17. 开发环境

本章介绍 Linux 下常用的开发环境，以及在 openSUSE 下安装和配置这些软件。由于这些软件中的每一个都有非常多的学问，这里只是简单介绍一下，开个门，让大家知道有这些东西的存在。实际学习编程语言和集成开发环境的用法需要参考各自相应的文档和教程书籍。

主要涵盖的编程语言：C/C++/Java/Perl/Python;

代码阅读和编辑器：Vim/Emacs/KWrite/Gedit;

图形界面库： Qt/Gtk+/wxWidgets;

集成开发环境：QtCreator/KDevelop/Anjuta/Code::Blocks/Eclipse;

Linux 下的 Android 开发环境；

版本控制软件：Git/Mercurial/CVS/SVN/Bazaar;

Open Build Service 软件打包（18 章介绍）。

17.1 代码阅读和编辑器

Vim 是 vi 的进化版，与 Emacs 都是老牌的代码编辑器。这两个编辑器已经纵横了几十年了。vim 本身是命令行下的编辑器，有个图形界面的包：gvim。这两个再 openSUSE 的发行 DVD 里面都有。openSUSE 默认安装了 vim，但没有 gvim，可以用 Yast 的软件管理搜索软件包的名字安装，或者执行安装命令：

zypper in gvim

Emacs 默认没被安装，可以自己装：

zypper in emacs emacs-info

Kwrite 是 KDE 桌面默认的文本和代码编辑器，KWrite 还有个高级版 Kate，功能更丰富。Gedit 是 GNOME 桌面默认的文本和代码编辑器。如果是 KDE 桌面，需要装 Gedit 的话，可以执行：

zypper in gedit gedit-plugins gedit-plugins-lang

如果是 GNOME 桌面尝试 KWrite/Kate，执行：

zypper in kwrite kate

几乎所有的 Linux 下的编辑器都带有各种编程语言的语法高亮显示，还有各种编辑器的插件（如 gedit-plugins ）。可以自己按照喜好定制编辑器。比如这个帖子：

Emacs 中打造强大的 Python IDE

http://www.cnblogs.com/coderzh/archive/2009/12/26/emacspythonide.html

不管喜好什么编辑器和编程语言，都可以类似地定制。这就是开源的好处之一。

17.2 C/C++和其他编程工具

17.2.1 Linux 软件包简介

国内学校计算机编程的课程很多都是从 C 语言开始的。openSUSE 默认没装 gcc 和 g++等编译器。所以很多初次接触 openSUSE 的会觉得比较坑。这个容易，可以自己安装这些编译器和必要的 devel 包。这

里先简单介绍一下，Linux 软件包的名字通常有四种，以 audacious 播放器为例：

audacious
audacious-devel
audacious-debuginfo
audacious-debugsource

第一个包 audacious 是正式的发行包，里面有可执行的二进制文件。audacious-devel 是开发包，里面通常是一些头文件和库文件，如果要用到 audacious 自带的一些头文件、库，那就需要装第二个。后面两个是用于调试的信息和源码，这通常只有调试 audacious 本身或者调试一些 audacious 插件才要用，后面两个通常是可以忽略的。

开发软件的时候，除了需要编译、链接工具外，还要一些常用库的发行包和开发包，如 glibc 和 glibc-devel。如果系统里明明看到有如 libnotify.so.4 的库文件，而编译相关源代码时，却提示找不到 libnotify 库，那很可能是没装 libnotify-devel 开发包。

17.2.2 C/C++编译器和开发库

下面进入正题，安装 gcc 和 g++等工具。可以载 Yast 的软件管理里面，搜索包的名字来安装，也可以用 root 命令行执行命令：

zypper in gcc gcc-c++

GNU 编译器里还有其他编程语言的，如 gcc-ada，gcc-fortran ，gcc-java ，gcc-objc 等等，可以根据自己需要安装。

安装这两个软件包，软件包管理器会自动安装它们的依赖包：

binutils, gcc47, gcc47-c++ , glibc-devel, libstdc++47-devel , linux-glibc-devel .

第一个 GNU Binutils 是一些编程工具的集合，如其中的 ar 用于打包静态库，ld 是链接器。这些是编程必备的。第二个和第三个是 gcc 和 g++的实际的软件包，最后三个是标准 C 库和 C++库的开发包，包含各种头文件和基础库。

安装好后，就可以执行如下编译命令：

```
gcc hello.c -o hello.out
g++ hello.cpp -o hcpp.out
```

在当前目录执行 hello.out 和 hcpp.out，需要用 ./ 打头：

```
./hello.out
./hcpp.out
```

17.2.3 其他常用编程工具和库

除了编译器，还有其他编程或编译源码常用的工具。这里简单介绍一些：

gdb：调试器，默认已装，可以调试各种源代码，包括 C/C++/Java/Fortran/Object-C 等。

kdbg：gdb 的图形界面。

cpp： 系统 GNU 预处理器，默认已装。

autoconf：自动配置源码和 makefile 文件的工具。

autogen： 自动文本文件生成器，用于生成程序的文件，简化程序的管理。

automake：自动生成 GNU 风格的 Makefile.in 文件。
bison：类似 Yacc 的语法解析生成器。学编译原理的应该知道。
flex： 快速词法分析生成器。学编译原理的应该知道。
m4：是传统 Unix 宏解析器的一个实现。
make ：是编译项目代码必备的，包含 make 命令行工具和文档。
cmake： 跨平台的 make 工具。如果是使用 cmake 的 Qt 项目，还需要包 automoc4。
ccache： 快速 C/C++编译器缓存，加速编译器重复编译的过程。
checkinstall：“make install”命令安装的跟踪器，从源码编译安装程序时常用。
nasm：x86 汇编工具。
boost-devel：Boost 是 跨平台的 C++标准库的扩展库，包含大量的通用功能代码。这个是其开发包。
gsl：GNU 科学计算库，如数值计算等。其开发库是 gsl-devel。
opencv：Intel 的开源图像视频处理库。其开发库是 opencv-devel。
sqlite3-devel：SQLite 3 数据库的开发包。
zlib-devel：压缩函数库 zlib 的开发包。
tcllib, tcl-devel, tclx, tix, tk, tk-devel, vtcl ： 这些是 Tcl/Tk 编程的工具和库文件，编译源码的时候可能遇到。
上面介绍的当然只是冰山一角，还有茫茫多的工具和开发库。实际需要哪些就装哪些，可以随时在 Yast 软件管理里面搜索相关的东西。命令行安装也可以，比如：

```
zypper in kdbg autoconf autogen automake automoc4 bison flex m4 make cmake ccache checkinstall nasm gsl-devel sqlite3-devel zlib-devel
```

