

Xenomai 的 LoongArch 平台移植与性能优化

本科毕业论文答辩

郝淼

指导老师：张福新

中国科学院大学计算机科学与技术学院

2023 年 5 月 25 日



① 选题背景及意义

② 研究现状

③ 主要研究内容

④ 总结与展望

① 选题背景及意义

② 研究现状

③ 主要研究内容

④ 总结与展望

选题背景及意义

- 随着**嵌入式市场**的不断繁荣，**实时操作系统**作为嵌入式平台上运行的基础软件将在未来发挥更大的作用
- 目前 LoonArch 平台支持的实时操作系统**数量较少**
- 基于现有成熟、易用的实时操作系统进行 **LoongArch 平台的移植与优化**是很有必要的

① 选题背景及意义

② 研究现状

③ 主要研究内容

④ 总结与展望

研究现状-实时操作系统的分类

实时操作系统要求外部事件发生时，对应的实时程序能够在截止时间前完成处理。

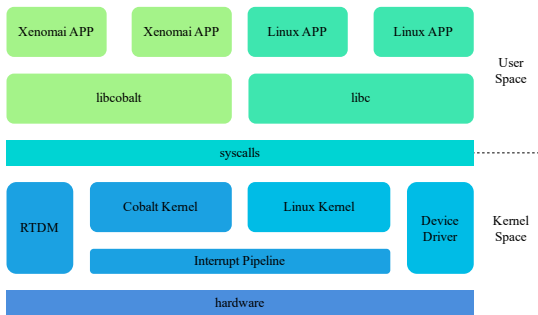
- **硬实时：**不允许处理时间超过截止时间
- **软实时：**允许部分处理时间超过截止时间

研究现状-实时操作系统的实现方式



微内核： 采用一个单独的实时内核，如 RT-Thread

研究现状-实时操作系统的实现方式



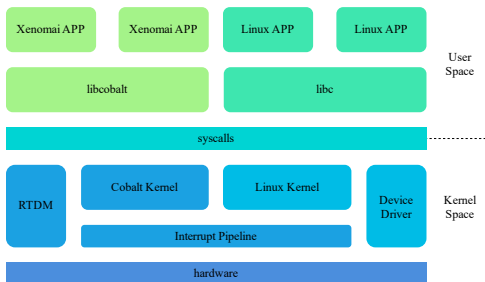
扩展内核：对通用内核进行修改，提升实时性，如 Xenomai

研究现状-实时操作系统的实现方式

目前，LoongArch 平台支持的实时操作系统包括：

- **微内核**：RT-Thread，SylixOS
- **扩展内核**：RTLinux

研究现状-Xenomai 项目



- 扩展内核架构
- 基于 ADEOS 的“双内核”系统
- l-pipe, cobalt, libcobalt

- 实时性: Xenomai 能够提供硬实时保障
- 兼容性: libcobalt 实现了与 VxWorks、POSIX 等兼容的 API

① 选题背景及意义

② 研究现状

③ 主要研究内容

实时任务模型

Xenomai 的 LoongArch 平台移植

LoongArch Xenomai 系统的性能优化

④ 总结与展望

① 选题背景及意义

② 研究现状

③ 主要内容

实时任务模型

Xenomai 的 LoongArch 平台移植

LoongArch Xenomai 系统的性能优化

④ 总结与展望

实时任务模型-简介

Hideyuki Tokuda 等在 1990 年为了研究可预测的实时调度器，建立了实时线程模型 (RT-Thread Model)，但该模型缺乏对**外部事件**和**实时操作系统能力**的刻画，导致无法用于描述 Xenomai 中实时任务运行的模型，因此本课题基于实时线程模型建立了实时任务模型。

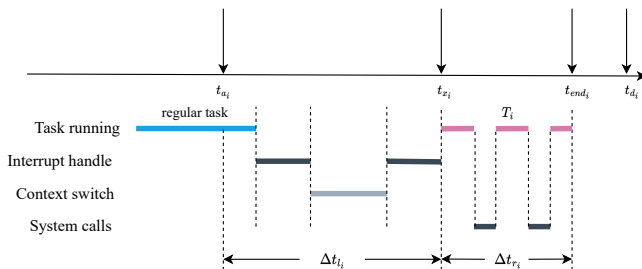
实时任务模型-简介

实时任务模型可以表述为：

$$\begin{cases} T_i = (\Delta t_{l_i}, \Delta t_{r_i}) \\ e_k = (C_{e_k}^1, C_{e_k}^2, C_{e_k}^3) \end{cases} \quad (1)$$

实时任务模型-实时任务 T_i

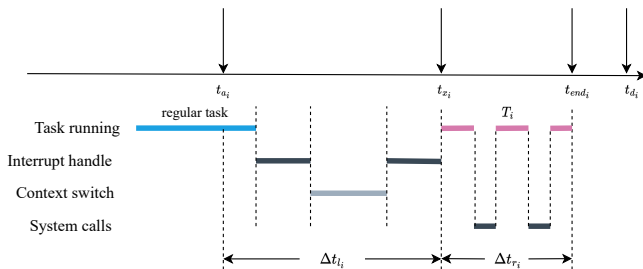
$$T_i = (\Delta t_{l_i}, \Delta t_{r_i})$$



