Paging and Speeding up Process Creation

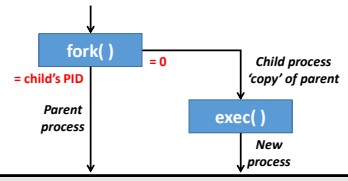
Dr Andrew Scott

a.scott@lancaster.ac.uk

1

Running a New Application

- fork( ) creates **duplicate copy of parent**
  - Often immediately run *exec( )* to start new application
    - Replaces all of child's pages with those for application
  - Doesn't make sense to copy all parent's memory
    - Only to immediately replace with new application code + data



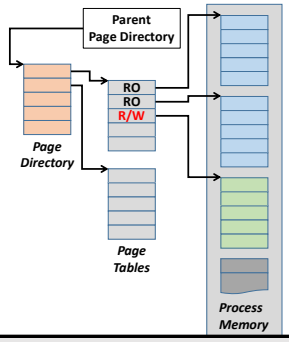
2

Saving Memory: *Copy on Write*

- Duplicate parent's page tables
  - Parent and child have identical tree of tables
    - Both sets of tables reference same memory pages
    - Child shares all parent's pages
  - Mark all pages as Read-Only in child **and** parent
    - But need to remember which should be writeable
      - Can use unused bit in page table entries
- Only works until one process needs to write
  - As pages Read-Only, any write will cause a *page fault*
    …we then copy page and re-run failed memory access

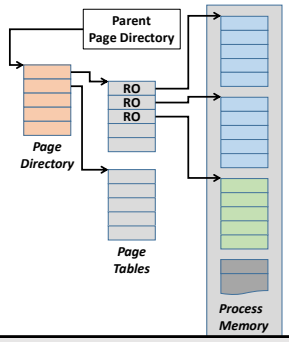
3

Creating a Child



4

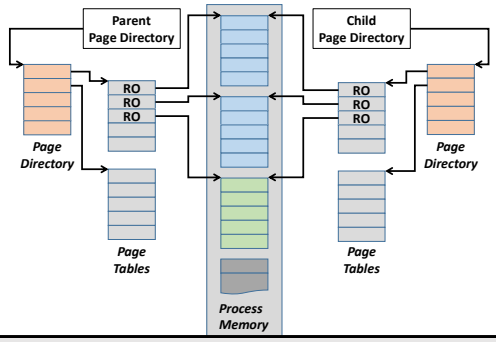
Creating a Child 2



- Notice
  - We set all page table entries to read-only
  - Must remember what access permissions were
    - We can use spare page table bits to hold old access permissions

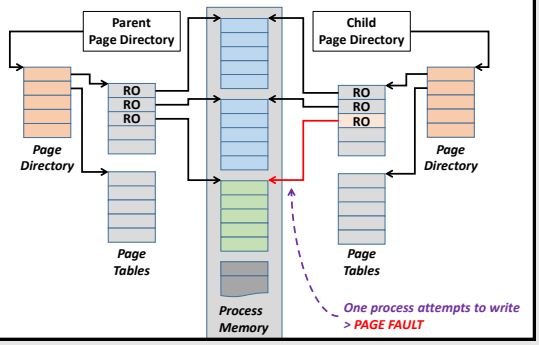
5

Creating a Child 3



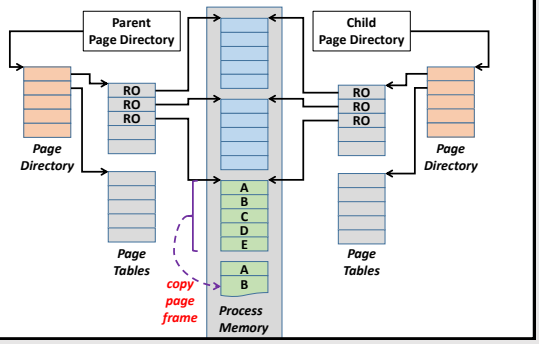
6

Creating a Child 4



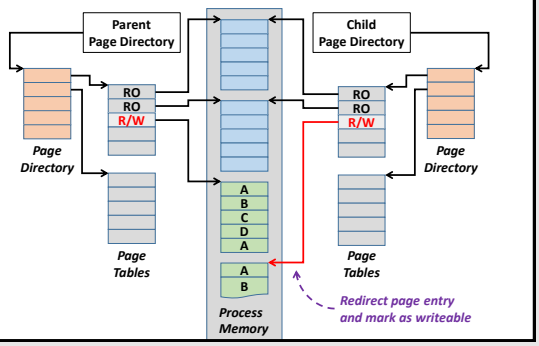
7

Creating a Child 5: *Copy on Write*



8

Creating a Child 6



9