第三部分 编程作业

第23章 MP3自动点唱机作业



第 23 章 MP3 自动点唱机作业

这一章中将会分阶段编写一个可作为 MP3 播放列表生成器和数据库管理器的主窗口程序。它将会搜索文件系统中的 MP3 歌曲,并据此生成和播放选择的 MP3 文件,同时该程序将允许基于存储在 ID3v2(元标记信息)中的数据进行过滤查询。

前提条件: 要继续本章的作业, 需要你看完第 13 章。

我们将要实施的功能是受到 Amarok, aTunes 这样的开源项目和 iTunes, Musicmatch Jukebox 这样的商业程序的启发。所有这些程序都提供了类似的功能,但用户界面却大不相同。还有很多创意也可以用到播放器上。其中一些用键盘或鼠标会一样简单。只用鼠标或只用键盘工作试试。

在\$QTDIR/examples/phonon/qmusicplayer中可以找到一个媒体播放器示例,可以以此作为工作的起点。然后,在做下列作业的过程中,可以向它添加一些其他组件。到撰写这部分内容时为止,qmusicplayer允许向播放列表中添加文件并播放计算机上的歌曲列表。

如图 23.1 所示,这是一个最简单的媒体播放器,有如下两个主要的组件。

- 播放器视图,它的构成如下。
 - 一个停靠区,用来显示用户当前正在播放的歌曲,同时提供一些控制,用来改变歌曲的声音大小和位置。
 - Play/Pause 和 Stop 按钮(如果还有 Next 和 Previous 按钮会更好一些)。
- 用作中央窗件的歌曲列表视图,用来显示一列歌曲。



图 23.1 示例的屏幕截图

这些组件中的每一个都是显示数据视图,建议使用单独的类来保存实际数据,并始终保 持模型代码和视图代码的分离。

Qt 的 MultiMediaKit

另外还有一个库,Qt 的 MultiMediaKit^①,可以用作 Qt 的一个附加组件(add-on)。该 组件包含在 Qt 软件开发包(Qt Software Development Kit, SDK)中,可以从 Qt Mobility^②中下载。Qt 的 MultiMediaKit 提供用于媒体播放器方面的高级 API,包括保存记录和访问原始声音矢量数据的功能。它提供播放列表和许多流行播放列表格式的读取、写入实现。如果打算复用这个库而不是使用 Phonon,更好的起点将是 Qt Mobility 中的 demos/player示例。Qt Mobility 的一部分只能用在诸如手机这样的移动平台上,而 Qt 的 MultiMediaKit则可以用于所有平台。

Qt 的 MultiMediaKit 仍处于开发之中,但在未来的 Qt 版本中将可能打算替换 Phonon。 截至 Qt 4.6,有一个称为 libqtmultimedia 的库,包含于 Qt 安装包中。它含有之前提到的 libqtmultimediakit 库的一个子集。截至 Qt 4.7,这个库都没有包含 M3u 播放列表类以及其他一些你想用于媒体播放器的类。

23.1 Phonon/MultiMediaKit 配置

第一步要确保有一个带 Phonon (或 Mobility 的 MultiMediaKit) 的 Qt 版本,还要确保你的 计算机系统带有正确的可以播放 MP3 文件的编解码器。如果当前可以在计算机上播放 MP3 文件,那么设置就完成了。否则,对于 Qt 4.7 来说,Qt SDK 已经为大多数平台提供了这两个库现成的二进制文件,它们似乎都可以很好地工作。

构建并运行你所选择的起始范例程序,以确保你已经恰当地安装了可用于该工程的 Qt。如果无法播放预期格式的媒体,可以参阅"安装 Phonon"一节中的文档[®]。

23.2 播放列表

1. 从起始范例开始并使其可以加载、保存和清空当前的播放列表。用于播放列表的模型 应当保存在一个名称为 PlayListModel 的类中,该类(直接或间接)派生自 QAbstractItemModel。

PlayListModel 应当为 MetaDataValue(或类似的)实例提供一个模型。

不要像 Qt Phonon 中的例子那样使用 QTableWidget, 而是对 QTableView 进行扩展并称其为 PlayListView。图 23.2 给出了这些类之间的关系。

如果愿意,可以复用 MetaDataObject, DataObjectTableModel, SimpleDelegate 以及可以在 dist 中 libmetadata 处找到的其他类[®]。

2. 让应用程序在 QSettings 中记住当前播放列表的文件名、表列的宽度以及当用户退出时主窗口的大小/位置。

① 参见http://doc.qt.nokia.com/qtmobility/multimedia.html。

② 参见http://qt.nokia.com/products/qt-addons/mobility/。

③ 参见http://doc.qt.nokia.com/latest/phonon-overview.html#installing-phonon。

④ 参见http://www.distancecompsci.com/dist/。

- 3. 在启动时,恢复这些 QSettings 的值。
- 4. 实现以下播放器/播放列表的动作:后一个(next)、前一个(previous)、播放下一首 (playNext,在当前歌曲播放完成之后)、添加文件(addFiles)、清空(clear)、另存为(save as)、保存(save)和打开(open)。把这些动作放到主菜单和播放列表视图的上下文菜单中。确保给定名称的动作可以真正地控制播放器。
- 5. (选作):实现剪贴板和在表视图中进行多重选择的动作,即移除文件(removeFiles)、复制(copy)、剪切(cut)和粘贴到此处(paste here)。

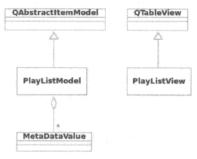


图 23.2 简单的播放列表

23.3 多种类型的播放列表

在这项作业中,会定义一些数据类型及加载和保存一些 M3U 格式播放列表的方式。 M3U 文件相当简单;它们每行都包括一个 URL,该 URL 可能是本地磁盘轨道上的一个 相对路径。可选的扩展信息和注释后出现在以#打头的一行中。

之前的作业是从一个用作自我提高的示例性媒体播放器程序开始的。现在,可以向该程序添加一些加载/保存播放列表的动作。

Phonon 库有没有用来处理 M3U 文件的程序代码, 所以需要自己编写一段序列化的代码。 Ot Mobility 示例中有一个 OMedia Playlist->load()方法, 它可以载人 M3U 文件。

src/handouts 中提供了一些加载和向磁盘保存 M3U 文件的示例代码。可以随意复用 这些代码。

MetaDataLoader 类使用了一些 dist 目录中的代码。有三种方式可以用: Phonon, taglib 或者 Qt Mobility。可以选择一种最适用于你的工作平台的方式。在 Linux 平台上,所有这三种方式都可以工作。截至本书出版时,无论是 Phonon 还是 Mobility,都不可以读取 Windows 上 ID3 格式的 MP3 文件,这就是为什么还要提供一个 Taglib 实现的原因。

如果想让播放器与其他播放器相兼容,在 http://help.mp3tag.de/en/main_tags.html 这个链接中所包含的信息 $^{\circ}$,可以帮助你确定流行的其他 MP3 播放器是使用哪一种标记来保存歌曲评分的。

用 QtTestLib 写一些测试用例,以验证可以读取和写人各种播放列表。

① 参见http://help.mp3tag.de/en/main_tags.html。

23.4 源选择器

如果能够从各种不同的源中选择歌曲,那么该播放器的功能将异常强大,也更为有用。 所谓的源可能是下列中的任意一种:

- 一个播放列表。
- 一个 MP3 库,存储在一个数据库中。
- 一个在线电台的列表。
- CD 上的音轨。
- 一个音轨文件夹。
- 一个音乐库的子集,可通过流派、艺术家或者唱片进行分类过滤。

有趣的是,所有这些都有一个共同点:每种都可以看作一个 QAbstractItemView。图 23.3 中,停靠在左侧的 SourceSelector 窗件允许用户选择当前的播放源,这种情况下,它对应的是一个许多歌曲的列表。

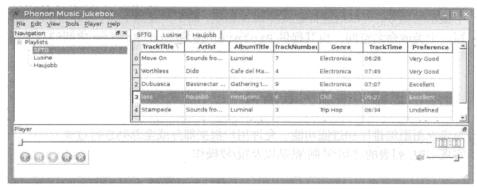


图 23.3 源选择器视图

任何时候,如果用户在窗件中单击另一个源,那么"当前播放列表"就应当切换到选中的那一个,且 MP3 播放器应当开始播放这个新的播放列表中的歌曲。

通过在选择器模型中单击一个源,应用程序应当改变目前可见的表或者中央窗件中的列表视图。图 23.4 给出了一种可能的方式来设计能够提供视图和一个源集合选择器的那些类。

在任何给定的时间,用户都可以执行一个动作,显示正在播放的(Show Now Playing)歌曲。它会改变中央窗件,以便可以显示当前正在播放的,而不是用户在源选择器中那个最后的选择。

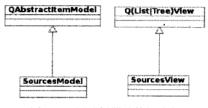


图 23.4 源选择器及其相关类

23.5 各播放列表数据库

要可以在数据库中加载和存储各个播放列表。如果不打算编写 SQL,可以复用 \$CPPLIBS/sqlmetadata 中的代码。

- 添加一个队列和一个历史播放列表。
- 给放入队列和移出队列的音轨添加 QAction。
- 在播放列表视图中把播放位置显示成一个装饰条。
- 在播放一首曲目的同时,测试 next/previous 是否可以如预期的那样工作。

23.6 星号评分

- 1. 在 Qt 的例子中,有一个 stardelegate 的应用程序,它提供了一个用于 StarRating 的 QItemDelegate 实例。
- 2. 评价一首歌曲时,不要使用类似于"差"(bad)或者"优"(great)这样的词,用表格中显示的星号,可以给这首歌从零到 N 颗星,而不只是除 Preference 以外的评价。
- 3. 写一个类或命名空间,由其提供 Preference 和 StarRating 之间的双向转换。
- 4. 写一个 StarDelegate, 让它用于 Preference。

23.7 排序,过滤和播放列表编辑

- 1. 为表视图添加排序和过滤功能。允许用户根据部分或全部列进行过滤。
- 2. 实现 URL 列表的剪切/复制/粘贴以及布/放操作。



附录 A C++的保留关键字

关键字是一种标志符,它是编程语言基本语法的一部分。关键字的名称具有固定的含义, 不能用在不是这种固定含义的任何场合。

以下是 C++的关键字列表, 其中以粗体显示的同时也为 ANSI C89 的关键字。

and	extern	signed
and_eq	FALSE	sizeof
asm	float	static
auto	for	static_cas
bitand	friend	struct
bitor	goto	switch
bool	if	template
break	inline	this
case	int	throw
catch	long	TRUE
char	mutable	try
class	namespace	typedef
compl	new	typeid
const	not	typename
const_cast	not_eq	union
continue	operator	unsigned
default	or .	using
delete	or_eq	virtual
do	private	void
double	protected	volatile
dynamic_cast	public	wchar_t
clse	register	while
enum	reinterpret_cast	xor
explicit	return	xor_eq
export	short	

附录 B 标准头文件

本书中用到了标准模板库(STL,也称为标准库)的一个小子集。书中所用的标准头文件在下面列出。为了有效地使用这些类和函数,知道去哪里查找文档是有用的。

表 B.1 中给出了一些标准头文件,其中有些头文件带有手册页,另外一些情况下,也可以找到某个函数的手册页。

如果使用的是 Microsoft Developer 中的 Studio,则标准库的文档会随 MSDN 文档一起提供。 对于其他平台,尽管文档能够提供帮助,但是没有必要在本地保存手册页的副本,因为在 Web 上存在大量的文档副本 $^{\circ}$ 。

