seL4: Threads

廖东海

ctrlz.donghai@gmail.com

Background

- Thread Control Block
  - seL4 内核提供线程的抽象，用于管理任务的执行，在内核中被表示为Thread Control Block。
  - 每个任务绑定一个TCB，TCB是CPU调度的基本单位。
- TCB中的主要内容：
  - 寄存器上下文。
  - 线程执行状态。
  - 线程的CSpace、VSpace。
  - 优先级信息。
  - IPC buffer信息。
  - 错误处理。
  - 调度队列中的前后指针。
  - EndPoint等待队列中的前后指针。

Scheduling Model

- 在seL4内核中，调度器是一个基于优先级的轮询调度器，会选取最高优先级的线程在特定的处理器核心上执行。
- Priorities：调度器选取最高优先级的可运行线程进行调度，内部提供的优先级范围是 0～255 。255是被编码为 seL4_MaxPrio 的常量。
- TCB内部还有一个 maximum control priority （MCP），当设置优先级时，必须提供显示的 TCB 对应的 capability 来获取设置权限。如果设置的优先级大于 MCP，则设置操作失败。 root task 的 优先级和 MCP 都被设置为 seL4_MaxPrio
- Round robin
  - 当有多个可运行的相同优先级的 TCB 时，调度器会采用 FIFO 的策略。而调度的时机则是使用时间片。
  - 每个 TCB 都有一个时间片字段，标识 TCB 有资格执行的周期数，而内核的计时器则会进行一个周期性的时钟中断来抢占时间片，当时间片用尽后则会撤销 TCB，调度其他有时间片的 TCB。
  - 当然，线程也可以使用 seL4_Yield 系统调用来主动交出他们当前的时间片。

User Interface

- 寄存器相关
  - seL4_TCB_ReadRegisters/seL4_TCB_WriteRegisters/seL4_TCB_CopyRegisters
- 优先级相关
  - seL4_TCB_SetPriority/seL4_TCB_SetMCPriority/seL4_TCB_SetSchedParams
- CNode配置：
  - seL4_TCB_Configure
  - seL4_TCB_SetIPCBuffer
  - seL4_TCB_SetSpace
- 线程启动暂停：seL4_TCB_Suspend/seL4_TCB_Resume
- Notification相关：
  - seL4_TCB_BindNotification/seL4_TCB_UnbindNotification

Example

- seL4_TCB_Configure(seL4_TCB _service, seL4_Word fault_ep, seL4_CNode cspace_root, seL4_Word cspace_root_data, seL4_CPtr vspace_root, seL4_Word vspace_root_data, seL4_Word buffer, seL4_CPtr bufferFrame)。
  - service：操作的TCB对象。
  - fault_ep：线程出现错误时传递错误消息的endpoint。
  - cspace_root：设置TCB的Root CSpace。
  - cspace_root_data：可选地设置Root CNode的guard和guard size。
  - vspace_root：设置TCB的Vspace。
  - vspace_root_data：暂时没用。
  - buffer：IPC Buffer的虚拟地址。
  - bufferFrame：IPC Buffer frame对应的CPtr。
- seL4_TCB_SetSchedParams(seL4_TCB _service, seL4_TCB authority, seL4_Word mcp, seL4_Word priority)
  - authority: TCB 在设置优先级和 MCP 时被限制的最大 MCP对应的TCB。

seL4: Schedule

廖东海

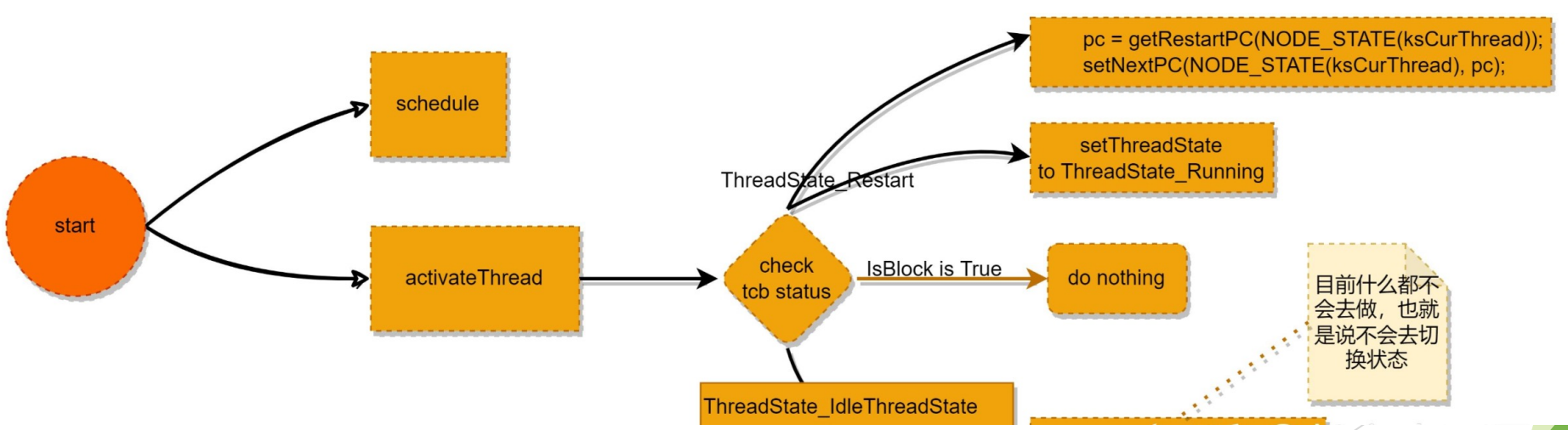
ctrlz.donghai@gmail.com

线程调度

- 普通调度
  - 指的是带有优先级的RR调度。
  - chooseThread(void)函数根据区域和优先级查到相应的就绪队列，再从ready queue的head找到要执行的线程 A。
  - switchToThread()函数负责将当前线程ksCurThread切换到A。
  - SCHED_APPEND 把线程插入就绪队列最后面，SCHED_ENQUEUE把线程插入就绪队列最前面。
- 特殊调度。
  - 调度器action有下面2种状态：
    - #define SchedulerAction_ResumeCurrentThread ((tcb_t*)0)
    - #define SchedulerAction_ChooseNewThread ((tcb_t*) 1)
  - 把要运行的线程的TCB指针赋值给ksSchedulerAction。
    - 通过possibleSwitchTo来赋值。

线程调度

- 调度一般分2步：
  - 第一步调用schedule()配置下个线程A的上下文。然后将ksCurThread指针指向A，意义是根据调度算法找到了下个要运行的线程了。
  - 第二步调用activateThread()切换到A上下文, setNextPC 会将EL指向A的第一条指令。



Thanks for Listening.