TCSS 422: OPERATING SYSTEMS

Three Easy Pieces:
Translation Lookaside Buffer,
Paging – Smaller Tables

Wes J. Lloyd
Institute of Technology
University of Washington - Tacoma

May 16, 2018

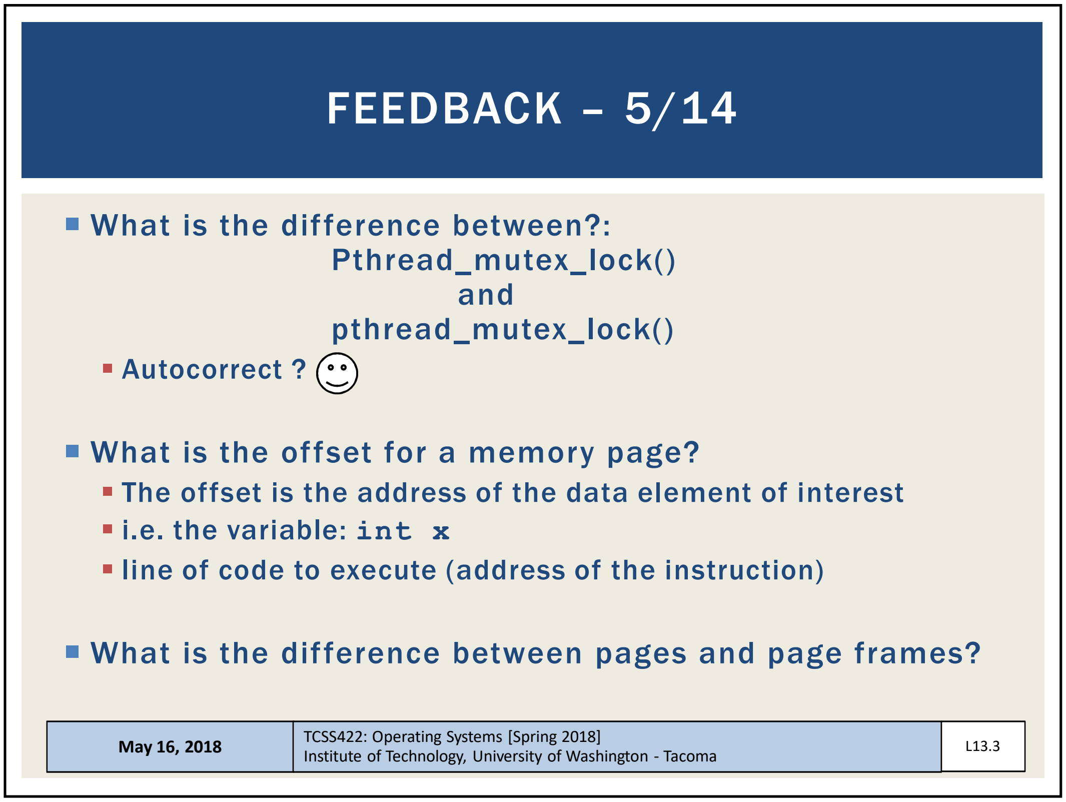
TCSS422: Operating Systems [Spring 2018]
Institute of Technology, University of Washington - Tacoma

OBJECTIVES

- Assignment 2 – Matrix Task Processor
- Assignment 3 – Page Table Walker
- "Group" Quiz #5 –in class

- Memory Virtualization
- Translation Lookaside Buffer – Ch. 19
- Paging – Smaller Tables – Ch. 20

May 16, 2018	TCSS422: Operating Systems [Spring 2018] Institute of Technology, University of Washington - Tacoma	L13.2



