13365)

[14.1. 第15章 桌面office应用移植 8](#_Toc10574)

[15. Atitit.常用语言的常用内部api 以及API兼容性对源码级别可移植的重要性 总结 8](#_Toc27014)

[15.1. 1.1. 要兼容的重要语言api 1 8](#_Toc5061)

[15.2. 1.2. 常用基础api分类 core api 1 8](#_Toc14343)

[15.3. 1.3. 比较常用的扩展库api ext api 1 8](#_Toc30511)

[15.4. 1.4. 标准函数库函数数量统计，共有多少个函数c c++ vbs js php 2 8](#_Toc14108)

[15.5. 2. 范例，给java扩展其他语言的api 2 8](#_Toc15257)

[16. 参考资料 8](#_Toc27656)

[16.1. 《Windows信息系统开放源码平台移植指南》书籍目录 9](#_Toc16630)

# .3.2 移植的策略

## 3.2.1 完全移植策略几乎总是可以找到功能类似的产品

移植实例表明，几乎总是可以找到功能类似的OSS 和（或）COLS产品作为对微软产品进行完全替换性移植的解决方案

## 3.2.2 部分移植策略

尽管选择性移植通常不会带来明显的成本优势，但在实际决策时综合考虑紧迫性等一些因素，选择性移植可以作为一种经常采用的移植策略。

# 可移植性标准 GB/T29833

GB/T29833的本部分规定了系统与软件可移植性指标体系及相关定义。

GB/T29833的本部分参照GB/T16260—2006《软件工程 产品质量》提出了系统与软件可移植性指标体系。

信息技术标准化技术委员会(SAC/TC 28)

GB/T29833在《系统与软件可移植性》总标题下,分为如下三部分:   
———第1部分:指标体系;   
———第2部分:度量方法;   
———第3部分:测试方法。

# 俩大可移植

## 源码可移植性

## 二进制可移植性

# 移植运行环境

## 跨语言可移植

## 跨语言体系可移植 锁定板os相当于

## 跨os可移植

## 软件的平台移植

# 可移植性目标

## 语法可移植性

## Api可移植

## 类库可移植

# Atitit.源码可移植性 与兼容 规范java php c# js

变量定义 $符号开始

数组dic,list的使用，要使用get函数方式。不是有括号语法方式

委托的不使用，使用内部类。

各种异常类的转换

# 可移植性最佳实践

## .分层设计策略，隔离平台相关的代码。

就像可测试性一样，可移植性也要从设计抓起。一般来说，最上层和最下层都不具有良好的可移植性。最上层是GUI，大多数GUI都不是跨平台的，如Win32 SDK和MFC。最下层是操作系统API，大多部分操作系统API都是专用的。

## 使用h5做界面ui

## 使用cmd命令行 api 代替 普通api

比如图像处理使用im代替语言api

## 尽可能使用第三方工具

## 避免平台环境特点api

## 分离界面表现与内部逻辑也是非常有用的

## 最好清楚不同平台的资源限制

## 标准的sdk接口。正是这些标准的接口使得应用开发和硬件以及底层相分离。

windows平台也好还是ios,android平台也罢，他们的应用是不同的编程语言实现的，但是他们有一个共同的特点：給应用开发人员提供标准的sdk接口。正是这些标准的接口使得应用开发和硬件以及底层相分离。大大提高了应用程序的可移植性。

## 要做到无关就得定义统一的接口规范。让应用程序调用接口规范规定的接口，底层软件根据不同的平台封装不同的程序来实现其接口功能。

如此大大提高应用程序不同硬件平台的可移植性。这样做也规范了编程，增加了程序的可读性。

嵌入式软件设计之提高代码可移植性 - KenoTu - 51CTO技术博客.html

# 硬件 传感器api移植

硬件体系相关

　　这次聊的话题主要是和硬件体系有关的。比如你的程序需要支持不同类型的CPU（x86、SPARC、PowerPC），或者是同种类型不同字长的CPU（比如x86和x86-64），这时候你就需要关心一下硬件体系的问题。

