R • I • T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences and Technologies

4002-XXX
Software Development on Linux Systems
Lab 04 – Using Development Tools and Frameworks; Exploring IDEs

Name: _______________________________________ Section: _____________

Activity – Installing Base Software Kits

Ubuntu:
1) Open a terminal
2) Type "sudo apt-get install build-essential"

Fedora:
1) Open a terminal
2) Su to root : Type "su -"
3) Type "yum groupinstall "Development Tools"

This will install tools for packaging and software development, as well as gcc, g++ and make. These will install all the tools necessary for building packages on Fedora or Ubuntu. This also includes all the information needed by the system to build package.

Activity – Installing Software Development Kits

C++, C, Perl and Python will have already been installed. Now we will install other common compilers.

1) Installing Java. We will use the open source version of the Java Developer Kit
   Using your preferred method of installation install the openJDK
   These are the package names-

   Ubuntu: openjdk-6-jdk
   Fedora: java-1.6.0-openjdk-devel

2) Installing OpenGL. This implement of OpenGL is called Mesa

1) OpenGL

Using your preferred method of installation install OpenGL.
These are the package names-

Ubuntu: mesa-common-dev
Fedora: mesa-libGL-devel

2) OpenGL Utility Library (GLU). This is not the same as GLUT

Using your preferred method of installation install libGLU
These are the package names-

Ubuntu: libglu1-mesa-dev
Fedora: mesa-libGLU-devel

3) Mesa Utilities

Using your preferred method of installation install mesa utils
These are the package names-

Ubuntu: mesa-utils
Fedora: glx-utils

3) Installing Simple DirectMedia Layer (SDL). This is a powerful graphics library for openGL that has become common for building applications and games. SDL by default works with c++ and c, but can also be used with C#, Java, Perl and Python with the appropriate libraries.

Using your preferred method of installation install SDL
These are the package names-

Ubuntu: libsdl1.2-dev
libsdl1.2-gfx1.2-dev
libsdl1.2-image1.2-dev
libsdl1.2-mixer1.2-dev
libsdl1.2-net1.2-dev
libsdl1.2-ttf2.0-dev

Fedora: SDL-devel
SDL_gfx-devel
SDL_image-devel
SDL_mixer-devel
SDL_net-devel
SDL_ttf-devel

4) Installing Boost Libraries. Boost libraries are a collection of C++ libraries to overcome programming difficulties, such as filesystem integration, geometry, python calls, regex and threads.

Using your preferred method of installation install the boost collection
These are the package names-

Ubuntu: libboost-all-dev
Fedora: boost-devel

5) Installing C#. This is referred to as Mono on Linux

Using your preferred method of installation install mono

These are the package names-

Ubuntu: monogmcs
mono-devel
mono-utils
monodoc-browser

Fedora: mono-core
mono-devel
monodoc

Activity – Compiling

You may use either Ubuntu or Fedora for this section.

-Download the lab 4 files from mycourses.

-Unzip the file

-Open terminal

-Cd to the unzipped directory

1) Compiling C

a) In terminal type "gcc Hello.c -o Hello"
(-o means output to file with the following name)

b) Type "./Hello"   (That's dot slash)

2) Compiling C++

a) In terminal type "g++ Hello.cpp -o Hello"

b) Type "./Hello"

3) Compiling Java

a) In terminal type "javac Hello.java"

b) Type "java Hello"

4) Running Perl

a) In terminal type "perl Hello.pl"

Note: perl can be compiled with the "pp" command

5) Running Python
   a) In terminal type “./Hello.py”

   Note: python can be compiled with “python -m compileall fileName.py”

6) Compiling C#
   a) In terminal type “gmcs Hello.cs”
   b) Type “./Hello.exe”

7) Compiling C++/C with boost libraries
   Libraries are added with the -l (that's an L) flag.

   Open the file ThreadTest.cpp and notice that it has an include for boost thread

   a) In terminal type “g++ ThreadTest.cpp -o ThreadTest -lboost_thread”
   b) Type “ThreadTest”

8) Compiling C++/C with SDL

   a) Untar the file sdl-colorkey.tar.gz
   b) Cd to the sdl-colorkey directory
   c) Type “g++ -o colorkey colorkey.cpp -lSDL -lSDL_image”
   d) Type “./colorkey”

Note: The colorkey demo is a tutorial that is GNU Free Documentation Licensed and the code is GPL licensed. You may find it here: http://ubuntu-gamedev.wikispaces.com/Drawing+Sprites+Using+SDL

Show the results of all of these to your instructor/TA to get credit

Instructor/TA Sign-Off ________________________________

R • I • T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences and Technologies

Activity – Installing Frameworks

You will be installing various software development frameworks. You may decide to use one of these for your project.