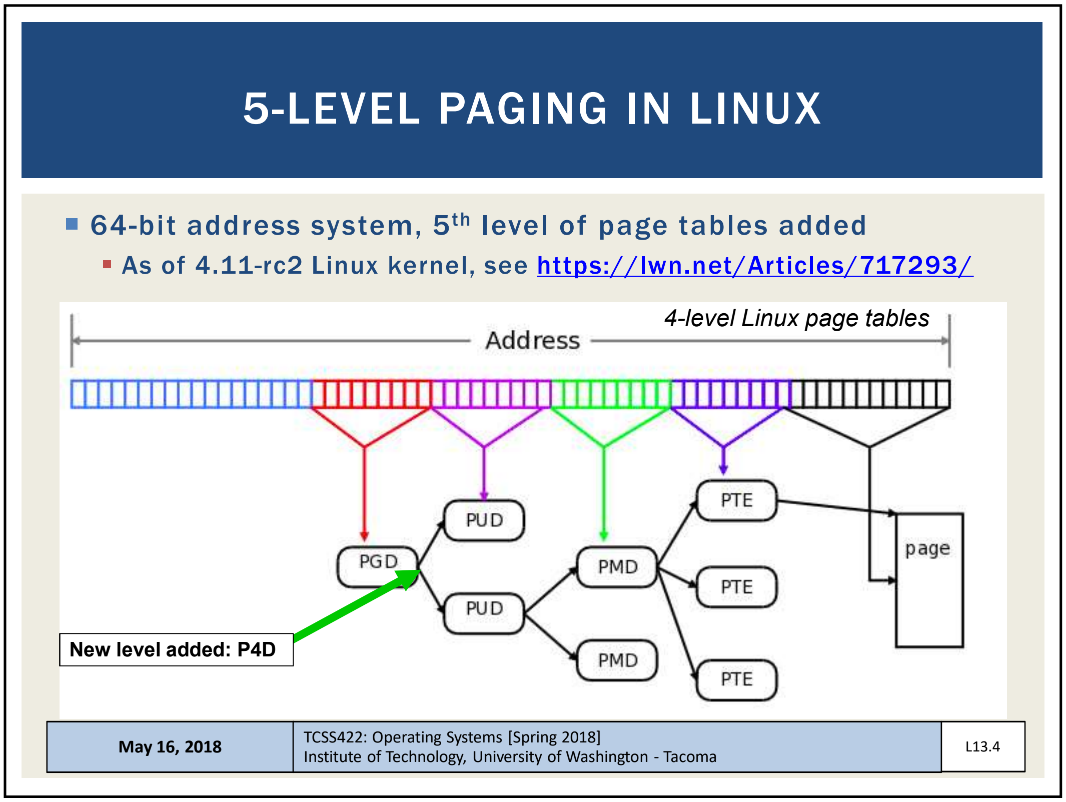
FEEDBACK – 5/14

- What is the difference between?:
  Pthread_mutex_lock()
  and
  pthread_mutex_lock()
  - Autocorrect ? ☺

- What is the offset for a memory page?
  - The offset is the address of the data element of interest
  - i.e. the variable: `int x`
  - line of code to execute (address of the instruction)

- What is the difference between pages and page frames?

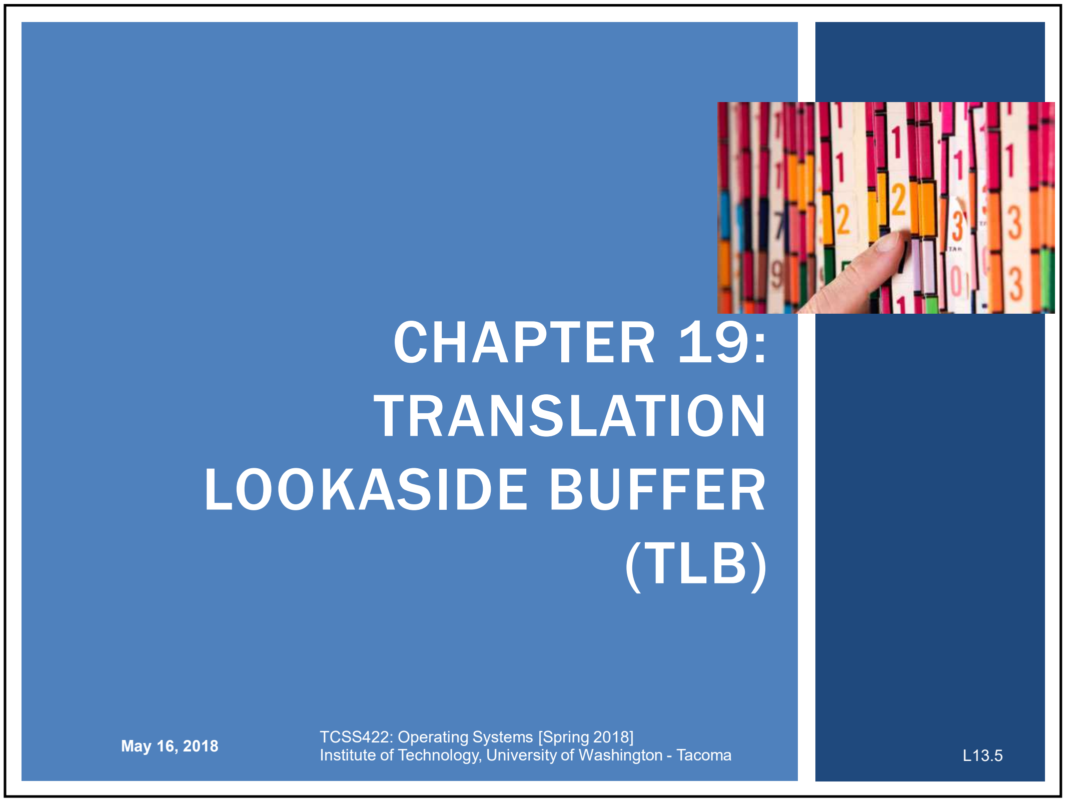
May 16, 2018 | TCSS422: Operating Systems [Spring 2018] Institute of Technology, University of Washington - Tacoma | L13.3



