

Windows_x86_64上为龙芯OpenHarmony平台开发应用全流程指南

文档导览

本文档旨在为开发者提供一个完整的操作指南，详细介绍如何在 x86_64 架构的 Windows PC 上使用 DevEco Studio 开发 OpenHarmony 应用，并最终将其成功部署到龙芯（LoongArch）设备上。

一、准备工作

1. 开发环境要求

- **操作系统:** Windows 10 或 Windows 11
- **开发工具:** DevEco Studio NEXT 5.1
- **SDK 版本:** SDK API Level 12+

2. 预备知识

在开始之前，建议开发者对 OpenHarmony 应用开发有基本的了解。您可以参考以下官方文档，快速掌握基础知识：

- **华为开发者联盟文档:** 《从零开始开发 HarmonyOS 应用（ETS 语言）》
- **OpenHarmony 社区文档:** 《快速入门（ArkTS）》
- **NDK 开发指南:** 《NDK 开发概述》

二、软件安装与配置

1. 下载 DevEco Studio for LoongArch

- **下载链接:** [devecostudio-windows-5.1.0.828-la.zip](#)
- **安装步骤:** 下载完成后，将压缩包解压到任意本地路径。`bin/devecostudio64.exe` 即为启动程序。

2. 下载 OpenHarmony SDK for LoongArch

- **下载链接:** [ohos-sdk.tar.gz](#)
- **安装步骤:**
 1. 在本地任意位置创建一个名为 `Sdk` 的文件夹，并在其中再创建一个名为 `12` 的子文件夹 (此处的 `12` 对应 SDK API Level 12)。
 2. 将下载的 `ohos-sdk.tar.gz` 压缩包中的五个包 (`ets`, `js`, `native`, `previewer`, `toolchains`) 解压到 `Sdk/12` 目录下。
 3. 解压后的文件目录结构应如下所示：

```
Sdk
├── 12
│   ├── ets
│   ├── js
│   └── native
```

```
├─ previewer
├─ toolchains
```

三、应用开发与构建

1. 创建与配置新工程

启动 DevEco Studio，创建一个新的应用工程，然后进行以下关键配置。

2. 配置工程 `build-profile.json5`

修改工程根目录下的 `build-profile.json5` 文件，确保应用能够在 OpenHarmony 系统上运行。

- **修改内容:** 将 `products` 标签下的 `runtimeOS` 字段修改为 `"OpenHarmony"`，并将 `compatibleSdkVersion` 与 `compileSdkVersion` 调整为 `12`。

```
"products": [  
  {  
    ...  
    "compatibleSdkVersion": 12,  
    "compileSdkVersion": 12,  
    "runtimeOS": "OpenHarmony",  
  }  
]
```

3. 配置 SDK Manager (首次运行配置)

重要提示: 为防止 IDE 自动下载官方 x86 SDK 覆盖龙芯 SDK，请在**断网状态**下执行此操作。

1. 打开 DevEco Studio，导航至 `File -> Settings -> OpenHarmony SDK`。
2. 在 `SDK Location` 栏点击 `Edit`。
3. 选择您之前创建并解压好的 SDK 路径 (例如 `path\to\your\Sdk`)。
4. 点击 `Finish` 完成配置。

4. 配置模块 `module.json5`

修改模块级别的 `module.json5` 文件，以适配目标设备类型。

- **修改内容:** 在 `deviceTypes` 字段中添加 `"default"` 和 `"tablet"`。

```
"deviceTypes": [  
  "default",  
  "tablet"  
],
```

5. (可选) 导入 PCID 系统能力文件

如果您的应用需要调用特定的系统权限，则需要导入 PCID (Permission, Component, and Interface Declaration) 文件。

- **导入方法:** 请参考官方文档的 [PCID 导入章节](#)。
- **注意:** 导入后会生成一个 `syscap.json` 文件。如果在构建过程中遇到与此文件相关的错误（通常是关于 "SystemCapability.Base"），可以直接在 `syscap.json` 中删除报错的对应行。

四、应用签名

应用安装前必须进行签名。签名需要三个核心文件：**密钥库 (.p12)**、**数字证书 (.cer)** 和 **Profile 文件 (.p7b)**。

方式一：自动签名 (推荐)

DevEco Studio 的自动签名功能目前可以正常工作，是首选方式。具体操作请参考华为官方指南：[《使用入门级签名证书进行签名》](#)。

方式二：手动签名

此处详细介绍如何使用 OpenHarmony SDK 中提供的证书进行手动签名。

1. 准备签名文件：

- **密钥库 (.p12):** 使用 `Sdk/12/toolchains/lib/OpenHarmony.p12`。
- **数字证书 (.cer):** 使用您从 3A5000 设备获取的 `OpenHarmonyApplication.cer` 文件。
- **签名信息:**
 - **Key Password / Store Password:** 123456
 - **Key Alias:** `openharmony application release`

2. 生成 Profile 文件 (.p7b):

- 所需执行的命令及说明已在 `3A5000/sign_prepare.txt` 文件中提供。
- **执行环境:** 建议在类 Unix 的 shell 环境下执行这些命令（如 Git Bash, WSL 等）。
- **Java 环境:** 可以使用 DevEco Studio 内置的 JRE (`jbr/bin/java.exe`)。
- **重要:** 命令中的 `com.example.ndk` 需替换为您应用的包名 (`bundleName`)，该名称定义在 `AppScope/app.json5` 文件中。

3. 在 IDE 中配置签名：

1. 获得全部三个文件后，在 DevEco Studio 中打开 `File -> Project Structure -> Signing Configs`。
2. **取消勾选** `Automatically generate signature` 选项。
3. 在下方的表单中，依次导入您准备好的 `.p12`, `.cer`, 和 `.p7b` 文件，并填写密码和别名。
4. 配置完成后，IDE 会在 `build-profile.json5` 文件中自动添加 `signingConfigs` 字段。

4. 构建带签名的 HAP：

- 重新执行构建 (Build)，即可在输出目录中找到已签名的 HAP 文件。

五、已知问题与说明

- **runtimeOS 兼容性问题:** 当 `build-profile.json5` 中的 `runtimeOS` 设置为 `HarmonyOS` 时, 生成的 HAP 包目前无法在 OpenHarmony 龙芯设备上安装。请务必使用 `OpenHarmony`。
- **调试限制:** DevEco Studio 目前暂不支持对 LoongArch 架构的 AOT (Ahead-Of-Time) 编译模式进行调试。
- **安装错误码参考:** 如果在安装应用时遇到失败, 可以参考此文档查询错误码详情: [《BM 工具使用指导 - 安装错误码》](#)。