Due Reminder

• Lab 2 due Monday 10/30 11:59 PM
  • Read document thoroughly and carefully
  • Read lab slides and watch lab tutorial videos
  • Don’t forget to submit written questions (as .txt)
  • Don’t forget to update the student.info and tag!!!
    • Make sure your tag includes your code
Quiz 1...
Lab 1 Extra Credit

Their Heads Will Roll!

this is ar!
Lab 1 Extra Credit

Printed colored strings: Red Green Blue Yellow Cyan Magenta White

buying whip 5m zezima
Lab 1 Extra Credit
Lab 1 Extra Credit

GO BEAVERS
Lab 1 Extra Credit
Recap: struct PageInfo

<table>
<thead>
<tr>
<th>idx</th>
<th>pp_ref</th>
<th>pp_link</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Struct PageInfo * pages (array)

Physical memory
Recap: struct PageInfo

- struct PageInfo *pp
  - The variable typed as struct PageInfo* will point to a struct PageInfo object in pages array

- You can access
  - pp->pp_ref
  - pp->pp_link

- But you cannot access
  - Physical page via pp

<table>
<thead>
<tr>
<th>idx</th>
<th>pp_ref</th>
<th>pp_link</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Recap: struct PageInfo

• How to get the physical address that
  • is represented by a struct PageInfo *pp?

• page2pa(pp)

• page2kva(pp)
  • Take a look at the implementation of those functions!!!
  • e.g., (pp – pages) << PGSHIFT ← why is this the physical address??
  • Physical page number = (pp-pages)

• memset(page2kva(pp), 0, PGSIZE)
  • This will zero out the corresponding physical page of pp..
CAVEAT for Lab 2

• Boot_map_region()

```c
static void
boot_map_region(pde_t *pgdir, uintptr_t va, size_t size, physaddr_t pa, int perm)
```

• Make sure that va + size do not overflow the 32-bit limit

• E.g.,
  • va = 0xf0000000, size = 0x10000001

• Then, va + size = 1

```python
>>> va = 0xf0000000 # KERNBASE
>>> size = 0x10000001
>>> hex(va + size)
'0x10000001'
>>> hex((va + size) % 2**32) # in 32-bit machine, we only store 32 bits..
'0x1'
```