# 第4章 操作系统环境的移植

# 第5章 文件系统移植 开发中容易碰上的FS差异主要有如下几个：

## 目录分隔符的差异；大小写敏感的差异；路径中禁用字符的差异。

1、文件和目录命名要规范

　　在给文件和目录命名时，尽量只使用字母和数字。不要在同一个目录下放两个名称相似（名称中只有大小写不同，例如foo.cpp与Foo.cpp）的文件。不要使用某些OS的保留字（例如aux、con、nul、prn）作文件名或目录名。

　　补充一下，刚才说的命名，包括了源代码文件、二进制文件和运行时创建的其它文件。

2、#include语句要规范

　　当你写#include语句时，要注意使用正斜线“/”（比较通用）而不要使用反斜线“\”（仅在Windows可用）。#include语句中的文件和目录名要和实际名称保持大小写完全一致。

## 3、代码中涉及FS操作，尽量使用现成的库

　　已经有很多成熟的、用于FS的第三方库（比如boost::filesystem）。如果你的代码涉及到FS的操作（比如目录遍历），尽量使用这些第三方库，可以帮你省不少事情。

## ★文本文件的回车CR/换行LF

　　由于几个知名的操作系统对回车/换行的处理不一致，导致了这个烦人的问题。目前的局面是：Windows同时使用CR和LF；Linux和大部分的Unix使用LF；苹果的Mac系列使用CR。

　　对于源代码管理，好在很多版本管理软件（比如CVS、SVN）都会智能地处理这个问题，让你从代码库取回本地的源码能适应本地的格式。

　　如果你的程序需要在运行时处理文本文件，要留意本文方式打开和二进制方式打开的区别。另外，如果涉及跨不同系统传输文本文件，要考虑进行适当的处理。

## ★文件搜索路径（包括搜索可执行文件和动态库）

　　在Windows下，如果要执行文件或者加载动态库，一般会搜索当前目录；而Posix系统则不尽然。所以如果你的应用涉及到启动进程或加载动态库，就要小心这个差异。

## ★环境变量

　　对于上述提到的搜索路径问题，有些同学想通过修改PATH和LD\_LIBRARY\_PATH来引入当前路径。假如使用这种方法，建议你只修改进程级的环境变量，不要修改系统级的环境变量（修改系统级有可能影响到同机的其它软件，产生副作用）。

## ★动态库

　　如果你的应用程序使用动态库，强烈建议动态库导出标准C风格的函数（尽量不要导出类）。如果在Posix系统中加载动态库，切记慎用RTLD\_GLOBAL标志位。这个标志位会Enable全局符号表，有可能会导致多个动态库之间的符号名冲突（一旦碰到这种事，会出现匪夷所思的运行时错误，极难调试）。

# 第7章 网络服务移植 dhcp ftp http等 主要讨论了WINS、DNS、DHCP等典型网络服务的移植方案

# 第11章 Web服务移植

# 第12章 数据库移植

# 其他

## 第15章 桌面office应用移植

# Atitit.常用语言的常用内部api 以及API兼容性对源码级别可移植的重要性 总结

## 1.1. 要兼容的重要语言api 1

## 1.2. 常用基础api分类 core api 1

## 1.3. 比较常用的扩展库api ext api 1

## 1.4. 标准函数库函数数量统计，共有多少个函数c c++ vbs js php 2

## 2. 范例，给java扩展其他语言的api 2

2.1. 目录结构 2

2.2. 调用 3

3. 参考 3

# 参考资料

《可移植的远程计算环境及其技术》(曾志勇　著)【简介\_书评\_在线阅读】 - 当当图书.mhtml

《Windows信息系统开放源码平台移植指南》(谢永强 主编)【简介\_书评\_在线阅读】 - 当当图书.mhtml

1. 绪 论  
   第1章 信息系统移植基础  
   第2章 信息系统移植中的关键问题  
   第3章 信息系统移植的决策建议  
   第二部分 信息系统移植技术指南  
   第4章 操作系统环境的移植  
   第5章 文件系统移植  
   第6章 打印服务移植  
   第7章 网络服务移植  
   第8章 系统审计与管理服务移植  
   第9章 目录服务移植  
   第10章 认证服务移植  
   第11章 Web服务移植

