Dovetail移植：

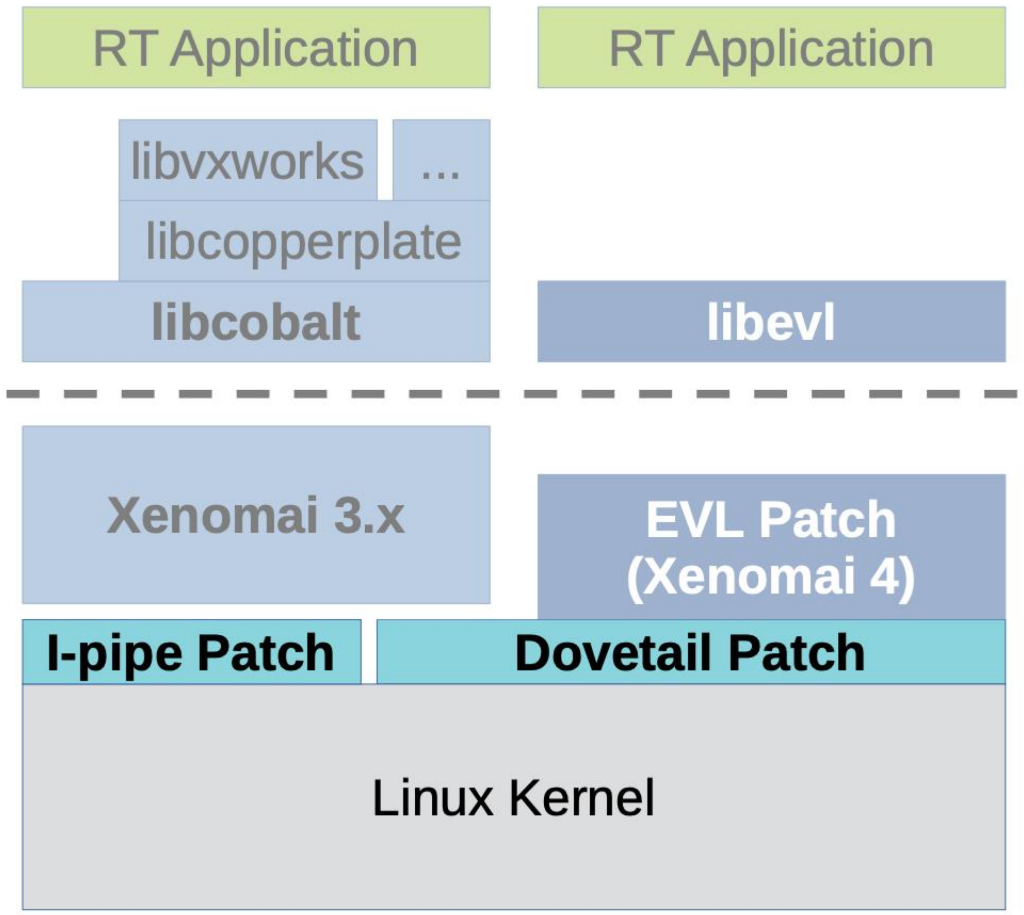
RISC-V与LoongArch实践

- 山宇轩

目录

双内核架构与Dovetail

- 双内核操作系统，一般由实时内核，和通用内核组成。实时内核用于处理需要超低延迟与有非常严格响应时间限制的任务。
- 中断虚拟化组件Dovetail通过接管所有的中断，引入带内(inband),带外(out of band)与一个两级中断流水线的概念来保证带外高实时需求的中断总是能抢占带内功能内核的执行来保证实时性。
- 除中断与异常外，Dovetail还引入了一套alternate scheduling机制，提供了实时核心控制Linux tasks调度器的能力。

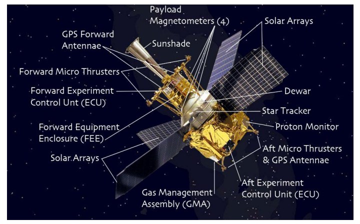


1 Dovetail移植：RISC-V与LoongArch实践

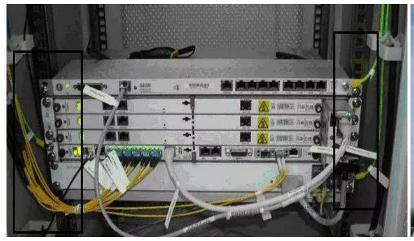
背景介绍

Scenarios

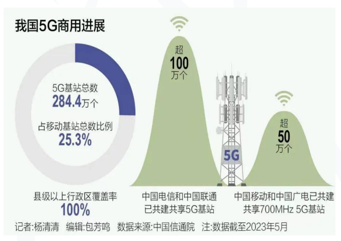
Satellite Internet

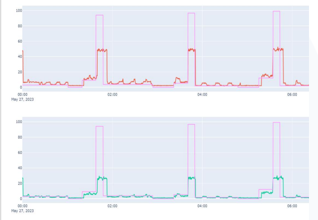

Payload Magnetometers (4)
Solar Arrays
GPS Forward Antennas
Sunshade
Forward Micro Thrusters
Dewar
Forward Experiment Control Unit (ECU)
Star Tracker
Forward Equipment Enclosure (FEE)
Proton Monitor
Solar Arrays
Aft Micro Thrusters & GPS Antennae
Gas Management Assembly (GMA)
Aft Experiment Control Unit (ECU)

Symptom 1: 5G BBUs are idle after white-boxing



Symptom 2: The base station is rich in resources and the load timing is jitter





Embodied intelligent robots

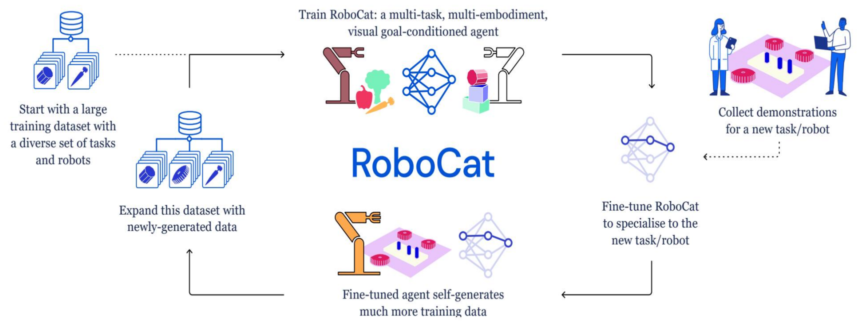
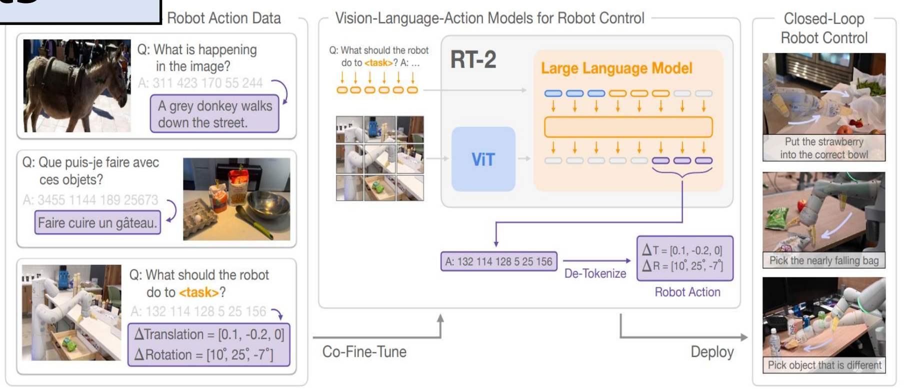

Start with a large training dataset with a diverse set of tasks and robots
Train RoboCat: a multi-task, multi-embodiment, visual goal-conditioned agent
RoboCat
Collect demonstrations for a new task/robot
Expand this dataset with newly-generated data
Fine-tune RoboCat to specialise to the new task/robot
Fine-tuned agent self-generates much more training data

Figure 1: **The self-improvement process.** RoboCat is a multi-task, multi-embodiment visual goal-conditioned