下面的这个表是通过运行 doxygen 而得到的标准头文件。

表 B.1 标准头文件

头 文 件	库	手 册 页
	C++ STL	
String	STL 字符串类型	std::string
sstream	stringstream,用于写人字符串流	std::stringstream
iostream	C++标准流库	std::ios, std::iostream
memory	C++中与内存相关的例程	std::bad_alloc, std::auto_ptr
	C标准库	
estring, string.h	用于 C char *字符串的函数	string, strepy, stremp
cstdlib, stdlib.h	C标准库	random, srandom, getenv, setenv
cstdio, stdio.h	标准输入/输出	stdin, stdout, printf, scanf
cassert	断言宏	assert

注意

默认情况下,C++标准库文档不会安装在系统中。用包管理器搜索字符串 libstdc,这样就能够安装某个文档,比如 libstdc++6-4.5-doc。

① 例如,可以查看 cplusplus.com(http://www.cplusplus.com/ref/) 或者 Dinkumware (http://www.dink umware.com/manuals/reader.aspx? lib=cpp)。

附录C 开发工具

本附录包含的中是一些有关 C++开发环境的文章。

C.1 make 和 Makefile

QMAKE_TARGET = qapp

DESTDIR TARGET

first: all

以前曾讲过:不带参数的 qmake 能读取工程文件并构建出一个 Makefile。示例 C.1 给出了对 Makefile 进行简要的回顾,该文件是从一个名称为 qapp 的简单工程中生成的。

```
示例 C.1 src/qapp/Makefi le-abbreviated
# Exerpts from a makefile
###### Compiler, tools and options
CC
              = gcc
                       # executable for C compiler
CXX
                        # executable for c++ compiler
LINK
                        # executable for linker
              = g++
# flags that get passed to the compiler
CFLAGS
             = -pipe -g -Wall -W -D_REENTRANT $ (DEFINES)
CXXFLAGS
              = -pipe -g -Wall -W -D_REENTRANT $ (DEFINES)
INCPATH
              = -I/usr/local/qt/mkspecs/default -I. \
                -I$(QT4)/include/QtGui -I$(QT4)/include/QtCore \
                -I$(QT4)/include
# Linker flags
             = $(SUBLIBS) -L$(QT4)/lib -lQtCore_debug -lQtGui_debug -lpthread
LFLAGS
              = -W1, -rpath, $(QT4)/lib
# macros for performing other operations as part of build steps:
QMAKE
              = /usr/local/qt/bin/qmake
###### Files
HEADERS
                   # If we had some, they'd be here.
SOURCES
              = main.cpp
OBJECTS
              = main.o
(snip)
```

to build "first," we must build "all"

= qapp # default target to build

1.

```
###### Implicit rules
.SUFFIXES: .c .o .cpp .cc .cxx .C
.cpp.o:
        $(CXX) -c $(CXXFLAGS) $(INCPATH) -o $@ $<
## Possible targets to build
all: Makefile $(TARGET) # this is how to build "all"
$(TARGET): $(OBJECTS) $(OBJMOC)
                                   # this is how to build qapp
        $(LINK) $(LFLAGS) -o $(TARGET) $(OBJECTS) $(OBJMOC) $(OBJCOMP) \
    $(LIBS)
       qmake: FORCE
                                    # "qmake" is a target, too!
        @$(QMAKE) -o Makefile qapp.pro # what does it do?
dist.
                                    # Another target
        @mkdir -p .tmp/qapp \
    && $(COPY FILE) --parents $(SOURCES) $(HEADERS) \
       $(FORMS) $(DIST) .tmp/qapp/ \
    && (cd `dirname .tmp/qapp` \ && $(TAR) qapp.tar qapp \
         && $(COMPRESS) gapp.tar) \
    && $(MOVE) `dirname .tmp/qapp`/qapp.tar.gz . \
    && $(DEL_FILE) -r .tmp/qapp
clean:compiler clean
                                    # yet another target
        -$(DEL FILE) $(OBJECTS)
        -$(DEL FILE) *~ core *.core
###### Dependencies for implicit rules
```

main.o: main.cpp

make 命令会检查依赖并执行 Makefile 中指定的每一个构建步骤。最终结果的名称和位置可以用(.pro)工程中的变量 TARGET 和 target.path 设置。如果没有给定 TARGET,则会以工程文件所在位置的目录名会成为默认的 TARGET 名称。如果没有给定 target.path,则用工程文件所在的目录作为默认的位置。

清空文件

qmake 生成的 Makefile 文件使得 make 可执行若干个有用的命令,有时也称为"伪" (phony)目标[©]。这些命令中的第一个, make clean,会移除在构建或者执行工程的过程中可能产生的全部的目标文件和核心转储(core dump)文件[©]。

① 之所以称为"伪",是为了和"真"目标文件进行区分。"真"目标通常是一些文件名,比如构建过程产生的可执行文件名。

② 核心转储文件有时是在执行程序遇到运行时错误并强制中断时产生的。它们给出了程序崩溃时主要内存的快照,或许可以用来确定造成崩溃的原因。

src/early-examples/example0> make clean

rm -f fac.o

rm -f *~ core *.core

src/early-examples/example0> ls

example0 example0.pro fac.cpp Makefile

src/early-examples/example0>

第二个命令, make distclean,除了会移除 make clean 移除的全部文件之外,还会移除 Makefile 和可执行文件,只剩下在其他机器上构建工程必须的那些文件。

src/early-examples/example0> make distclean

rm -f fac.o

rm -f *~ core *.core

rm -f example0

rm -f Makefile

src/early-examples/example0> 1s

example0.pro fac.cpp

src/early-examples/example0>

qmake 会在 Makefile 中生成其他的"伪"目标,但刚介绍过的那两个将会是最常用的。通过检查 Makefile 并查看那些标签,即以标示符开头且后跟一个冒号(比如,clean:)那些行,就可以熟悉其他的目标。要了解有关 make 的更多信息,可以参阅作者推荐的[Rehman03] 这本书。

注意

如果在 make 执行后又修改了项目文件,则应该在下次调用 make 前运行 qmake 以重新 生成 Makefile。

2 提示

make dist 命令会创建一个包含工程文件所知的全部源文件的压缩包(dirname.tar.gz)。

C.2 用于#include 文件的预处理器

在 C++中,代码的复用是在源文件中由预处理器命令#include 来指明的。被包含的头文件中包含诸如类或命名空间的定义、常量的定义、函数原型等内容。这些文件在编译器开始编译代码之前就被逐字地包含到自己的文件中了。

只要编译器发现某个标志符被定义了不止一次,它就会报告错误。编译器可以容忍重复声明,但不能容忍重复定义[©]。为了防止重复定义,我们总是谨慎地用一个#ifdef 将每个头文件包起来。它会告诉 C 预处理器跳过已经出现过的内容。下面看一下示例 C.2 中的类定义。

示例 C.2 src/preprocessor/constraintmap.h

#ifndef CONSTRAINTMAP_H #define CONSTRAINTMAP H

① 20.1 节讨论了声明和定义之间的区别。

```
#include <QHash>
#include <QString>
class Constraint;
                                                            2
class ConstraintMap : public QHash<QString, Constraint*> {
private:
                                                            3
   Constraint* m_Constraintptr;
                                                            4
   Constraint m_ConstraintObj;
                                                            5
   void addConstraint(Constraint& c);
} ;
#endif
            // #ifndef CONSTRAINTMAP H
1 前置声明。
2 需要 QHash 和 QString 的定义,但仅需要 Constraint 的声明,因为它是一个指针。
3 没问题, 这只是一个指针。
```

4 错误:不完整的类型。

#include "constraintmap.h"

5 使用前置声明。

正如所看到的,在函数的参数列表中,可以使用指针或者引用指向那些仅声明没定义的类。指针的解引用和成员的访问操作是在示例 C.3 中给出的实现文件中进行的。这里,必须使用#include 包含出所用到的全部类型的完整定义。

示例 C.3 src/preprocessor/constraintmap.cpp

```
ConstraintMap map;
#include "constraintmap.h"

Constraint * constraintP;

Constraint p;
#include <constraint.h>
Constraint q;

void ConstraintMap::addConstraint(Constraint& c) {
    cout << c.name();

6
```

- 1 没问题,已经包含了 ConstraintMap。
- 2 尽管多余,但只要已用#ifndef包住就会成为多余但无害的东西。
- 3 从 constraintmap.h.那里使用前置声明。
- 4 错误:不完整类型。
- 5 现在,这是一个完整类型了。
- 6 这里需要完整类型。

为了使头文件之间"强依赖"(strong dependency)的头文件最少,应当尽可能多地使用类的声明而不是在只要需要的地方就使用#include 头文件。为了判断到底是需要一个前置声明还是需要将整个头文件都包含到头文件中,这里有几条指导原则:

- 如果 ClassA 派生自 ClassB, 那么在处理 ClassA 的定义之前编译器必须知道 ClassB的定义。这样, ClassA的头文件中必须包含 ClassB的头文件。
- 如果 ClassA 的定义中包含一个成员, 而它是 ClassD 的对象, 那么 ClassA 的头文件必须包含 ClassD 的头文件。如 ClassA 的定义包含一个函数, 其参数之一或返回值是 ClassD 的对象, 那么 ClassA 的头文件也必须包含 ClassD 的头文件。
- 如果 ClassA 的定义仅包含未被解引用的 ClassE 指针,那么在 ClassA 的头文件中 有对 ClassE 的前置声明就足够了: class ClassE;。

只进行声明而没有进行定义的类称为不完整类型(incomplete type)。对不完整类型,试图解引用其指针或定义它的对象都会产生一个编译错误[©]。

ClassA的实现文件 classa.cpp 应当既包含 classa.h 又包含 ClassA 所使用的所有类的头文件(除非该头文件已经包含进了 class.h)。所有的指针解引用都应当在.cpp 文件中进行,这将会减小类之间的依赖性并且提高编译速度。

- 一个.cpp 文件永远不应该包含另一个.cpp 文件。
- 一个头文件应该尽可能少地包含其他头文件,这样它就能被快速地包含且只带有少量的依赖。
- 应该总是使用#ifdef 把头文件包起来,以防止它被多次包含。

循环依赖和双向关系

每当一个文件#include 另一个文件时,二者之间就会产生强依赖。当这样的依赖性出现在头文件之间时,它就不可能是双向的:预处理器没有能力处理头文件之间"彼此都包含对方"的这种循环依赖(circular dependency)。其中的某一个#include 语句必须替换成前置类声明。

前置声明有助于消除类之间的循环依赖,并且这个过程中允许它们存在双向关系。

C.3 了解链接器

图 C.1 显示了链接器如何接收由编译器生成的二进制文件并创建可执行二进制代码作为输出结果。链接器的可执行程序,在*nix 机器上称为 1d。在成功编译完所有的源文件之后,g++就会运行该命令。

上述所有的这些步骤都是在运行 make 命令时执行的, 而 make 命令通常会在执行每条命令之前将其打印出来。通过阅读 make 的输出结果, 就可以观察到哪些参数被传递给编译器和链接器。而当出现错误时, 错误提示将会紧跟产生错误的代码行被打印出来。

