Section 7: Cow Fork
Announcements

- Lab 3 has been graded, please let us know if you have any questions
- Lab 4B deadline is May 15th
- Lab 4C deadline is May 22nd
  - Final deadline, this is the version that will be graded
- Lab X proposal Due May 18th
User-Exception
Stack
COW Fork
(copy-on-write)

Sometimes being lazy is the right thing to do
Lab 4: Before dumbfork

Virtual Address Space A

VPage A

VPage B

VPage C

VPage D

Physical Memory

PPage 1

PPage 2

PPage 3

PPage 4

PPage 5

PPage 6

PPage 7

PPage 8

Read/Write Virtual Page

Read Only Virtual Page

Physical Page In Use
Lab 4: After dumbfork

Virtual Address
Space A

Virtual Address
Space B

Physical Memory

PPage 1
PPage 2 (Copy of 1)
PPage 3 (Copy of 4)
PPage 4
PPage 5 (Copy of 6)
PPage 6
PPage 7
PPage 8 (Copy of 7)
COW Fork: before a copy-on-write fork()

Virtual Address
Space A

VPage A

VPage B

VPage C

VPage D

Physical Memory

PPage 1

PPage 3

PPage 5

PPage 6

Read/Write Virtual Page

Read Only Virtual Page

Physical Page in Use
COW Fork: after a copy-on-write fork()

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Memory</th>
<th>Virtual Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space A</td>
<td></td>
<td>Space B</td>
</tr>
<tr>
<td>VPage A</td>
<td>PPage 1</td>
<td>VPage A</td>
</tr>
<tr>
<td>VPage B</td>
<td>PPage 3</td>
<td>VPage B</td>
</tr>
<tr>
<td>VPage C</td>
<td>PPage 5</td>
<td>VPage C</td>
</tr>
<tr>
<td>VPage D</td>
<td>PPage 6</td>
<td>VPage D</td>
</tr>
</tbody>
</table>
COW Fork: After Environment A writes to Virtual Page A
COW Fork

- Use PTE_COW to mark a page as a copy on write page (these pages must also be read only, no PTE_W)
- Can/should we use COW semantics for all parts of the environment’s address space?
  - User stack?
  - User exception stack?
  - User heap?
  - Kernel’s space (above UTOP)?