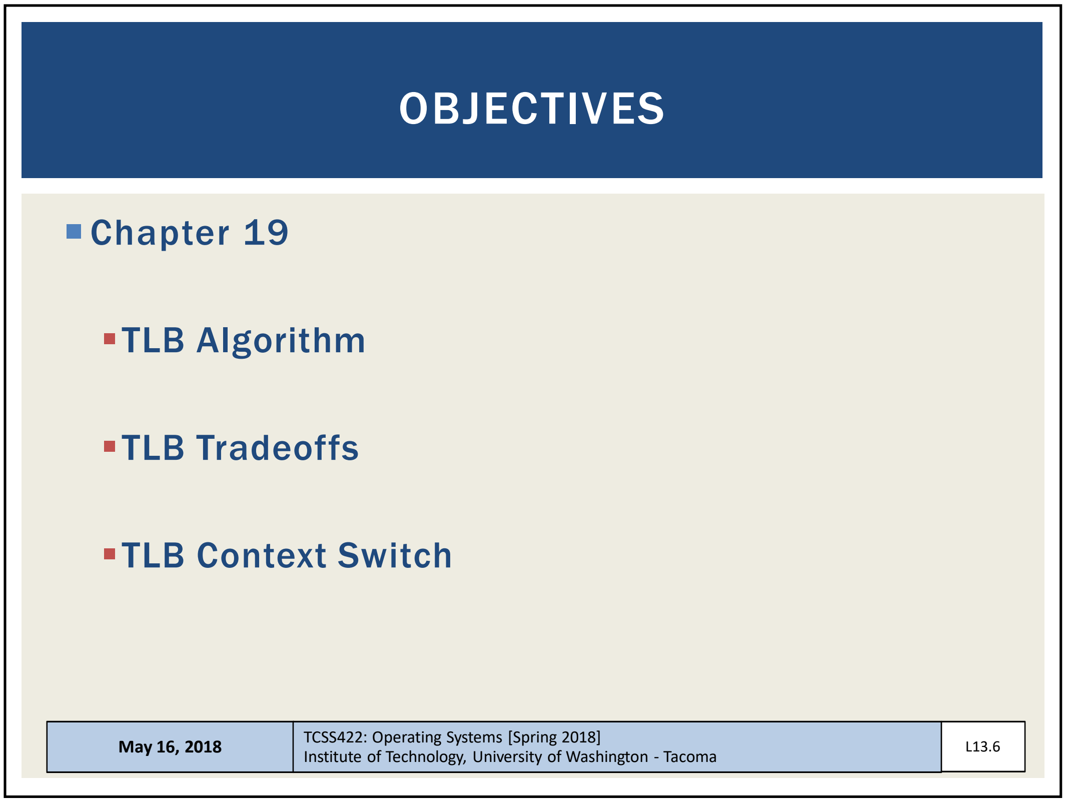
5-LEVEL PAGING IN LINUX

- 64-bit address system, 5th level of page tables added
  - As of 4.11-rc2 Linux kernel, see https://lwn.net/Articles/717293/

May 16, 2018 | TCSS422: Operating Systems [Spring 2018] Institute of Technology, University of Washington - Tacoma | L13.4