示例 C.4 给出了传递给 g++的命令行选项, 并且试图展示 g++如何运行链接器 1d 并给它传递一些参数。

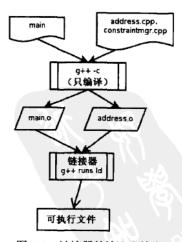


图 C.1 链接器的输入和输出

① 实际出错信息不一定总是很清晰,而且对于 QObject, 错误可能源于 MOC 生成的代码而不是自己的代码。

示例 C.4 linker-invocation.txt

g++	-Wl,-rpath,/usr/local/qt-x11-free-3.2.3/lib	1
	-o hw7	2
	.obj/address.o .obj/ca_address.o .obj/constraintgroup.o	
	.obj/customer.o .obj/dataobject.o .obj/dataobjectfactory.o	
	.obj/hw07-demo.o .obj/us_address.o .obj/moc_address.o	3
	.obj/moc_ca_address.o .obj/moc_customer.o .obj/moc_dataobject	.0
	.obj/moc_us_address.o	
	-L/usr/local/qt-x11-free-3.2.3/lib -L/usr/X11R6/lib	
	-L/usr/local/utils/lib	4
	-lutils -lqt -lXext -lX11 -lm	5

- 1 告诉 g++运行链接器, 并把这些选项传递给 1d。
- 2 指定输出文件的名称为 hw7。
- 3 把这些目标文件链接为可执行文件。
- 4 添加另外一个位置作为链接器搜索库的位置。
- 5 将此应用程序再与 4 个库进行链接: qt 工具库、ext、X11 和 m。

链接命令的详细步骤如下。

- 对于每个带有开关-1 的库名称,搜索库路径以及所有开关-L 的参数(由 qmake 通过 指定 qmake 变量 LIB 来生成),来找到它所对应的库文件。
 - 对于静态库,它包含了将被链接进可执行文件的二进制代码。
 - 对于动态库,它是一个目录列表(通常采用可读的 ASCII 格式),其中描述了每个标签定义所对应的实际共享对象的位置。链接器将检查并确认那些共享对象确定位于正确的位置,否则就会报错。
- 对于正在链接的代码中调用的所有函数,找到其代码所在的对象并执行一个简单快速的检查,以确定在该位置确实存在一个定义完整且具有恰当名称/函数签名的函数。
 若找不到,或者名称/类型/大小有误,则报错。
- 对于变量名的每个引用,找到该变量所处的对象地址,并执行一个简单快速的检查, 以确定这个地址对于该类型的对象来说是有效的。

总体的思路是:链接器解析对名称的引用,途径是查找它们在文件中的实际地址并检查 该地址对该类型是否有效。这类似于为 C++编译器提供了一个字典查找服务。

C.3.1 常见的链接器错误信息

C++程序员经常花费大量时间来试图理解并修复编译和链接错误。如果不能理解那些消息,就无法继续。对于编译错误,往往可以很容易地诊断问题所在,因为它只与一个源码模块和它所包含的头文件的编译有关。编译器通常会告诉用户它所侦测到的每个错误的精确位置。而对于链接错误,问题涉及到源代码模块是如何链接到一起的。当到达链接阶段时,所有单个模块都已被无错地编译完毕。链接错误可能产生于 C++代码中的 bug,但也可能是工程文件问题所导致的。以下是四种错误类型。

1. Error: Unable to find libxxx.so.x

对于 Win32 用户

在创建过程中, IDE 需要能够找到.DLL 文件。要避免这种麻烦,研究一下菜单结构直到你找到 project->properties->c/c++build->libraries。在这里,你可以添加一个第三方的库,当然此时会有一个对话框来询问头文件和.DLL 文件的位置。

在运行时刻,系统环境变量 PATH 必须包含所需.DLL 文件所在的路径。

安装一个库意味着使其对系统上不只一个用户可用。我们有可能复用一个库而不去安装它,复用的所有库或者必须进行安装,或者必须位于 LD_LIBRARY_PATH 所指定的目录中。

当第一次复用一个库时,可能会看到这条出错的消息,它说明链接器无法找到库。当 gnu 链接器查找共享对象时,它至少会检查两个地方:

- 1. LD_LIBRARY_PATH 指定的目录。
- 2. 名称为/etc/ld.so.cache 的 cache 文件中列出的已安装库。

Cache 文件: ld.so.cache

cache 文件对在/etc/ld.so.conf 指定的目录中发现的共享对象提供快速查找。在此处能够发现的目录包括:

/lib
/usr/lib
/usr/X11R6/lib
/usr/i486-linuxlibc1/lib
/usr/local/lib
/usr/lib/mozilla

如果使用一个 Linux 包安装程序来安装一个库,它将会对 Id.so.conf 做出恰当的 改动并重建 cache 文件。然而如果是手动编译并安装库,那么必须修改这个文件。之后可以通过 ldconfig 命令来重建 cache 文件。

2. undefined reference to identifier

这是最常见的也几乎是最讨厌的链接错误。它的意思是链接器无法找到代码中某个命令 实体的定义。下面是 make 命令的一些输出。

.obj/ca_address.o(.gnu.linkonce.t._ZN10DataObject16getConstraintGroupEv+0x4):
In function `DataObject::getConstraintGroup()':

/usr/local/qt-x11-free-3.2.3/include/qshared.h:50:

undefined reference to `DataObject::s Cm'

collect2: ld returned 1 exit status

make: *** [hw7] Error 1

编译器找到了声明,但链接器却找不到相应的定义。在代码中的某部分引用了一个符号, 但是找不到其定义。下面是一些有用的信息:

- 它找不到的那个符号是 DataObject::sm Cm。
- 试图使用它的函数是 DataObject::getConstraintmgr。

解决问题的第一步是确认,你能否找到那个丢失的定义。如果无法找到,那么链接器如何可以呢?如果在.cpp 文件中找到了它,那么你必须确认下面的两件事情。

● .cpp 和.h 文件都在工程中。

● 这个文件被我们正在连接的某个库包含。

由于正在使用良好的命名约定(参见 3.1 节), 随意马上可以断定 sm_Cm 是 DataObject 类的一个静态数据成员。编译器发现了声明,但链接器无法找到定义。

而由于它是静态的(参见 2.9 节), sm_Cm 的定义隶属于 dataobject.cpp。编译器期望找到一个如下形式的定义语句:

ConstraintGroup DataObject::s_Cm;

如果它在那里,但链接器仍旧无法找到它,那么最有可能的原因是:

- 包含该定义的.cpp 文件没有在.project 文件中的 qmake 的 SOURCES 中列出。
- 代码位于其他库中但链接器无法找到那个库。这时可以通过在工程文件中添加一个遗漏的 LIBS 参数来解决问题。
- -lmyLib 添加一个要链接的库。
- -LmyLibDir 向链接器的库搜索路径列表中添加一个目录。
- 3. Error: unresolved external symbol

当使用 Microsoft 编译器链接自己的库时,可能会出现这样的错误:

customer.obj : error LNK2001: unresolved external symbol
"public: virtual bool __thiscall DataObject::readFrom(class QObject const &)"
(?readFrom@DataObject@@UAE NABVQObject@@@Z)

这通常意味着,在构建 DLL 时该符号未被导出。确保在该类或者函数声明之前有一个 Q_DECL_EXPORT 宏,然后再次构建该 DLL。参见 7.1.2 节。

4. Error: undefined reference to vtable for ClassName

这是众多最令人费解的错误之一,它通常表示缺少一个虚函数定义。从字面意思来说,就是该类的 vtable(它存有每个虚函数的地址)无法完整地创建。

代码中缺少了函数定义就会产生这种错误,而在make/project文件的HEADERS或SOURCES中缺少某一项也会导致这种问题。如果你最近在某个已存在的头文件中添加了 Q_OBJECT 宏,那么需要重新运行 qmake, 因为 Makefile 需要重新生成。这方面的更多细节请参阅 8.4 节。

2 提示

在这类链接器错误后,首先检查所有文件是否都恰当地列在了工程文件中。所有的 QObject 都有一个 Q_OBJECT 宏,试着执行命令 qmake && make clean,然后看看是否还会生成这个错误。

完全内联类(All-Inline Classes)

对于多态类,在与头文件相对应的源程序文件(.cpp)[©]中至少要有一个非内联的定义(函数或静态成员)。如果没有,则很多链接器将无法找到它的任何一个虚方法定义并报出类似的错误。

完全内联类在 C++中是合法的, 但是当与多态混用时它将不能按预想的方式工作。

① 多态是指拥有至少一个虚方法的类。

C.4 调试

编译器可以定位并描述出语法上的错误,而链接器可以揭示出程序组件之间的不一致并 有助于我们正确地定位它们。在 C++中最具有挑战性的方面是学习如何找到并修复各种各样 的运行时错误。

运行时错误(run-time errors)是在一个语法正确且不包含未定义对象和函数的程序逻辑错误。通过有效地使用调试器(一个专门设计的、用于追踪运行时错误的程序),能够极大减少花在处理这些错误上的时间。

一个调试器允许单步地执行代码,还允许查看对象值。由于调试器是工作在编译后的代码之上的,早期的版本只能由熟悉汇编语言的程序员使用。现在的调试器能够同时在编译后的机器代码和原始程序代码上面工作。GNU 的开发工具家族包含了 gdb,这是一个可以用于C/C++程序的源码级 GNU 调试器。gdb 设计有一个命令行接口,这个接口非常强大但对用户不甚友好。幸运的是,gdb 有了一些开源的图形界面外壳,我们下面会讨论其中的一个。商业的 C++ IDE(如 Visual Studio)通常有内建的源码级调试器。

C.4.1 建立一个可调试的目标

为了让 gdb 能够工作,就必须在编译时把调试符号创建到代码当中,否则机器指令将无法正确映射到 C++原文件中的对应位置上。我们可以在调用编译器时使用适当的命令行开关 (-g)来简单地完成这件工作:

g++ -g filename.cpp

这通常会生成一个尺寸大得多的可执行文件。一般来说,这个尺寸增长是与源代码的尺寸和复杂程度成正比的。这个扩展了的可执行文件包含符号表信息,调试器可以用它来找到与机器码对应的源代码。要使得 qmake 生成的 makefile 开启调试选项,在 qmake 工程文件中加入下面一行:

CONFIG += debug

当创建带有调试符号的 Qt 库时,就可以像操作自己的代码一样单步调试 Qt 的源代码。 有些程序会包含由 Qt 库直接进行调用的代码,在调试这些特定的程序时,可能需要创建带有 调试符号的 Qt 库。

提示

在 Win32 中,这是一个可以单击的菜单选项。而在*nix 平台上,首先解开源代码的 tar 压缩文件,然后在创建之前给 configure 脚本传递一个参数,这样在创建 Qt 库时就会包含调试符号了。

./configure --enable-debug make

make install

C.4.1.1 练习:建立一个可调试的目标

为了更好地理解调试的代价:

- 比较在工程文件中有和没有 CONFIG+=debug 的情况下创建的可执行文件大小。
- 记得稍候后用一个更复杂的应用再试一下。

C.4.2 gdb 入门

设想你正在运行一个程序, 但不知什么原因, 程序崩溃了。

```
[lazarus] app> ./playlistmgr
Segmentation fault
[lazarus] app>
```

