VLC开发准备工作

本笔记在本人完成VLC源码编译并成功运行后的总结笔记，可以作为以后相同工作的参考。本人在进行这部分工作时，主要参考了VLC官方开发[wiki](https://wiki.videolan.org/Developers_Corner)。

1. VLC开发介绍

根据官方wike的[Getting Started At Coding](https://wiki.videolan.org/Getting_Started_At_Coding/)的内容，VLC的开发工作将按照获以下顺序进行：

1. 获取源码并编译；
2. 理解源码；
3. 编写自己的代码。
4. 获取VLC源码

获取VLC源码的方式主要有两种，通过VLC官方的git远程仓库clone到本地，也可以从官网直接下载。前者的好处是可以获取整个VLC源码的版本管理仓库，也可以生成自己的版本分支，为VLC项目做贡献；而后者下载速度更快。这里我们不需要各个版本分支，也并非给VLC项目做贡献，推荐用使用后者。

源码最新版下载：<http://www.videolan.org/vlc/download-sources.html>（推荐）

源码各个版本下载：<http://download.videolan.org/pub/vlc/>

各个获取方法：<https://wiki.videolan.org/GetTheSource/#How_to_get_VLC_media_player.27s_latest_source_tree>

本人下载到的是目前的最新的2.2.4版本源码vlc-2.2.4.tar（181MB）。

1. 准备开发环境

对VLC不同的运行环境，需要的开发环境是不同的，本文只针对运行于Windows平台的VLC源码开发环境搭建进行说明。

有三种源码编译方法，见下页表。官方推荐使用**MinGW** on Linux的方法，需要电脑运行Microsoft Windows，在虚拟机中运行Linux，在Linux进行编译。官方也只对这种方法进行了详细的步骤说明。故采用**MinGW** on Linux的方法。

|  |  |
| --- | --- |
| Method | Notes |
| **MinGW** on Linux | Preferred method involving cross-compilation from Linux. On computers running Microsoft Windows, a virtual machine is necessary to run Linux. |
| **[MSYS+MinGW](http://www.mingw.org/)** on Windows | Native compilation method. MSYS is a minimal build environment to compile Unix-style projects directly on Microsoft Windows. This isnot officially supported by VideoLAN. Best of luck. |
| **Cygwin** on Windows | Cygwin method. Cygwin emulates a POSIX environment to compile on Microsoft Windows. This is error-prone and slow and therefore not recommended. |

对于虚拟机和Linux，官方并未指定，但指出Ubuntu 14.04版本太旧，不能编译VLC。本人使用的是VMware® Workstation 12 Pro虚拟机和ubuntu-16.04.2-desktop-amd64。

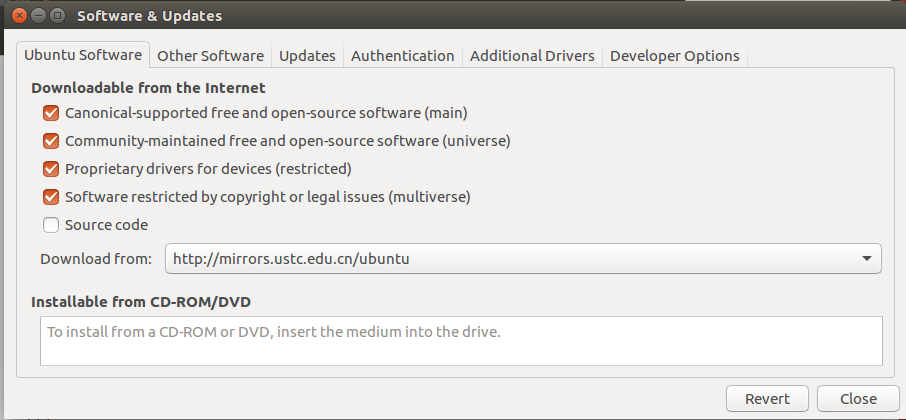
* 1. 设置Ubuntu

为了方便后面的工作，需要对Ubuntu进行一些设置。

首先为了能方便的进行虚拟机和主机直接文件和文本的复制粘贴和拖动，需要安装VMware Tools，可以参考[百度经验——Ubuntu下正确安装VMware Tools](http://jingyan.baidu.com/article/93f9803f0d9d9be0e46f55ce.html)。

然后，由于Ubuntu默认从美国服务器下载更新软件，会导致后续各种开发环境下载速度过慢甚至失败，需要在Ubuntu的设置中更改为中国的镜像服务器。

System Setting --> Software & Updates --> Downlod from:框中点击Other...，在新的对话框中，找到China点击，再选择mirror.ustc.edu.cn，这是中科大的镜像服务器。然后点击Choose Server，然后稍等系统更新即可。等待到最后可能会报错，无视即可。



设置完成后，如上图所示。下载更新速度将会从原本的每秒几KB提高到近1MB。

* 1. 编译连接工具Mingw-w64

对于编译64-bit版和32-bit版的VLC，打开终端，输入命令

32-bit版本：

sudo apt-get install gcc-mingw-w64-i686 g++-mingw-w64-i686 mingw-w64-tools

64-bit版本：

sudo apt-get install gcc-mingw-w64-x86-64 g++-mingw-w64-x86-64 mingw-w64-tools

输入后，将自动进行获取并安装Mingw-w64.

