设备管理器和服务管理器 (此部分不清楚答案是什么)

* 设备管理器：device.exe（device.dll in CE6）
  + 作用和流程

负责加载和管理windows ce 下绝大多数的设备驱动程序，包括网卡驱动，电池驱动，声卡驱动，串口驱动，usb 设备驱动及pcmcia 驱动

流程： 内核-> device.exe->i/o 资源管理器 Iorm.lib->电源管理器Pm.dll-> 总线枚举器 BusEnum.dll –> PCIBUS.dll

* + 驱动程序的编写：动态链接库
  + 相应的注册表项
* 服务管理器：services.exe
  + 作用和流程:

Services.exe 负责加载所有的服务

服务的启动有两种方式，通过注册表中设置相应的注册表项，系统启动时让services.exe 加载服务;另外一种是应用程序使用函数手动加载服务。

* + 服务程序的编写：动态链接库
  + 相应的注册表项（Context项目，可配置为普通服务，独立服务，超级服务）

如果 某个服务dll 在单独的进程中被加载，那么此服务被称为独立服务。

超级服务侦听某个网络端口，当有客户连接这些端口时，服务程序会给客户提供相应的服务。

（只需把HKEY\_LOCAL\_MACHINE\Services\<servernname>\下的  
Context项设置为1）

6应用程序的开发

（Why Platform Manager?

WinCE应用程序开发与Windows桌面应用程序开发的主要不同：

* 在开发桌面应用程序时，应用程序同时在桌面操作系统中运行;
* 在开发WinCE应用程序时，需要将WinCE应用程序download到WinCE目标设备上运行，并且调试的情况也是一样。）

#include <windows.h>

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR szCmdLine, int nShow)

{

MessageBox(NULL, TEXT("Hello, Win32"), TEXT("HelloMsg"), MB\_OK);

return 0;

}

* WinMain() prototype

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPWSTR szCmdLine, int nShow)

Why?

* + Under Windows CE, the command-line string is a Unicode string. In all other versions of Windows, the string is always ASII.

7 系统定制与开发：

* 系统定制与开发的整体流程：

系统定制：1得到并安装bsp ;2 定制操作系统; 3下载到开发板上运行调试：3.1得到并安装bootLoader 3.2配置网络连接 3.3配置调试串口 3.4配置platform builder 3.5连接设置 3.6下载运行映像 ; 4发布操作系统

开发应用程序大致可以分三个步骤：

1 安装合适的sdk 2 编写代码和调试3 发布应用程序

* Build System – 4 steps
  + 与Build有关的文件：Source，DIRS，Makefile
  + 与系统初始化有关的文件：.bib, .reg, .dat, .db

1Sysgen 阶段（生成阶段）2 Feature build 阶段 （编译阶段）

3 Release copy 阶段（release 文件夹复制阶段） 4 Make Image 阶段（镜像打包阶段）

8 BSP

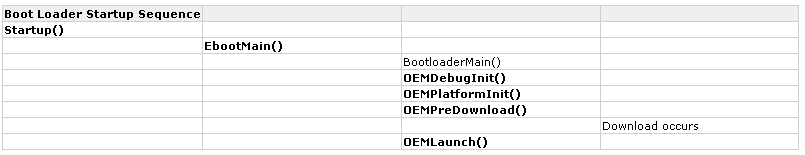
* BSP的概念与组成：4部分

BSP：Board Support Package

* + 主板硬件和操作系统之间的一层软件系统。严格地说，BSP属于操作系统的一部分
  + 解决操作系统跨不同CPU体系结构的方法之一：抽象操作系统和硬件之间的交互接口



* Bootloader的简单分析:功能：初始化目标硬件设备，控制启动过程，下载并执行操作系统映像
  + 代码框架：blcommon oem 代码 eboot 存储管理 EDBG 驱动程序
  + 程序流程



* 黑体字标识的函数需要由OEM厂商来实现.
* 用户需做的：

1实现OEM的应用程序接口（API）.

2连接Microsoft提供的库

引导程序 – StartUp函数

* 硬件复位和运行时复位需要执行的第一条指令
* 设置为超级用户模式
* 执行必须的硬件初始化:
  + CPU 内存控制器 系统时钟 串口 缓存 快表 (TLBs)
* 根据使用的CPU修改Startup.s

引导程序 – EbootMain

* EbootMain是C代码运行的入口
* 调用BLCOMMON库
* BLCOMMON 库 源文件在 Blcommon.c 文件中，路径为%\_WINCEROOT%\Public\Common\Oak\Drivers\Ethdbg directory

引导程序 – OEMDebugInit

* 用来初始化串行口，作为调试输出
* OEMDebugInit初始化完成后, 一个Windows CE的标记会出现，表示这个接口可以使用了.

引导程序 – OEMPlatformInit

* 各种OEM 硬件平台初始化函数，包括时钟, PCI接口,或者NIC接口.
* NIC接口用于下载映象，另外服务于后面一些函数.

引导程序 – OEMPreDownload

* 在加载一个运行时映象时首先被BLCOMMON调用.
* 查找硬件设备的IP地址，并与宿主机相连
* 如果出错返回-1

引导程序 – OEMLaunch

* OEMLaunch 是引导程序的最后一个需要运行的函数.
* 负责跳转的到需要运行的映象.
* 跳转到由dwLaunchAddr指定的第一条指令，这条指令在运行时映象的启动函数里.