当应用程序中止或崩溃时,如果精确地知道(尽可能快地)事故在哪里发生将是很有用的。 我们可以使用 gdb 来快速简单地定位故障点。下面是一个 gdb 命令行会话的例子。

```
[lazarus] app> gdb playlistmgr
GNU gdb 6.3-debian
Copyright 2004 Free Software Foundation, Inc.
This GDB was configured as "i386-linux"...Using host libthread db library "/lib/
tls/libthread db.so.1".
(gdb) r<sup>0</sup>
Starting program: ftgui/app/playlistmgr
[Thread debugging using libthread_db enabled]
[New Thread -1227622176 (LWP 17021)]
Qt: gdb: -nograb added to command-line options.
         Use the -dograb option to enforce grabbing.
This is a debug message
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread -1227622176 (LWP 17021)]
0xb7f03320 in FormDialog::createActions (this=0x80ae2a0) at formdialog.cpp:53
            delete m_Okaction;
53
(gdb)
```

gdb 不仅可以向用户显示文件名和行号,还可以显示相应的源代码行。然而,我们可能仍然想要得到这个错误的上下文。list 命令显示当前文件中该错误周围的一些代码:

```
(qdb) list
       void FormDialog::createActions() {
51
52
53
           delete m OkAction;
           delete m_CancelAction;
54
           m_OkAction = new OkAction(m_Model, m_View);
55
           m_CancelAction = new CancelAction(m_Model, m_View);
56
           QHBoxLayout *buttons = new QHBoxLayout(0);
57
where 命令的作用是显示栈的内容或者显示如何到达该处。
(qdb) where
#0 0xb7f03320 in FormDialog::createActions (this=0x80ae2a0) at formdialog.cpp:53
#1 0xb7f03058 in FormDialog::setModel (this=0x80ae2a0, fmodel=0x80c80d0)
   at formdialog.cpp:34
```

① r表示的是"运行"命令。

- #2 0x080664bd in SettingsDialog (this=0x80ae2a0, parent=0x0) at settingsdialog.cpp:14
- #3 0x0805f313 in MainWindow (this=0xbfffdec8) at mainwindow.cpp:42
- #4 0x08066f14 in Controller (this=0xbfffdec0, argc=1, argv=0xbfffdfe4) at controller.cpp:25
- #5 0x0805a8a4 in main (argc=1, argv=0xbfffdfe4) at main.cpp:7
 (gdb)

诸如 Eclipse 和 QtCreator 这样的多数开源 IDE 在后台使用的都是 gdb,它们分别提供了一个用户接口,以使某些功能更易于学习和使用。ccdebug[©]是使用 Qt 编写的,且专门设计用于调试 Qt 应用程序,并提供 QString 支持。

提示

在一些调试器内 QString 是很难看到的,因为它们是间接指向 Unicode 数据的指针。调试器要能正确显示 OString 的内容需要知道额外的信息。

如果你使用的是安装了带有 Qt 集成工具的 IDE, 应当可以在调试器内看到 QString。

如果使用的是 QtCreator 并且无法看到 QString, 可以到 Qt settings 中,检查是否可以 为你所使用的 Qt 构建调试帮助。

如果正在使用的是 gdb 命令行,可以下载 kde-devel-gdb[©],这是 KDE subversion 仓 库中的一个 Qt 帮助宏集合,然后把以下内容放在~/.gdbinit 中。

source /path/to/kde/kde-devel-gdb

define pqs

printq4string \$arg0

end

现在应该就能够在调试时使用 pqs 宏来显示 QString 了。

提示

可以使用.gdbinit 文件自动加载一个可执行文件,设置一些断点,并开始调试任务。 一旦发现在运行 gdb 时会重复输入相同的命令,可以试着将该命令序列加到工程所在目录下的.gdbinit 文件。

C.4.3 查找内存错误

若没有运行时分析工具的帮助内存错误是非常难以跟踪的。用来分析程序运行性能的程序称为 profiler(分析器)。Valgrind 是一个开源的 Linux 分析工具,用于跟踪代码的内存和 CPU 使用率并检测各种错误。这些错误包括:

- 内存泄漏(memory leak)——不可再被使用却又没有被删除的内存。
- 堆内存的非法指针使用,比如
 - 索引越界
 - 分配和释放的语法不匹配(例如,用 new[]分配却用 delete 释放)

① 参见http://sourceforge.net/projects/ccdebug。

② 参见http://websvn.kde.org/*checkout*/trunk/KDE/kdesdk/scripts/kde-devel-gdb。

使用未初始化的内存

debugging/wrongdelete>

刚才列出的每个错误都能在一小段程序中产生灾难性的结果。分析器也能够用于性能的 调试以及确定哪些代码对拖慢程序的速度负有责任(即找速度瓶颈)。

示例 C.5 的程序包含了一个有意为之的内存使用错误。

示例 C.5 src/debugging/wrongdelete.cpp

```
void badpointer1(int* ip, int n) {
    ip = new int[n];
    delete ip;
}

int main() {
    int* iptr;
    int num(4);
    badpointer1(iptr, num);
}

1 错误的删除语法。

为了使输出结果更易读,我们加入了调试符号(-g)。
debugging/wrongdelete> g++ -g -Wall wrongdelete.cpp
debugging/wrongdelete> ./a.out
```

编译器并没有报错,而且甚至是在程序运行以后,也没表现出错误行为。尽管如此,该 程序破坏了内存。

让我们看一下 valgrind 对我们程序的分析(略为删减),我们去掉了每行开头的 valgrind 作业的进程 id。这个进程 id 在每次运行 valgrind 时都会不同。

```
src/debugging> valgrind a.out
--3332-- DWARF2 CFI reader: unhandled CFI instruction 0:50
--3332-- DWARF2 CFI reader: unhandled CFI instruction 0:50
Mismatched free() / delete / delete []
    at 0x401C1CB: operator delete(void*) (vg_replace_malloc.c:246)
    by 0x80484BD: badpointer1(int*, int) (wrongdelete.cpp:3)
    by 0x80484F4: main (wrongdelete.cpp:9)
Address 0x4277028 is 0 bytes inside a block of size 16 alloc'd
    at 0x401BBF4: operator new[] (unsigned) (vg_replace_malloc.c:197)
    by 0x80484AC: badpointer1(int*, int) (wrongdelete.cpp:2)
    by 0x80484F4: main (wrongdelete.cpp:9)
```

valgrind 发现了错误,且根据调试符号就能够指出问题代码的位置。示例 C-6 稍微有趣一些,因为它存在内存泄漏和数组索引错误。

示例 C.6 src/debugging/valgrind-test.cpp

```
#include <iostream>
int badpointer2(int k) {
  int* ip = new int[3];
  ip[0] = k;
```

```
1
  return ip[3];
                                                                2
}
int main() {
 using namespace std;
  int* iptr;
                                                                3
                                                                4
  int num(4), k;
  cout << iptr[num-1] << endl;
                                                                5
  cout << badpointer2(k) << endl;
                                                                6
1 越界的索引。
2 内存泄漏:分配的内存已不再可用。
3 iptr没有初始化。
4 k 没有初始化。
5 iptr 的状态是什么?
6 向函数发送了未初始化的 arg。
用 valgrind 运行示例 C.6 得出了一些错误的精确位置:
For more details, rerun with: -v
--2164-- DWARF2 CFI reader: unhandled CFI instruction 0:50
--2164-- DWARF2 CFI reader: unhandled CFI instruction 0:50
Use of uninitialised value of size 4
   at 0x80486AF: main (valgrind-test.cpp:17)
68500558
Invalid read of size 4
   at 0x804867C: badpointer2(int) (valgrind-test.cpp:8)
   by 0x80486DD: main (valgrind-test.cpp:18)
 Address 0x4277034 is 0 bytes after a block of size 12 alloc'd
   at 0x401BBF4: operator new[](unsigned) (vg replace malloc.c:197)
   by 0x8048667: badpointer2(int) (valgrind-test.cpp:6)
   by 0x80486DD: main (valgrind-test.cpp:18)
ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 19 from 1)
malloc/free: in use at exit: 12 bytes in 1 blocks.
malloc/free: 1 allocs, 0 frees, 12 bytes allocated.
For counts of detected errors, rerun with: -v
searching for pointers to 1 not-freed blocks.
checked 120,048 bytes.
LEAK SUMMARY:
   definitely lost: 12 bytes in 1 blocks.
     possibly lost: 0 bytes in 0 blocks.
   still reachable: 0 bytes in 0 blocks.
        suppressed: 0 bytes in 0 blocks.
Use --leak-check=full to see details of leaked memory.
```

如果根据这些信息还不足够找到内存在哪泄漏,可以重新运行 valgrind 并使用 --leak-check=full 开关。在示例 C.7 中,我们修复了其中的一些错误。

示例 C.7 src/debugging/valgrind-test2.cpp

```
int notSoBadPointer(int k) {
  int* ip = new int[3];
 ip[0] = k;
                                                                         1
 delete[] ip;
                                                                         2
 return k;
}
int main() {
 using namespace std;
                                                                         3
 int* iptr;
                                                                         4
 int num(4), k(4);
 cout << iptr[num-1] << endl;</pre>
 cout << notSoBadPointer(k) << endl;</pre>
}
```

1 清除内存泄漏。

#include <iostream>

- 2 一个可返回值。
- 3 未初始化的指针!
- 4. 至少 k 不再未初始化。
- 5. 这里有问题!

在不使用 valgrind 的情况下,编译并运行这个稍经修正的测试程序不会产生警告或者错误且输出了一个没有意义的值:

```
src/debugging> g++ -g -Wall valgrind-test2.cpp
src/debugging> ./a.out
-1078391036
使用 valgrind 运行它生成的错误比先前少一些:
src/debugging> valgrind ./a.out
For more details, rerun with: -v
Use of uninitialised value of size 4
   at 0x8048794: main (valgrind-test2.cpp:18)
-1096641724
4
ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 18 from 1)
malloc/free: in use at exit: 0 bytes in 0 blocks.
malloc/free: 1 allocs, 1 frees, 12 bytes allocated.
For counts of detected errors, rerun with: -v
All heap blocks were freed -- no leaks are possible.
src/debugging>
```

最后,在示例 C.8 中,我们清除了最后一个错误。

示例 C.8 src/debugging/valgrind-test3.cpp

```
#include <iostream>
int notSoBadPointer(int k) {
  int* ip = new int[3];
  ip[0] = k;
 delete[] ip;
                                                                2
  return k;
int main() {
 using namespace std;
                                                               3
  int num(4), k(4);
 int* iptr = new int[num] ;
  for (int i = 0; i < num; ++i)
    iptr[i] = i;
                                                                5
 cout << iptr[num-1] << endl;
 cout << notSoBadPointer(k) << endl;
 delete[] iptr;
}
1 清除内存泄漏。
2 一个可返回值。
3 至少 k 不再是未初始化。
4 没有未初始化指针。
5 不再有问题了。
编译、运行并再次用 valgrind 运行:
src/debugging> g++ -g -Wall valgrind-test3.cpp
src/debugging> ./a.out
src/debugging> valgrind ./a.out
4
 ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 18 from 1)
 malloc/free: in use at exit: 0 bytes in 0 blocks.
 malloc/free: 2 allocs, 2 frees, 28 bytes allocated.
 For counts of detected errors, rerun with: -v
 All heap blocks were freed -- no leaks are possible.
src/debugging>
```

在 Mac OS X 下, Valgrind 尚无法使用, [©]但是在 Mac Developer Tools 中包含了一个称为 MallocDebug.app 的图形化工具,它可以代替 valgrind 的一些功能。

C.5 开源开发工具、库和集成开发环境

下面是一些构建与 Qt 之上的开源库, 提供了一些额外的可复用组件:

① 可以用源代码进行安装,源代码可以在 http://valgrind.org/downloads/repository.html 中找到。

- Qwt[©]——用于科技类应用程序的 Qt (Qt Widgets for Technical Applications)。
- Oxt^②——Ot 扩展库 (Ot eXTension Library)。