- $\Delta t_{l_i} = t_{x_i} - t_{a_i}$: 延迟
- $\Delta t_{r_i} = t_{end_i} - t_{x_i}$: 运行时间

实时任务模型-外部事件 e_k

$\{T_i\}_{e_k}$ 为响应 e_k 的实时任务集合, $e_k = (C_{e_k}^1, C_{e_k}^2, C_{e_k}^3)$



- $C_{e_k}^1 = t_{d_i} - t_{a_i}$: 刻画了外部事件的截止时间
- $t_{a_{i+1}} - t_{a_i} \in [C_{e_k}^2, C_{e_k}^3]$: 刻画了外部事件的周期性

实时任务模型-实时约束条件

在实时操作系统中，希望 $\{T_i\}_{e_k}$ 能满足约束条件

$$t_{end_i} \leq t_{d_i} \Leftrightarrow (t_{end_i} - t_{x_i}) + (t_{x_i} - t_{a_i}) \leq t_{d_i} - t_{a_i} \quad (2)$$

$$\Leftrightarrow \Delta t_{r_i} + \Delta t_{l_i} \leq C_{e_k}^1 \quad (3)$$

$$\Leftrightarrow \Delta t_{l_i} + \Delta t_{r_i} \leq C_{e_k}^1 \quad (4)$$

当条件 (4) 不成立时，称系统中发生了超时故障

实时任务模型-Xenomai 实时原理

对于 Xenomai 而言，应尽可能缩短 $\Delta t_{l_i} + \Delta t_{r_i}$ ：

- 使用专用的实时内核进行调度 \Rightarrow 缩短 Δt_{l_i} ；
- 实时内核提供实时系统调用 \Rightarrow 缩短 Δt_{r_i} ；
- 提供区别于普通 API 的实时 API \Rightarrow 帮助程序员从宏观上缩短 $\Delta t_{l_i} + \Delta t_{r_i}$ ；

① 选题背景及意义

② 研究现状

③ 主要研究内容

实时任务模型

Xenomai 的 LoongArch 平台移植

LoongArch Xenomai 系统的性能优化

④ 总结与展望

Xenomai 的 LoongArch 平台移植

- l-pipe: 3A5000 的中断结构, LoongArch Linux 的中断处理过程, l-pipe 的实现原理
- cobalt: LoongArch 中的时间硬件资源, cobalt 内核的工作原理, cobalt 中的体系结构相关代码
- libcobalt: LoongArch ABI

① 选题背景及意义

② 研究现状

③ 主要内容

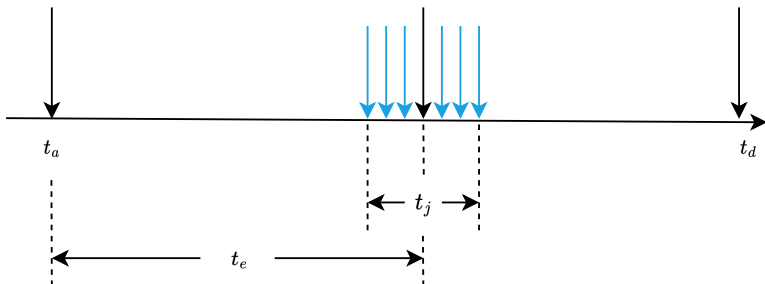
实时任务模型

Xenomai 的 LoongArch 平台移植

LoongArch Xenomai 系统的性能优化

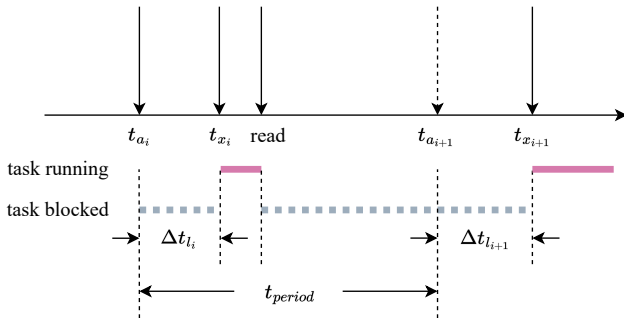
④ 总结与展望

LoongArch Xenomai 系统的性能优化-性能评价指标



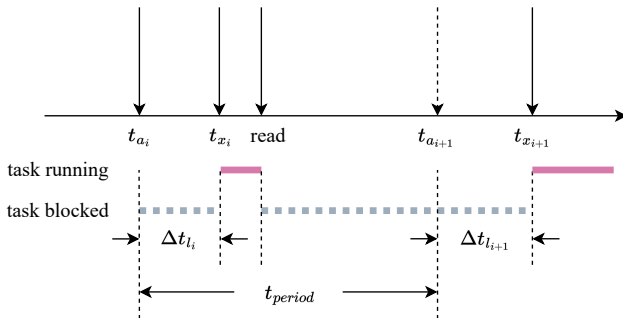
- 延迟期望 $t_e \rightarrow$ 实时性
- 抖动 $t_j \rightarrow$ 稳定性
- 硬实时约束 $\rightarrow \max_{T_i \in \{T_i\}_{e_k}} (\Delta t_{l_i} + \Delta t_{r_i}) \leq C_{e_k}^1$