* 1. 开发工具
* lua (5.2)
* all autotools: libtool, automake, autoconf, autopoint, make, gettext, pkg-config
* qt4-dev-tools, qt5-default (or qt4-default if qt plugin build fails)
* git, subversion cmake, cvs if you want to rebuild contribs
* wine-dev for creating Win32 packages
* zip [for creating .zip package], p7zip [for .7z package], nsis [for .exe auto-installer], bzip2 [for 'make prebuild]

在终端输入：

sudo apt-get install lua5.2 libtool automake autoconf autopoint make gettext pkg-config

sudo apt-get install qt4-dev-tools qt5-default git subversion cmake cvs

sudo apt-get install wine64-development-tools zip p7zip nsis bzip2

sudo apt-get install yasm ragel ant default-jdk protobuf-compiler dos2unix

将会自动完成各个开发工具的下载、配置和安装。

* 1. 复制源码

将之前下载的源码包复制入虚拟机。并将源码的内容解压到要放置的目录中。

* 1. 准备第三方库

可以根据需要，选择预编译的第三方库（速度快）或自己编译第三方库（速度慢）。预编译库只能对VLC 2.2.x版本的源码工作，更多细节可以参见：[Prepare 3rd party libraries](https://wiki.videolan.org/Win32Compile/)。本文采用预编译的第三方库。

在源码的根目录下，打开终端，输入以下指令（其中的**XXX**，对32bit版VLC替换为i686-w64-mingw32，对64bit版VLC替换为x86\_64-w64-mingw32）：

mkdir -p contrib/win32

cd contrib/win32

../bootstrap --host=**XXX**

make prebuilt

在../bootstrap --host=**XXX**这一步时，会下载文件：vlc-contrib-x86\_64-w64-mingw32-latest.tar，速度可能较慢，可以从<http://ftp.vim.org/mediaplayer/vlc/contrib/x86_64-w64-mingw32/>下载相应的压缩包后，将其复制到win32文件夹中，并重新输入指令

../bootstrap --host=XXX

make prebuilt

* 1. Linux 64-bit

对于64bit的Linux，还需要删除一些文件。

保持终端在contrib/win32下，输入命令：

rm -f ../i686-w64-mingw32/bin/moc ../i686-w64-mingw32/bin/uic ../i686-w64-mingw32/bin/rcc

最后，让终端回到源码目录。

* 1. 配置并编译

在源码根目录，打开终端，输入：

./bootstrap

mkdir win32 && cd win32

然后配置 PKG\_CONFIG\_LIBDIR，根据自己的目录调整下面指令其中的**XXX**，对32bit版VLC替换为i686-w64-mingw32，对64bit版VLC替换为x86\_64-w64-mingw32）：

export PKG\_CONFIG\_LIBDIR=$HOME/vlc/contrib/**XXX**/lib/pkgconfig

配置，**XXX**含义同上：

sudo ../extras/package/win32/configure.sh --host=**XXX** --build=x86\_64-pc-linux-gnu

sudo ../configure --host=**XXX** --build=x86\_64-pc-linux-gnu

配置完成后，编译

make

* 1. VLC打包

当编译完成后，可以通过下表的方式建立自己打VLC包：

|  |  |
| --- | --- |
| Command | Description |
| make package-win-common | Creates a subdirectory named vlc-x.x.x with all the binaries. You can run VLC directly from this directory. |
| make package-win-strip | Same as above but will create 'stripped' binaries (that is, smallest size, unusable with a debugger). |
| make package-win32-7zip | Same as above but will package the directory in a 7z file. |
| make package-win32-zip | Same as above but will package the directory in a zip file. |
| make package-win32 | Same as above but will also create an auto-installer package. You must have NSIS installed in its default location for this to work. |

本人只成功了第一种方式，输入

make package-win-common

将在源码根目录/win32/下生成一个vlc-x.x.x的文件夹，这个文件夹可以手动打包，拷贝到Windows中，解压后，双击vlc.exe即可运行。

1. 后续工作

后续将开始进行选择性的理解源码工作。

在下周，了解源码的结构模块划分和结合方式，以及具体的VLC开发方法。