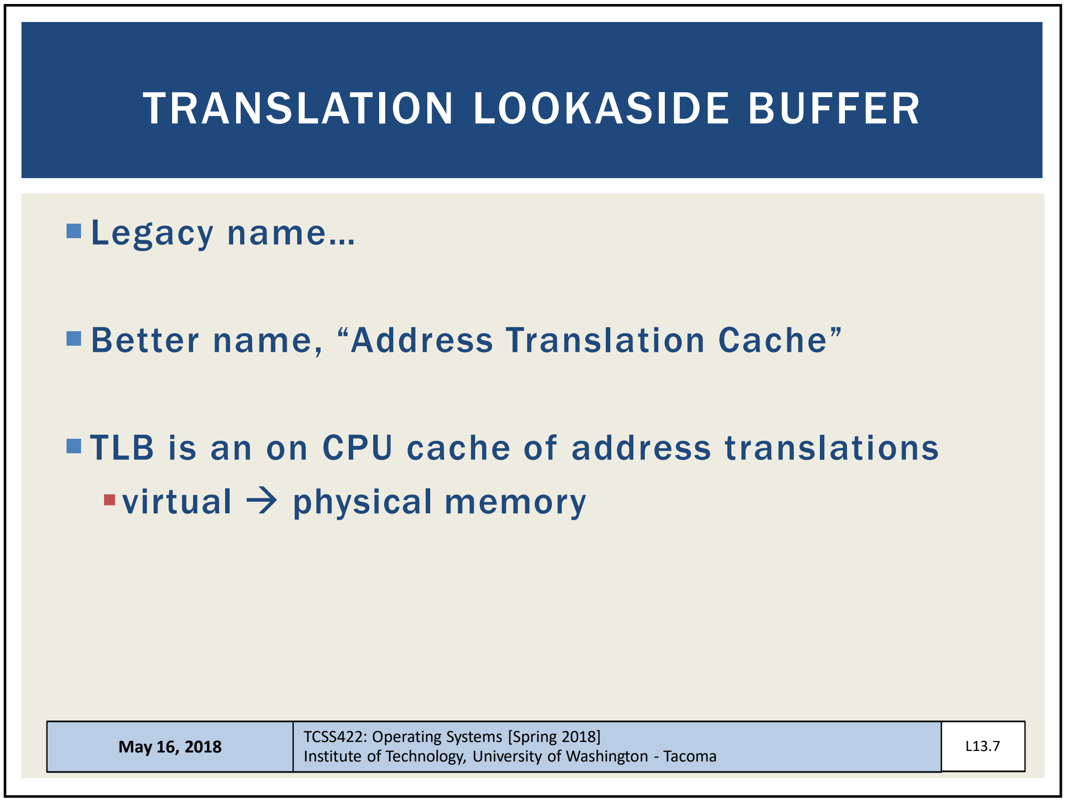


CHAPTER 19: TRANSLATION LOOKASIDE BUFFER (TLB)



OBJECTIVES

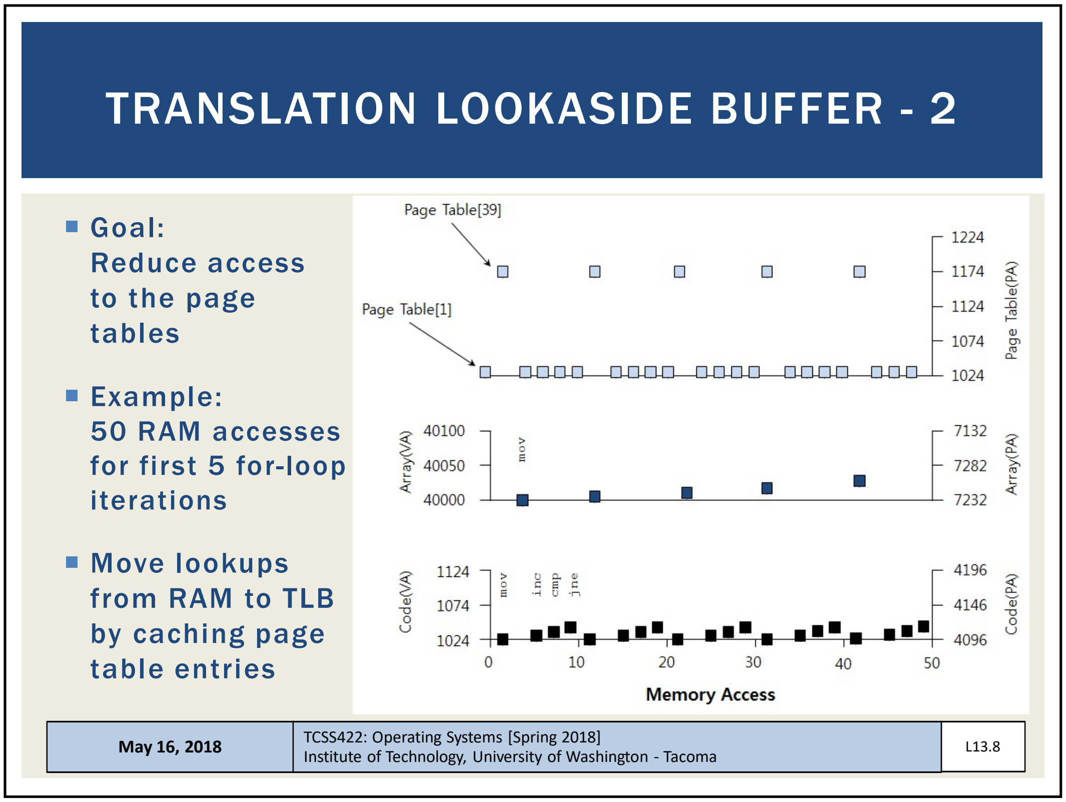
- Chapter 19
  - TLB Algorithm
  - TLB Tradeoffs
  - TLB Context Switch