Robot Action Data
Q: What is happening in the image?
A: 311 423 170 55 244
A grey donkey walks down the street.
Q: Que puis-je faire avec ces objets?
A: 3455 1144 189 25673
Faire cuire un gâteau.
Q: What should the robot do to <task>?
A: 132 114 128 5 25 156
ΔTranslation = [0.1, -0.2, 0]
ΔRotation = [10°, 25°, -7°]
Vision-Language-Action Models for Robot Control
Q: What should the robot do to <task>? A: ...
RT-2
Large Language Model
ViT
A: 132 114 128 5 25 156
De-Tokenize
ΔT = [0.1, -0.2, 0]
ΔR = [10°, 25°, -7°]
Robot Action
Co-Fine-Tune
Deploy
Closed-Loop Robot Control
Put the strawberry into the correct bowl
Pick the nearly falling bag
Pick object that is different

RoboCat System: ROS System

Optimus Gen2 System: ROS System

RT-2 system: ROS system

Real-time: robot control

Generality: AI large model tasks

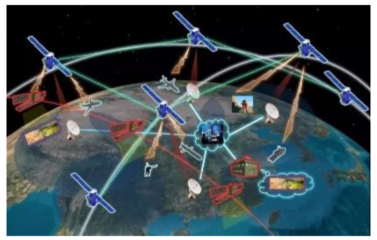
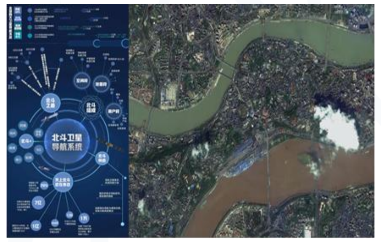
1 Dovetail移植：RISC-V与LoongArch实践

背景介绍

Scenario 1: Complication of satellite mission/payload tasks

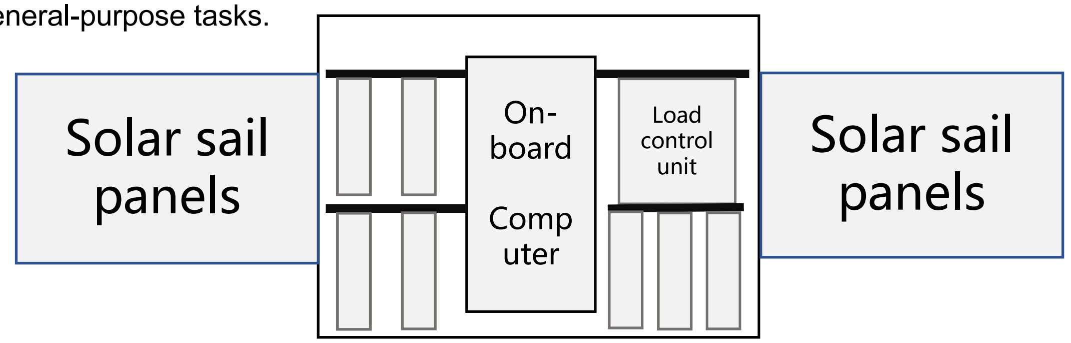
Satellite Internet

Satellite constellations

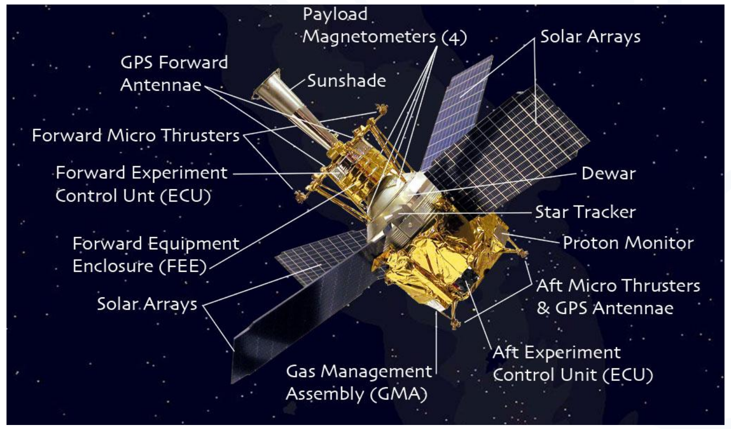
Complication of satellite missions

∅ Satellite hardware heterogeneity, functions are configured, controlled and operated through software;
There are underutilized computing and storage resources between multiple payloads of satellites;
The Star Computer/Payload Control Unit has to handle both traditional real-time tasks and general-purpose tasks.



Generalization of satellite missions

Satellites face various generalized tasks such as communication, navigation and positioning, earth observation, scientific experiments, emergency response, and resource exploration.

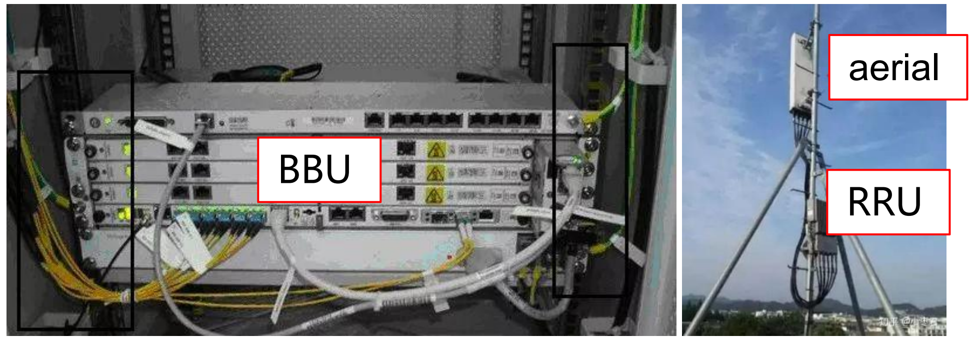

Payload Magnetometers (4)
Solar Arrays
GPS Forward Antennae
Sunshade
Forward Micro Thrusters
Forward Experiment Control Unt (ECU)
Dewar
Star Tracker
Proton Monitor
Forward Equipment Enclosure (FEE)
Aft Micro Thrusters & GPS Antennae
Solar Arrays
Aft Experiment Control Unit (ECU)
Gas Management Assembly (GMA)

1 Dovetail移植：RISC-V与LoongArch实践

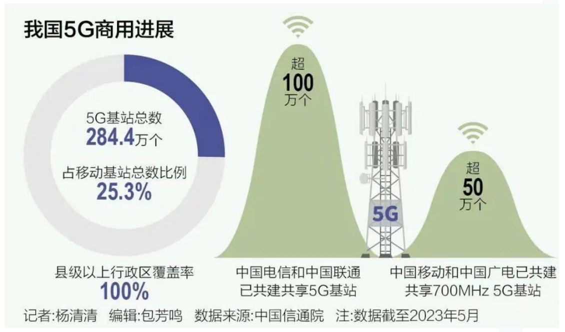
背景介绍

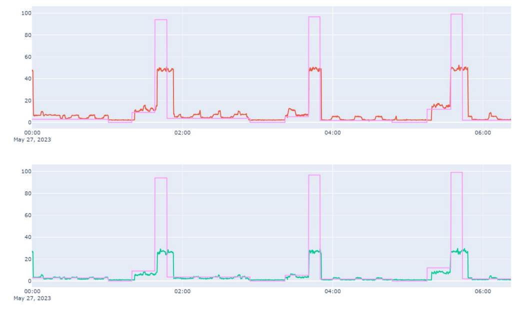
Scenario 2: 5G base stations share computing resources

Symptom 1: 5G BBUs are idle after white-boxing



Symptom 2: The base station is rich in resources and the load timing is jitter





BBU（Base band Unite)

- Function: Add packet protocol header;
  Attributes: real-time tasks, requiring hard real-time;
  Hardware: white-box BBU (50% idle resources);

User-specific task loads

- Function: User Defined;
  Attributes: General tasks, no real-time requirements;
  Hardware: COTS architecture can be satisfied;