## **《Windows信息系统开放源码平台移植指南》书籍目录**

第一部分 绪 论

第1章 信息系统移植基础

1.1 windows平台信息系统

1.1.1 windows操作系统的发展历史

1.1.2 一个典型的windows信息系统组成

1.2 开放源码平台信息系统

1.2.1 开放源码平台的发展历史

1.2.2 一个典型的开放源码平台信息系统组成

1.3 移植的主要原因与范围

1.3.1 引起移植的主要原因

1.3.2 移植的范围

1.4 小结

第2章 信息系统移植中的关键问题

2.1 移植的技术可行性

2.2 移植过程中人的因素

2.3 移植的成本问题

2.4 移植过程对新技术发展的考虑

2.5 小结

第3章 信息系统移植的决策建议

3.1 决策的过程

.3.2 移植的策略

3.2.1 完全移植策略

3.2.2 部分移植策略

3.3 一般性建议

3.4 小结

第二部分 信息系统移植技术指南

第4章 操作系统环境的移植

4.1 windows域中新增加gnu/linux桌面

4.1.1 增加简单配置的gnu/linux桌面的设置

4.1.2 增加复杂配置的gnu/linux桌面的设置

4.2 windows域中增加gnu/linux服务器

4.3 windows域控制器移植

4.4 gnu/linux桌面替换windows客户端

4.5 小结

第5章 文件系统移植

5.1 windows环境文件系统描述

5.1.1 文件系统的功能

5.1.2 windows文件系统

5.1.3 ntfs文件系统权限管理

5.1.4 ntfs访问控制

5.1.5 文件服务功能

5.2 文件系统linux平台移植方法

5.2.1 移植功能需求

5.2.2 samba介绍

5.2.3 文件服务器的比较

5.2.4 利用samba进行文件服务的移植

5.3 小结

第6章 打印服务移植

6.1 windows环境打印服务描述

6.1.1 windows打印服务介绍

6.1.2 windows的两种打印方法

6.1.3 打印服务的特性

6.2 打印服务linux平台移植方法

6.2.1 移植功能需求

6.2.2 打印系统cups体系描述

6.2.3 打印数据传输的标准

6.2.4 打印机控制协议

6.2.5 驱动程序的实现方法

6.2.6 使用cups/samba进行打印服务移植

6.3 小结

第7章 网络服务移植

7.1 windows环境网络服务描述

7.1.1 windows因特网名字服务wins

7.1.2 域名系统（dns）

7.1.3 动态主机配置协议（dhcp）

7.2 网络服务linux平台移植方法

7.2.1 移植功能需求

7.2.2 samba的nmbd支持因特网名字服务

7.2.3 bind9支持域名系统（dns）

7.2.4 isc的dhcpd

7.3 小结

第8章 系统审计与管理服务移植

8.1 windows环境系统审计与管理服务描述（含系统管理）

8.1.1 sms 2.0

8.1.2 服务器监控

8.2 系统审计与管理服务linux平台移植方法

8.2.1 移植功能需求

8.2.2 软件管理移植

8.2.3 网络管理移植

8.2.4 服务器管理移植

8.2.5 更高复杂度系统的移植

8.3 小结

第9章 目录服务移植

9.1 windows环境目录服务描述

9.1.1 目录服务介绍

9.1.2 windows活动目录

9.1.3 kerberos认证机制

9.1.4 目录结构的新特性

9.1.5 dns名字空间

9.2 目录服务linux平台移植方法

9.2.1 移植功能需求

9.2.2 ldap和openldap

9.2.3 使用linux/openldap移植目录服务

9.3 小结

第10章 认证服务移植

10.1 windows环境认证服务描述

10.1.1 windows域介绍

10.1.2 windows域的目录服务

10.1.3 windows域的认证机制

10.2 认证服务linux平台移植方法

10.2.1 移植功能需求

10.2.2 使用openldap和samba移植认证服务

10.3 小结

第11章 web服务移植

11.1 windows环境web服务描述

11.1.1 web服务及xml语言

11.1.2 windows上的web服务

11.2 web服务linux平台移植方法

11.2.1 移植功能需求

11.2.2 apache的功能描述

11.2.3 从iss移植到apache

11.3 小结

第12章 数据库移植

12.1 windows环境数据库系统描述