安装时，软件包管理器通常会自动装些依赖库和语言包，选“确定”就行了。

17.3 Java 和 Eclipse 集成开发环境

Java 毫无疑问也是非常流行的编程语言。Oracle 官方有适合各种操作系统的 JDK 7.0 的下载（如果是要装 Android 开发环境，那么本节的内容可以完全跳过，后面 17.8 节有更简单的方法）。
openSUSE 自带的是开源社区维护的 OpenJDK 7，默认装了 OpenJDK 7，但是开发包、文档、例子没装，如果要搞 Java 开发，建议都装上：

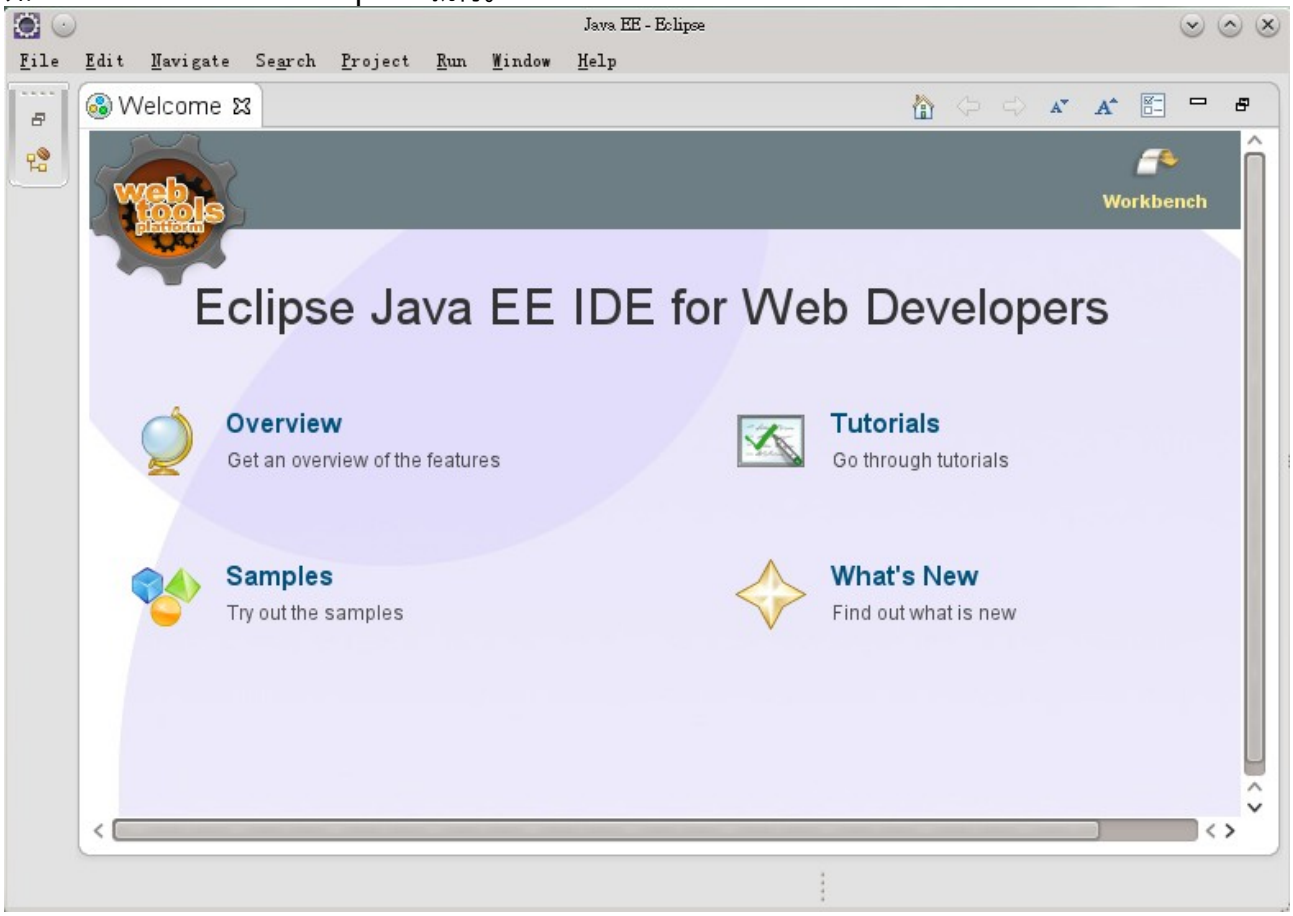
```
zypper in java-1_7_0-openjdk-devel java-1_7_0-openjdk-demo java-1_7_0-openjdk-javadoc
```

由于 openJDK 很多东西或快捷方式装在系统路径上，所以不需要配置。直接可用来编译 Java 程序。

```
javac hello.java
java hello
```

Eclipse 是 IBM 公司引领的杰作，很多 Java 开发都用 Eclipse 集成开发环境，包括 Android 开发调试。Eclipse 本身是一个跨平台的框架，可以通过插件形式，支持 Java、C/C++、Perl、Ruby、Python、数据库等开发。Eclipse 需要去官网下载：
http://www.eclipse.org/downloads/
一般 Java 开发下载：Eclipse IDE for Java EE Developers ，根据自己系统选择 Linux 32bit 或 Linux 64bit。如果用于 C/C++或建模语言，可以下载其他的版本。

下载后，比如 64 位系统的 eclipse-jee-juno-SR2-linux-gtk-x86_64.tar.gz，可以直接解压缩到用户目录的 bin 文件夹里，比如解压到 /home/suse123/bin/ 。然后在/home/suse123/bin/eclipse/目录下面直接运行 eclipse 就行了。初次运行会比较慢，后面会好些。Eclipse 工作空间用默认的如/home/suse123/workspace 就行。