TRANSLATION LOOKASIDE BUFFER

- Legacy name…
- Better name, “Address Translation Cache”
- TLB is an on CPU cache of address translations
  - virtual → physical memory

May 16, 2018 | TCSS422: Operating Systems [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L13.7



TRANSLATION LOOKASIDE BUFFER - 2

- Goal:
  Reduce access to the page tables
- Example:
  50 RAM accesses for first 5 for-loop iterations
- Move lookups from RAM to TLB by caching page table entries

May 16, 2018 | TCSS422: Operating Systems [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L13.8

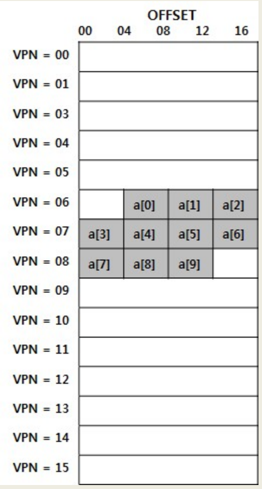
TLB – ADDRESS TRANSLATION CACHE

- Key detail:
- For a TLB miss, we first access the page table in RAM to populate the TLB… we then **requery** the TLB
- **All address translations go through the TLB**

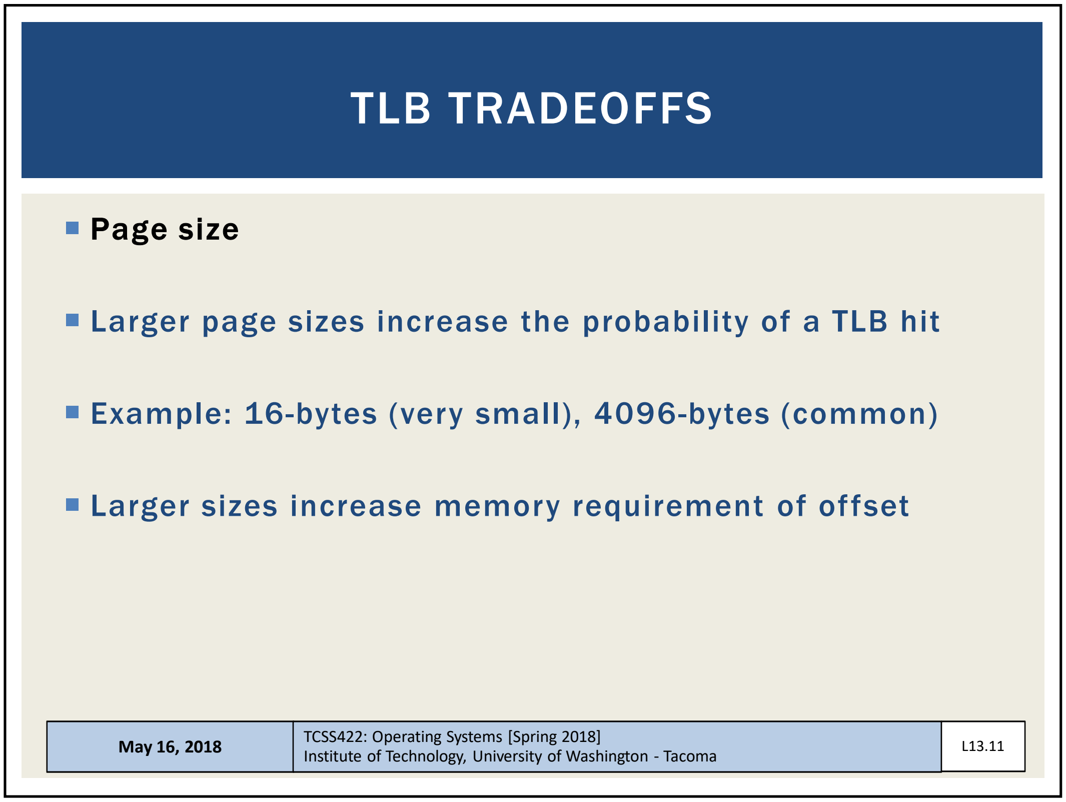
TLB EXAMPLE - 4

```
int sum = 0 ;
for( i=0; i<10; i++){
        sum+=a[i];
}
```

