**嵌入式Linux内核热补丁功能**

# 背景

运营商对电信设备有着极高的连续稳定运行需求；而我们的软件系统构建自上亿行代码，庞大、复杂。不断会有有BUG被发现、修复之后需要发布补丁应用于现网运行的设备中，以解决潜在的隐患。要求打补丁不能影响现网运行的设备（避免需要重启等导致长时间业务中断）。

# 现有解决方案

用户态应用程序已经有了较好的热补丁机制来完成BUG修复，设备/单板无需重启。我们可以通过向目标进程插入补丁动态库，修改目标进程中函数入口指令、使之运行时可以跳转到新的补丁函数，从而实现热补丁功能。

# 问题&改进方案

当前的热补丁方案，并不适用于Linux内核及内核驱动；当前如果发现内核及内核驱动的BUG，只能通过修改代码，以冷补丁方式应用于现网设备（需要重启生效）。

1. 如果由你来设计用户态进程热补丁功能，你将如何设计？列举出在做热补丁时的关键流程并加以说明。这个过程中存在什么风险、是否有规避方案？
2. Linux内核模块（.ko）在被加载时，如何关联到内核接口？请结合内核/模块符号表管理机制加以说明。
3. 请你来设计Linux内核热补丁方案（包括补丁制作、安装激活、卸载回退三个流程）。
4. 在你的热补丁方案中，针对那些场景有效，哪些场景无效？在哪些场景下会产生风险？请加以说明；如有应对方案，也请加以说明。

要求：请您以Word输出整体运作方案，并将其中要点以PPT形式进行输出，在极致挑战环节进行宣讲。