C.5.1 集成开发环境

使用普通的文本编辑器进行面向对象开发是不太实际的。面向对象开发工作涉及到众多的类以及各种其他的文件(头文件和源代码文件)。在编辑窗口中写代码只是开发进程中的一部分,一个优秀的程序编辑器或者 IDE(集成开发环境)应当支持下列众多特性:

- 每个文件中对象/成员的树状结构导航。
- 重构(Refactoring)帮助,用于成员的移动和改名。
- 集成的调试器。
- 上下文敏感的链接到 API 文档的帮助。
- 内建的命令行外壳窗口,这样就能在不离开环境的情况下运行程序。
- 一个工程管理器,用来管理相关文件分组和子分组。
- 可用于其他语言的编辑模式。
- 易用的键盘制定能力,可以让任何键击都可以执行任何一项功能(尤其是光标移动, 当然也包括窗口的移动)。
- 开放的插件构架,以便添加其他组件。
- 若能集成版本控制功能则更好,尤其是在窗口环境中。可参看 Subversion[®]或其他更高级的分布式版本控制系统,如 bzr, mercurial, git, monotone 或者 darcs。
- 易学习、可编码的宏。
- 根据语言特性在不同文件之间跳转(有套找声明、套找定义、套找引用等快捷方式)。

C.5.2 开源集成开发环境

Nokia/Qt Software 已发布其自由、开源的 QtCreator[®], 现在已包含在 Qt SDK 中。QtCreator 是一个用 Qt 和 C++ 编写的 IDE。它为 qmake/cmake 工程、上下文自动补全、代码导航、重构和集成化代码调试等提供完整支持。我们认为,无论是对初学者还是专家来说,它都是一个在所有主流平台上开发基于 Qt 的 C++编程集成开发环境的理想 IDE。

在使用 C++/Ot 开发前, Mac OS X 的用户或许需要先安装有 XCode。

对于所有平台,还有一个基于 Java 的自由、开源集成开发环境: Eclipse[®]。诺基亚提供了一个 Eclipse 集成软件包[®],提供了许多功能,使 Eclipse 成为一个适用于 Qt 开发的集成开发环境。首先需要安装 C/C++开发工具(C/C++ Development Tools, CDT)或者一些用于 C++开发的插件[©]。Qt 集成工具使得你可以直接将 qmake 的.pro 文件导入成 Eclipse 的工程文件,并可将设计师的可停靠窗件停靠在 Eclipse 的主窗口中,以便可以使用设计师的全部功能而无须离开 Eclipse。此外,将 Qt API 的文档集成到 Eclipse 的上下文敏感帮助系统中也是可行的。

① 参见http://qwt.sourceforge.net/。

② 参见http://docs.libqxt.org/index.html。

③ 参见http://subversion.tigris.org/。

④ 参见http://qt.nokia.com/products/developer-tools。

⑤ 参见http://www.eclipse.org。

⑥ 参见http://qt.nokia.com/developer/eclipse-integration。

⑦ 参见http://www.eclipse.org/cdt/。

C.5.3 UML 建模工具

为了能够使用统一建模语言(UML)创建本书中的图表,我们使用了两个开源工具: Umbrello^①和 Dia^②。这两个工具都使用某种 XML 方言(XML dialect)作为其原生文件格式。

如图 C.2 所示, Umbrello 是 KDE 中的 UML 建模器 (Modeler)——它可以直接导入 C++ 代码, 这使得拖拽导入类到图表中变得非常容易。

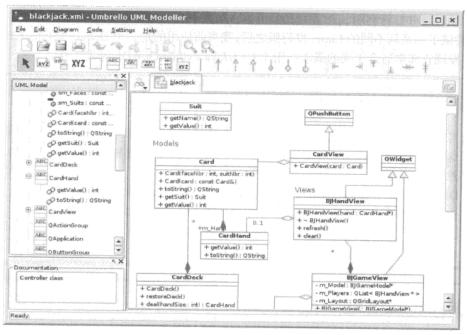


图 C.2 Umbrello 的屏幕截图

Dia 是一个 Gnome 实用工具,它是一个带有某些 UML 特性的通用性更强的图表工具。有很多插件和实用工具可用于往/从 Dia 中导入代码、导出图形到其他语言和格式。

① 参见http://uml.sourceforge.net/index.php/。

② 参见http://www.gnome.org/projects/dia/。

附录 D Alan 的 Debian 程序员快速指南

这篇文章总结了我用来建立用于开发工作的新 Debian 桌面的步骤。这对不熟悉 Debian Linux 桌面环境并打算利用 apt 系统的新开发人员来说会非常有用。

在尝试了一系列不同的 Linux 发行版之后,我当前使用的是一个 Debian Testing KDE 安装工具[©],发现它非常适合于我的需求:用于 C++/Java/Python/Qt/KDE 开发工作。



问题: 为什么要用 Debian

- 1. 可用软件包的数量庞大的令人觉得可笑——许多软件包都难以从其他发行版中找到。 这就意味着,你可以大把抓到几乎每一个库已编译的二进制软件包,而不是你自已从 源代码开始构建。
- 2. Debian 测试版 (Debain testing)似乎包含了最新的 Qt 和 KDE 软件包。
- 3. 许多 KDE/Ot 开发人员都是在 Debian 或者 Kubuntu 中完成工作的。
- 4. Debian 和 Ubuntu 的发行版提供了一个强大的名称为 apt 的软件包管理系统,它用起来相当容易,而且在出现问题时可以很轻松地进行修复。



问题: 为什么要用 KDE

KDE 是这本书的两位作者最喜爱的桌面环境。我们两个都使用和喜欢 KDE 的桌面环境 许多工具,而不是许多其他桌面环境(例如, GNOME)中相类似的应用程序。除了设计精良和 用户环境友好之外,它还拥有自身是用 Qt 和 C++编写的这一优势。

推荐基于 Debian 的 Linux 发行版

以下给出的各发行版都有一些重要的共同点:它们是可自引导的,这就意味着可以从 CD 或 USB 记忆棒中的映射直接引导你的计算机并运行 Linux,而无须安装任何东西到你的硬盘中。但你也可以用同一个 CD 或 USB 记忆棒中在你的硬盘上安装一份完整的基于 Debian 的 Linux。举荐的所有这些发行版都是自由(免费)的,可以很容易地从各种网站下载到:

1. Debian[®]一直是过去几年中主要的几个"基本"发行版之一。最初,Debian 的卓越性能是公认的,但对于初学者来说,它的安装太困难了。其他几个发行版基本上都是对Debian 的简化,提供简单、用户友好的安装脚本。硬件检测和识别一直是安装系统时主要的难点领域之一。Debian 的优势之一(可以自定义安装,以便可以准确适用于硬

① KDE 是一个桌面和一系列工具集,与基于 Gtk 的 Gnome 不同,该工具集是完全用 Qt 编写的。

② 参见http://www.debian.org。

件) 也是它的主要缺点(安装系统时必须知道如何找到硬件驱动程序)。Debian 的近期版本已经有了更为友好的安装脚本。可以下载一个"netinst"ISO 映像,将其刻录到CD 或者 USB 记忆棒中,从它启动,并只需一点点儿工作即可安装到硬盘中。推荐使用 debian.org[©]中最新的 debian-live-KDE-desktop.iso CD 映像安装包,因为它是以liveCD 的形式启动并可安装一个最新的 KDE 桌面。

- 2. Ubuntu[©]可能是本书出版之时最为流行的基于 GNOME 的发行版。
- 3. Kubuntu[®](基于 KDE 的 Ubuntu)和 Ubuntu 类似,但它是基于 KDE 的,而不是基于 GNOME 的。



问题

稳定? 测试? 不稳定?

这听起来很吓人。为什么会有人愿意使用其他的而不是稳定的操作系统?许多 Debian 新手都会被用来形容任意 Linux 发行版的"不稳定"一词所吓倒。因为其他两个可用的选项是"测试"和"稳定",人们自然会尽量先去试用稳定版。遗憾的是,稳定版中提供的大多数应用程序和库都已经相当陈旧(至少一年,有时甚至更久远),还有一些你最喜欢的软件包可能尚不可用。对于桌面/开发系统来说,可能更倾向于使用较新的软件。"测试"是一个处于不稳定和稳定之间的妥协产物。通常情况下,Debian 测试版的安装程序也支持更新一些的硬件。在开发过程中,会给 Debian 的版本取些昵称,这些名称源于自玩具总动员(Toy Story)电影系列^⑤。

稳定版(三眼仔, squeeze)是当前的稳定版本。它包括了 Qt 4.6。这些可用的版本都经过了彻底的测试。

测试版(企鹅吱吱, wheezy, 2011 年)目前正在使用 Qt 4.7, 并含有稳定版中没有但将被包含在下一个主要发行版中的版本。

不稳定版(席德, sid)含有比测试版更多的更新版本,根据决定升级系统的不同时间,或许会发现需要自己修复那些无法安装的软件包(broken package)。只有在使用 apt 系统相当顺手以后才建议你使用"席德"版。

D.1 apt 系统

apt 系统是一个用来管理软件包及其依赖关系的工具。下列程序可以用来管理基于 apt 的软件库:

- apt-get,一个让获得和升级软件包变成轻松事情的简便程序。
- dpkg, 软件的低层。许多简单的 apt 命令都会被转换成更为复杂的命令并由 dpkg 加以执行。
- apt-cache, 一个帮助你在本地 apt 软件包信息数据库中搜索的工具。

① 参见http://cdimage.debian.org/debian-cd/current-live。

② 参见http://www.ubuntu.com/。

③ 参见http://www.kubuntu.org/。

④ 文章 http://en.wikipedia.org/wiki/Debian 较为详细地介绍了这一点。

● aptitude, 一个稍微智能些的 apt-get。当 apt 无效时,它可以用来获得和移除 软件包。这是 dpkg 之上的另一个简便层。

以下是一些有用的提示和关于 apt 系统以及如何使用 apt 的基本知识。

etc/apt/sources.list

这个文件包含了一个 apt 用来检查软件包的资源列表。在安装新软件之后,应当对这个资源列表做两件事情:

修改主镜像(main mirror)文件,以便让它们指向本地镜像[©]而不是指向远程镜像。实现这一点可以有很多方法,但最为用户友好的方式是运行 apt-setup。

可选项:如果打算让发行版中的每样东西都是最新的,可以把来源设置成"不稳定" (Debian)或者"edgy"(Kubuntu)[©]。edgy 是 Kubuntu 中大约 6 年前的一个老版本。

apt-get 更新会从软件来源中下载软件列表,以便可以有一个清单、依赖关系和自己dpkg 数据库说明的本地副本。

有时,可能需要某个软件包但没有记住它的具体名称。命令 apt-cache search search-string 会对本地已下载的软件包清单进行搜索,从每个软件包的名称或者描述中查找该字符串的出现地方。如果返回的清单中含有所查找的软件包,就可以从网络上把它轻松拉取下来并予以安装。

一旦知道了软件包的名称,就可以使用命令 apt-cache show pkgname 阅读它的完整说明。

apt-get install packageName 是功能最强大的命令,在基于 Debian 的系统中,只有拥有 root 权限才可以使用该命令。

apt-get remove packageName, 顾名思义,会完全移除给定名称的软件包。

为了节约时间,下面列出了一些在 Debian 系统上进行 C++开发时建议安装的软件包。注意插入注释的"#"的用法。

alias for following commands:

alias agi='apt-get install -y'

Database Stuff

agi mysql-server mysql-client libmysqlclient15-dev

agi sqlite3 libsqlite3-dev

For development

agi build-essential manpages-dev # manual pages for stdlib

agi global cscope exuberant-ctags # For C++ development - navigation and doc generation

agi libqt4-dev qt4-dev-tools libphonon-dev

agi gstreamer0.10-fluendo-mp3 # for playing mp3 files in phonon

agi libtag1-dev libtag1-doc # taglib used by libfiletagger

agi gdb # gnu debugger

agi umbrello # UML Diagramming tool that reads/writes XMI and imports C++ source

① 参见http://www.debian.org/mirror/list。

② 仅 Ubuntu: 确保在 main 之后加一个"universe",以便可以获得由 Ubuntu 提供的完整软件包。

更多的 apt 中提示

apt-get (dist-|dselect-) upgrade

随着时间的流逝,会有新的安装包出现在你的软件仓库中。在你打算升级系统时,执行一下 dist-upgrade 会很合适。为了给 apt 赋予权限以移除那些陈旧软件包并替换成更新的软件包,可以使用 apt-getdselect-upgrade 命令(这个命令与 aptitude upgrade 命令相似但不完全一样)。

apt-get source packageName

除非需要的是一个软件包源无法提供的特定版本,只需通过简单请求 apt 即可方便地获得任何可以获得的软件包的源代码压缩包。

我刚安装了一样东西。它跑哪儿去了?

