Paging: Smaller Tables

CS 537: Introduction to Operating Systems

Louis Oliphant

University of Wisconsin - Madison

Fall 2024

Administrivia

- Project 3 out – Due Oct 8th @ 11:59pm
- Midterm 1 Scheduled:
  - Regular Time: Oct 15th, 5:45-7:15, Van Vleck B102, Ingraham B10
  - Unable to attend? Fill out this form
  - Alternate Time: Oct 16th, 5:45-7:15, Psych Bldg 121
  - McBurney Time: Oct 15th, 5:45-8:45pm, CS 1257
  - Bring **#2 Pencil** and **UW Student ID**

Review: TLB

- The **TLB** is a **fully associative** cache of a few (e.g. 32) PTEs to **reduce the number of memory accesses** during address translation.
- Cache's take advantage of **spatial** and **temporal** locality
- TLB misses can be handled by hardware (the MMU) or software (the OS)
- Flushing or ASID portion in TLB handles issues involving context switches
- Common replacement policies are Random and **LRU**

Quiz 7: TLBs

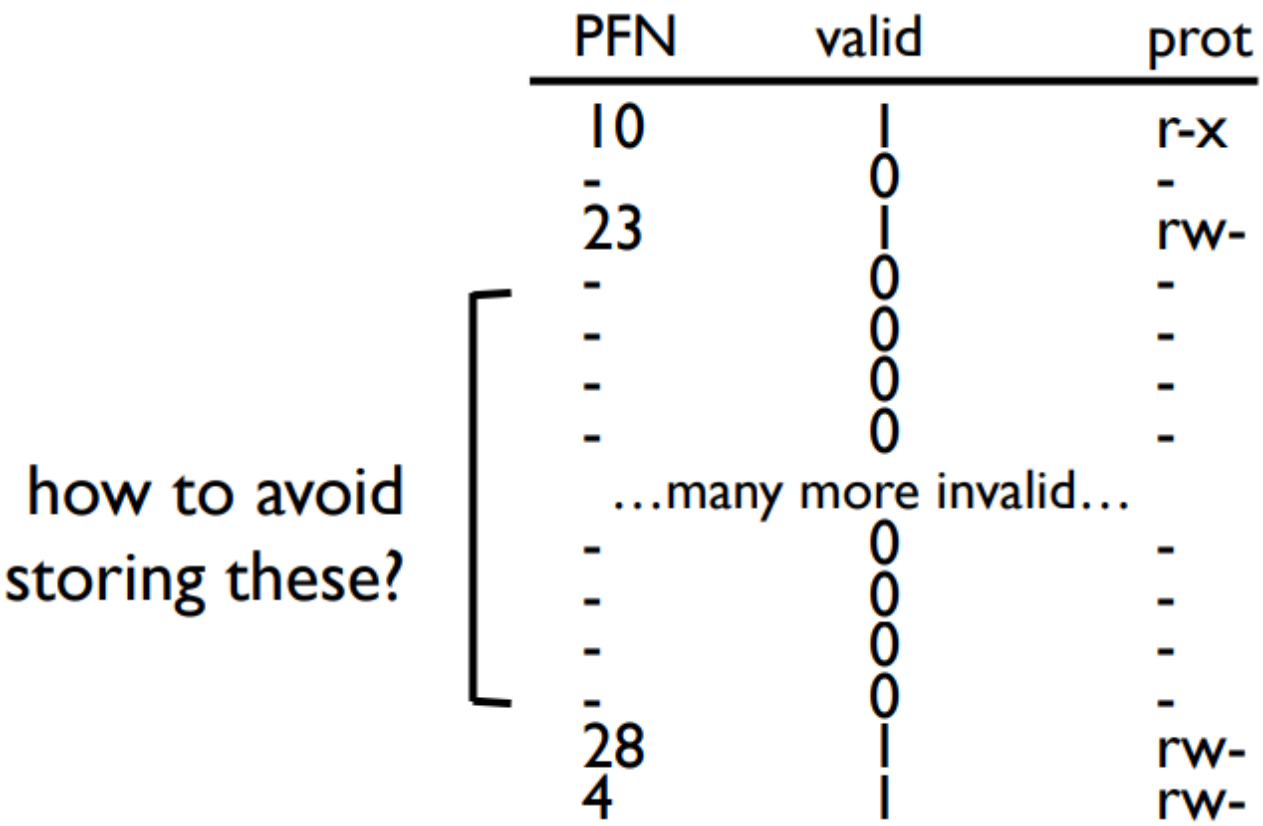
http://tinyurl.com/cs537-fa24-q7

Agenda

Ways of Addressing Large Page Tables:

- Larger Pages
- Hybrid Approach: Combining Segmentation and Page Tables
- Multi-level Page Tables

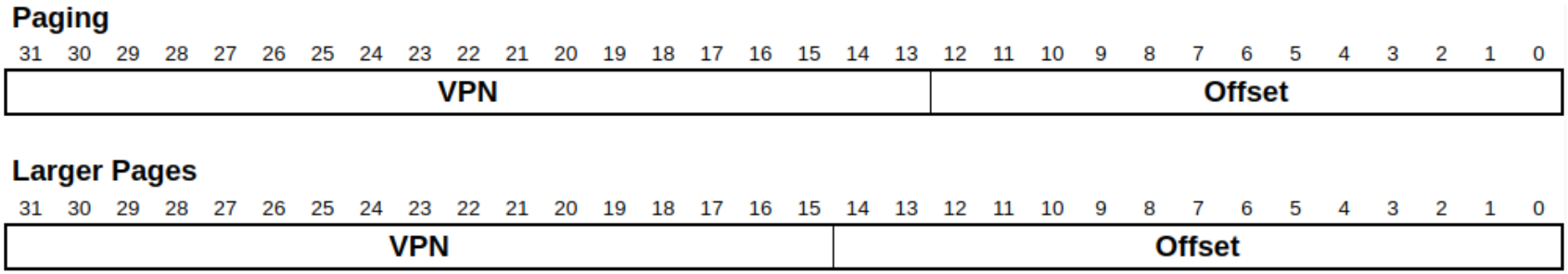
Why Are Page Tables Large?



Larger Pages

4 bytes per entry, 4KB Pages

with 32 bit virtual addresses and 12-bit offset: 1 M entries = 4 MB

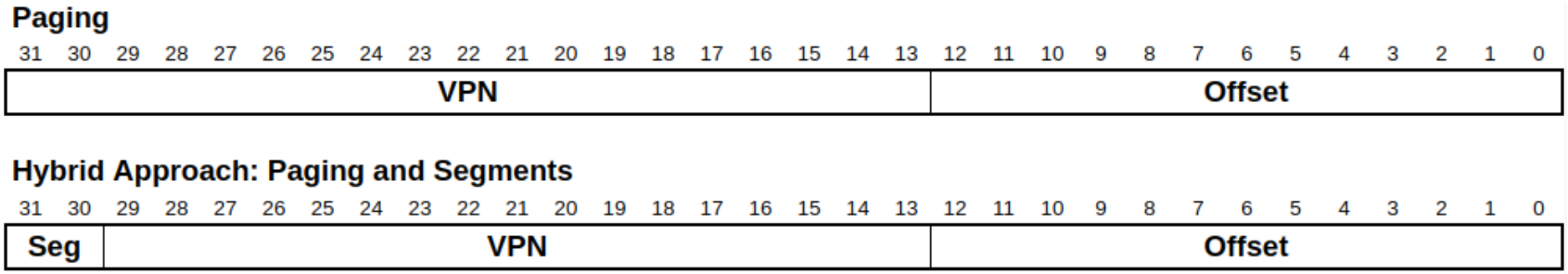


4 bytes entry, 16 KB Pages

with 32 bit virtual addresses and 14-bit offset: 256 K entries = 1 MB

Larger Pages = Waste *within* pages (internal fragmentation)