1) QT
   a) Install QT 4
      These are the package names-
      Ubuntu: libqt4-dev
      Fedora: qt-devel

   b) Test QT
      I) In terminal cd to the QT directory in the folder you downloaded in mycourses
      II) In that folder type "qmake-qt4 -project"
      III) Type "qmake-qt4"
      IV) Type "make"
      V) Type "ls" and you will now see various files
      VI) Now we need to make the file executable. Type "chmod 744 QT"
      VII) Type "./QT" and you should see a window with a button

2) GTK+
   GTK+ is what the Gnome desktop is built on. Because Ubuntu and Fedora run on Gnome, GTK+ is installed by default. Many languages can integrate with GTK+, but for this exercise we will use python.
   a) Install Python GTK+ (pyGTK)
      These are the package names-
      Ubuntu: python-gtk2-dev
      Fedora: pygtk2-devel

   b) Test GTK
      I) In terminal cd to the GTK directory in the folder you downloaded in mycourses
      II) In that folder type "chmod 744 HelloWorld.py"
      III) Type "./HelloWorld.py"

Show the results of all of these to your instructor/TA to get credit

Instructor/TA Sign-Off ________________________________

Activity – Installing Integrated Development Environments (IDEs)
There are many IDEs available on Linux for almost any purpose or need. We will install a few of the more popular IDEs.

1) VIM

Vim was used in the last lab and will not be covered here. It is however, a very powerful IDE and is completely acceptable to use for this course.

2) Gedit

Gedit comes with Ubuntu and Fedora by default. It is a basic text editor with capabilities of color coding and working with code in many languages.

To turn gedit into an IDE install the developer plugins
Package names-
Ubuntu: gedit-developer-plugins
Fedora: gedit-plugins

3) Eclipse

Eclipse is a major IDE for developing in many languages. Eclipse is rich in plugins and integrates easy into most environments.

Install Eclipse
These are the package names-

Ubuntu: eclipse
Fedora: eclipse

4) MonoDevelop

MonoDevelop is a powerful IDE for programming in C#. It is specific to C#, but is an interactive IDE that allows rapid GUI programming and integration with the .NET framework.

Install MonoDevelop
These are the package names-
Ubuntu: monodevelop
            monodevelop-versioncontrol

Fedora: monodevelop

5) Anjuta

Anjuta is a powerful IDE similar to Eclipse intended for use with C, C++, Java, Python and Vala. Though it is less powerful than Eclipse, it is much lighter and integrates well with gcc/g++.

Install Anjuta
These are the package names-
Ubuntu: anjuta
Fedora: anjuta

6) Bluefish

Bluefish is a lightweight IDE intended for web development. It is full of powerful and rich web developer tools. Bluefish can handle HTML, XML, CSS, PHP, Javascript, Java, SQL, Perl, Python and many other languages.

Install bluefish
These are the package names-
Ubuntu: bluefish
Fedora: bluefish

R · I · T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences and Technologies

Activity – Configuring IDEs

1) Configuring Gedit

Now that you have the plugins installed, open Gedit

1) Go to edit → preferences
2) Check “highlight matching bracket” in the view tab
3) Go to the plugins tab
4) Check “code comment”
5) Check “embedded terminal”
6) Check “snippets”
7) Click the editor tab
8) You may check “Automatic indentation”, “Create backup before saving” or “Autosave” if you would like to.

2) Configuring Eclipse

Now that you have installed Eclipse, open it

We will use PHP developer tools as an example

1) Choose a location for the workspace
2) In Eclipse, click help from the menu and choose "install new software"
3) In the "work with" field type

   http://download.eclipse.org/tools/pdt/updates/2.0

   If this gives any errors try
   http://download.eclipse.org/tools/pdt/updates

4) Click "add"
5) Check the box marked PDT SDK with the highest number after it
6) Hit "next"
7) Hit "next"
8) Mark "agree" to the terms of service and hit "finish"
9) After it finishes, hit "restart now"
10) Choose a location for the workspace
11) Choose File → new → project
12) We will choose a PHP project. Scroll down to PHP and choose "PHP Project"
13) Enter a name for the project
14) Choose "create new project in workspace"
15) Choose "use default PHP settings"
16) Hit "next"
17) Here you would be able to choose which external libraries to include in the project. For now, we will not include any. Hit "next"
18) This provides the PHP build path. We will leave it as the default for now. Click "finish"
19) When asked to open the PHP perspective choose "Yes"
20) On the left right click on the folder for your project and choose " new → PHP File "
21) Name the file and hit "finish"

You should now have a PHP file you can code with

3) Configuring MonoDevelop

Now that you have MonoDevelop installed, open it

1) On the left choose “start a new solution”
2) Name your solution and hit “next”
3) On the menu bar, choose “file → new → file”
4) Under C# choose “general → general → empty file” and click “new”

4) Configuring Anjuta

Now that you have Anjuta installed, open it

1) Click “new file”

5) Configuring Bluefish

Now that you have Bluefish installed, open it

1) You should be able to just start using Bluefish once it is opened

Instructor/TA Sign-Off ______________________________