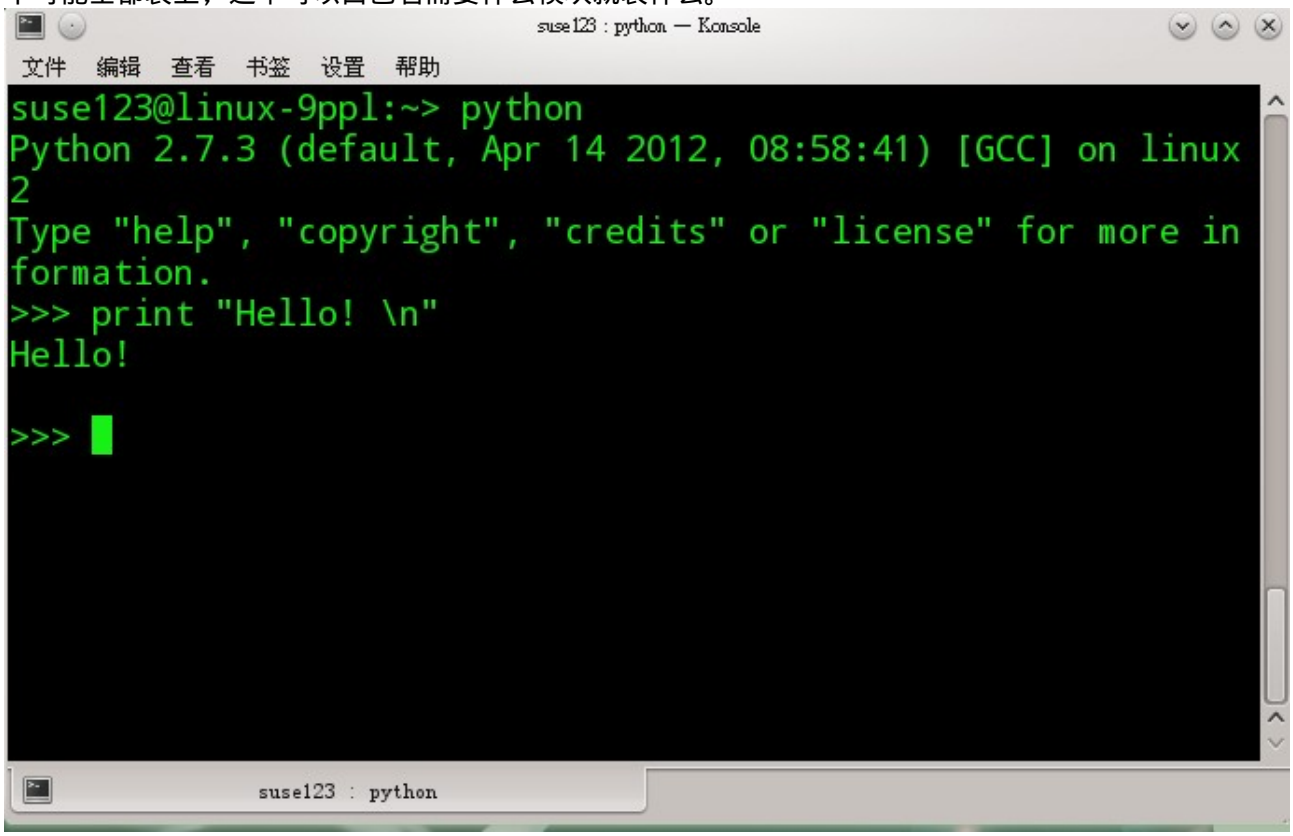
可以在 Eclipse 里面新建各种项目，包括 Java 本身的和 Web 的都可以。

Eclipse 汉化包可以通过在 Eclipse 里面添加源，然后自动下载中文语言插件。参看这个：
http://zhidao.baidu.com/question/179934518.html
http://www.eclipse.org/babel/downloads.php
对于最新的 Juno 版本，打开 Eclipse 菜单——>Help——>Install New Software...，
在打开的对话框里，第一行里输入：
http://download.eclipse.org/technology/babel/update-site/R0.10.1/juno
名字取为 Babel，点击 OK。然后等待列出一堆语言包，选上
Babel Language Packs in Chinese(Simplified) 和
Babel Language Packs in Chinese(Traditional)，一直点 Next 和 Accept，
等下载安装完就搞定了。

17.4 Perl 和 Python

Linux 下著名的 3P 脚本语言就是 Perl、Python、Php。Php 放到 19 章搭建 LAMP 和 LNMP 网站介

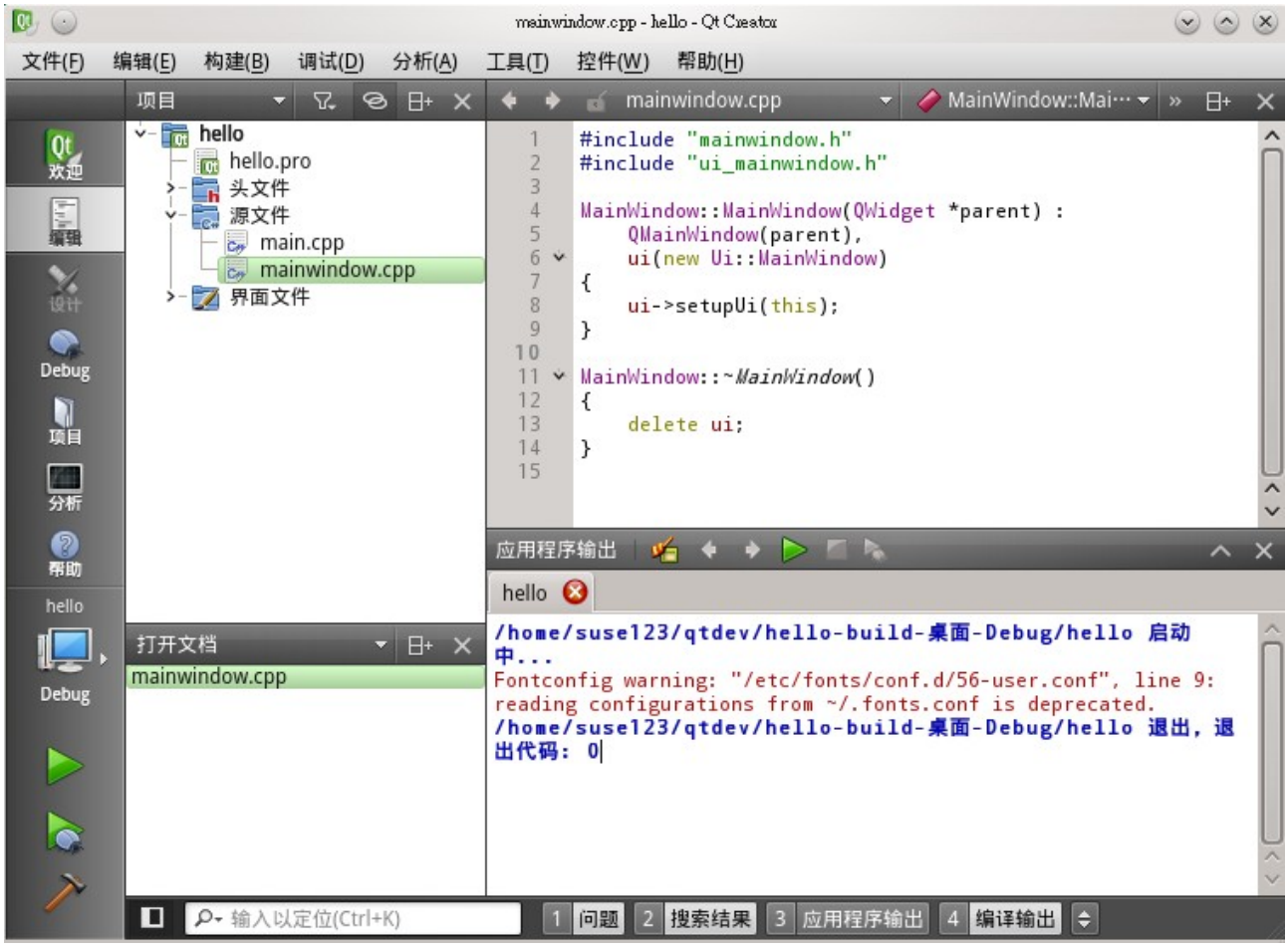
绍。openSUSE 默认就安装了 Perl 和 Python。当然，Perl 和 Python 还有非常多的模块，数不胜数，不可能全都装上，这个可以自己看需要什么模块就装什么。