对于这个命令,需要深入研究一下 dpkg。某种意义上说,dpkg 是一个功能强大的工具,但大多数情况下,我们只是借助 apt-get 而间接用到它。dpkg -L packageName 是其中一个很有用的选项,这个命令会列出从这个软件包中提取的全部文件并给出它们在系统中的当前位置。

apt-get build-dep packageName

有时,在编译大的软件包(如 Qt)时,一些库(或者它们的开发包)的缺失会造成构建的失败。当我还在以自己的方式研究我的第一个基于 Debian 系统时,我是通过以下这种暴力性质的重复过程来构建应用程序和库的: 配置(configure),遇到并检查每一个错误信息,试着找出遗漏了什么库,安装它,继续重复直到没有错误为止。在我得知 apt-get build-dep之后,即刻意识到我原本可以节约那么多的时间,这是多么微不足道的一件事情啊!

你已经知道 apt 系统可以知道哪个软件包会依赖另外哪一个软件包,但使用 apt-get build-dep packageName 命令,apt 就会自动下载所有那些依赖的库及其开发包,以便可以拥有构建 packageName 所需的全部东西。换句话说,再也不需要去跟踪是哪些头文件被遗漏了!

为了能够看到各个软件包之间的依赖关系,可以试一下命令 apt-cache showpkg packageName。

aptitude:在 apt 系统无法使用时

有时,在试图安装东西时遇到了错误,但此时 apt 也已经处于一种无法使用的状态了。 在运行非稳定版时,这非常常见,在使用测试版时会少一些。或许,你会打算试一下 apt-get -f install,但试验后,还是需要试着移除那些让人不舒服的软件包,但你还得继续忍着。

仔细阅读错误信息很重要: 绝大多数情况下,你会看到一些指向造成这一问题的具体软件包的名称。通过在同一行内使用 apt-get remove pkg1 pkg2 pkg3 移除它们,有时可以把系统重新带回可用的状态。

修复这一问题的另一种方法,是对每个软件包都使用 aptitude remove pkgN。 aptitude 可以帮助你找到并移除造成问题的相关软件包集。

D.2 update-alternatives

因为在 Debian 系统上可同时存在同一个程序的多个替代版本(尤其是诸如 qmake 和 java 这样的程序),会有一些从/usr/bin 到这些程序所需版本的符号链接。为了管理这些链接并将其修改指向所需的版本,root可以使用 update-alternatives 命令。

例如,为了选择要运行的 qmake 的默认版本, root 可以使用

update-alternatives --config qmake

为了查看每个已安装的有可替换版本的程序清单, root 可以键入命令

update-alternatives --all

并在所有的替换版本中选中默认版本。



附录 E C++/Qt 配置

为充分用好这本书,需要把 Qt 4.6 或更新的 Qt 版本以及 C++编译器安装在你的计算机上。 以下各节介绍了我们所推荐的在三大桌面平台上安装 Qt 的流程。

E.1 C++/Qt 配置: 开源平台

在大多数 UNIX 的派生平台上,都自带了开源开发工具 $(ssh, bash, gcc)^{0}$ 。

在讨论那些基于 UNIX 派生的平台 (Linux, BSD, Solaris 等)时,我们会以*nix 来简称"大致类似于 UNIX 的操作系统"。

另外一个重要的缩略语是 POSIX,表示 Portable Operating System Interface for UNIX,即 UNIX 可移植操作系统接口。这个应用程序接口 (API) 系列标准是由 IEEE (Institute of Electrical and Electronics Engineers,美国电气与电子工程师学会) 倡导的。IEEE 是一个面向工程师、科学家和学生的组织,以制定计算机和电子工业的开发标准而著称^②。

这一节主要面向在计算机上安装并使用*nix 操作系统的读者。

针对本书设置计算机的第一步是要确认系统中完全安装了 Qt。这不仅包括源代码和编译库的代码,还包括 Qt 帮助文档系统、编程示例以及 QtCreator 程序。

Qt SDK

Qt SDK 包括 Qt 库、帮助程序、设计师以及 QtCreator,可用特殊的编译器进行构建。 建议在选定的开发平台上使用快速安装的方式。

要查看在计算机系统中安装的是哪一个版本的 Qt(如果有的话),可以用命令 which qmake qmake -v

第一条命令的输出会显示可执行文件 qmake 在系统中的位置。如果输出类似于 "bash: qmake: command not found",则可能的情况是:

- 1. Qt没有"完全"安装(包括开发工具)。
- 2. 安装了 Qt, 但 PATH 环境变量中没有包含/path/to/qt/bin 路径。
- 3. 是通过诸如 qmake-qt4 这样的安装包管理器安装的,以避免与 Qt 3 中的可执行 文件名冲突。

如果该命令可以运行,那么 qmake -v 会提供版本信息。如果报告信息是 "Using Qt version 4.x.y",那么就可以使用类似的命令分别查看是否其他的 Qt 工具也是可用的:moc, uic, assistant, designer 和 qtcreator。

① 在 Mac OS X 上,为了获得 C++编译器和 make 工具,需要安装 Xcode。

② 如果打算写一个 POSIX 正则表达式(参见 14.3 节),对于*nix 来说,看起来会是这个样子的: (lin|mac-os|un|solar|ir|ultr|ai|hp)[iu]?[sx]。

如果所有这些可执行命令都可以找到并且版本匹配,那么就说明已安装了 Qt,并且可以使用了。

如果这些测试表明你安装了早前的版本,或者根本没有安装 Qt,再或者在安装 Qt 未安装某些组件,那么无论是哪种情况,都需要构建或者安装 Qt 的最新发行版,并把 Qt 的可执行文件放到默认路径中。

从安装包安装 Qt

使用*nix 系统的安装包管理器(例如,apt,zypper,aptitude,kpackage,synaptic,yum等),可以轻松快捷地从包含 Qt 的安装包进行安装。只是需要记住的是,Qt 有可能会被分隔成许多小块,而需要的则可能是它们所有。以下是可用于 2011 年 Debian wheezy 系统的 Qt 4.7 的安装包清单:

```
[ROOT@lazarus]# apt-cache search qt4
libqt4-assistant - transitional package for Qt 4 assistant module
libqt4-core - trans pkg for Qt 4 core non-GUI runtime libraries
libqt4-dbg - Qt 4 library debugging symbols
libqt4-dbus - Qt 4 D-Bus module
libqt4-declarative - Qt 4 Declarative module
libqt4-declarative-folderlistmodel - Qt 4 folderlistmodel QML plugin
libqt4-declarative-gestures - Qt 4 gestures QML plugin
libqt4-declarative-particles - Qt 4 particles QML plugin
libqt4-designer - Qt 4 designer module
libqt4-dev - Qt 4 development files
libqt4-gui - transitional package for Qt 4 GUI runtime libraries
libqt4-help ~ Qt 4 help module
libqt4-network - Qt 4 network module
libqt4-opengl - Qt 4 OpenGL module
libqt4-opengl-dev - Qt 4 OpenGL library development files
libqt4-phonon - Qt 4 Phonon module
libqt4-qt3support - Qt 3 compatibility library for Qt 4
libqt4-script - Qt 4 script module
libqt4-scripttools - Qt 4 script tools module
libqt4-sql - Qt 4 SQL module
libqt4-sql-ibase - Qt 4 InterBase/FireBird database driver
libgt4-sql-mysql - Qt 4 MySQL database driver
libqt4-sql-odbc - Qt 4 ODBC database driver
libqt4-sql-psql - Qt 4 PostgreSQL database driver
libqt4-sql-sqlite - Qt 4 SQLite 3 database driver
libqt4-sql-sqlite2 - Qt 4 SQLite 2 database driver
libqt4-sql-tds - Qt 4 FreeTDS database driver
libqt4-svg - Qt 4 SVG module
libqt4-test - Qt 4 test module
libqt4-webkit - transitional package for Qt 4 WebKit module
libqt4-webkit-dbg - trans pkg for Qt 4 WebKit debugging symbols
libqt4-xml - Qt 4 XML module
libqt4-xmlpatterns - Qt 4 XML patterns module
libqt4-xmlpatterns-dbg - Qt 4 XML patterns library debuqqing symbols
qt4-demos - Qt 4 examples and demos
```

```
qt4-demos-dbg - Qt 4 examples and demos debugging symbols qt4-designer - graphical designer for Qt 4 applications qt4-dev-tools - Qt 4 development tools qt4-doc - Qt 4 API documentation qt4-doc-html - Qt 4 API documentation (HTML format) qt4-qmake - Qt 4 qmake Makefile generator tool qt4-qmlviewer - Qt 4 QML viewer qt4-qtconfig - Qt 4 configuration tool [ROOT@lazarus]#
```

正如上面所示,在 Debian 中,Qt 4 被划分成多个独立的安装包,这样开发者可以在配置时获得更好的灵活性。在开发时,需要全部的 Qt 4 安装包,如果打算使用 Qt 发行版进行程序调试,就尤其需要-core, -dev, -doc, -dev-tools, -designer 和用于调试的-dbg 符号。

从源代码安装

可以从 qt.nokia.com[©]下载最新的源文件 tarball, 然后进行解压缩和编译。如果有 Qt SDK, 可以运行 Updater 并选择 Qt Sources 来将其下载到 Qt SDK 目录。要确保也选择了 demos 和 examples。

