# 29. 端口热插拔框架

端口热插拔框架为DPDK应用程序提供了运行时添加、移除端口的能力。 由于框架一来PMD实现，所以热插拔的端口必须是PMD支持的端口才行。 此外，从DPDK程序中移除端口之后，框架并不提供从系统中删除设备的方法。 对于由物理网卡支持的端口，内核需要支持PCI热插拔功能。

## 29.1. 概述

端口热插拔框架的基本要求：

使用端口热插拔框架的DPDK应用程序需要管理其自己的端口。

端口热插拔矿机被实现为允许DPDK应用程序管理自己的端口。 例如，当应用程序调用添加端口的功能时，将返回添加的端口号。 DPDK应用程序也可以通过端口号移除该端口。内核需要支持待添加、移除的物理设备端口。

为了添加新的物理设备端口，设备首先被内核中的用户框架IO驱动识别。 然后DPDK应用程序可以调用端口热插拔功能来连接端口。 移除过程步骤刚好相反。 移除之前，必须先停止并关闭端口。

DPDK应用程序在移除端口之前，必须调用 “rte\_eth\_dev\_stop()” 和 “rte\_eth\_dev\_close()” 函数。 这些函数将启动PMD的反初始化过程。 本框架不会影响传统的DPDK应用程序的行为。

如果端口热插拔的功能没有被调用，所有传统的DPDK应用程序仍然可以不加修改地工作。

## 29.2. 端口热插拔API概述

添加一个端口

“rte\_eth\_dev\_attach()” API 将端口添加到DPDK应用程序，并返回添加的端口号。 在调用API之前，设备应该被用户空间驱动IO框架识别。 API接收一个类似 “0000:01:00.0” 的pci地址或者是 “net\_pcap0,iface=eth0” 这样的虚拟设备名称。 在虚拟设备名称情况下，格式与DPDK的一般‘-vdev’选项相同。

移除一个端口

“rte\_eth\_dev\_detach()” API 从DPDK应用程序中移除一个端口，并返回移除的设备的pci地址或虚拟设备名称。

## 29.3. 引用

“testpmd” 支持端口热插拔框架。

## 29.4. 限制

* 端口热插拔API并不是线程安全的。
* 本框架只能在Linux下使能，BSD并不支持。
* 为了移除端口，端口必须是igb\_uio或VFIO管理的设备端口。
* 并非所有的PMD都支持移除功能。要知道PMD是否支持移除，请搜索 rte\_eth\_dev::data::dev\_flags 中的 “RTE\_ETH\_DEV\_DETACHABLE” 标志。 如果在PMD中定义该标志，则表示支持。