17.5 Qt 和 QtCreator、KDevelop

Linux 下常用的 C/C++语言的图形库就数 Qt、Gtk+和 wxWidgets 了。这些都是可以跨平台的，Windows 下也可以有。常用的集成开发环境，除了上面的 Eclipse，还有 KDevelop、QtCreator、Anjuta、Code::Blocks，Gtk+有个可视化编辑的工具 glade 。

本节介绍 Qt 和 QtCreator。Qt 最初是奇趣公司开发的跨平台图形库。后来整个公司被 Nokia 买了，Nokia 本打算利用 Qt 开发新的智能手机操作系统 Meego，Meego 不幸夭折了。Qt 也被 Nokia 出卖给芬兰 IT 服务公司 Digia。虽然遭遇各种变故，Qt 的活力不减当年。
Qt 是 KDE 桌面的基石，开发 Qt 程序可以用简单的文本编辑器，也可用各种复杂的集成开发环境。
QtCreator 算是后起之秀，现在做的界面和功能都不错，建议使用。Qt 开发库和 QtCreator 安装命令：
zypper in libqt4-devel qt-creator libqt4-devel-doc
需要注意磁盘空间，最后一个文档包体积较大，当然所带的英文资料也够丰富。

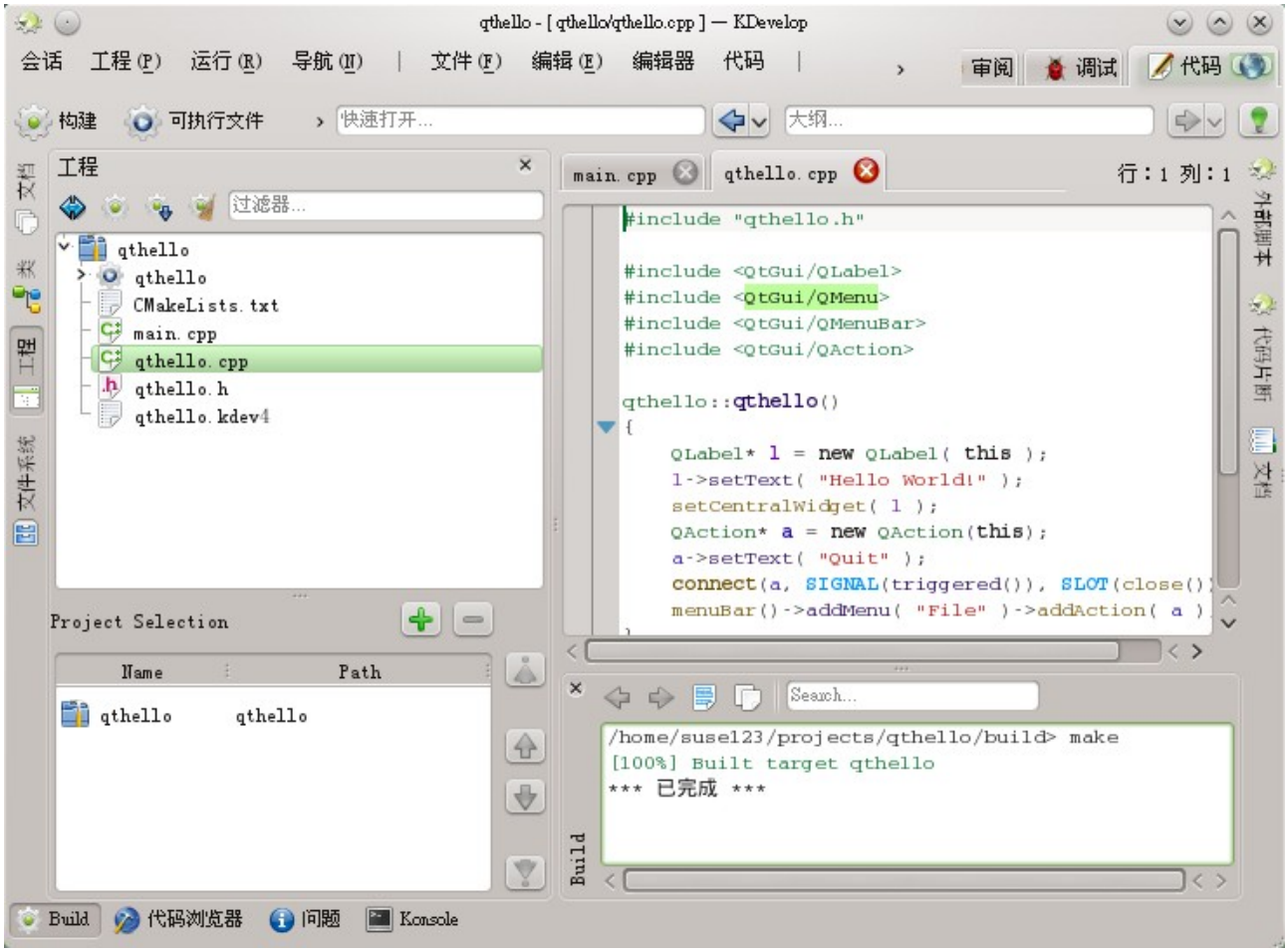


KDevelop 是比较全面的集成开发环境，KDevelop 支持多种不同语言，其中包括 C、C++、Perl、Python、PHP、Java、Fortran、Ruby、Ada、Pascal、SQL 和 Bash 脚本。