IPhone 开发自学笔记

# 环境搭建

## 前言

Windows 7 X64 + VMware Workstation8.0.2 + MAC OS X Lion 10.7.3 + xcode\_4.2\_and\_ios\_5\_sdk\_beta\_7\_for\_lion 安装

开发iPhone程序必须要在MacOS下利用Xcode开发，可一个apple MacBookPro可要不少大洋，新的MBP少说8000，对于拿着基本生活工资的我们简直就是一边天文数字，没有办法，用小黑顶替吧！下面就说说如何 在windows环境下搭建iPhone开发环境。

## 硬件要求：

1.CPU支持VT，称作虚拟技术，如果没有VT只能运行Mac OS X10.5了，10.6Snow Leopard以上必须要支持VT！怎么知道自己的cpu支不支持VT？<http://ark.intel.com/VTList.aspx> ，英特尔会告诉的！

2.内存怎么个也要4g，2g内存装个虚拟机会卡的，硬盘要留40g及以上给虚拟机硬盘，所以确保你的硬盘剩余空间大于40gb。MacOsX镜像大概4g多，加上Xcode大概4g多，注意磁盘空间容量。

## 软件需求：

1.Vmware：VMware-workstation-full-8.0.2-591240

2.苹果操作系统：Mac OS 10.7.3。下载地址暂且不提供。在google上搜索Mac OS X Lion 10.7.3会出来结果的。

3.darwin.iso。如果不安装VMware的mac补丁,则需要使用这个软件来引导iso安装mac,这个iso是用来引导安装和启动mac操作系统的，非常重要，mac装到虚拟机里不能自己启动，需要引导，这个镜像就是干这个的。下载：<http://115.com/file/e65vi0nu>。

小编在安装中发现，在安装mac 10.6时可以使用darwin.iso来引导，但是需要修改VM文件夹中的vmx文件，修改guestOS = "darwin10"，建议安装vm 8 mac补丁，在百度上搜索VMware.unlock.rar，应该可以下载到，详细安装过程参见VMware 8.0.2 安裝 Mac OS X Lion 10.7.3.docx,文档是在网上搜到的，安装过程不再赘述，这里记录一些关于安装过程中遇到的常见问题，希望能在大家安装mac os遇到问题时，提供一些帮助。

## 在安装中的常见问题:

1. Mac OS X Lion 10.7.3 Retail.dmg 只能使用7z将InstallESD.dmg加压之后转换ISO,之后才可以使用InstallESD.iso安装mac os
2. 1 dmg文件转换ISO,使用UltraISO\_9可以将mac InstallESD.dmg文件转换为ISO文件
3. darwin.iso安装mac时会出现 EBIOS Read Error，此时别着急，按回车，在按c选择 DVD启动，稍等片刻即可继续安装
4. 如果安装10.5 10.6则一定要记得修改vmx文件中guestOS = "darwin10"
5. 如果安装 10.7时则需要下载安装vm 8 mac选项补丁，之后VM中在安装os选项中即可出现Apple MAC OX选项，按照文档中的步骤就可以安装了
6. 在安装xcode\_4.2\_and\_ios\_5\_sdk\_beta\_7\_for\_lion时需要将mac中系统时间设置到2011年1月1号，否则不能通过验证，安装不上

安装好了之后,让我们开始iphone开发之旅，先来学习一下Objective-C基础吧

# Objective-C基础

Objective-C是C语言的一个扩展集。Object-C以C语言作为基础，在该语言中添加了一些微妙但意义重大的特性。使用过C++或者Java的同学一定会惊叹Objective-C程序的简短。

## HelloWorld Objective-C

1 使用Xcode创建Objective-C project

打开Xcode（4.2）之后，选择create new project,在Applation项目类型中选择Command Line Tool ,之后项目命名，选择目录就不详细说明了，这里要注意的一点是Type要选择Foundation(基础框架)

2 解构Objective-C程序

工程创建好后，我们会在xcode的project视图中看到工程结构，其中生成的main.m文件为程序的入口，打开main.xml我们会看到下面的代码：

|  |
| --- |
| #import <Foundation/Foundation.h>  int main (int argc, const char \* argv[])  {  NSAutoreleasePool \* pool = [[NSAutoreleasePool alloc] init];  NSLog(@“Hello World”);  [pool drain];  return 0;  } |

Xcode的.m扩展名表示文件含有Objective-C代码，应由Objective-C编辑器处理。C编辑其处理名称以.c结尾的文件。而C++编辑器处理.cpp文件。在Xcode中所有这些编译工作全部由GCC(GNU Compiler CollectionmGNU编译器合集)处理，这个编译器能够理解C语言的全部三个变体。

这里代码很简单只是打印helloworld,先来介绍一下特性 #import，为了防止文件重复被引用，通常在C中我们会使用#ifdef命令来避免这个问题，但是在Objective-C中使用#import可确保头文件只被包含一次，而不论此命令实际上在哪个文件中出现了多少次。

原理:在Prefix.pch文件中包含如下定义

|  |
| --- |
| #ifdef \_\_OBJC\_\_  #import <Foundation/Foundation.h>  #endif |

@符号 + “”在Objective-C中可以用作表示引用的字符串应该作为Cocoa的NSString元素来处理。NSLog方法参数要使用NSString元素，否则可能会崩溃。

NSString功能： 告知其长度；将自身与其他字符串比较；将自身转换为整型值或浮点值。