- What factors affect the hit/miss rate?
  - Page size
  - Data locality
  - Temporal locality

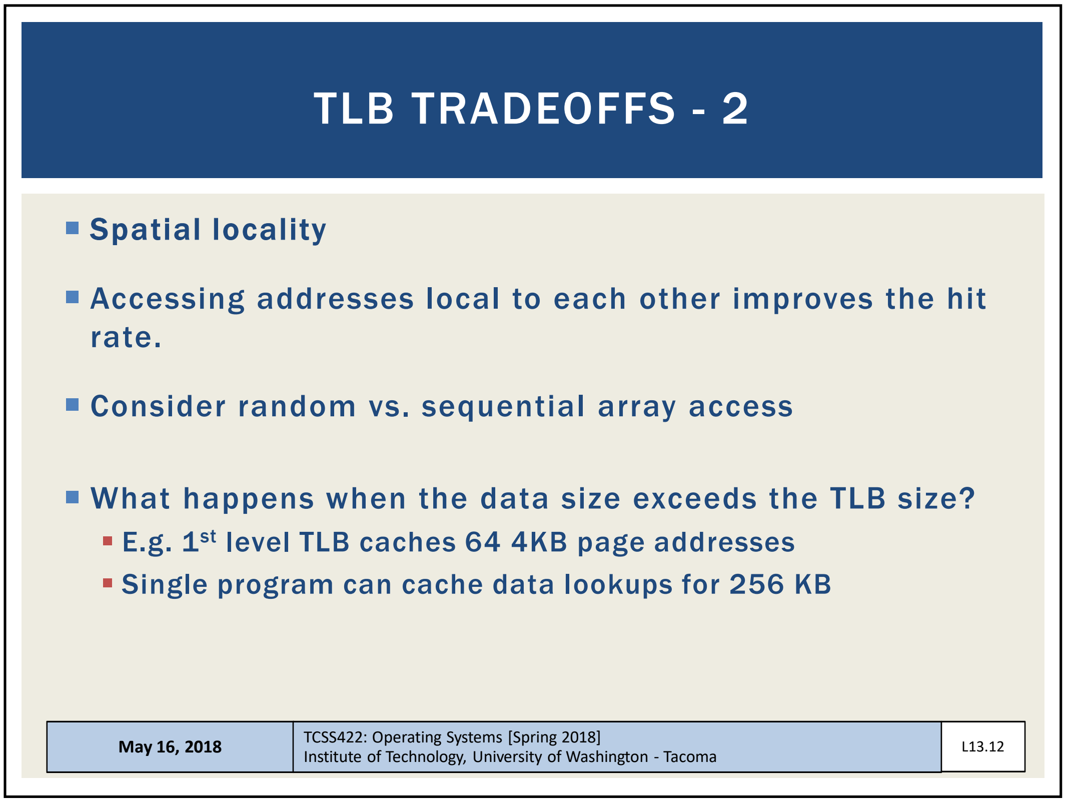


	OFFSET 00	04	08	12
VPN = 00				
VPN = 01				
VPN = 03				
VPN = 04				
VPN = 05				
VPN = 06		a[0]	a[1]	a[2]
VPN = 07	a[3]	a[4]	a[5]	a[6]
VPN = 08	a[7]	a[8]	a[9]	
VPN = 09				
VPN = 10				
VPN = 11				
VPN = 12				
VPN = 13				
VPN = 14				
VPN = 15				