支持各种图形库编程，如，支持的构建系统包括 GNU（automake）、cmake、qmake 和自定义项目的 make（如果您想使用自己的 Makefile，KDevelop 不会破坏它们的）以及不需要 Makefile 的脚本项目。安装命令：

zypper in kdevelop4 libkde4-devel kdevelop4-devel kdevelop4-plugins-php kdevelop4-doc

安装后运行 KDevelop 可能会提示 Git 插件被禁用，因为没装 Git，Git 等版本控制系统后面介绍，这里先不管。默认的项目模板不多，其他语言支持的应该需要具体配置，参看其文档和主页：

http://www.kdevelop.org/



17.6 Gtk+和 Anjuta

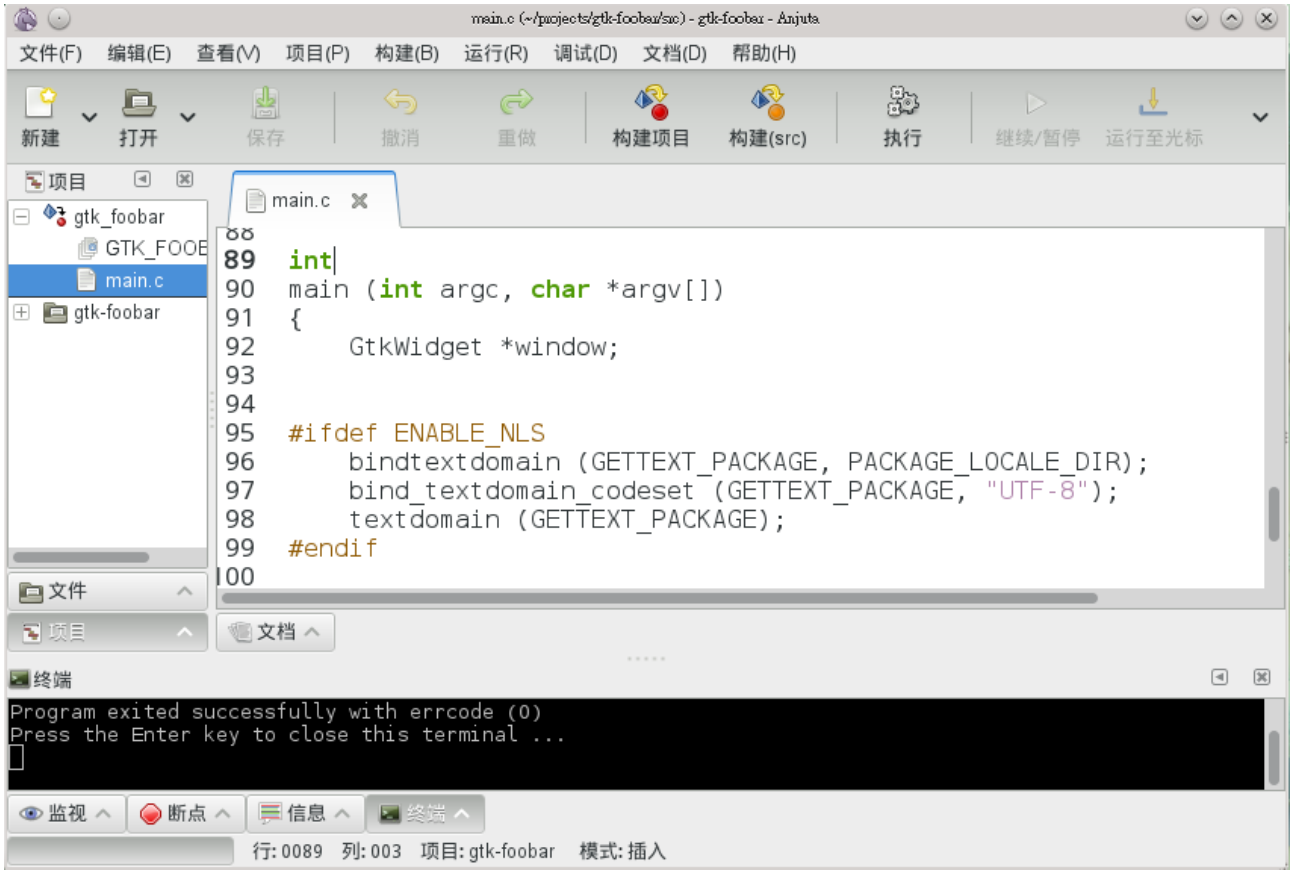
Gtk+当然也是非常著名的图形库，是 GNOME、Xface、LXDE 等桌面环境的基石。openSUSE 发行光盘里面有 Gtk+ 版本 2 和版本 3 的开发包，还有许多其他 Gtk+系列的库。Gtk+有一个可视化编辑工具 Glade，当然更推荐 GNOME 桌面集成开发环境 Anjuta，Anjuta 还有一些插件可以自己装。Gtk+和 Anjuta 安装命令：

zypper in gtk3-devel gtk-doc anjuta anjuta-lang intltool gtkmm3-devel gtkmm3-doc gtkmm3-tutorial gtkmm3-tutorial-lang