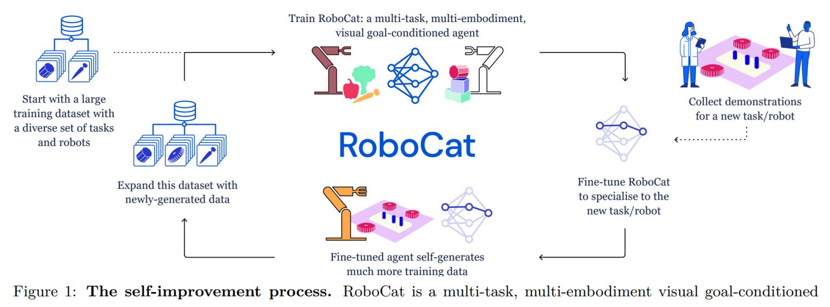
The BBU white-box unit processes both real-time and general-purpose tasks to improve resource utilization[1]

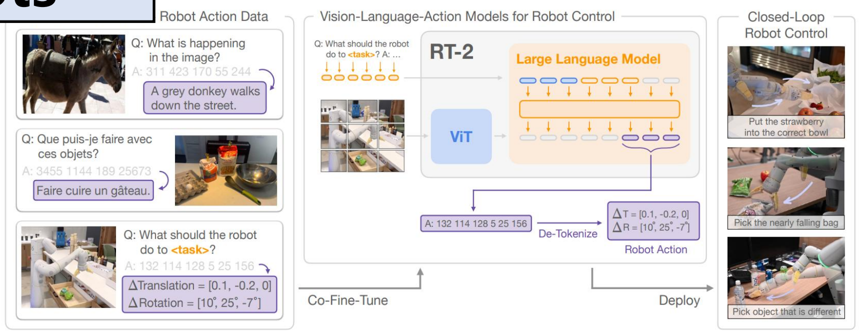
[1] Foukas, Xenofon, and Bozidar Radunovic. "Concordia: Teaching the 5G vRAN to share compute." Proceedings of the 2021 ACM SIGCOMM 2021 Conference. 2021.

1 Dovetail移植：RISC-V与LoongArch实践
背景介绍

Scenario 3: Embodied intelligent robots

Embodied intelligent robots


Train RoboCat: a multi-task, multi-embodiment, visual goal-conditioned agent
Start with a large training dataset with a diverse set of tasks and robots
Expand this dataset with newly-generated data
RoboCat
Collect demonstrations for a new task/robot
Fine-tune RoboCat to specialise to the new task/robot
Fine-tuned agent self-generates much more training data
Figure 1: The self-improvement process. RoboCat is a multi-task, multi-embodiment visual goal-conditioned


Robot Action Data
Q: What is happening in the image?
A grey donkey walks down the street.
Q: Que puis-je faire avec ces objets?
Faire cuire un gâteau.
Q: What should the robot do to <task>?
ΔTranslation = [0.1, -0.2, 0]
ΔRotation = [10°, 25°, -7°]
Vision-Language-Action Models for Robot Control
RT-2
Large Language Model
ViT
A: 132 114 128 5 25 156
De-Tokenize
Robot Action
Co-Fine-Tune
Deploy
Closed-Loop Robot Control
Put the strawberry into the correct bowl
Pick the nearly falling bag
Pick object that is different

RoboCat System: ROS System

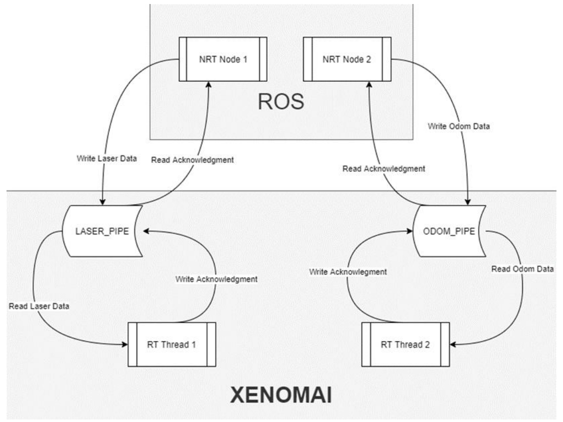
Optimus Gen2 System: ROS System

RT-2 system: ROS system

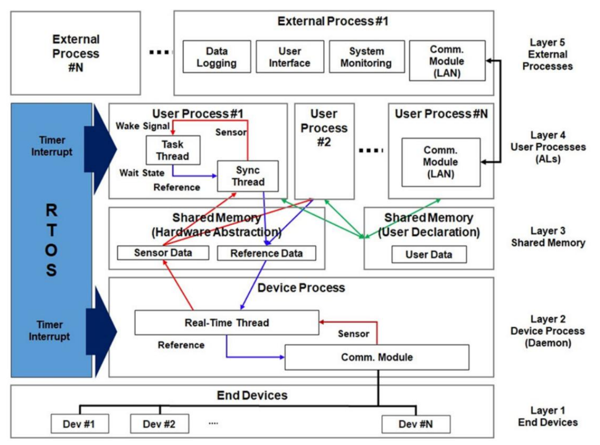
Real-time: robot control

Generality: AI large model tasks

There are already dual-kernel architecture robots.



The ROS system enhances real-time performance with Xenomai



Real-time enhanced robot DRC-HUBO+ architecture (published in TRO)

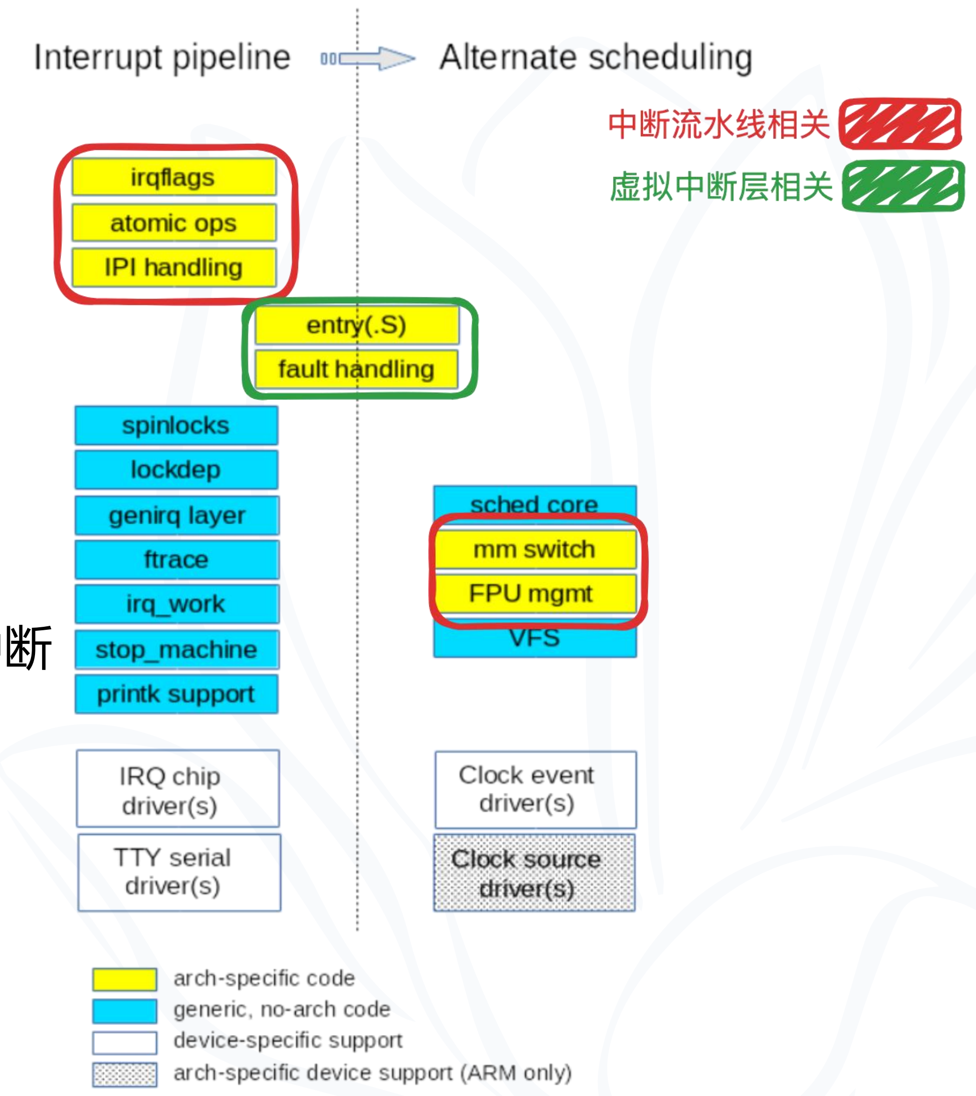
2 Dovetail移植：RISC-V与LoongArch实践

移植方法及步骤

移植Dovetail的主要组成部分

- 虚拟中断层
  1. 硬件中断状态位设置
  2. 带外IPI中断的实现
  3. 带内关于原子操作的部分还是要屏蔽中断，需要特殊处理
  4. 对于切换进程地址空间的switch_mm()等 =>需要关闭硬件中断