提示

在 Debian 中,在另外编译一个 Debian 安装包后,也是有可能用一个简单的命令即可自动安装所有的工具和库。当打算从源代码编译任何流行的开源工具时,可以充分利用这一点。更多详细信息,可以参阅附录 D。这个命令为:

apt-get build-dep libqt4-dev

注意

tarball 是利用 tar 命令(tape archive, 磁带归档的缩写)生成的一个文件,为了便于存储和传送,一般使用 tar 命令把多个文件结合生成一个文件(扩展名通常是.tar)。得到的文件往往再使用诸如 gzip 或者 bzip2 等压缩工具进行压缩,形成一个扩展名为.gz或者.bz 的压缩文件。

tarball 的解压命令取决于其建立的方式。一般可以通过文件的扩展名来进行判断:

```
tar -vxfwhatever.tar//使用verbose开关tar -zxfwhatever.tar.gz//使用gzip压缩tar -zxfwhatever.tgz//同样也是使用gzip压缩tar -jxfwhatever.tar.bz2//使用bzip2压缩
```

tar 文件可以保留目录结构,同时也有很多选择和开关,可以通过输入下面的命令来阅读在线文档:

info tar info gzip info bzip

① 参见http://qt.nokia.com/downloads。

Qt 的源文件 tarball 包含完整的 Qt 库源文件,外加无数的例子、教程和完整的参考文献。该 tarball 包含简单的安装命令(在 README 和 INSTALL 文件中)以及一个 configure - help帮助信息。从源文件 tarball 安装软件时,一定要仔细阅读 README 文件。

从源文件进行编译可能要花费 2~3 小时的时间(根据系统的速度),但花费这个时间是值得的。示例 E.1 给出了配置 Qt 4.7 时的选项。

示例 E.1 /bin/qtconfi gure

#!/bin/sh

specify -phonon if you want to build the audiojukebox exercise or any of the Phonon examples.

replace username with your username, and Qt473 with your version of Ot

./configure -phonon -fast -prefix /home/username/Trolltech/Qt473

在Qt配置后, 键人 make 编译, 然后再安装它。

提示

如果有一颗四核处理器,可以试着键入

make ~j 4

这样编译器即可同时运行四个编译进程,可以充分利用额外的其他处理器核。

在最后一步中, make install 会把可执行文件和头文件从 tarball 解包后的目录复制到另外的一个位置。如果在一个公共位置进行安装,则必须以 root 身份来执行这一步。

检查 Qt 的安装

提示

在安装之后,键入 qmake -v 命令来确定 shell 中能够找到的是哪个版本的 qmake。对于安装有多个版本的系统,这是务必要做的一步。

[ezust@stan] /home/ezust> which qmake

which qmake

/usr/local/Trolltech/QtSDK/Desktop/Qt/473/gcc/bin/qmake

[ezust@cerberus] /home/ezust> qmake -version

QMake version 2.01a

Using Qt version 4.7.3

in /usr/local/Trolltech/QtSDK/Desktop/Qt/473/gcc/lib

环境变量

安装完成后,检查一下环境变量,确保 PATH 中含有指向已安装 Qt 的合适的引用。

2 提示

使用 bash 命令 env 可以显示目前系统中的所有环境变量。7.2 节讨论并使用过环境变量。

示例 E.2 中的脚本文件给出了如何利用 bash 命令设置环境变量,但实际值需取决于系统中文件的具体位置。

示例 E.2 /bin/qt.sh

- # Using the Qt SDK 1.1
- # None of the variables below are required by Qt
- # I just like having variables pointing to these locations for easy access:

export QTSDK=/opt/QtSDK

export QTCREATOR=\$QTSDK/QtCreator

export QTDIR=\$QTSDK/Desktop/Qt/473/gcc

export QTSRC=\$QTSDK/QtSources/4.7.3

- # make sure SDK's qmake and qtcreator are found first in the path: export PATH=\$OTDIR/bin:\$QTCREATOR/bin:\$PATH
- # Location of your shared libraries: export CPPLIBS=~/cs331/projects/libs
- # Where to search for shared object libraries at runtime: export LD LIBRARY PATH=\$CPPLIBS

E.2 C++/Qt 配置: Win32

有两个 Qt 版本可用于 Win32 平台:

- 1. mingw[©]版,编译需使用 MinGW (Minimalist Gnu for Windows, Windows Gnu 最小包)中的 g++,可以下载并可自由用于开源项目中。
- 2. Microsoft Visual C++版,编译需使用诸如 Microsoft Visual Studio 2008 这样的 Microsoft 编译器。

无论安装的是哪个版本的 Qt, 都很容易: Win32 安装程序会引导你完成包括注册扩展名和设置环境变量在内的整个过程。

提示

如果还没有编译器或者安装 IDE, 那么可以安装 Qt SDK, 在安装包中含有各个库、mingw编译器和 QtCreator。

Qt 安装后,还需要单击 Start->Programs->Qt by Nokia->Build debug symbols。 这可能需要花上几个小时。

然后,通过单击 Start->Programs->Qt by Nokia->Command Prompt 打开一个 shell 窗口。

现在,可以在这个命令提示符控制台中运行 qmake -v 来查看当前安装的 Qt 的版本。此时,qmake, assistant, designer, qtdemo, qtconfig, g++和 make 都应当在系统的搜索路径中。也可以试一下"开始"菜单中的 qtdemo。

① 参见http://www.mingw.org/。

E.3 C++/Qt 配置: Mac OS X

- 1. 从 Developer.apple.com 安装 xcode[©]。
- 2. 从 qt.nokia.com 安装 qtsdk²。
- 3. 从之前安装的 qtsdk 那里运行 qtcreator。

gmake 的特别之处

- qmake 默认创建 xcode 工程而不是利用 makefile。
- 使用 qmake -spec macx-g++生成一个命令行型的 Makefile.
- 在.pro 文件中放一个 CONFIG -= app_bundle, 以免在 Mac OS 的应用程序目录 结果中生成可执行文件。
- 如果使用 fink 安装的 make, 请阅读帮助文档中有关如何适当重新编译/设置 Qt 环境部分的内容。



② 参见 qt.nokia.com。



参考文献

C++参考文献

[Stroustrup97] The C++ Programming Language. Special Edition. Bjarne Stroustrup 1997. Addison Wesley. 0-201-70073-5

[Meyers] Effective C++. Scott Meyers. 1999-2005. Addison Wesley Professional Software Series. 0321334876. [Morris06] "The C++ Interpreter Pattern for Grammar Management" Stephen Morris. 2006. informit.com.

Qt 参考文献

[Blanchette08] C++ GUI Programming with Qt 4. Jasmin Blanchette and Mark Summerfield. Second Edition. 2008 Prentice Hall.

[Summer11] Advanced Ot Programming, Mark Summerfield. 2011. Prentice Hall.

[Thelin07] Foundations of Qt Development. Johan Thelin. 2007. Apress.com.

[Molkentin06] The Book of Qt 4. Daniel Molkentin 2006. No Starch Press.

[kdestyle] "KDE Style Guidelines." Coding Style.kde.org".

[qtapistyle] "Designing Ot-Style C++ APIs." Matthias Ettrich. 2005. Trolltech.

[qtapistyle] "Ot Coding Style" Community Project. 2010. Trolltech.

[qttestlib] "Writing Unittests for Qt 4 and KDE4 with QtTestLib." Brad Hards. 2005. developer.kde.org®.

OOP 参考文献

[Buschmann96] Pattern-Oriented Software Architecture. First Edition. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. 1996. John Wiley & Sons. 0-471-95869-7.

[Fowler04] UML Distilled. Third Edition. Martin Fowler. 2004. Addison Wesley. 0-321-19368-7.

[Gamma95] Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. Addison-Wesley 0-201-63361-2.

[Koenig95] Patterns and Antipatterns. Andrew Koenig. 1995. Journal of Object-Oriented Programming 8(1) pgs 46-48.

[Martin98] Pattern Languages of Program Design 3. Robert C. Martin, Dirk Riehle, and Frank Buschmann. 1998. Addison-Wesley. pgs 293-312. 0-201-31011-2.

¹ http://techbase.kde.org/Policies/Kdelibs Coding Style.

② http://doc.trolltech.com/qq/qq13-apis.html.

³ http://qt.gitorious.org/qt/pages/QtCodingStyle.

⁴ http://developer.kde.org/documentation/tutorials/writingunittests/writingunittests.html.

Docbook 参考文献

[docbook] *Docbook: The Definitive Guide*. Norman Walsh. 2006. O'reilly Associates[©]. [docbookxsl] *Docbook XSL: The Complete Guide*. Bob Stayton. 2005. SageHill Enterprises[©].

其他参考文献

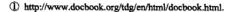
[xhtmlw3c] "w3c Recommendation: XHTML 1.0 The Extensible HyperText Markup Language."2005. W3C (World Wide Web Consortium)[©].

[dist] The web accessible site from which you can download example source code and other useful resources: http://informit.com/title/9780132826457.

[Friedl98] Mastering Regular Expressions. Second Edition. Jeffrey Friedl. 1998. O'Reilly. 1-56592-257-3.

[Rehman03] The Linux Development Platform. Rafeeq Ur Rehman. Christopher Paul. 2003. Prentice Hall. 0-13-009115-4.

[Guzdial07] Introduction to Computing & Programming with Java, A Multimedia Approach. Mark Guzdial and Barbara Ericson. 2007. Prentice Hall. 0-13-149698-0.



② http://www.sagehill.net/docbookxsl/

³ http://www.w3.org/TR/xhtml1/

更多尽在http://shop102083147.taobao.com

请有空看看,现货直接拍

关键词可以店里搜索

plc的书 http://shop102083147.taobao.com/?q=plc&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

数控的书 http://shop102083147.taobao.com/?q=%CA%FD%BF%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

单片机的书 http://shop102083147.taobao.com/?q=%B5%A5%C6%AC%BB%FA&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=y&initiative_id=itemz_20130608

嵌入式的书 http://shop102083147.taobao.com/?q=%C7%B6%C8%EB%CA%BD&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=y&initiative_id=itemz_20130608

有限元的书 http://shop102083147.taobao.com/?q=%D3%D0%CF%DE%D4%AA&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=v&initiative id=itemz 20130608

ansys的书 http://shop102083147.taobao.com/?q=ansys&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

建筑的书 http://shop102083147.taobao.com/?q=%BD%A8%D6%FE&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

电动机的书 http://shop102083147.taobao.com/?q=%B5%E7%B6%AF%BB%FA&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=y&initiative_id=itemz_20130608

电路的书 http://shop102083147.taobao.com/?q=%B5%E7%C2%B7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

EDA的书 http://shop102083147.taobao.com/?q=eda&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

 $VHDL的书 \qquad http://shop102083147.taobao.com/?q=VHDL\&searcy_type=item\&s_from=newHeader\&source=\&ssid=s5-e\&search=y\&initiative_id=itemz_20130608$

FPGA的书 http://shop102083147.taobao.com/?q=FPGA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

MATLAB的书 http://shop102083147.taobao.com/?q=MATLAB&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y &initiative id=itemz 20130608

labview的书 http://shop102083147.taobao.com/?q=Lab&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

建模的书 http://shop102083147.taobao.com/?q=%BD%A8%C4%A3&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

变频器的书 http://shop102083147.taobao.com/?q=%B1%E4%C6%B5%C6%F7&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=v&initiative id=itemz 20130608

PHP的书 http://shop102083147.taobao.com/?q=PHP&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

SQL的书 http://shop102083147.taobao.com/?q=SQL&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

压力容器的书 http://shop102083147.taobao.com/?q=%D1%B9%C1%A6%C8%DD%C6%F7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

焊接的书 http://shop102083147.taobao.com/?q=%BA%B8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

希特勒 http://shop102083147.taobao.com/?q=%CF%A3%CC%D8%C0%D5&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=y&initiative_id=itemz_20130608

第三帝国 http://shop102083147.taobao.com/?q=%B5%DA%C8%FD%B5%DB%B9%FA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=v&initiative id=itemz 20130608

二战 http://shop102083147.taobao.com/?q=%B6%FE%D5%BD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

LINUX的中 http://shop102083147.taobao.com/?q=LINUX&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