安装 Anjuta 的时候，软件管理器会装一堆依赖包（好像少了个 intltool，一定要补上），其中有 subversion，就是 SVN 版本控制系统。

这些集成开发环境都附带有版本控制系统的插件功能，比较省事。

除了支持 C/C++/Gtk+，Anjuta 还支持 Java/JavaScript/Python 等语言和 WxWidgets 图形库编程。



17.7 wxWidgets 和 Code::Blocks/CodeLite

wxWidgets 是模仿 Windows MFC 编程风格的跨平台图形库。它执行效率比较高，在 Windows 下像 MFC，Linux 下就像 Gtk+。网页编辑工具 Amaya、FTP 软件 FileZilla 、集成开发环境 Code::Blocks/CodeLite 就是使用 wxWidgets 开发的。

wxWidgets 的开发库需要连网下载。安装命令：

zypper in wxWidgets-devel wxWidgets-docs

wxWidgets 的组件比较散，还有一些插件和其他的插件开发包，可以自己选择安装。

Code::Blocks 也是流行的开发环境，只是在 openSUSE 上用的比较少，在 openSUSE 官方发行 DVD 和源里面没有。CodeLite 也是类似的开发环境，也是类似地没在 openSUSE 官方源里。需要自己去搜索：

http://software.opensuse.org/search

Code::Blocks 需要安装 libcodeblocks0 和 codeblocks 本身，一键安装的页面：

http://software.opensuse.org/package/libcodeblocks0

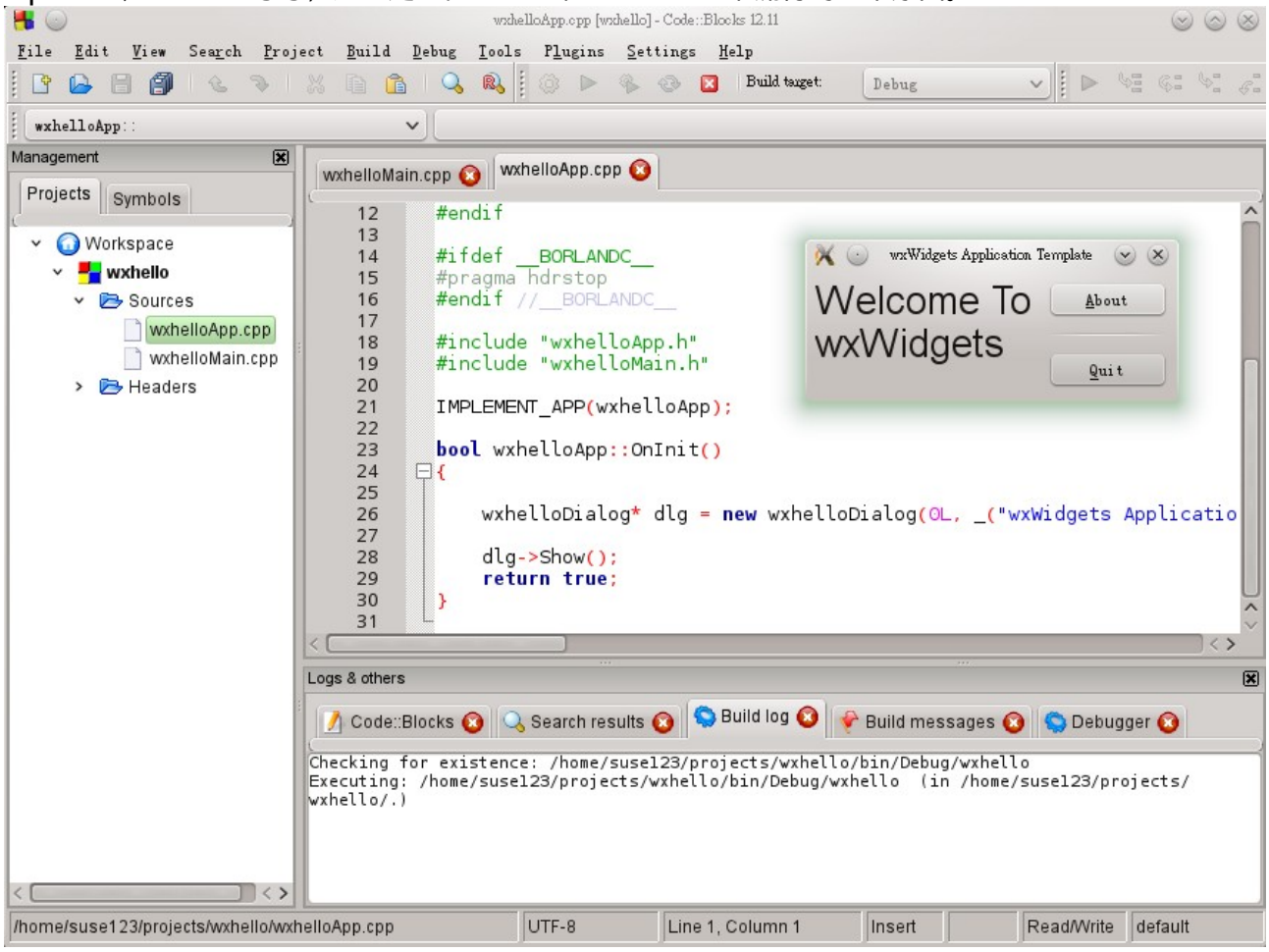
http://software.opensuse.org/package/codeblocks

或者使用 root 命令行安装：

zypper ar -f http://download.opensuse.org/repositories/home:/NarkoZ:/devel:/tools:/ide/openSUSE_12.3/ IDE

zypper in libcodeblocks0 codeblocks

Code::Blocks 运行比较快，默认就带很多项目模板如 C/C++/Fortran、Gtk+、Qt、wxWidgets、OpenGL、Matlab 等等，这也是它在 Linux 和 Windows 下流行的一个原因。