- 中断流水线
  1. 中断入口与异常处理需要接入中断流水线的控制逻辑

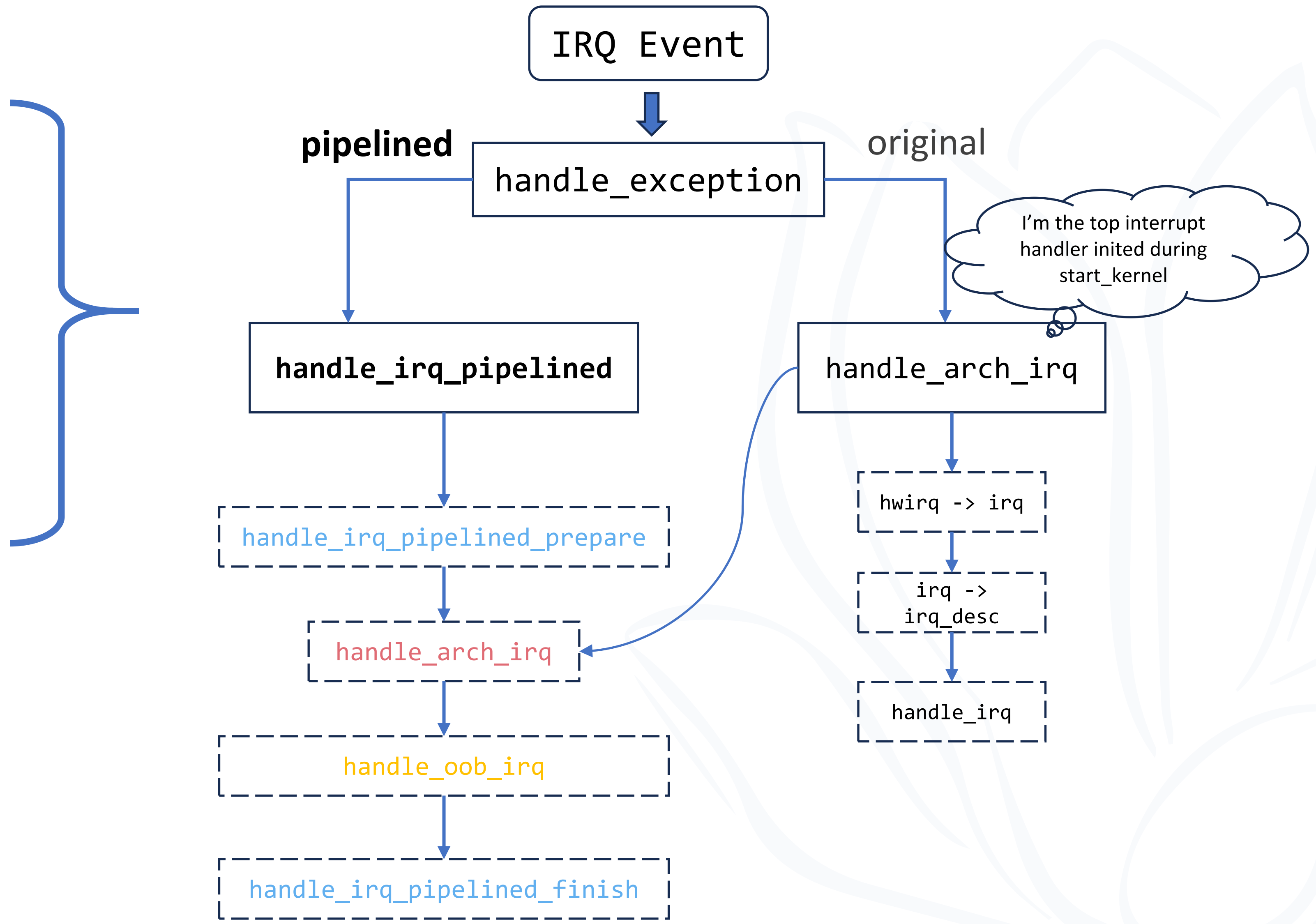


2 **Dovetail移植：RISC-V与LoongArch实践**

移植方法及步骤

中断的类型：

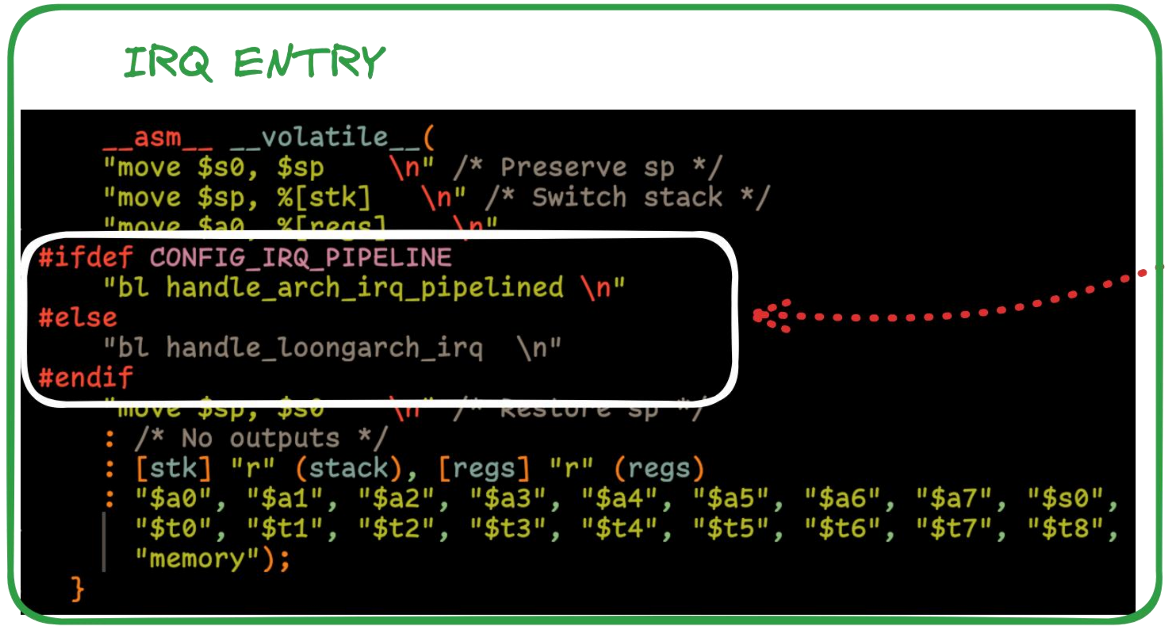
- External Interrupt
- Software Interrupt, e.g. IPI
- Timer Interrupt
- Exception



