
ZYNQ RTOS

1.总述

支持 FAST 开源架构的 OpenBox-S4 平台的核心器件采用的是 Xilinx 公司的 ZYNQ 系列芯片的产品，该平台目前是 FAST 开源社区的主要适配平台，支持所有的 FAST 开源项目，如 SDN、TSN 等。时间敏感网络（TSN：Time Sensitive Networking）是以以太网为基础的新一代网络标准，具有时间同步、延时保证等确保实时性的功能。

FAST 开源社区目前已经完善了在真实物理网络传输环境下的 TSN 功能适配，湖南新实也推出了一体化的 TSN 综合演示方案。随着我们对 TSN 技术的不断深耕，支持 TSN 功能、适配 TSN 业务软件的综合系统应用环境也将逐步推出。

ZYNQ RTOS 顾名思义就是在 ZYNQ 平台上运行 RTOS（Real Time Operating System，简称 RTOS），更准确的说就是在 OpenBox-S4 上运行实时操作系统，更好的支持 TSN 技术。

2.RTOS 背景

How fast is fast enough? 实时是一种相对概念。根据应用场景与用户需求不同，实时的响应时间要求不同，其实现方法也各有差异。支持 ZYNQ 的实时解决方案分为如下几种：

1. SMP 的亲 and

SMP（对应多处理器）的亲 and 是指将业务处理逻辑亲近在某些或某个特定的 CPU 核上进行处理，减少 CACHE 失效带来不确定的查表延时。CPU 的亲 and 包括软件逻辑的亲 and 与中断的亲 and。中断的亲 and 其实也是为了让软件逻辑亲 and 在指定的核上工作。通过亲 and 的方式可以对逻辑的性能带来一定的提升，处理延时也会得到一些改善。但亲 and 改善的主要关键在于 CACHE 的失效，故只对一些数据 IO 密集型业务有改善，延时的稳定性也与数据分布相关。

2. PREEMPT_RT

PREEMPT_RT 是 Linux 内核的一个实时补丁。主要目的是为了提高 Linux 的延时响应时间。补丁代码需要替换内核中成千上万行的代码，需要适配不同的支持设备，而且为了保证实时，也牺牲了系统整体的吞吐量。由于这些不尽理想的代价权衡，至今为止，该补丁仍未融入到 Linux 内核的主线版本中来。

3. Enea LWRT

Enea LWRT 是一种商业解决方案，主要是在用户空间提供无中断（上下文切换）的代码执行环境。整个系统分为两部分，一部分为实时分区，另一部分为非实时分

区。其中实时部分既可以使用部分系统屏蔽的 CPU 核来运行，也可以在内核中嵌入实时模块来支持。实时的用户空间提供了一些轻量级的线程 API 和管理模块。

4. 裸机 AMP

AMP（非对称多处理器）的最直接方案就是 Xilinx 提供的裸机 APP 应用。Linux 系统运行在一个专门的 CPU 核上，通常是 CPU0，然后通过 FSBL 将裸机 APP 或可执行代码加载到 CPU1 上运行。由于操作系统运行在 CPU0 上，在 CPU1 上仅运行业务逻辑代码，不存在上下文切换，调度等任何其他带来的延时开销，故该方法可以较好的保证业务逻辑处理的实时性。

5. FreeRTOS

FreeRTOS 主要由 Amazon 开发与维护，属于 AMP 架构。Linux 系统运行在 CPU0 上，在 CPU1 上运行支持整个实时应用环境的操作系统。实时任务与非实时任务均只在自己的核上运行，但两者之间可以相互访问。实时访问非实时要根据 MMU 的地址定义来决定，而非实时访问实时可以像访问其他设备存储一样读写。

6. Xenomai

Xenomai 是一种采用双内核机制的 Linux 内核的强实时扩展。他与 Linux 共享硬件资源，优先级高于 Linux 内核，负责处理系统的实时任务。其运行方式更像一个勾子系统，系统支持的内部或外部中断触发后，先由 Xenomai 执行，属于实时操作的任务在 Xenomai 环境下全部执行完成，由于该环境的最高优先级，处理过程不会被 Linux 中断。只有非实时处理任务会被分派到 Linux 系统执行。这样就可以完全保证 Xenomai 环境任务的实时要求不受 Linux 的影响，提供系统的实时可确定性。

3, 我们的选择

FAST 开源架构主要面向网络通信领域，TSN 技术更是为了支持确定性的网络通信传输。通过上述的几种对实时处理的解决方案对比，裸机 AMP、FreeRTOS 和 Xenomai 各有优势与不足。结合目前我们在 FAST 架构上的应用与开发，我们倾向选择 Xenomai。一方面由于该方案更适合网络应用环境，包含了从驱动、协议栈到用户应用的完整的网络应用开发示例。另一方面该方案仍然支持并使用 SMP 架构的通用 Linux 系统，对用户原有系统架构和 FAST 原有支持系统与架构均不会带来任何影响。

4, 测试与验证

1. 使用 RTOS

从 FAST 网站下载支持 RTOS 的系统环境，并在 OpenBox-S4 上更新。下载地址与更新操作说明如下：

<http://abc>

2. 测试运行

更新成 RTOS 系统环境后，正常登录 OpenBox-S4 设备，并输出命令：ABC，输出如下打印说明 RTOS 系统运行成功。

关于 RTOS 上的实时应用开发，敬请关注 FAST 开源社区的推出的技术文章。FAST 开源社区微信公从号如下：

海阔天空