Part-B //2017/12/20

**Android**

Android是一种基于Linux的自由及开放源代码的操作系统。主要使用于移动设备，如智能手机和平板电脑，由Google（谷歌）公司和开放手机联盟领导及开发。

**JVM/DVM/JIT/ART/AOT**

1、JVM是什么？

JVM是Java Virtual Machine

JVM本质是软件，是计算机硬件的一层软件抽象，屏蔽了底层实现的差异

Java编译的程序会生成JVM字节码，字节码需要通过JVM被翻译成机器指令

JVM本质作用是将平台无关的.class内的字节码翻译成平台相关的机器码-实现跨平台

2、Dalvik虚拟机(DVM)是什么

是android中使用的虚拟机。所有的Android程序都运行在Android系统进程中，每个进程都对应一个Dalvik虚拟机实例

DVM和JVM都提供了对象生命周期管理、堆栈管理、线程管理、安全和异常管理以及垃圾回收等重要功能

3、DVM和JVM的区别

JVM运行的是JAVA字节码，通过解释class中内容来运行程序。

DVK运行的是Dalvik字节码, 均由Java字节码转换而来，并都被打包到DEX(Dalvik Executable)执行文件，DVM通过解释DEX文件来执行这些字节码.

JVM生成的class文件，每个class文件都会有一个常量池，会导致信息冗余。体积过大。

DVM生成的Dex文件共享一个常量池能减少体积。

JAVA虚拟机基于栈结构，程序在运行时虚拟机需要频繁的从栈上读取写入数据，这个过程需要更多的指令分派与内存访问次数，会耗费很多CPU时间。

Dalvik虚拟机基于寄存器架构，数据的访问通过寄存器间直接传递，这样的访问方式比基于栈方式要快很多。

4、JIT(Just in time)和AOT(Ahead of time)是什么？

Just In Time，即时编译技术

Ahead Of Time，预编译技术

JVM的做法是将字节码逐条解释翻译，比C/C++编译的可执行二进制字节码程序慢。通过JIT会将一些方法编译成对引得汇编代码，存储起来，以后调用的时候不需要解释可以直接运行。(Dalvik虚拟机在2.2中增加了JIT)

AOT就是指C/C++这类语言，编译器在编译时直接将程序源码编译成目标机器码，运行时直接运行机器码。

5、DVM和ART(Android Runtime)虚拟机的区别与联系

Dalvik虚拟机执行的是dex字节码，ART虚拟机执行的是本地机器码

DVM原理：依靠JIT编译器去解释执行，运行时动态地将执行频率很高的dex字节码翻译成本地机器码然后在执行，但是将dex字节码翻译成本地机器码是发生在应用程序的运行过程中，并且应用程序每一次重新运行的时候，都要重新做这个翻译工作.

ART原理：应用程序仍然是一个包含dex字节码的apk文件。在安装应用的时候，dex中的字节码将被编译成本地机器码，之后每次打开应用，执行的都是本地机器码。移除了运行时的解释执行，效率更高，启动更快。（安卓在4.4中发布了ART运行时）

6、ART的优点

系统性能显著提升

应用启动更快、运行更快、体验更流畅、触感反馈更及时

续航能力提升

支持更低的硬件

7、ART的缺点

更大的存储空间占用，可能增加10%-20%

更长的应用安装时间

**Xamarin**

Xamarin始创于2011年，旨在使移动开发变得难以置信地迅捷和简单。Xamarin的产品简化了针对多种平台的应用开发，包括iOS、Android、Windows Phone和Mac App。Xamarin由许多著名的开源社区开发者创立和参与，而且也是Mono项目的主导者——C#与·NET框架的开源、跨平台实现。

**打包党**

打包党，一些个人或者网络公司推出一些为方便个人下载和安装或商业利益，对别人或其他公司的产品进行再加工然后提供给用户下载，我们称之为打包党

**activity/ intent/ service/ layout/ findViewById/ toast**

**adb**

adb工具即Android Debug Bridge（安卓调试桥） tools。它就是一个命令行窗口，用于通过电脑端与模拟器或者真实设备交互。

**ndk**

NDK就是帮助我们可以在Android应用中使用C/C++来完成特定功能的一套工具.

**net**

NET是 Microsoft XML Web services 平台。XML Web services 允许应用程序通过 Internet 进行通讯和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web services 并将这些服务集成在一起之所需。对个人用户的好处是无缝的、吸引人的体验

**c#**

C#是一种安全的、稳定的、简单的、优雅的，由C和C++衍生出来的面向对象的编程语言。它在继承C和C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了VB简单的可视化操作和C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET开发的首选语言。

**kotlin**

Kotlin 是一个用于现代多平台应用的静态编程语言 [1]  ，由 JetBrains 开发。

Kotlin可以编译成Java字节码，也可以编译成JavaScript，方便在没有JVM的设备上运行。

Kotlin已正式成为Android官方支持开发语言。

**swift**

Swift是一款易学易用的编程语言，而且它还是第一套具有与脚本语言同样的表现力和趣味性的系统编程语言。Swift的设计以安全为出发点，以避免各种常见的编程错误类别