2 Dovetail移植：RISC-V与LoongArch实践

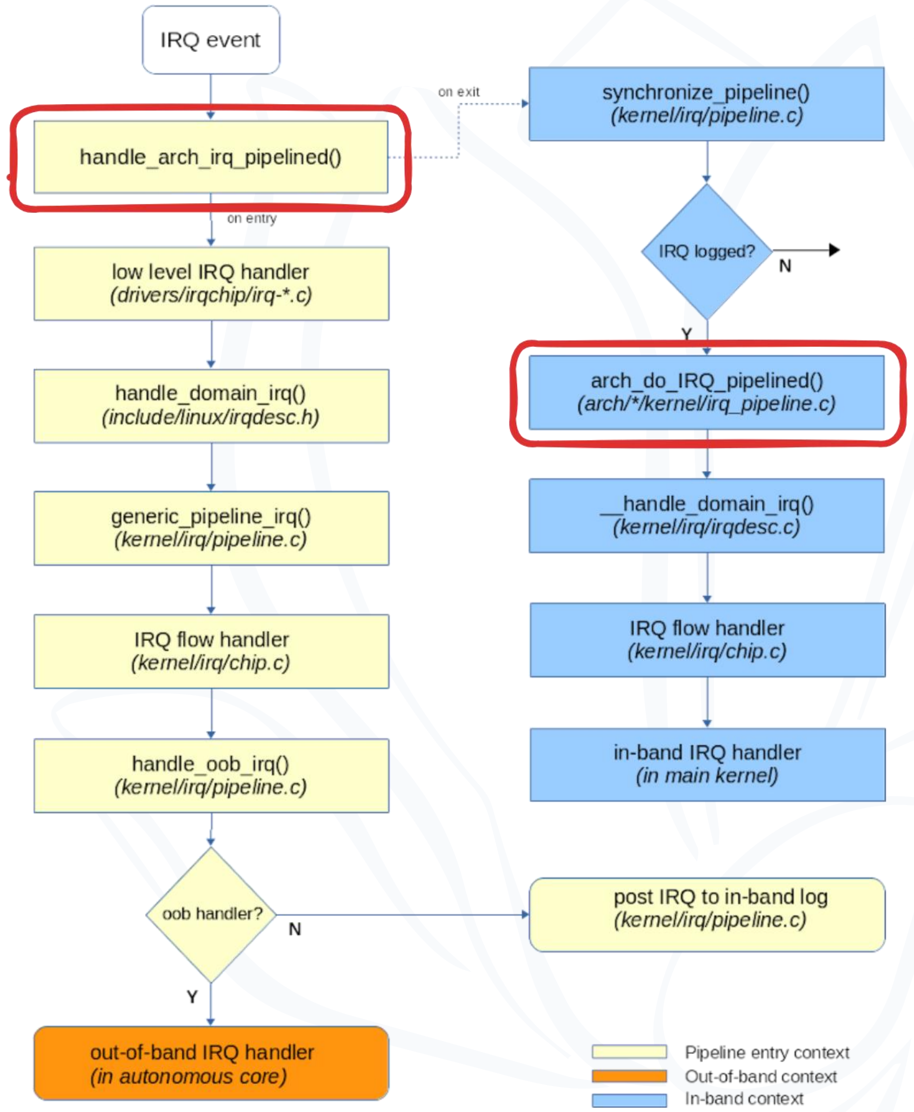
移植方法及步骤

中断流水线



IRQ ENTRY

```
    __asm__ __volatile__(
    "move $s0, $sp     \n" /* Preserve sp */
    "move $sp, %[stk]    \n" /* Switch stack */
    "move $a0, %[regs]     \n"
#ifdef CONFIG_IRQ_PIPELINE
    "bl handle_arch_irq_pipelined \n"
#else
    "bl handle_loongarch_irq  \n"
#endif
    "move $sp, $s0     \n" /* Restore sp */
    : /* No outputs */
    : [stk] "r" (stack), [regs] "r" (regs)
    : "$a0", "$a1", "$a2", "$a3", "$a4", "$a5", "$a6", "$a7", "$s0",
      "$t0", "$t1", "$t2", "$t3", "$t4", "$t5", "$t6", "$t7", "$t8",
      "memory");
  }
```



修改CPU架构对应的中断入口，接入Dovetail中断流水线的逻辑，相比于原来直接调用中断处理函数，带内中断会被Dovetail记录下来，直到带外的中断都处理完了才会处理带内的中断事件。

中断从带外流向带内，通过一个两阶段的流水线保证中断执行的序列化，与虚拟中断层共同保证带外中断的响应速度

2 Dovetail移植：RISC-V与LoongArch实践

移植方法及步骤

中断的类型：

- External Interrupt
- **Software Interrupt, e.g. IPI**
- Timer Interrupt
- Exception

ARM64	RISC-V

Original
`enum ipi_msg_type {`
`    IPI_RESCHEDULE,`
`    IPI_CALL_FUNC,`
`    IPI_CPU_STOP,`
`    IPI_CPU_CRASH_STOP,`
`    IPI_TIMER,`
`    IPI_IRQ_WORK,`
`    IPI_WAKEUP,`
`    NR_IPI`
`};`
- One SGI per ipi type
- Need 8 SGIs
- Have SGI0-15
	Original
	`enum ipi_message_type {`
	`    IPI_RESCHEDULE,`
	`    IPI_CALL_FUNC,`
	`    IPI_CPU_STOP,`
	`    IPI_CPU_CRASH_STOP,`
	`    IPI_IRQ_WORK,`
	`    IPI_TIMER,`
	`    IPI_MAX`
	`};`
	• msip register per cpu for triggering ipi
	• Use `ipi-mux` to identify ipi type

With OOB
• Send ipi message
  • Use PER_CPU to multiplexed over SGI0
  • Use SGI1 – 3 for OOB IPIs
• Handle ipi message
  • Bitwise Decoding for inband
  • One SGI per ipi type
	With OOB
	• Send ipi message
	  • Do not need multiplexed
	• Handle ipi message
	  • Do not need bitwise decoding
	  • Just need add:
	`#define TIMER_OOB_IPI       (ipi_virq_base + OOB_IPI_OFFSET)`
	`#define RESCHEDULE_OOB_IPI  (TIMER_OOB_IPI + 1)`
	`#define CALL_FUNCTION_OOB_IPI   (RESCHEDULE_OOB_IPI + 1)`

