第 3 讲进程与调度

第五节：ch3：分时多任务系统

向勇、陈渝、李国良

清华大学计算机系

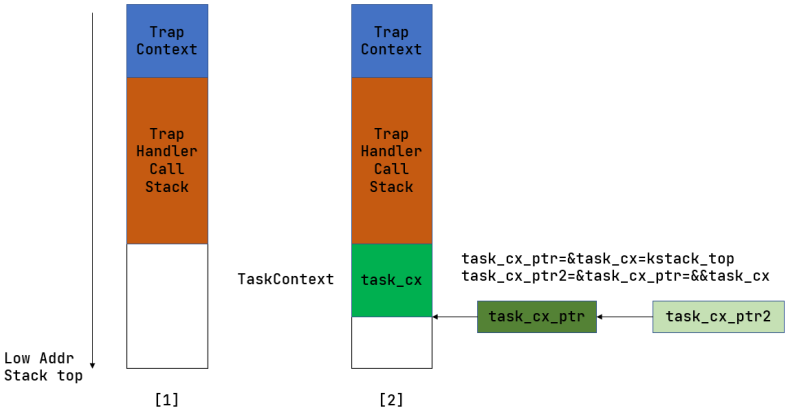
xyong,yuchen,liguoliang@tsinghua.edu.cn

2021 年 9 月 18 日

系统调用：中断上下文保存与恢复

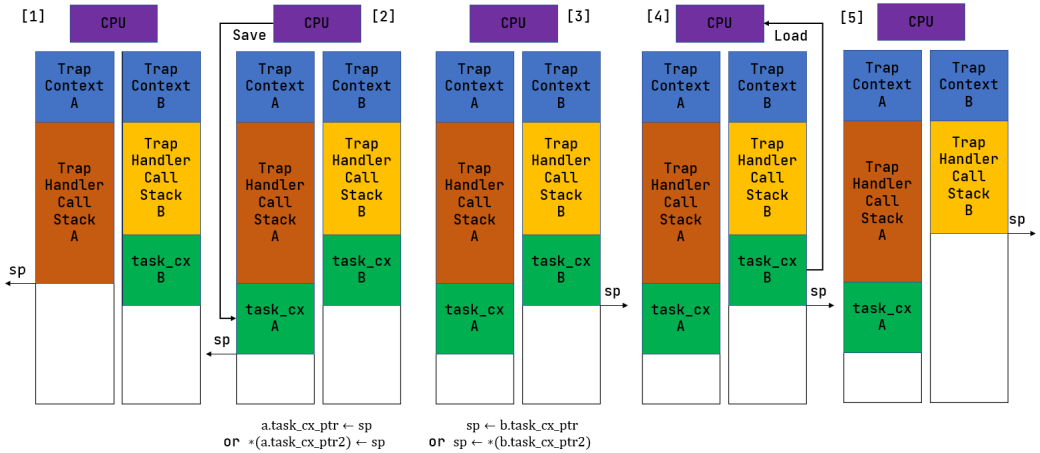
- TrapContext 结构体
- __alltraps 的实现
- 上下文恢复的 __restore 的实现

任务切换：任务上下文 (Task Context)



- TaskContext 数据结构

进程切换过程



- ___switch 的实现

进程切换的实现

如何进入用户态第一次执行应用程序？

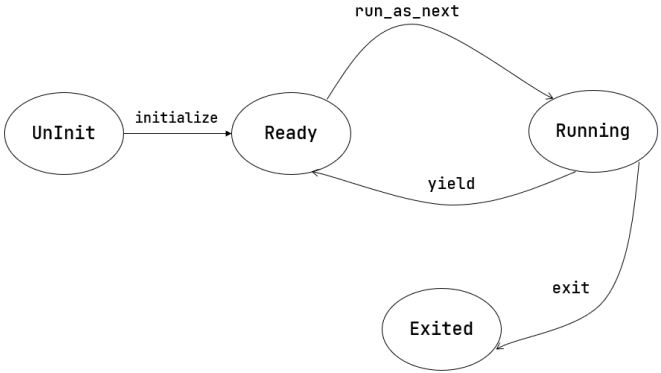
- run_next_app 函数
- app_init_context 函数

多道批处理系统中的程序加载

- load_apps 函数

进程管理：任务运行状态

简单的进程控制块数据结构和三状态进程模型



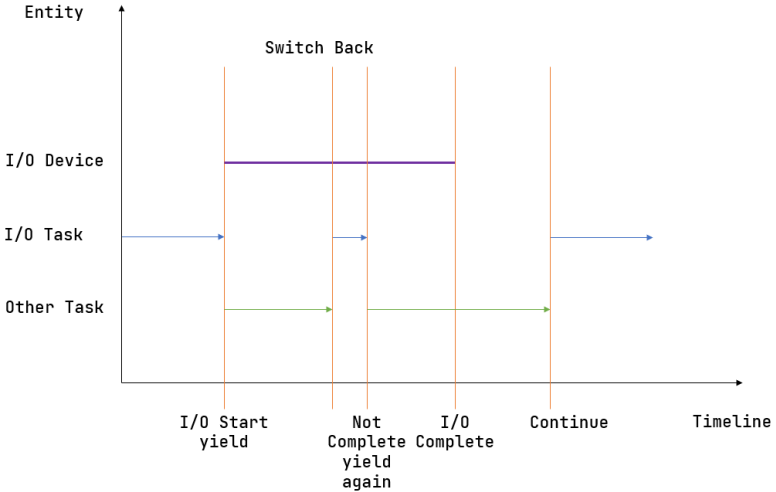
- TaskStatus 数据结构

进程管理：任务控制块

- **任务控制块** (Task Control Block)：TaskControlBlock **数据结构**

协作式调度：主动让出 CPU

主动调用 sys_yield 来交出 CPU 使用权。



sys_yield 和 sys_exit

- sys_yield 系统调用
- sys_yield 的实现
- sys_exit 的实现

第一次进入用户态

多进程下的第一次进入用户态；

- init_app_cx 的实现
- task::run_first_task 的实现
- task::run_next_task 的实现

第一次进入用户态

- timer 模块
- suspend_current_and_run_next 的引用和实现