GIS的书 http://shop102083147.taobao.com/?q=gis&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

组态的书 http://shop102083147.taobao.com/?q=%D7%E9%CC%AC&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

触摸屏 http://shop102083147.taobao.com/?q=%B4%A5%C3%FE%C6%C1&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=v&initiative id=itemz 20130608

齿轮 http://shop102083147.taobao.com/?q=%B3%DD%C2%D6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

网页 http://shop102083147.taobao.com/?q=%CD%F8%D2%B3&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=v&initiative_id=itemz_20130608

 $Php \\ http://shop102083147.taobao.com/?q=Php&searcy_type=item\&s_from=newHeader\&source=\&ssid=s5-e\&search=y\&initiative_id=itemz_20130608$

Arm http://shop102083147.taobao.com/?q=Arm&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

lsp http://shop102083147.taobao.com/?q=Isp&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Proteus http://shop102083147.taobao.com/?q=Proteus&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y &initiative_id=itemz_20130608

 $\label{lem:complex} Er lang $$ $ http://shop102083147.taobao.com/?q=Er lang $$ searcy_type=item $$_from=new Header $$ source= $$ sid=s5-e $$ search=y $$ initiative_id=itemz_20130608 $$$

开关 http://shop102083147.taobao.com/?q=%BF%AA%B9%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

电源 http://shop102083147.taobao.com/?q=%B5%E7%D4%B4&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

多核 http://shop102083147.taobao.com/?q=%B6%E0%BA%CB&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Windows http://shop102083147.taobao.com/?q=Win&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Altium Designer http://shop102083147.taobao.com/?q=Altium&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Phone http://shop102083147.taobao.com/?q=Phone&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

通信

程序员网址导航www.daocode.com

Indesign http://shop102083147.taobao.com/?q=Indesign&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

经典战史回眸http://shop102083147.taobao.com/?q=%BE%AD%B5%E4%D5%BD%CA%B7%BB%D8%ED%F8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

曲面http://shop102083147.taobao.com/?q=%C7%FA%C3%E6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

木马

Cut http://shop102083147.taobao.com/?q=Cut&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

uml http://shop102083147.taobao.com/?q=uml&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

mac http://shop102083147.taobao.com/?q=mac&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

ArcGIS http://shop102083147.taobao.com/?q=ArcGIS&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

GIS http://shop102083147.taobao.com/?q=GIS&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

FANUC http://shop102083147.taobao.com/?q=FANUC&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

VoIP http://shop102083147.taobao.com/?q=VoIP&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

 $\label{lem:cisco} {\tt http://shop102083147.taobao.com/?q=cisco\&searcy_type=item\&s_from=newHeader\&source=\&ssid=s5-e\&search=y\&initiative_id=itemz_20130608$

MATLAB http://shop102083147.taobao.com/?q=LAB&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

局域网

路由器

数控车床

电气信息工程丛书

数控 http://shop102083147.taobao.com/?q=%CA%FD%BF%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

数据恢复

主板

显示器

平面设计

建筑

汽车

可编程序控制器

宽带

iр

自动化

开关

电源

驱动

服装

flash

fluent

Fortran

ABAQUS

adams

access

发动机

```
仿真
 继电器
 传感器
 影视
太阳能
现场总线
机器人
                                                                    http://shop102083147.taobao.com/?q=%BB%FA%C6%F7%C8%CB&searcy_type=item&s_from=newHeader&source=&
ssid=s5-e&search=y&initiative_id=itemz_20130608
加工中心
ZigBee
CATIA V5
 逆向工程
 模具
冲压
CAE
压力容器
 钣金
 编程
路由器
Oracle
 高分子
TCP/IP
游戏
纺织
DIV+CSS
ZBrush
 电磁
 摄像
摄影
 i Phone
 钢结构
 布线
 电池
动漫
 卡诵
动画
 非线性
PCB
CMOS
封装
 半导体
 纳米
碳纤维
不孕
 单反
 土木
 激光
 电动机
精密
 新能源
动力
                                    http://shop102083147.taobao.com/?q=\%87\%FE\%CE\%F1\%C6\%F7\&searcy\_type=item\&s\_from=newHeader\&source=\&ssid=s5-e\&stid=s5-e\&stid=s5-e\&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-e&stid=s6-
search=y&initiative_id=itemz_20130608
软件
广告
 处理器
机床
 夹具
 桥梁
隧道
 公路
 微机
WEB
```

数据库 飞机. Visual Studio 程序员 锻浩 热处理 轿车 铁路 材料 虚拟 化工 SOL 包装 数字图像 数字 服饰 人体摄影 仿生 自动 器折 焊接 工程 ansvs abadus

hyperwork

plc的书http://shop102083147.taobao.com/?q=plc&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

数控的书http://shop102083147.taobao.com/?q=%CA%FD%BF%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

arm的书http://shop102083147.taobao.com/?q=arm&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

单片机的书http://shop102083147.taobao.com/?q=%B5%A5%C6%AC%BB%FA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

嵌入式的书http://shop102083147.taobao.com/?q=%C7%B6%C8%EB%CA%BD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

LINUX的书http://shop102083147.taobao.com/?q=LINUX&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

有限元的书http://shop102083147.taobao.com/?q=%D3%D0%CF%DE%D4%AA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=v&initiative_id=itemz_20130608

ansys的节http://shop102083147.taobao.com/?q=ansys&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

CATIA的书http://shop102083147.taobao.com/?q=CATIA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

Workbench的书http://shop102083147.taobao.com/?q=Workbench&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

abaqus的书http://shop102083147.taobao.com/?q=abaqus&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Hyperh的书http://shop102083147.taobao.com/?q=Hyper&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

LS-DYNA的书 http://shop102083147.taobao.com/?q=LS-DYNA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

NASTRAN的中 http://shop102083147.taobao.com/?q=NASTRAN&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

动力的书 http://shop102083147.taobao.com/?q=%B6%AF%C1%A6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=v&initiative_id=itemz_20130608

汽车的书http://shop102083147.taobao.com/?q=%C6%FB%B3%B5&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

建筑的书http://shop102083147.taobao.com/?q=%BD%A8%D6%FE&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

土木的书http://shop102083147.taobao.com/?q=%CD%C1%C4%BE&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

电动机的书http://shop102083147.taobao.com/?q=%B5%E7%B6%AF%BB%FA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

电路的书http://shop102083147.taobao.com/?q=%B5%E7%C2%B7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

EDA的书http://shop102083147.taobao.com/?q=eda&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

VHDL的书http://shop102083147.taobao.com/?q=VHDL&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

FPGA的书http://shop102083147.taobao.com/?q=FPGA&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

Protel的书http://shop102083147.taobao.com/?q=Protel&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

控制的书http://shop102083147.taobao.com/?q=%BF%D8%D6%C6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

仿真的书http://shop102083147.taobao.com/?q=%B7%C2%D5%E6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

MATLAB的书http://shop102083147.taobao.com/?q=MATLAB&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

labview的书http://shop102083147.taobao.com/?q=labview&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

建模的书http://shop102083147.taobao.com/?q=%BD%A8%C4%A3&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

变频器的书http://shop102083147.taobao.com/?q=%B1%E4%C6%B5%C6%F7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

PHP的书http://shop102083147.taobao.com/?q=PHP&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

SQL的书http://shop102083147.taobao.com/?q=SQL&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

压力容器的书http://shop102083147.taobao.com/?q=%D1%B9%C1%A6%C8%DD%C6%F7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

焊的书http://shop102083147.taobao.com/?q=%BA%B8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

希特勒http://shop102083147.taobao.com/?q=%CF%A3%CC%D8%C0%D5&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

第三帝国http://shop102083147.taobao.com/?q=%B5%DA%C8%FD%B5%DB%B9%FA&searcy_type=item&s_from=newHeader&source=&ssid=s 5-e&search=y&initiative_id=itemz_20130608

二战http://shop102083147.taobao.com/?q=%B6%FE%D5%BD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

纳粹德国的兴亡http://shop102083147.taobao.com/?q=%C4%C9%B4%E2%B5%C2%B9%FA%B5%C4%D0%CB%CD%F6&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

GIS的书http://shop102083147.taobao.com/?q=gis&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

组态的书http://shop102083147.taobao.com/?q=%D7%E9%CC%AC&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

触摸屏的书http://shop102083147.taobao.com/?q=%B4%A5%C3%FE%C6%C1&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

开关的书http://shop102083147.taobao.com/?q=%BF%AA%B9%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

电源的书http://shop102083147.taobao.com/?q=%B5%E7%D4%B4&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

LED的书http://shop102083147.taobao.com/?q=LED&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

LCD的书http://shop102083147.taobao.com/?q=LCD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

机器人的书http://shop102083147.taobao.com/?q=%BB%FA%C6%F7%C8%CB&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

 $FANUC的书http://shop102083147.taobao.com/?q=FANUC\&searcy_type=item\&s_from=newHeader\&source=\&ssid=s5-e\&search=y\&initiative_id=itemz_20130608$

数控的书http://shop102083147.taobao.com/?q=%CA%FD%BF%D8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Android的书 http://shop102083147.taobao.com/?q=Android&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

影视的书http://shop102083147.taobao.com/?q=%D3%B0%CA%D3&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Window的书http://shop102083147.taobao.com/?q=Window&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Visio的书http://shop102083147.taobao.com/?q=Visio&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&

initiative id=itemz 20130608

黑客的书http://shop102083147.taobao.com/?q=%BA%DA%BF%CD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y &initiative id=itemz 20130608

Apache的书http://shop102083147.taobao.com/?q=Apache&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

PHP的书http://shop102083147.taobao.com/?q=PHP&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

男人的书http://shop102083147.taobao.com/?q=%C4%D0&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

ArcObjects给书http://shop102083147.taobao.com/?q=ArcObjects&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

伺服的学http://shop102083147.taobao.com/?q=%CB%C5%B7%FE&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

控制器的书http://shop102083147.taobao.com/?q=%BF%D8%D6%C6%C6%F7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

PID的书http://shop102083147.taobao.com/?q=PID&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

量子的书http://shop102083147.taobao.com/?q=%C1%BF%D7%D3&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

Python的书http://shop102083147.taobao.com/?q=Python&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

模具http://shop102083147.taobao.com/?q=%C4%A3%BE%DF&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

Maya的书http://shop102083147.taobao.com/?q=Maya&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

动画的书http://shop102083147.taobao.com/?q=%B6%AF%BB%AD&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

卡通的书http://shop102083147.taobao.com/?q=%BF%A8%CD%A8&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

WinCC的书http://shop102083147.taobao.com/?q=WinCC&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative id=itemz 20130608

服装的书http://shop102083147.taobao.com/?q=%B7%FE%D7%B0&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

公路的书http://shop102083147.taobao.com/?q=%B9%AB%C2%B7&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

DSP的书http://shop102083147.taobao.com/?q=DSP&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

java的书http://shop102083147.taobao.com/?q=java&searcy_type=item&s_from=newHeader&source=&ssid=s5-e&search=y&initiative_id=itemz_20130608

更多尽在http://shop102083147.taobao.com 请有空看看,现货直接拍

好书多多 欢迎再来 收藏我的店吧

http://shop102083147.taobao.com 有你要的

目录是推荐 关键词可以店里搜索 看看推荐 有你要的

直接拍 还可以定制图书哦

http://www.szdnet.org.cn/ 上面的 要有试读

定制图书2元1本

白天要上班 我晚上在线的时候会上架 其余时间下架 有人捣乱 见谅