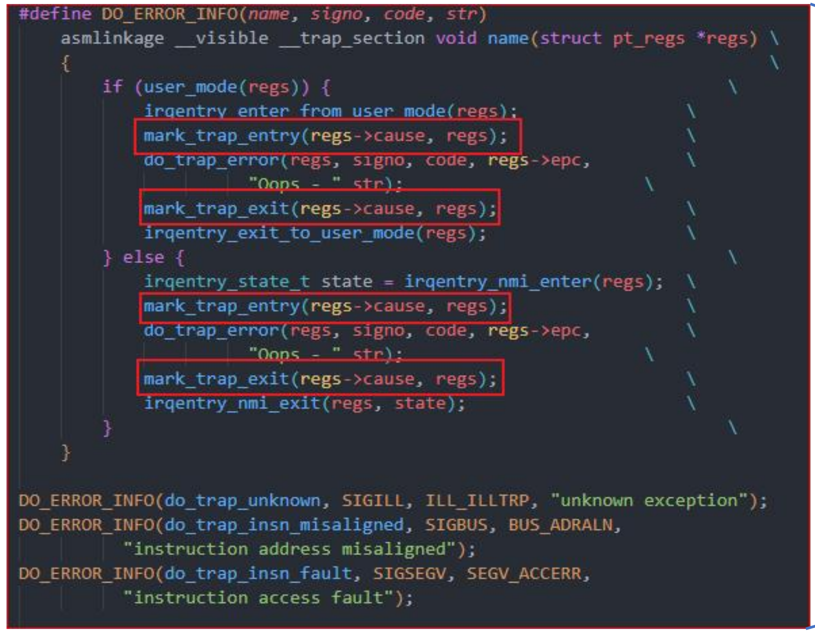
2 Dovetail移植：RISC-V与LoongArch实践

移植方法及步骤

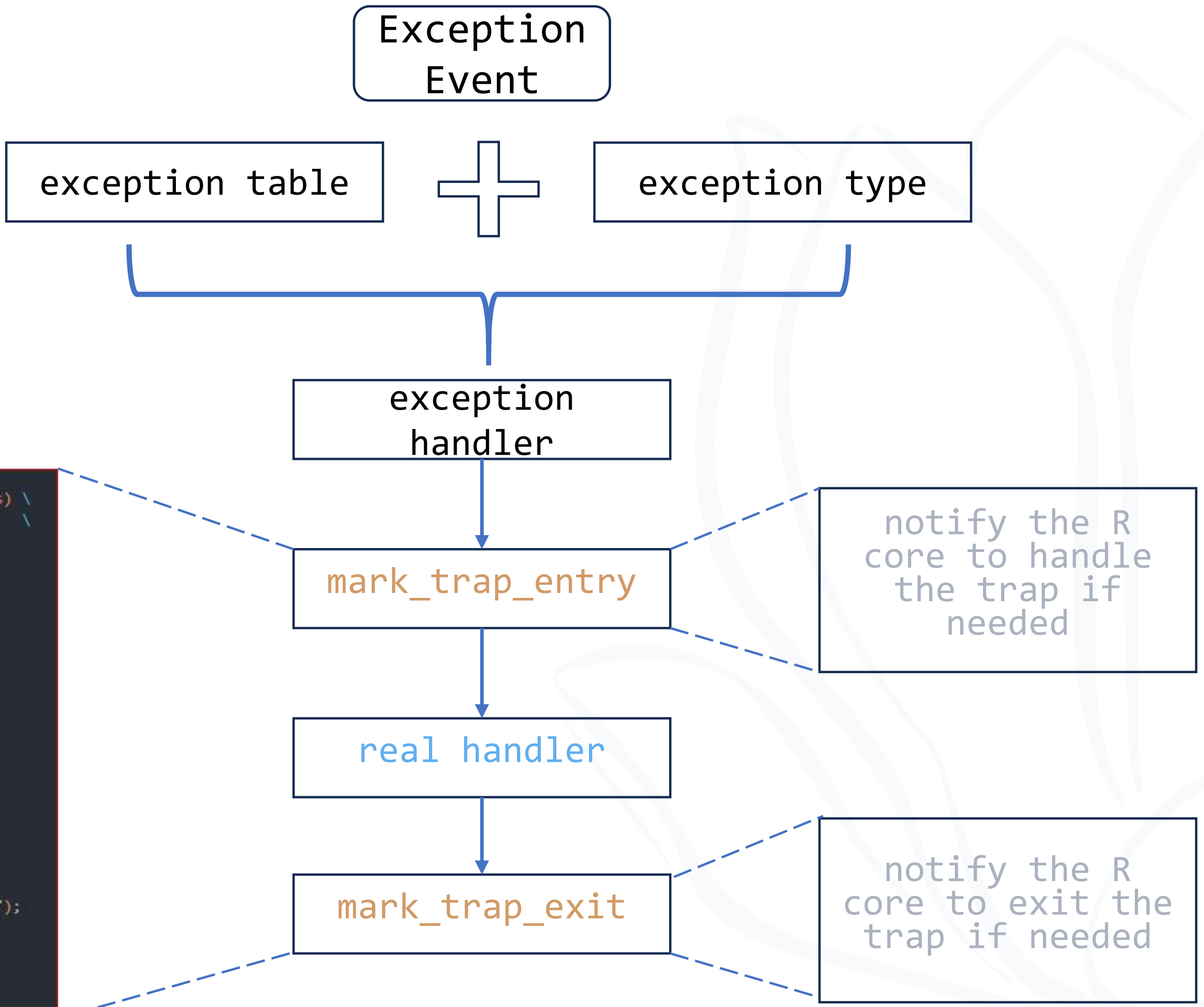
中断的类型：

- External Interrupt
- Software Interrupt, e.g. IPI
- Timer Interrupt

- Exception



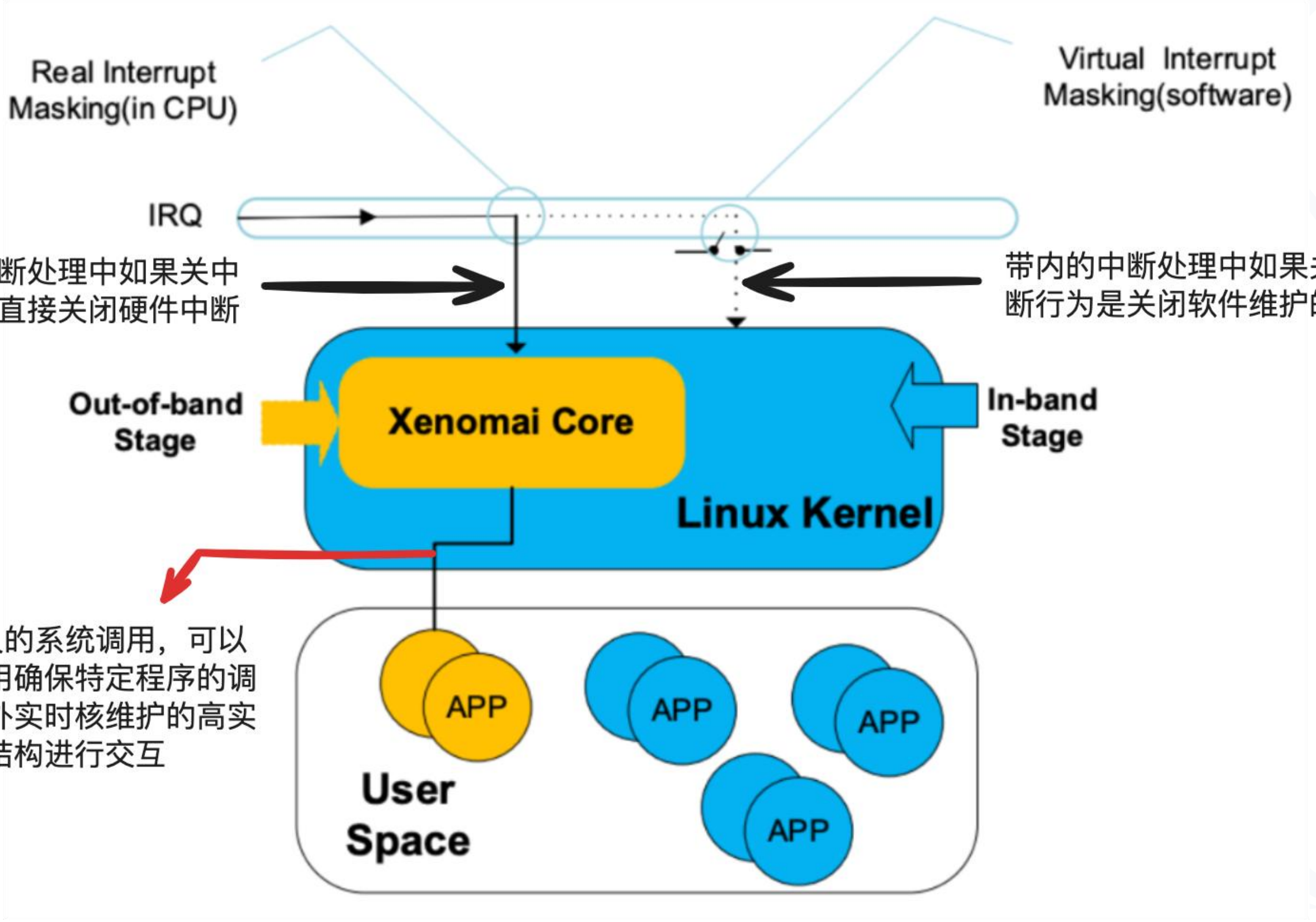
```
#define DO_ERROR_INFO(name, signo, code, str)
    asmlinkage __visible __trap_section void name(struct pt_regs *regs) \
    {                                                                    \
        if (user_mode(regs)) {                                          \
            irqentry_enter_from_user_mode(regs);                        \
            mark_trap_entry(regs->cause, regs);                         \
            do_trap_error(regs, signo, code, regs->epc,                 \
                    "Oops - " str);                                     \
            mark_trap_exit(regs->cause, regs);                          \
            irqentry_exit_to_user_mode(regs);                           \
        } else {                                                        \
            irqentry_state_t state = irqentry_nmi_enter(regs);          \
            mark_trap_entry(regs->cause, regs);                         \
            do_trap_error(regs, signo, code, regs->epc,                 \
                    "Oops - " str);                                     \
            mark_trap_exit(regs->cause, regs);                          \
            irqentry_nmi_exit(regs, state);                             \
        }                                                               \
    }

DO_ERROR_INFO(do_trap_unknown, SIGILL, ILL_ILLTRP, "unknown exception");
DO_ERROR_INFO(do_trap_insn_misaligned, SIGBUS, BUS_ADRALN,
        "instruction address misaligned");
DO_ERROR_INFO(do_trap_insn_fault, SIGSEGV, SEGV_ACCERR,
        "instruction access fault");
```



