Project 1: 
Fake Demand Paging Device

SWE3015

Sung-hun Kim

Most contents of slides borrowed from Jaeho Hwang's slides
Our Goal
Our Goal

Fake Memory Assignment!!!
• Create /dev/osp device file
  – Providing only mmap operation

• When mmap system call is called
  – **DO NOT Allocate the Page NOW!!**

• If the access is attempted
  – Page fault must be raised.
  – Then allocate the memory page
    • BUT...
We use only ONE memory page for whole mmap!

- The “Scrooge” paging device
- So, we can see same contents for all pages
  - From /dev/osp

Option 1: implement it as a kernel module.

Option 2: allow ONLY ONE mapping at the same time

- If somewhere would make a page fault to access the page, the previous mapping for the page should be unmapped.
• Refer to our class page
<table>
<thead>
<tr>
<th>open /dev/fake_device successfully !</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>map1</strong></td>
</tr>
<tr>
<td>content: 0x6561a000</td>
</tr>
<tr>
<td>content: 0xfffff8804</td>
</tr>
<tr>
<td>content: 0x38</td>
</tr>
<tr>
<td>content: 0x7f</td>
</tr>
<tr>
<td>content: 0x6f0101</td>
</tr>
<tr>
<td>content: 0x2a</td>
</tr>
<tr>
<td>content: 0x6d6f632f</td>
</tr>
<tr>
<td>content: 0x7562752f</td>
</tr>
<tr>
<td>content: 0x2f75746e</td>
</tr>
<tr>
<td>content: 0x74737055</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>modify contents</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>map1</strong></td>
</tr>
<tr>
<td>content: 0x0</td>
</tr>
<tr>
<td>content: 0x1</td>
</tr>
<tr>
<td>content: 0x2</td>
</tr>
<tr>
<td>content: 0x3</td>
</tr>
<tr>
<td>content: 0x4</td>
</tr>
<tr>
<td>content: 0x5</td>
</tr>
<tr>
<td>content: 0x6</td>
</tr>
<tr>
<td>content: 0x7</td>
</tr>
<tr>
<td>content: 0x8</td>
</tr>
<tr>
<td>content: 0x9</td>
</tr>
</tbody>
</table>

| **map2**                            |
| content: 0x6561a000                  |
| content: 0xfffff8804                |
| content: 0x38                        |
| content: 0x7f                        |
| content: 0x6f0101                    |
| content: 0x2a                        |
| content: 0x6d6f632f                  |
| content: 0x7562752f                  |
| content: 0x2f75746e                  |
| content: 0x74737055                  |
• Insert some logs when entering your code

```
[ 81.349178] mmap_fake: begin: d68f4000 end: d68f5000
[ 81.349181] mmap_fake: begin: d68f3000 end: d68f4000
[ 81.349186] fault_fake:page fault occurred
[ 81.349187] fault_fake:fault address is d68f4000 page_address 68e38000
[ 81.349188] fault_fake: fault handling complete
[ 81.349238] fault_fake:page fault occurred
[ 81.349239] fault_fake:fault address is d68f3000 page_address 68e38000
[ 81.349241] fault_fake: addr(139710295916544) has been unmapped
[ 81.349242] fault_fake: fault handling complete
[ 81.349281] fault_fake:page fault occurred
[ 81.349281] fault_fake:fault address is d68f4000 page_address 68e38000
[ 81.349282] fault_fake: addr(139710295912448) has been unmapped
[ 81.349283] fault_fake: fault handling complete
[ 81.349354] fault_fake:page fault occurred
[ 81.349355] fault_fake:fault address is d68f3000 page_address 68e38000
[ 81.349356] fault_fake: addr(139710295916544) has been unmapped
[ 81.349356] fault_fake: fault handling complete
```
Assignment

• Assignment submission
  – Due date: APR 17th (Fri) 24:00
  – Mail to shkim@csl.skku.edu
    • The title should be “[SWE3015] project1 name”
  – The mail must contains following
    • Screenshots of the Kernel log which shows your page fault handling
    • Screenshots of the test program
    • Source code you written
    • A brief report (<= 3 pages) <- important!
      – Source code snippets you written
      – Explain how you did it briefly
      – Contribution for each
  – Each person submits report
Presentation

- Two times for each team
  - Intermediate presentation (4/13)
    - Progress (5 min) + specific topic (15 min)
  - Final presentation (after midterm)
    - Project review (20 min)
  - Presenter MUST be different for each presentation
• Adding a new device file (김성영)
• Handling page fault (양석우)
• mmap function (윤지아)
Adding A New Dev File

• HINT
  – See "drivers/char/mem.c"
  – You can track the string of well known dev file
    • E.g. /dev/zero, /dev/mem

• Assignment 1
  – TEAM 1 prepare a presentation how to make a dev file
    • What for devfs is?
    • How to create a dev file?
    • Common examples of devfs
    • Don’t have to show to make exact file for this project.
Handling Page Fault

• You should implement proj1 in the demand paging implementation
  – In the end of page fault handler, you can find the code for it

• Assignment 2:
  – TEAM 2 prepare a presentation
    • Explaining which code handles “LINEAR” fault
    • How demand paging is done
• Actually, it is not directly related to this project
  – But if you would do good, it might be added?

• Assignment 3:
  – TEAM 3 prepare a presentation
    • Explaining mmap function and usage
    • Illustrating related kernel codes

• All presentation should not exceed 20 min.
Kernel Version

- Kernel 3.18 (long-term)