TLB TRADEOFFS

- **Page size**
- Larger page sizes increase the probability of a TLB hit
- Example: 16-bytes (very small), 4096-bytes (common)
- Larger sizes increase memory requirement of offset



TLB TRADEOFFS - 2

- Spatial locality
- Accessing addresses local to each other improves the hit rate.
- Consider random vs. sequential array access
- What happens when the data size exceeds the TLB size?
  - E.g. 1st level TLB caches 64 4KB page addresses
  - Single program can cache data lookups for 256 KB

TLB TRADEOFFS - 3

- Temporal locality

- Higher cache hit ratios are expected for repeated memory accesses close in time

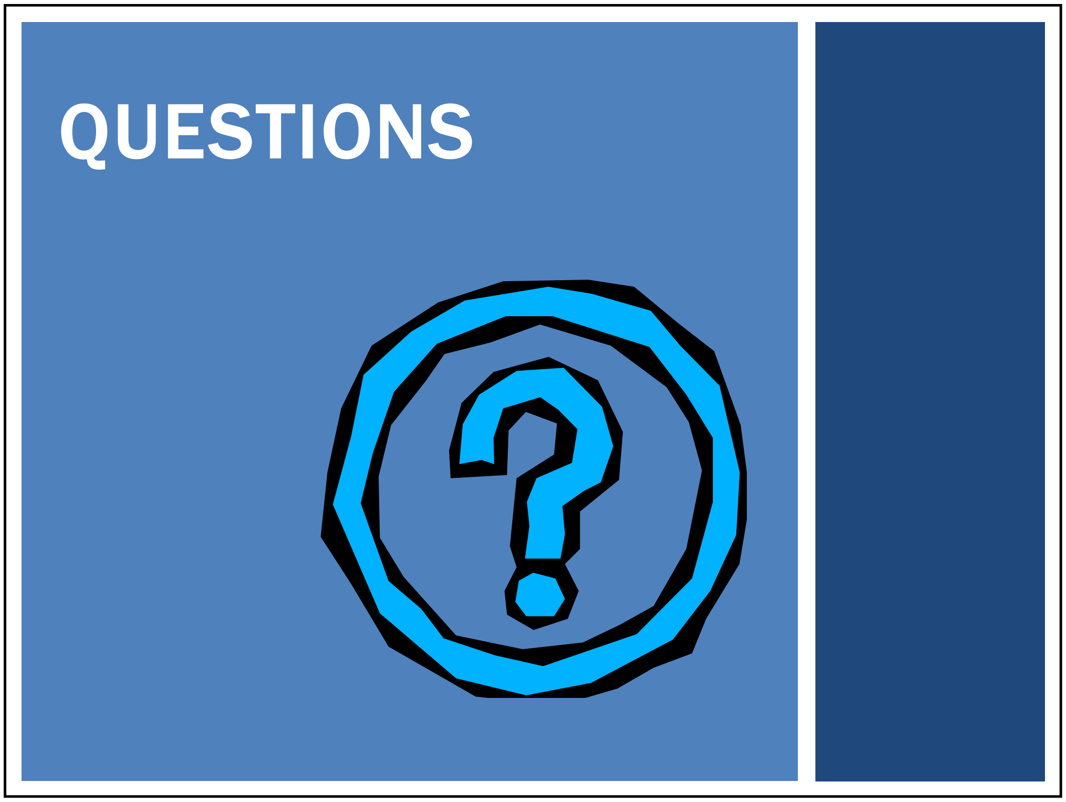
- Can dramatically improve performance for "second iteration"

EXAMPLE: LARGE ARRAY ACCESS

- Example: Consider an array of a custom struct where each struct is 64-bytes. Consider sequential access for an array of 8,192 elements stored contiguously in memory:

- 64 structs per 4KB page
- 128 total pages
- TLB caches stores a maximum of 64 - 4KB page lookups

- How many hits vs. misses for sequential array iteration?
  - 1 miss for every 64 array accesses, 63 hits
  - Complete traversal: 128 total misses, 8,064 hits (98.4% hit ratio)



QUESTIONS