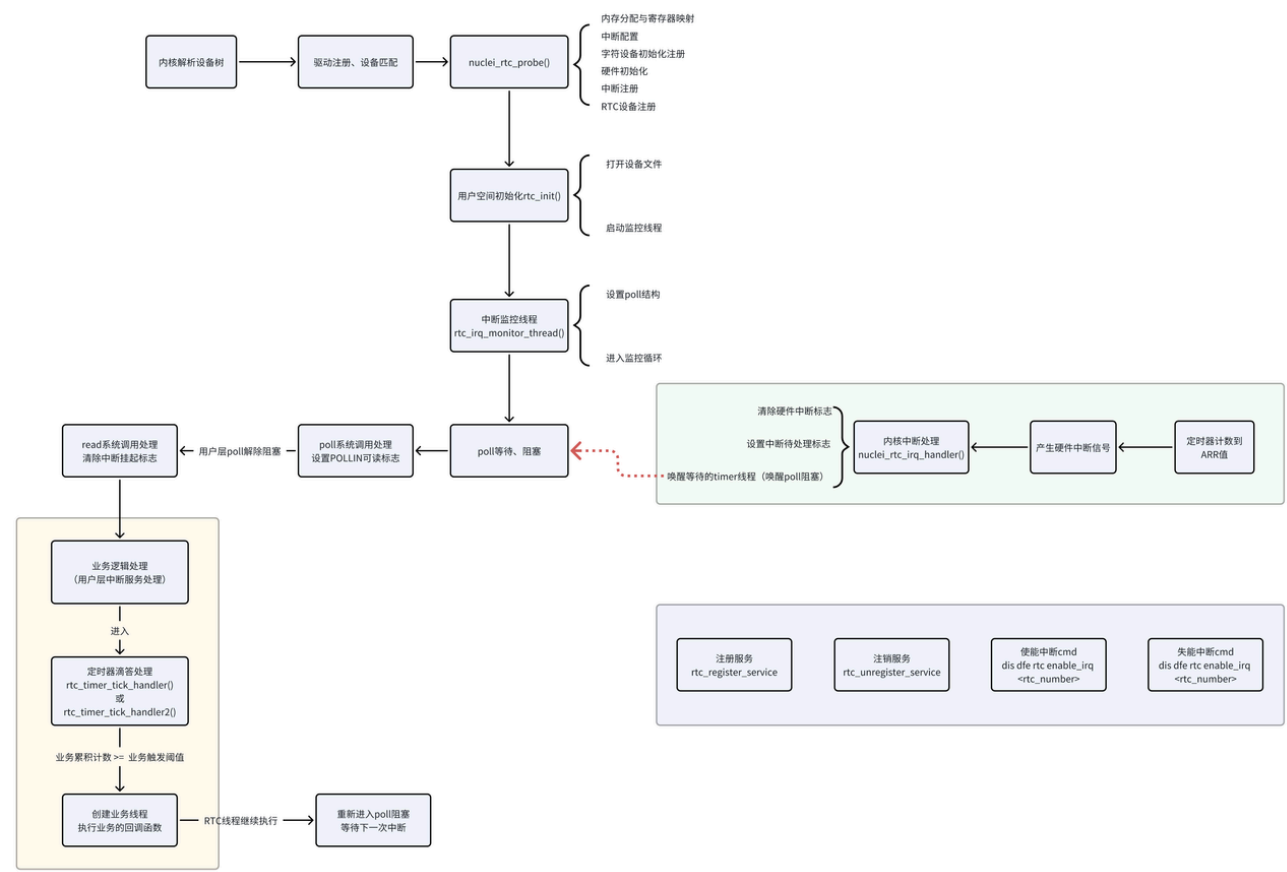


定时器驱动配置说明

一、RTC驱动完整时间线



二、RTC驱动配置

2.1 设备树匹配

文件位置: `nuclei-linux-sdk/conf/digs-chance/nuclei_rv64imafdc_kernel.dts`

▼

复制代码

```
#ifndef TIMER
btimer0:btimer@f01e0000 {
    compatible = "nuclei,rtc";
    reg = <0x0 0xf01e0000 0x0 0x1000>;
    clocks = <&periphclk>;
    counter-value = <1000000>; //单位为us
    interrupt-parent = <&plic0>;
    interrupts = <13>;
}
```

```

    status = "okay";
};

btimer1:btimer@f01e2000 {
    compatible = "nuclei,rtc";
    reg = <0x0 0xf01e2000 0x0 0x1000>;
    clocks = <&periphclk>;
    counter-value = <1000000>; //单位为us
    interrupt-parent = <&plic0>;
    interrupts = <14>;
    status = "okay";
};
#endif

```

配置了两个定时器的设备节点，在/dev下表示为nuclei_rtc0，nuclei_rtc1，分别设置其硬件中断号为13和14，可在/proc/interrupt或/proc/irq下面查看。其中counter-value可配置定时器的中断触发间隔时间，单位为us。目前配置为1000'000us，即1s。推荐中断触发间隔设置为秒级，最多可到100ms，这样保证触发间隔误差控制到us级。

2.2 业务注册

首先在进程里面初始化RTC，使用接口函数**rtc_init()**。

在业务代码调用rtc_register_service()接口可以注册回调函数。完整接口如下：

```

int rtc_register_service(int timer_id, const char *name, int interval, void (*callback_func)(void))

```

其中：

timer_id表示注册到哪个定时器设备，0表示timer0，1表示timer1；

name表示服务线程的名字；

interval表示服务线程的唤醒阈值，定时服务的总时间间隔为： $\text{interval} * \text{counter-value}$ ；

callback_func表示注册的回调函数；

每个定时器最大管理MAX_SERVICES个业务，采用注册线程来执行业务回调函数的方式，保证了业务处理的即时性，同时添加了防重入机制。

2.3 RTC中断使能

提供了cmd命令来启用定时器中断，使用**dis dfe rtc enable_irq <rtc number>** 使能定时器中断；使用**dis dfe rtc disable_irq <rtc number>**失能定时器中断。