LoongArch Xenomai 系统的性能优化-latency 测试



- $t_j = t_{max} - t_{min}$
- $t_e = t_{avg}$
- $e_l = (t_{period}, t_{period}, t_{period})$

LoongArch Xenomai 系统的性能优化-latency 测试



在 latency 测试中， $\Delta t_{r_i} \approx 0$ ，硬实时约束简化为 $t_{max} \leq t_{period}$

LoongArch Xenomai 系统的性能优化-测试环境

- latency 配置: $e_l = (100\mu s, 100\mu s, 100\mu s)$, 硬实时约束为 $t_{max} \leq 100\mu s$
- 物理平台:
 - 处理器: 4 核 2.5GHz 龙芯 3A5000
 - 桥片: 龙芯 7A2000
 - 物理内存: 16GB
- 访存负载: `stress -m 4`

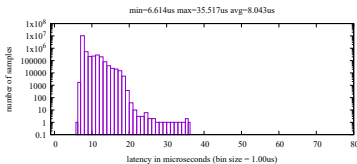
LoongArch Xenomai 系统的性能优化-优化方法

- 关闭页迁移：透明巨页 THP
- 关闭高延迟外设：声卡，实时时钟
- 关键路径优化：`xnclock_core_ns_to_ticks` 与 `xnclock_core_ticks_to_ns`

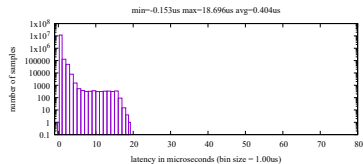
LoongArch Xenomai 系统的性能优化-对照设置

- ① 硬实时内核 RTLinux，内核版本 4.19
- ② 未优化 Xenomai，内核版本 4.19
- ③ 在 2. 的基础上关闭页迁移
- ④ 在 3. 的基础上关闭高延迟外设，并进行关键路径优化

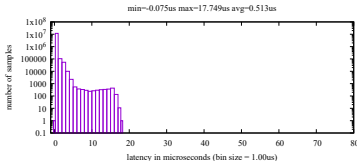
LoongArch Xenomai 系统的性能优化-无负载测试结果



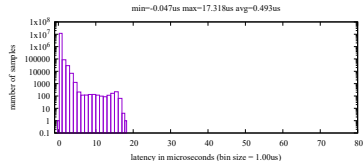
(a) RTLinux



(b) Xenomai

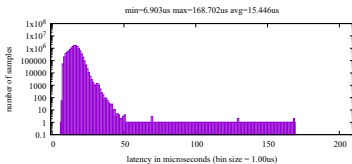


(c) Xenomai 关闭页迁移

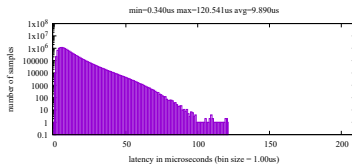


(d) Xenomai 关闭页迁移与高延迟外设并优化关键路径

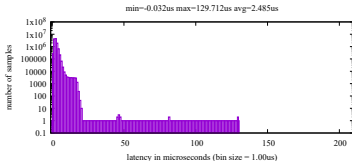
LoongArch Xenomai 系统的性能优化-访存负载测试结果



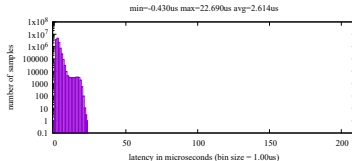
(e) RTLinux



(f) Xenomai



(g) Xenomai 关闭页迁移



(h) Xenomai 关闭页迁移与高延迟外设并优化关键路径

LoongArch Xenomai 系统的性能优化-结论

通过关闭页迁移，关闭高延迟外设和关键路径优化，可以使龙芯 3A5000+7A2000 平台上的 Xenomai 系统**达到硬实时要求**

① 选题背景及意义

② 研究现状

③ 主要研究内容

④ 总结与展望

总结

- 提出一种由外部事件驱动的**实时任务模型**
- 将 Xenomai 项目**移植到 LoongArch 平台**
- 根据 LoongArch 平台的具体特点，结合实时任务模型进行**分析优化**

展望

- **充分验证。**未来需要对系统进行大量测试以验证移植实现的正确性。
- **嵌入式平台。**未来将 Xenomai 系统运行在 LoongArch 嵌入式平台仍需要进一步的移植适配工作。
- **充分优化。**未来需要充分结合 LoongArch 体系结构与龙芯 CPU 的特点进行优化。

请各位专家老师批评指正！