LAB4 Keypoints

0011 0010 1010 1101 0001 0100 1011

OS LAB

2008-4-30

主要内容

0011 0010 1010 1101 0001 0100 1011

- 多任务调度
- FORK和COW
- 进程间通信（IPC）
- CHALLENGS

lab4任务清单

1. 实现进程调度
2. 实现用于创建进程的系统调用
3. 掌握fork系统调用的原理
4. 掌握并实现Copy-On-Write技术
5. 实现用户PGFAULT服务程序的注册与启用
6. 实现进程间通讯(IPC)

多任务调度

0011 0010 1010 1101 0001 0100 1011

- Round-robin调度算法
  - 把envs当做**循环数组进行搜索**,跳过idle进程
  - 只有当**没有进程可运行**时,运行idle进程
  - **调度时机**：进程自愿放弃CPU或者退出
  - Kern/shed.c中sched_yield()实现RR

fork的原理

0011 0010 1010 1101 0001 0100 1011

- 利用一些**系统调用**来实现一个用户空间的、写时复制的**库函数fork**
- 用户**创建进程相关**的系统调用：
  - sys_exofork
  - sys_env_set_status
  - sys_page_alloc
  - sys_page_map
  - sys_page_unmap

User level page fault

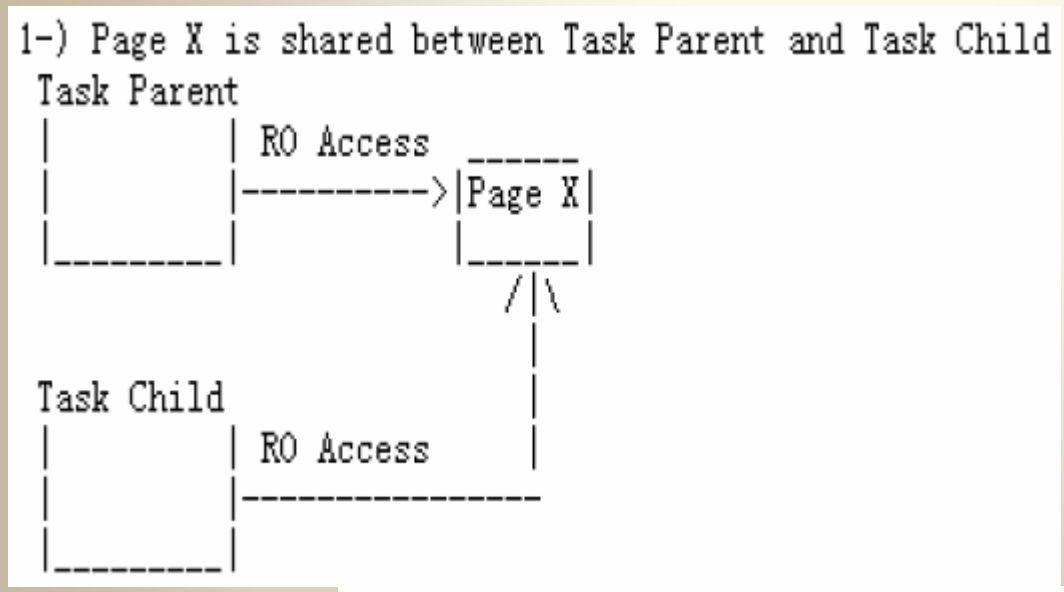
- 需要完成的几件事情:
  - **用户注册**user level的page fault**处理函数**到env环境中供内核调用（记得为exception stack分配空间）
  - **内核处理部分**为user level 的page fault处理函数**设置exception stack**使得user level处理函数处理完后直接返回用户态出错代码执行,并使得内核返回到user level的page fault处理函数处执行
  - user level的**page fault处理函数**:具体处理page fault

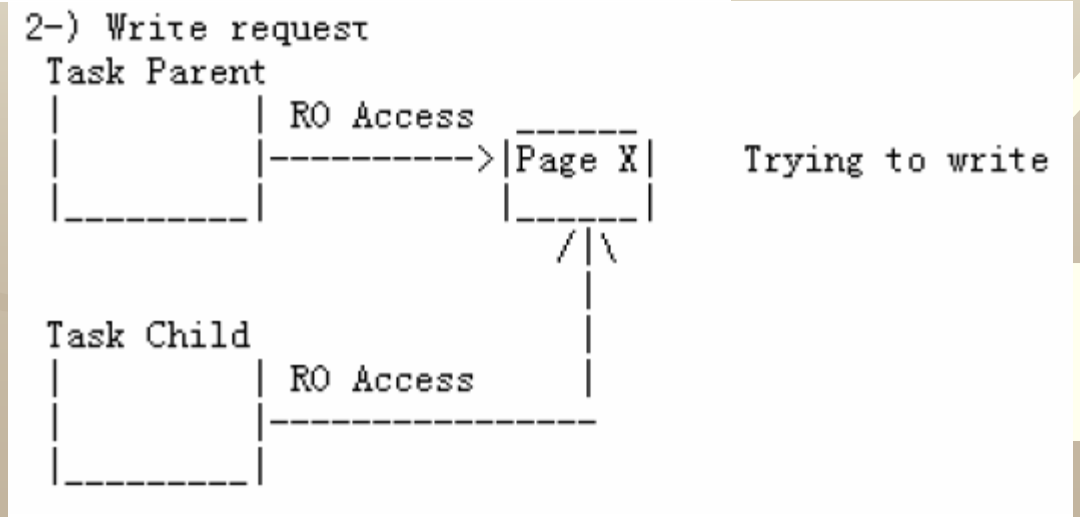
Fork流程

- set_pgfault_handler
- sys_exofork
  - env_alloc
  - Memmove
  - eax (0)
  - Status (ENV_NOT_RUNNABLE)
- 映射 writable or copy-on-write pages为copy-on-write
- 为子进程分配exception stack
- 为子进程设置user level page fault handler
- 标志子进程为runnable

Fork中的Copy On Write

0011





进程间通信

0011 0010 1010 1101 0001 0100 1011

- Env结构中与IPC相关的项:
  - env_ipc_recving
  - env_ipc_from
  - env_ipc_value
  - env_ipc_perm
  - env_ipc_dstva
- 实现两个系统调用:
  - sys_ipc_try_send
  - sys_ipc_recv

sys_ipc_recv()流程

0011 0010 1010 1101 0001 0100 1011

- 设置env中相应项:
  - env_ipc_dstva
  - env_ipc_recving
- 设置返回值eax = 0(为什么要返回值？)
- 设置status 为ENV_NOT_RUNNABLE
- 放弃CPU

时钟中断

0011 0010 1010 1101 0001 0100 1011

- 修改trapentry.S和trap.c初始化IDT中的相应项,来处理时钟中断
- 修改env_alloc()保证用户态下运行时允许时钟中断发生
- 可以用于抢先式进程调度

Challenges

- challenge1:
  - 实现较少循环查询的调度算法，比如固定优先级调度策略
  - 实现时注意 同优先级进程调度机会均等
  - 高优先级进程持续可运行不会造成低优先级进程饿死
- challenge2:
  - 内核在进程切换时为用户态程序保存MMX、FPU等辅助寄存器
  - 进程切换时大量保存、恢复寄存器效率较低
- challenge5：
  - 实现共享内存的sfork，除了stack外均共享
  - 两个进程可以共享bss段(全局变量)

0011 0010 1010 1101 0001 0100 1011

END