12.1.1 客户端与服务器体系

12.1.2 数据库存储与处理体系

12.1.3 用户控制管理体系

12.2 数据库系统linux平台移植方法

12.2.1 移植功能需求

12.2.2 选择目标数据库移植

12.3 小结

第13章 群件构建模型移植

13.1 windows环境群件构建模型描述

13.1.1 exchange服务器架构

13.1.2 通信协议与方式

13.1.3 用户功能

13.2 群件构建模型linux平台移植方法

13.2.1 移植功能需求

13.2.2 选择novell openexchange移植

13.2.3 选择kroupware移植

13.2.4 选择exchange4linux移植

13.2.5 选择samsung contact移植

13.2.6 选择novell groupwise移植

13.2.7 选择phpgroupware移植

13.2.8 移植解决方案比较

13.3 小结

第14章 终端服务器和瘦客户端移植

14.1 windows环境终端服务和瘦客户端描述

14.1.1 终端服务和瘦客户端的概念

14.1.2 windows终端服务和瘦客户端

14.2 终端服务和瘦客户端linux平台移植方法

14.2.1 移植功能需求

14.2.2 选择ltsp项目移植

14.2.3 选择nx终端服务移植

14.3 小结

第15章 桌面office应用移植

15.1 windows环境office应用描述

15.1.1 微软office概述

15.1.2 微软office编程环境

15.2 office应用linux平台移植方法

15.2.1 移植功能需求

15.2.2 linux平台office办公套件

15.2.3 选择openoffice/staroffice移植

15.3 小结

第16章 中间件技术移植

16.1 windows环境中间件技术描述

16.1.1 中间件技术概述

16.1.2 微软com技术和.net平台

16.2 中间件linux平台移植方法

16.2.1 移植功能需求

16.2.2 选择gnome的corba技术移植

16.2.3 选择j2ee移植

16.3 小结

第17章 高可用性系统移植

17.1 高可用性系统描述

17.1.1 高可用性目标与标准

17.1.2 高可用性方法与分类

17.2 linux平台ha移植方法

17.2.1 移植功能需求

17.2.2 linux平台高可用性方法

17.3 小结

第18章 其他功能组件技术移植

18.1 linux环境中的病毒防护

18.2 linux环境中的时间服务

18.3 linux环境中的用户界面

18.4 linux环境中的备份与恢复

18.5 小结

第三部分 信息系统移植实施指南

第19章 信息系统移植的组织与实施

19.1 移植过程概述

19.2 确定移植的目标

19.2.1 明确目标

19.2.2 为目标环境创造良好的用户接受度

19.3 决策者和选择用户的介入

19.3.1 管理决策层的介入

19.3.2 选择用户介入的时机

19.4 实施移植的技术前提

19.4.1 决定初始条件

19.4.2 覆盖功能需求

19.5 移植实施的组织结构

19.5.1 项目组

19.5.2 项目成员

19.5.3 外部专家顾问

19.5.4 定义项目的组织形式

19.6 建立移植实施的计划

19.7 移植的控制与管理

19.8 用户与系统管理员的培训

19.9 小结

第20章 完全替换类信息系统移植的策略与实例

20.1 完全替换性应用移植策略

20.1.1 工作站计算机

20.1.2 web服务器

20.1.3 文件系统

20.1.4 打印服务

20.1.5 网络服务

20.2 大中型组织机构的移植实例

20.2.1 数据库管理系统

20.2.2 组件

20.2.3 目录服务

20.2.4 系统管理服务

20.3 提供it服务的特殊机构移植实例

20.3.1 数据库管理系统

20.3.2 应用服务器

20.3.3 系统管理服务

20.4 小型组织机构移植实例

20.4.1 数据库管理系统

20.4.2 组件

20.4.3 目录服务

20.4.4 系统管理服务

20.5 小结

第21章 部分替换类信息系统移植的策略与实例

21.1 选择性应用移植策略

21.2 服务器端的移植实例

21.2.1 用户管理和认证

21.2.2 文件和打印服务

21.3 小结

附录a windows平台与linux平台功能软件对应表

参考文献



Windows信息系统开放源码平台移植指南(谢永强等)\_Windows书籍\_希赛网图书.mhtml