Hybrid Approach



- A page table for each logical segment
- Base/Bounds for each table
  - Base – Location of table
  - Bounds – Holds maximum valid page in segment

```
SN = (VirtualAddress & SEG_MASK) >> SN_SHIFT
VPN = (VirtualAddress & VPN_MASK) >> VPN_SHIFT
if VPN > Bounds[SN]
   Raise Segmentation Violation
else
   AddressOfPTE = Base[SN] + (VPN * sizeof(PTE))
```

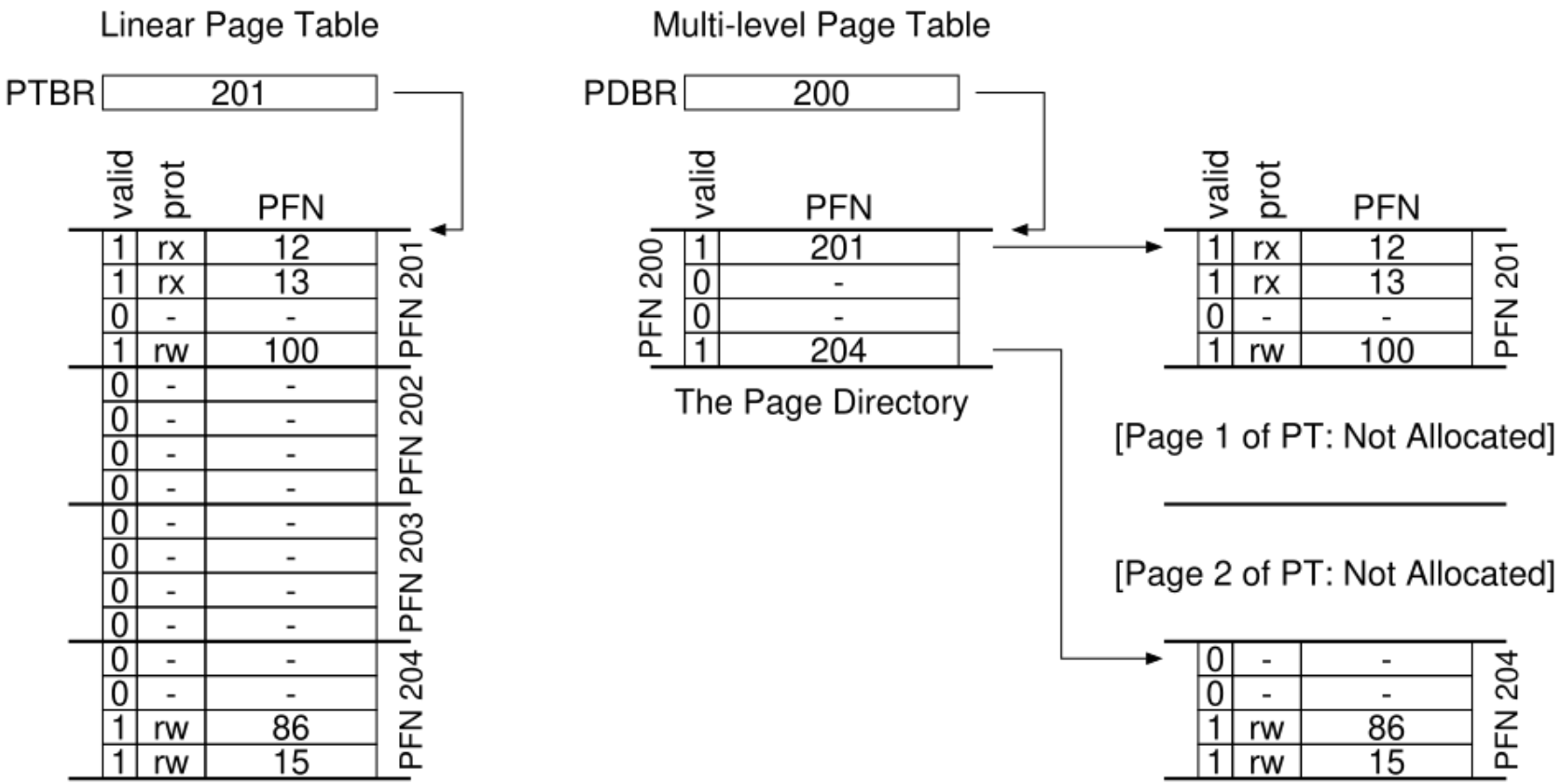
Hybrid Approach Drawbacks

- Large, sparsely used heap, still have a lot of page table waste
- Page tables now can be arbitrary in size (e.g. memory becomes hard to manage, external fragmentation)