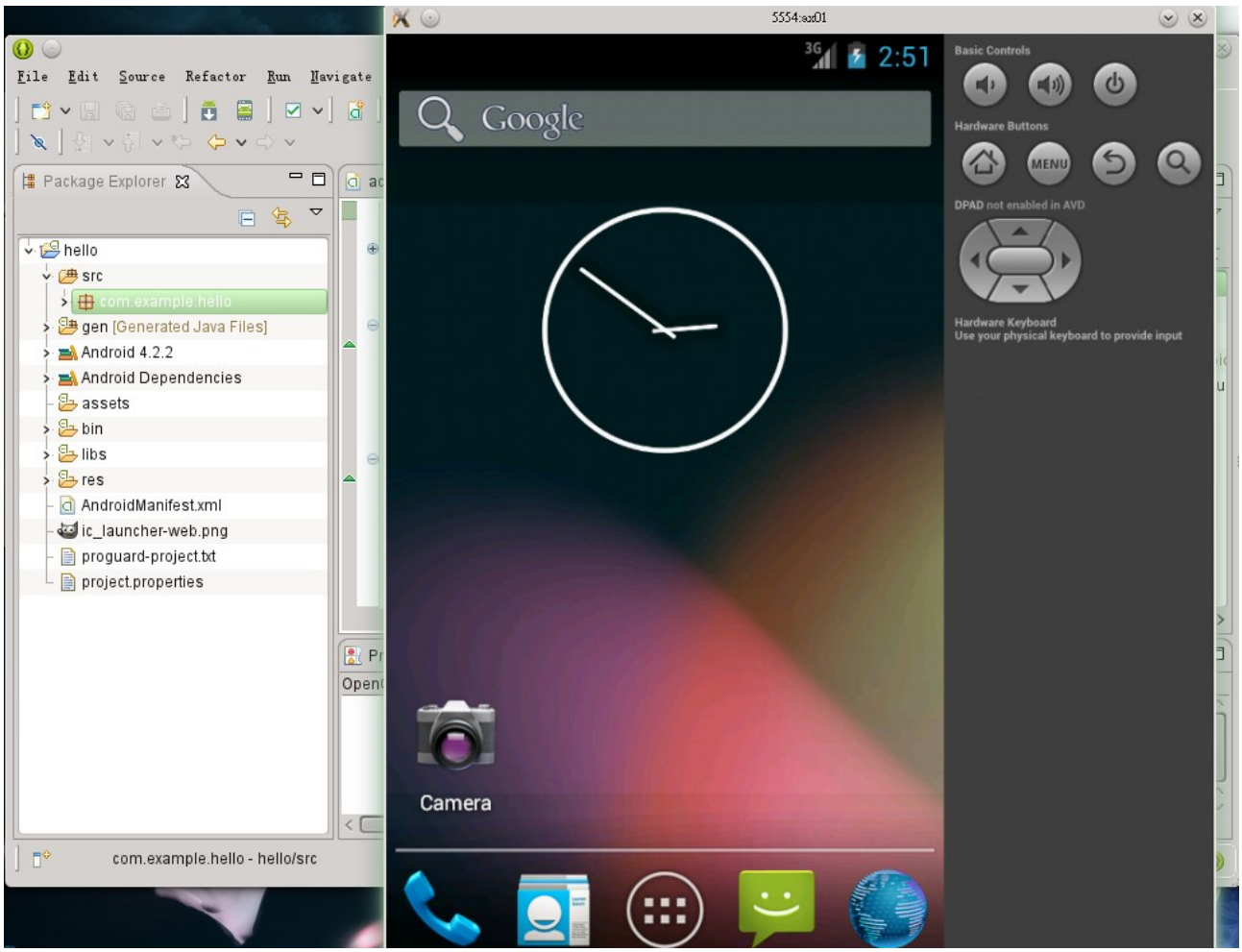


Code::Blocks 官网：
http://www.codeblocks.org/
CodeLite 的 openSUSE12.3 下的包好像还没打好，可能要去官网下载源码编译安装了：
http://www.codelite.org/

17.8 Android 开发环境

Android 是著名的开源智能手机操作系统，Android SDK 是其开发环境，如果要为 Android 开发软件包，可以在官网下载全套的 ADT (Android Developer Tools)：
http://developer.android.com/sdk/index.html
32 位系统下载 adt-bundle-linux-x86-*.zip，64 位系统下载 adt-bundle-linux-x86_64-*.zip 。

比如下载的是 adt-bundle-linux-x86_64-20130219.zip，直接解压缩到当前用户目录的 bin 文件夹里，如 /home/suse123/bin/ ,然后进入目录 /home/suse123/bin/adt-bundle-linux-x86_64-20130219/eclipse/ ，直接运行里面的 eclipse 即可。由于 openSUSE 默认就装了 openjdk 到系统路径，所以不需要配置，就可以使用 Android 开发平台。



上面方法是最简单的安装方式，ADT (Android Developer Tools)包含了 Eclipse IDE、Android 开发库、模拟器，应有尽有。还有一个稍微麻烦的安装方式。

如果系统里面已经有了现成的 openJDK 和 Eclipse，希望通过插件方式集成 Android SDK Tools 也是可以的，需要连网下载 Eclipse 插件，具体配置方式参考：

http://developer.android.com/sdk/installing/index.html

http://www.cnblogs.com/pharen/archive/2011/09/13/2174592.html

新手当然推荐第一种最简单的 ADT (Android Developer Tools)，不需要配置，直接使用。

17.9 版本控制软件

版本控制软件是开发软件、维护软件源码、软件代码更新、下载和发布的综合管理平台。比如 openSUSE 的源代码就完全通过 git 来开发、管理、下载、发布的（如 http://gitorious.org/opensuse ）。因为开源社区维护开源软件时，很多时候同一软件的开发者不可能始终凑一起开发，所以分布式版本控制软件就很必要。可以在各种环境开发和更新软件源码。常用的版本控制系统有：CVS（Concurrent Versions System）、SVN（Subversion）、Git、Mercurial、Bazaar 等。

关于 Linux 下的版本控制软件，推荐看这个帖子：

http://www.ibm.com/developerworks/cn/linux/l-vercon/index.html

17.9.1 CVS

CVS 是一个老牌的版本控制系统，根据维基百科介绍，很多开源或者自由软件项目都使用 CVS 作为其程序员之间的中心点，以便能够综合各程序员的改进和更改。这些项目包括：
Gnome、KDE、GIMP、Wine 等。CVS 的使用获 GNU 通用公共许可证授权。
CVS 安装命令：

```
zypper in cvs cvs-doc cervisia
```

查看 CVS 内置帮助说明的命令：

```
cvs --help
```

17.9.2 SVN

根据维基百科介绍，Subversion，简称 SVN，是一个开放源代码的版本控制系统，相对于的 RCS、CVS，采用了分支管理系统，它的设计目标就是取代 CVS。互联网上越来越多的控制服务从 CVS 转移到 Subversion。2009 年 11 月，Subversion 被 Apache Incubator 专案所接收。2010 年 1 月，正式成为 Apache 软件基金会的一个顶级专案。
SVN 安装命令：

```
zypper in subversion subversion-doc kdesvn
```

还有其他 Subversion 相关的包，比如 cvs2svn 是将 CVS 源转换成 SVN 的源， kdesvn 是 SVN 的图形客户端，subversion-server 和 subversion-tools 是搭建 SVN 服务器的 Apache 软件模块和管理工具。查看 SVN 内置帮助说明的命令：

```
svn help
```