2 **Dovetail移植：RISC-V与LoongArch实践**

移植方法及步骤

虚拟中断层



带外的中断处理中如果关中断行为是直接关闭硬件中断

带内的中断处理中如果关中断行为是关闭软件维护的中断

一组Xenomai定义的系统调用，可以通过这组系统调用确保特定程序的调度优先级，与带外实时核维护的高实时性的内核数据结构进行交互

虚拟中断层

- We have pipelined interrupt & exception handler now

- But for the irqflags operation in common tasks
    - Disable the interrupt
    - Uncontrollable critical section
    - Real-time task can not response in a bounded time

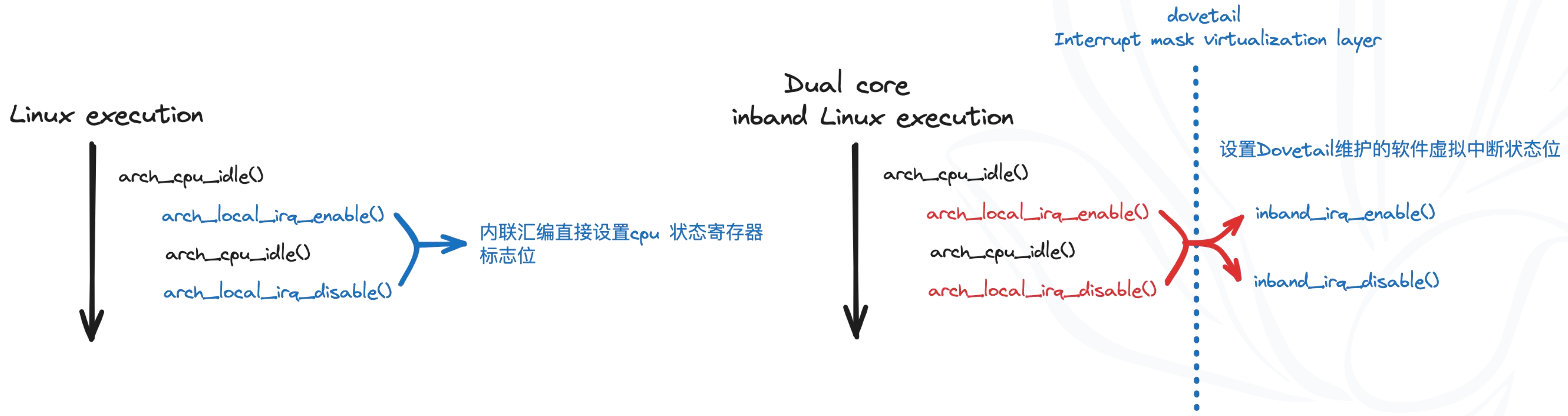
generic code		arch specific code
local_irq_enable	->	arch_local_irq_ebable
local_irq_disable	->	arch_local_irq_disable
local_irq_save	->	arch_local_irq_save
local_irq_restore	->	arch_local_irq_restore

2 Dovetail移植：RISC-V与LoongArch实践

移植方法及步骤

虚拟中断层

- 修改CPU状态位架构相关的函数被替换为设置虚拟中断层的实现
- 真正修改CPU状态位的函数由另一组函数继承
- 避免了在inband状态的Linux关闭硬件中断导致不可预测的延迟



2 **Dovetail移植：RISC-V与LoongArch实践**

移植方法及步骤

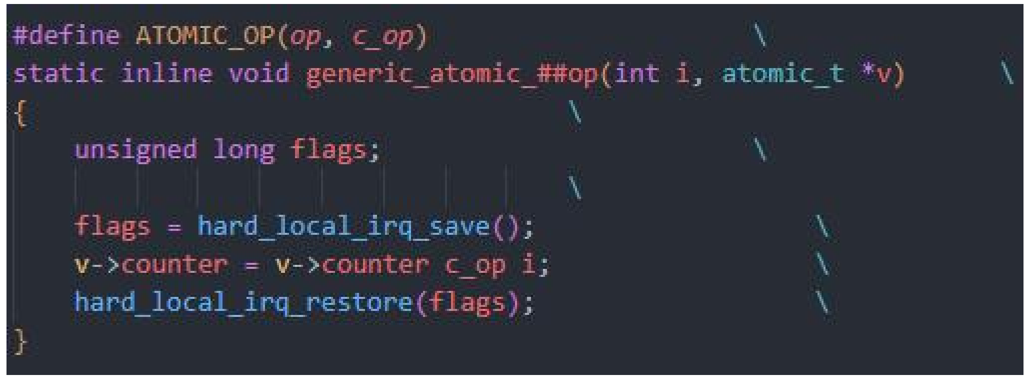
Atomic operations

Why we need to port Atomic operations?

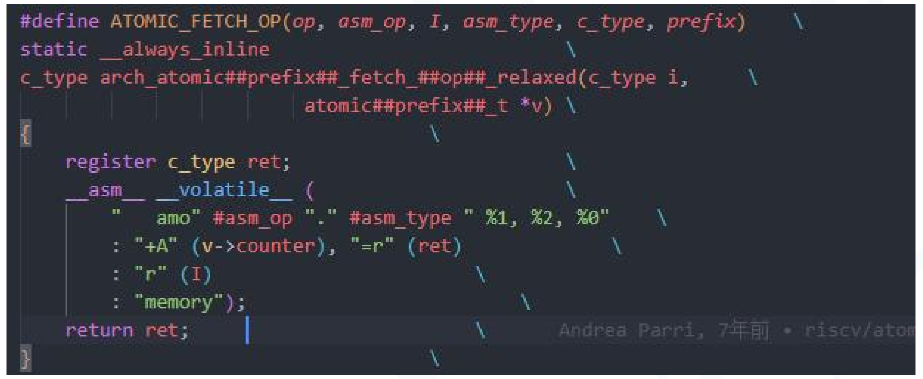
Some architectures(e.g. arm under SMP) use interrupt to implement the atomic operations

- Used `arch_local_irq_ebable` in the macro

- Replace arch_local_irq_xxx functions with hard_local_irq_xxx to make atomic operations correctly



```
#define ATOMIC_OP(op, c_op)                         \
static inline void generic_atomic_##op(int i, atomic_t *v)      \
{                                               \
    unsigned long flags;                        \
                                                \
    flags = hard_local_irq_save();                  \
    v->counter = v->counter c_op i;                 \
    hard_local_irq_restore(flags);                  \
}
```