Linux系统移植（第2版）\_百度百科.mhtml

Windows信息系统开放源码平台移植指南 txt免费下载\_读后感\_在线阅读\_读书人图书资料库.mhtml

C++的可移植性和跨平台开发 - 程序员深度学习 - CSDN博客.mhtml

编写可移植C\_C++程序的要点 - 郭晓倩 - 博客园.mhtml

Atitit usrQBN1305 软件移植规范标准化与解决方案.docx

Atitit. 跨语言ui api兼容性增强 与源码转换 源码级别移植方案.docx

Atitit.源码可移植性规范java php c# js.doc

作者:: 绰号:老哇的爪子claw of Eagle 偶像破坏者Iconoclast image-smasher

捕鸟王"Bird Catcher kok 虔诚者Pious 宗教信仰捍卫者 Defender Of the Faith. 卡拉卡拉红斗篷 Caracalla red cloak 万兽之王 纵火者

简称：： st Emir Attilax Akbar 圣 埃米尔 阿提拉克斯 阿克巴

全名：：st Emir Attilax Akbar bin Mahmud bin attila bin Solomon bin adam Al Rapanui 圣 埃米尔 阿提拉克斯 阿克巴 本 马哈茂德 本 阿提拉 本 所罗门 本亚当 阿尔 拉帕努伊

常用名：艾提拉（艾龙）， EMAIL:1466519819@qq.com

头衔：

|  |  |
| --- | --- |
| uke | Emir Uke部落首席大酋长，ati协会创始人  uke总部o2o负责人，全球网格化项目创始人，  圣阿提拉克斯国王 |
| 科技领域 | UTSC uke技术标准化委员会委员长 uke 首席cto 软件部门总监 技术部副总监 研发部门总监主管 产品部副经理 项目部副经理 uke科技研究院院长 uke软件培训大师  Ati组织科研研究院创始人 |
| 文艺领域 | ， ，， uke机车协会主任 uke纹身协会  uke交友协会会长 uke捕猎协会会长  Ati文艺协会会长 ati文学协会 |
| 行政领域 | Gchsp总裁 gchsp常委 GsP创始人 |
| 媒体传播领域 | uke出版社编辑总编 宣传布道总策划  Ati传媒总部 |
| 渔猎军事领域 | uke保安部首席大队长  Uke 户外运动协会理事长 度假村首席大村长  Ati打猎协会 |
| 法学 | 法学研究会 制度研究会 |
| 管理领域 | 工商管理学 公共管理与社会服务  ,uke制度检查委员会副会长 |
| 教育领域 | uec学院校长， uecip图像处理机器视觉专业系主任 uke文档检索专业系主任  Uke图像处理与机器视觉学院首席院长  uke终身教育学校副校长  靓号研究院 |
| 经济领域 | uke波利尼西亚区大区连锁负责人 汤加王国区域负责人 uke克尔格伦群岛区连锁负责人，莱恩群岛区连锁负责人，uke布维岛和南乔治亚和南桑威奇群岛大区连锁负责人  Uke软件标准化协会理事长理事长 Uke 数据库与存储标准化协会副会长  直达巴士西北区负责人 直达巴士长沙与西安分部部长  润昌通讯软件事业部总裁 执行长 分部负责人 执行委员会主席  Ati经济研究所 |
| 历史领域 | 历史事业部 ati历史研究院 |
| 社会科学领域 | 社科学院 ati文化部 |
| 自然科学领域 | Uke研究院院长兼首席研究员 科学家  Ati自然科学研究院 |
| 宗教神学领域 | uke宗教与文化融合事务部部长 大师master  uke制度与重大会议委员会委员长 ati宗教事务所 |
| 医学领域 | Uke医院 与医学院方面的创始人 |

转载请注明来源：attilax的专栏 <http://blog.csdn.net/attilax>

<http://www.cnblogs.com/attilax/>

Microblog

<http://weibo.com/u/5941179815> (common attilax)

<https://weibo.com/p/1005055941179815> （attilax201707,bek weibo）

<http://weibo.com/u/5487832265> (tech,for blog auto gene)

知乎空间

https://www.zhihu.com/people/ati-att/activities

Qq 1466519819 小号112237553

微信attilax 小号attilax201708

微博 attilax2016 小号attilax201707

--Atiend v19