

WINCE600 的目录组织及 PQOAL

在介绍 S3C2410 的 WinCE6.0 移植之前，我们先熟悉一下 WINCE600 的目录组织和 PQOAL 的相关概念。

这里简单说一下 WinCE6.0 开发环境的搭建。WinCE6.0 的开发环境是 Platform Builder6.0，但与 Platform Builder 5.0 不同的是，它作为 VS2005 的一个插件来用。所以，我们需要先安装 VS2005，并且安装相应的 SP1。PB6.0 目前应该是不支持 VS2008 的。然后顺序安装 PB6.0、PB6.0 的 SP1、R2。虽然微软没有严格说明，但还是建议将这些软件都默认安装到 C 盘，以免出现意外的情况。当然，这要求 C 盘足够大，至少得有 10G 的可用空间，以免安装的过程中出错。

安装完成后，C 盘的根目录下产生一个 WINCE600 的目录，这里包括 WinCE6.0 的源代码、BSP 和一些编译 WinCE 时需要用到的工具和库文件。另外还产生一个 C:\Program Files\Microsoft Platform Builder\6.00 的目录，这里主要包括 PB6.0 的开发环境工具、帮助文件和其他一些开发时用到的工具。

下面我们看一下 WINCE600 的目录组织。

目录	说明
CRC	存放了一个 PB6.0 安装时用到的校验文件 crc.ini
OSDesigns	用来存放 PB 的工程，对应 PB5.0 中的 PBWORKSPACES
OTHERS	存放了一些运行库、用于编译操作系统的二进制文件、注册表文件、批处理文件
PLATFORM	存放了和硬件平台相关的 BSP 及 MCU 相关的代码和其他一些文件
PRIVATE	存放了 WinCE6.0 开放的源代码
PUBLIC	存放了 WinCE6.0 的相关组件，这里应该是纯软件的代码和库
SDK	存放了用于编译 WinCE6.0 的相关工具和 DLL 文件

更详细的目录组织说明请参看 MSDN <http://msdn.microsoft.com/en-us/library/aa908702.aspx>。

一般来说，在移植 BSP 的过程中，我们只会修改 PLATFORM 下的相关目录和文件，而其他的除 OSDesigns 之外的目录，最好都不要修改，以免出错。PLATFORM 目录下除了 COMMON 目录外，其他都是特定硬件平台的 BSP，而 COMMON 目录中则包涵了不同体系结构的相关代码(ARM、X86、MIPS、SHX)、与平台无关的代码(COMMON)和 SOC 的相关代码(SOC)，这些代码都不能直接修改，如果需要修改，应该先 CLONE 出来，再做修改。

PQOAL 这个概念在 WinCE5.0 中就引入了，全称为 Production Quality OAL，产品级的 OAL。它的基本原则如下：

不同芯片或片上系统的代码必须分离开来；

BSP 中的代码应该是组件化的并且有逻辑的组织在一起；

芯片级代码、片上系统的代码和板级支持包(BSP)中的代码都应该是高质量的，以便于代码重用。

PQOAL 并不是一个硬性的规定，你可以按照自己的想法来组织 BSP 的目录及相关文件。但我还是建议尽量按照 PQOAL 的原则来组织，这会给后期的调试、移植和发布带来很大的便利。这是前人成功的经验，没有理由不用。

下面就从 PQOAL 的角度分析 PLATFORM 的目录组织。

PLATFORM\COMMON：这里存放了所有可重用的代码。这一部分代码将在 BSP 之前编译。

PLATFORM\COMMON\SRC\COMMON：这里存放了被 BSP 中重用的通用代码，这一部分代码是跟硬件平台无关的。譬如一般的 IOCTL 处理函数、与内核交互的公共的中断程序等

等，另外还包括一下库文件，如 OAL_IOCTL.lib、oal_intr.lib、oal_log.lib、kitl_log.lib 等等。这一部分代码由微软提供，一般不能修改。

PLATFORM\COMMON\SRC\<CPU>: CPU 表示 MCU 的不同体系结构，如 ARM、MIPS、SH 和 X86。这些目录分别存放了各体系结构的 MCU 的相关代码，如 CACHE 相关代码、物理地址和虚拟地址转换的代码等等。这部分代码只针对 MCU 的内核，不涉及具体的芯片。这一部分代码也由微软提供，不建议修改。

PLATFORM\COMMON\SRC\SOC: 该目录下存放了不同的 MCU 对应的代码，跟 BSP 对应，这里可看做是 CSP(CHIPSET SUPPORT PACKAGE)。这一部分的代码一般来说不能直接修改，如果需要移植类似平台的 BSP，应该复制一个，重命名后再做修改。这里的目录和其中链接后的库文件的命名也遵循一定的规则(芯片名称_厂商名称_版本号)。这里需要注意的是 SOC 目录下 dirs 文件需要包括体系结构的说明，如 PLATFORM\COMMON\SRC\SOC\PXA27X_MS_V1 中的 dirs 文件以 DIRS_ARM= 打头。这与 BSP 中的一般的 DIRS 文件不同。我在移植 BSP 的时候，并没有在这里做任何修改，只是将其中相关的文件拷贝到我自己的 BSP 目录下，这样方便 BSP 的发布。

PLATFORM\BSPName: 这里存放了跟开发板对应的相关代码。在编译 WinCE 操作系统时，它在 PLATFORM\COMMON 的目录之后编译。我们在针对一款新的硬件平台移植 WinCE6.0 时就是在这里做相应的添加和修改。

WinCE6.0 中 BSP 的目录组织如下：

目录	说明
CATALOG	存放 BSP 的 CATALOG file
CESYSGEN	编译的过程中，bib 和 reg 文件将被拷贝到这
FILES	存放 BSP 的 bib、reg、dat 等配置文件和无须编译的二进制文件
SRC	存放 BSP 的所有源代码
SRC\BOOTLOADER	存放 BOOTLOADER 的源代码
SRC\COMMON	存放共享代码，链接后的库可被 BOOTLOADER、OAL 和驱动使用
SRC\DRIVERS	存放设备驱动的代码
SRC\INC	存放相关头文件
SRC\KITL	存放 KITL 的相关代码
SRC\OAL\OALEXE	链接 OAL.lib 和其他一些库，生成 OAL.exe
SRC\OAL\OALLIB	存放 OAL 的源代码，编译生成 OAL.lib

本文分析了 WINCE600 的目录组织，并从 PQOAL 的角度分析了 PLATFORM 目录