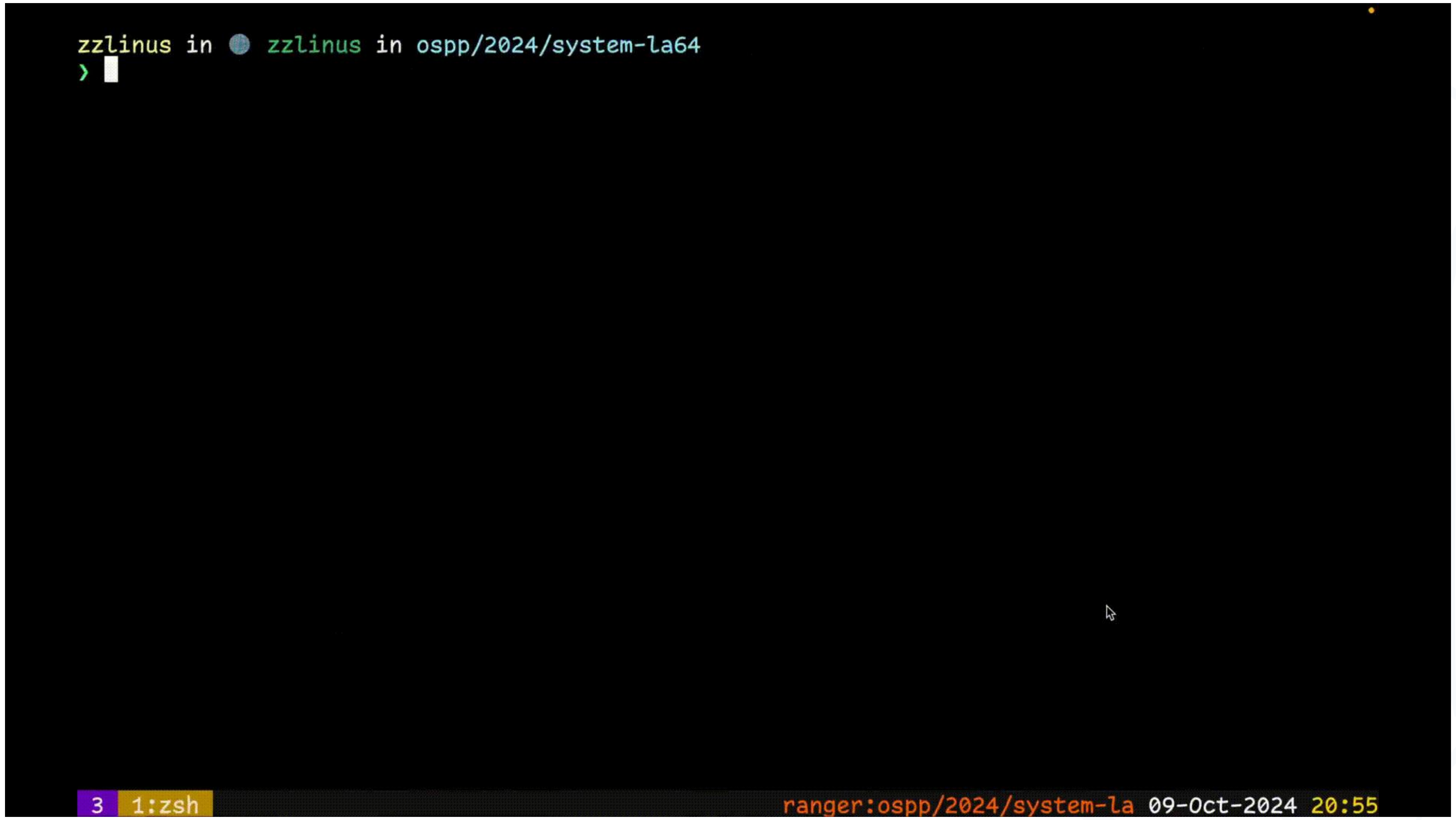


```
#define ATOMIC_FETCH_OP(op, asm_op, I, asm_type, c_type, prefix)    \
static __always_inline                                  \
c_type arch_atomic##prefix##_fetch_##op##_relaxed(c_type i,     \
                        atomic##prefix##_t *v) \
{                                               \
    register c_type ret;                                \
    __asm__ __volatile__ (                              \
        "    amo" #asm_op "." #asm_type " %1, %2, %0"    \
        : "+A" (v->counter), "=r" (ret)             \
        : "r" (I)                               \
        : "memory");                                \
    return ret;                                 \        Andrea Parri, 7年前 • riscv/atom
}                                               \
```

3

Dovetail移植：RISC-V与LoongArch实践

测试结果及展示



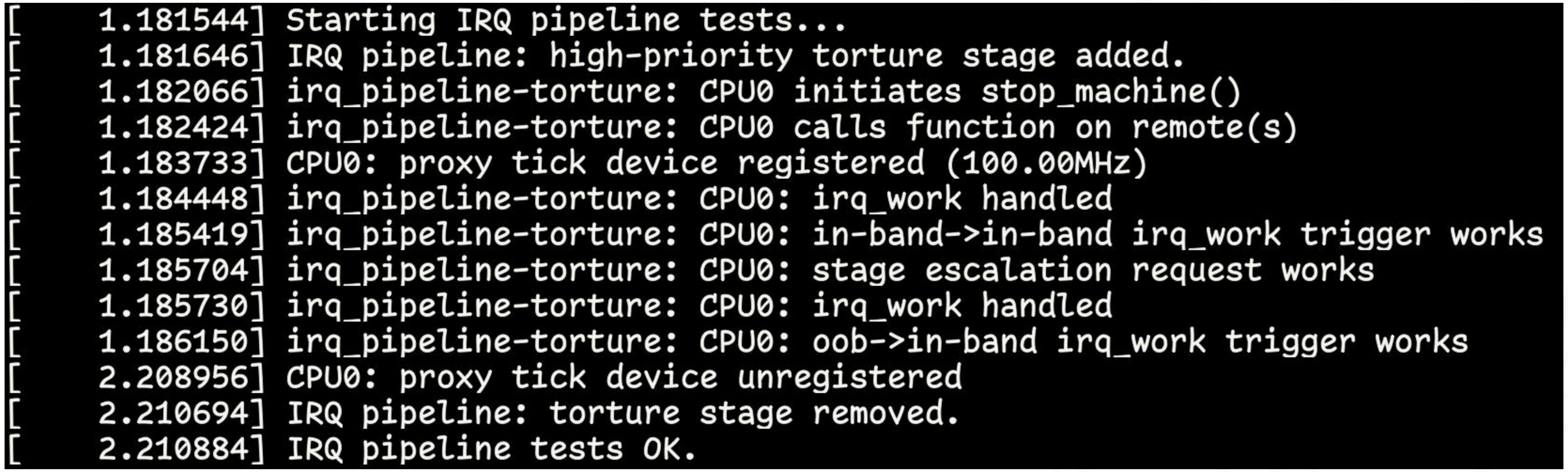
```
zzlinus in 🌐 zzlinus in ospp/2024/system-la64
❯ █

 3  1:zsh                                    ranger:ospp/2024/system-la 09-Oct-2024 20:55
```

可以在单核模式通过所有测试正常启动

3 Dovetail移植：RISC-V与LoongArch实践

测试结果及展示

IRQ pipeline tests



```
[    1.181544] Starting IRQ pipeline tests...
[    1.181646] IRQ pipeline: high-priority torture stage added.
[    1.182066] irq_pipeline-torture: CPU0 initiates stop_machine()
[    1.182424] irq_pipeline-torture: CPU0 calls function on remote(s)
[    1.183733] CPU0: proxy tick device registered (100.00MHz)
[    1.184448] irq_pipeline-torture: CPU0: irq_work handled
[    1.185419] irq_pipeline-torture: CPU0: in-band->in-band irq_work trigger works
[    1.185704] irq_pipeline-torture: CPU0: stage escalation request works
[    1.185730] irq_pipeline-torture: CPU0: irq_work handled
[    1.186150] irq_pipeline-torture: CPU0: oob->in-band irq_work trigger works
[    2.208956] CPU0: proxy tick device unregistered
[    2.210694] IRQ pipeline: torture stage removed.
[    2.210884] IRQ pipeline tests OK.
```

4

Dovetail移植：RISC-V与LoongArch实践

未来计划

- A better documentation for porting Dovetail onto new architecture

- Support EVL core on RISC-V & LoongArch

- Support EVL core on real hardware

  - RISC-V: starfive-visionfive2
  - LoongArch: 3A6000

- Build a EtherCAT demo on starfive-visionfive2 & 3A6000 with SV630 + MS1H4

Thanks
Q&A