Multi-Level Page Tables

- Chop up the page table into page-sized units.
- If an entire page of page-table entries are invalid, don't allocate that page of the page table at all.
- Many modern systems use this approach (e.g. x86)
- to track if a page of the page table is valid, use a **page directory**
- The page directory can be used to find a page of the page table OR indicate no valid pages

Multi-Level Page Table Example

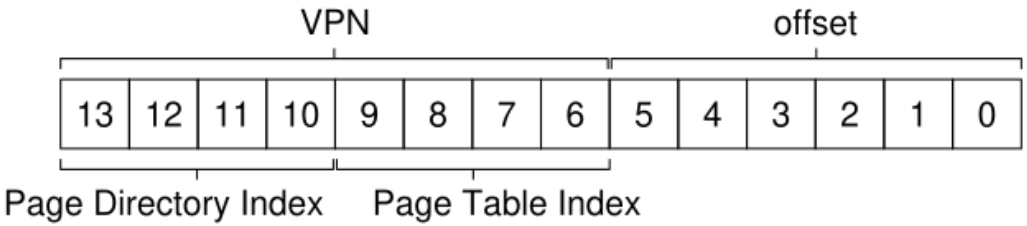


Multi-Level Page Table Advantages & Disadvantages

- Only allocates page table space in proportion to the amount of address space a process uses
- Generally compact and supports sparse address spaces
- If carefully constructed, each portion of the page table fits neatly within a page, making memory management easy
- Adds a level of indirection (**time-space tradeoff**)
  - Now a TLB miss requires 2 memory accesses to get the PTE:
    - First, lookup in the page directory where that portion of the page table is located
    - Second, lookup in that page of the page table the PTE
- Adds complexity

Detailed Multi-Level Example

- 16 KB Address Space (14 bits for a VA)
- 64 byte pages (offset is 6 bits)
- 256 # Pages (VPN is 8 bits)



VPN	Contents
0000 0000	code
0000 0001	code
0000 0010	(free)
0000 0011	(free)
0000 0100	heap
0000 0101	heap
0000 0110	(free)
0000 0111	(free)
.....	... all free ...
1111 1100	(free)
1111 1101	(free)
1111 1110	stack
1111 1111	stack

A Page Directory and Pieces of Page Table

Page Directory		Page of PT (@PFN:100)			Page of PT (@PFN:101)		
PFN	valid?	PFN	valid	prot	PFN	valid	prot
100	1	10	1	r-x	—	0	—
—	0	23	1	r-x	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	80	1	rw-	—	0	—
—	0	59	1	rw-	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	—	0	—
—	0	—	0	—	55	1	rw-
101	1	—	0	—	45	1	rw-

Address Translation With Multi-Level Page Table

Virtual Address: 0x3F80 (11 1111 1000 0000)
PD Index: **1111** PT Index: **1110** Offset **00 0000**
Lookup 1111 in Page Directory: Get PFN is 101
Lookup 1110 in Page 101 of PT: Get PFN is 55 (110111)
Physical Address: 55 (shifted) + offset = 00 1101 1100 0000 = 0x0DC0

More Than Two Levels

- Multi-Level page tables can be any number of levels
- With each extra level adds another memory access on a TLB miss

Summary

- Multiple approaches to shrink page tables
  - Larger pages
  - Hybrid approach
  - Multi-Level page tables
- Select the approach based upon constraints
- **Practice with ostep-homework repository on vm-smallertables**

- Next Time: Swapping (When memory is not enough)