17.9.3 Git

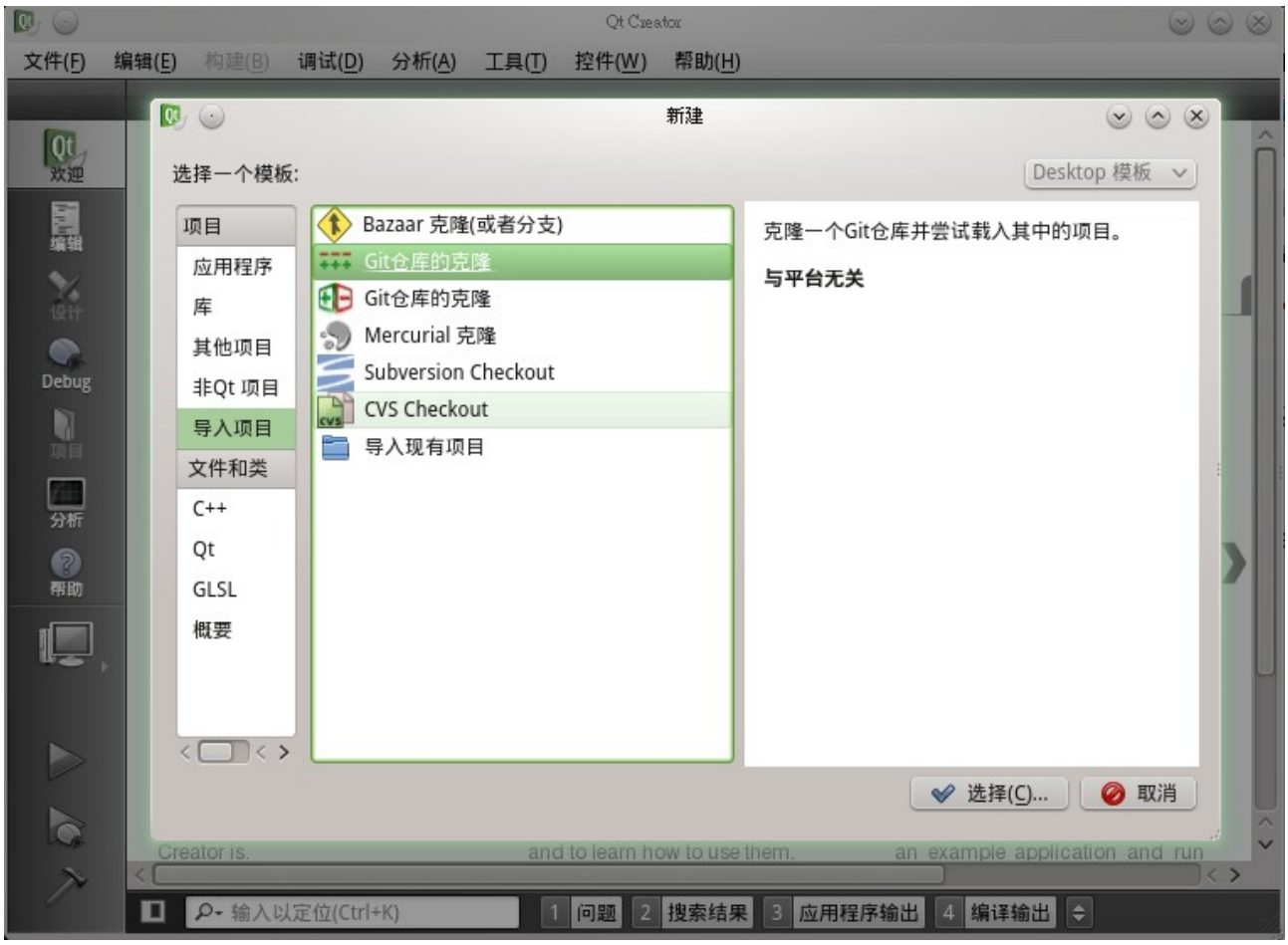
Git 是 Linux 内核开发老大 Linus Torvalds 创立的，是为了更好地管理 Linux 内核。当然现在不止是内核，发行版如 Fedora、openSUSE、Arch Linux，图形库如 Qt、Gtk+，还有很多基础软件源码也都是 Git 管理、下载和发布的。
Git 安装命令：

```
zypper in git qgit
```

软件包会自动安装依赖包：git-core、git-cvs、git-email、git-gui、git-svn 、git-web、gitk。除了默认的 git-gui，还有 Gtk+版的 gitg 图形界面和 Qt 版的 qgit 图形界面。
查看 Git 内置帮助说明的命令：

```
git help
```

安装了 Git 之后，前面介绍过，几乎所有的集成开发环境都带了各种版本控制软件的功能插件。比如装了 Git 之后，KDevelop 的 Git 插件就可以正常使用了，不会再报插件禁用的问题。QtCreator 可以从版本控制系统管理的项目导入现有项目，也可以通过 Git 插件发布代码到网上。



17.9.4 Mercurial

根据维基百科介绍，Mercurial 是一个跨平台的分布式版本控制软件。Mercurial 主要由 Python 语言实现，不过也包含一个用 C 实现的二进制比较工具。Mercurial 主要由一个命令行程序组成，但现在也有了图形用户界面。对 Mercurial 的所有操作都由用不同的关键字作为参数调用程序 hg 来实现，Hg 是参考水银的化学符号而取的名字。Mercurial 是以 GPLv2 发布的自由软件。

采用 Mercurial 管理项目的软件有

Mozilla、OpenJDK、OpenSolaris、Xen、Audacious、Netbeans 等。

Mercurial 安装命令：

zypper in mercurial mercurial-lang hgview-qt4

查看 Mercurial 内置帮助说明的命令：

hg help

17.9.5 Bazaar

根据百度百科，Bazaar 是一个分布式的版本控制系统，采用 GPL 许可协议，可运行于 Windows、GNU/Linux、UNIX 以及 Mac OS 系统之上。Bazaar 由 Canonical 公司（Ubuntu 母公司）赞助。 使用 Bazaar 的项目或组织包括 MySQL、 Mailman、Launchpad、Linux Foundation 等等，Ubuntu 和 Debian 的很多东西当然也用 Bazaar 管理。

Bazaar 安装命令：

zypper in bzr bzr-lang

查看 Bazaar 内置帮助说明的命令：

bzr help