Knoppix OS and Open Source Development Software Tools

Agenda
- What is Knoppix
- Knoppix Features
- Open Source Development Tools
  - QT
  - UML
  - Eclipse
  - Mono C#

What is Knoppix
- Knoppix, or KNOPPIX (k-NOP-iks), is an operating system based on Debian.
- It is designed to be run directly from a CD / DVD (Live CD) and a USB key (Live USB) One of the first of its kind for any operating system.
- Knoppix was developed by Linux consultant Klaus Knopper.

Knoppix Features
- Computers that support booting from USB devices can load Knoppix from a live USB flash drive or memory card.
- Aufs: When starting Knoppix program, it is loaded from the optical disc/USB and decompressed into a RAM drive. The decompression is transparent and on-the-fly.
- More than 1000 software packages on the CD edition.
- More than 2600 are included on the DVD edition. Up to 9 gigabytes on the DVD in compressed form.

Knoppix Features
- Knoppix was one of the first Live CDs available, and is known as the "original" Debian-based Live CD.
- Its extensive hardware detection allows most systems to start Knoppix without any configuration.
- Its ability to automatically connect to most kinds of networks.
- Its utilities for system repair and troubleshooting.

System Requirements
- Requirements to run Knoppix:
  - Intel-compatible processor (i486 or later)
  - 32 MB of RAM for text mode, at least 64 MB for graphics mode with LXDE (at least 128 MB of RAM is recommended to use the various office products)
  - Bootable CD-ROM drive, or a boot floppy and standard CD-ROM (IDE/ATAPI or SCSI)
  - Standard SVGA-compatible graphics card
  - Serial or PS/2 standard mouse or IMPS/2-compatible USB-mouse
Qt Designer

- Creating a Qt Application
- Creating Main Windows with Actions, Toolbars and Menus
- The Designer Approach
- Subclassing and Dynamic Dialogs
- Creating Custom Widgets
- Creating Database Applications
- Customizing and Integrating Qt Designer

Creating a Qt Application

- Introduces Qt Designer.
- Step by step through the creation of a small but fully functional application.
- Create a form and add widgets to it.
- Use the form and property editors to customize the application.
- Lay out a form using the layout tools.
- Use Qt’s signals and slots mechanism.

Creating a Qt Application

- Use Qt Designer’s built-in code editor to make the application functional.
- Use qmake to generate a Makefile so that you can compile and run the application.

Creating Main Windows with Actions, Toolbars and Menus

- Example: Build an application called ‘multiclip’, which allows you to store and retrieve multiple text clippings to and from the clipboard.

Creating Main Windows with Actions, Toolbars and Menus

- Learn how to create a simple text editor.
- Learn how to create a main window with menus and toolbars.
- Learn how to use Qt’s built-in functionality to handle common tasks (e.g. copy and paste handling).
- Learn how to create your own functionality for your own menu items and toolbar buttons.
The Designer Approach
- Information on the Qt Designer approach to developing applications.
- Explains some of the rationale behind Qt Designer.

Subclassing and Dynamic Dialogs
- Learn how to subclass a form.
- Learn how to clearly separate the user interface from the underlying code that implements its functionality.
- Additional information on qmake and uic.
- Explain how you can dynamically load dialogs from .ui files into your application using QWidgetFactory.
- Learn how to access the widgets and sub-widgets of these dialogs.

Creating Custom Widgets
- Explains how you can create your own custom widgets.
- Introduction to the simple method and the new more powerful method of Qt Designer.
- Using plugins in QT Designer.

Creating Database Applications
- Introduces Qt's SQL classes.
- Learn how to execute SQL queries.
- Learn how to set up master-detail relationships.
- Learn how to perform drilldown and handle foreign keys.

Customizing and Integrating Qt Designer
- Focuses on Qt Designer itself.
- Learn how to customize QT Designer.
- Learn how to integrate Designer with Visual Studio and how to create Makefiles.

Umbrello UML Modeller
- Umbrello UML Modeller is a Unified Modelling Language diagram programme for KDE.
- UML allows you to create diagrams of software and other systems in a standard format (Review already learned UML knowledge).
Working with Umbrello

- Working with Umbrello UML Modeller.
  - User Interface
  - Tree View
  - Documentation Window
  - Work Area
  - Code Import and Code Generation
  - Copying UML objects as PNG images, exporting to an Image

Please refer to the resource about Umbrello for more detailed look about this open source tool.

Working With Eclipse

- Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system.
- It is written primarily in Java and can be used to develop applications in Java.
- By means of various plug-ins, other languages including C, C++, COBOL, Python, Perl, PHP and Ruby (including Ruby on Rails framework).

The IDE is often called
- Eclipse ADT for Ada
- Eclipse CDT for C/C++
- Eclipse JDT for Java
- Eclipse PDT for PHP

Eclipse is a very popular development IDE, please refer to the contents of Web resources for its detailed introduction.
http://www.eclipse.org/

Mono C#

- Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Novell
- Mono is an open source implementation of Microsoft’s .NET Framework based on the ECMA standards for C# and the Common Language Runtime.
- A growing family of solutions and an active and enthusiastic contributing community is helping position Mono to become the leading choice for development of Linux applications.

Working with MonoDevelop IDE

- MonoDevelop is an IDE primarily designed for C# and other .NET languages.
- MonoDevelop enables developers to quickly write desktop and ASP.NET Web applications on Linux, Windows and Mac OSX.
- MonoDevelop makes it easy for developers to port .NET applications created with Visual Studio to Linux and to maintain a single code base for all platforms.
Feature Highlights(1)

- Multi-platform:
  - Supports Linux, Windows and Mac OSX.
- Advanced Text Editing
  - Code completion support for C# 3, code templates, code folding.
- Configurable workbench
  - Fully customizable window layouts, user defined key bindings, external tools
- Multiple language support
  - C#, Visual Basic.Net, C/C++, Vala

Feature Highlights(2)

- Integrated Debugger
  - For debugging Mono and native applications
- GTK# Visual Designer
  - Easily build GTK# applications
- ASP.NET
  - Create web projects with full code completion support and test on XSP, the Mono web server.
- Other tools
  - Source control, makefile integration, unit testing, packaging and deployment, localization

MonoDevelop Main Window

- Screenshot

Integrated Debugger

- Screenshot

NUnit add-in

- Screenshot

MonoQuery add-in

- Screenshot
Add-in Manager

• Screenshot

GUI designer

• Screenshot

MonoDevelop resource

For student how had already learned C# & .NET courses prior to this course.

Please refer to the contents of Web resources for the detailed introduction of MonoDevelop of C#.

http://monodevelop.com/
http://monodevelop.com/Documentation

Resources

- Knoppix官方网站：http://www.knopper.net/knoppix/
- Comparison of Linux distributions：http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions
- QT home page：http://qt.nokia.com/
- Umbrello UML：http://docs.kde.org/
- Eclipse：http://www.eclipse.org/
- Mono：http://monodevelop.com/

谢谢！