"cat /proc/$PID/maps": What Could Possibly Go Wrong?
What Could Possibly Go Wrong???
What Could Possibly Go Wrong???

Tiny slice of the system

- Not-So-Important Monitoring
  (reads /proc/$PID/maps)

The rest of the system

- Very Important CPU-Bound Application
What Could Possibly Go Wrong???

Tiny slice of the system

Not-So-Important Monitoring
(reads /proc/$PID/maps)

The rest of the system

Very Important CPU-Bound Application

Preempted holding mmap_sem  Blocks address-space change
Exactly How Does This Happen???

1) Not-so-important monitoring (NSIM) acquires mmap_sem to read /proc/$PID/maps

2) Very important CPU-bound application (VICBA) thread A invokes mmap() and blocks write-acquiring mmap_sem

3) VICBA thread B takes a page fault and blocks read-acquiring mmap_sem

4) Other VICBA threads and other unrelated work consume all available CPU, preventing NSIM from running.

5) VICBA threads A & B are blocked indefinitely!!!
Reproducer

- Problem happens in production, but rarely
- Helpful to have reproducer for testing:
  - One process maps and unmaps a region
  - Another repeatedly scans /proc/$PID/maps
  - Others consume all available CPU

https://github.com/paulmckrcu/proc-mmap_sem-test.git
## 24 Runs of the Reproducer on v5.4

<table>
<thead>
<tr>
<th>--nbusytasks</th>
<th>Median</th>
<th>Minimum</th>
<th>Maximum</th>
<th># &quot;hangs&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.097</td>
<td>0.036</td>
<td>0.141</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>27.296</td>
<td>23.932</td>
<td>116.081</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>123.514</td>
<td>119.402</td>
<td>179.284</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>357.379</td>
<td>307.146</td>
<td>1251.496</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>8019.600</td>
<td>4114.936</td>
<td>12020.700</td>
<td>23</td>
</tr>
</tbody>
</table>

```bash
./run-proc-vs-map.sh --nsamples 24
```
VMA Maple Tree

- Tree protected with a spinlock
  - Readers can use RCU
- VMAs are now RCU freed
- Visible inconsistencies are tolerable
  - May see overlapping VMAs
  - May miss newly added VMAs
Compare With Maple-Tree Prototype

<table>
<thead>
<tr>
<th></th>
<th>V5.4</th>
<th></th>
<th>Maple-Tree Prototype (Jan 2021)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Busy</td>
<td>Median</td>
<td>Minimum</td>
<td>Maximum</td>
<td>Median</td>
</tr>
<tr>
<td>0</td>
<td>0.039</td>
<td>0.036</td>
<td>0.088</td>
<td>1.329</td>
</tr>
<tr>
<td>1</td>
<td>27.037</td>
<td>26.955</td>
<td>76.058</td>
<td>2.007</td>
</tr>
<tr>
<td>2</td>
<td>27.577</td>
<td>27.243</td>
<td>31.574</td>
<td>1.797</td>
</tr>
</tbody>
</table>

RCU readers decouple /proc/$PID/maps scans from mmap()/munmap()
Page Table Issue

- /proc/$PID/smaps walks page tables
  - Reports presence of pages
- In RCU mode, can race with unmap
  - And the page tables can be freed under it
- Need to RCU free all page tables