



# Computer Systems B

## COMS20012

Introduction to Operating Systems and Security

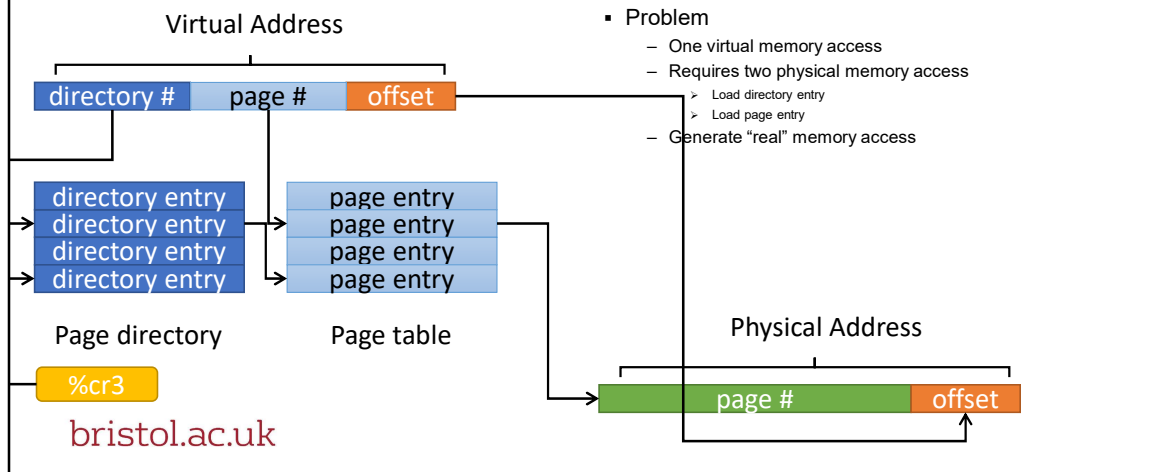
[bristol.ac.uk](http://bristol.ac.uk)

## Translation Lookaside Buffer

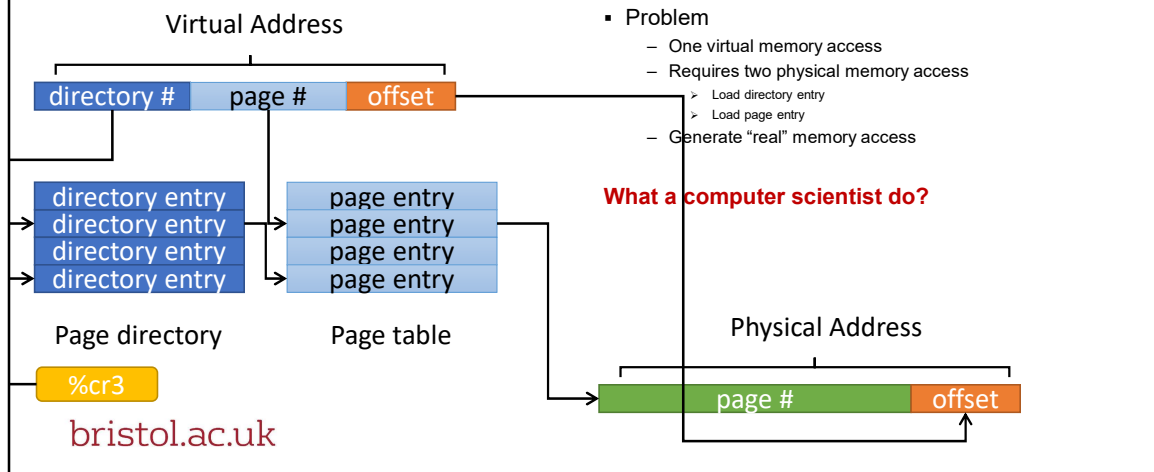
[bristol.ac.uk](http://bristol.ac.uk)



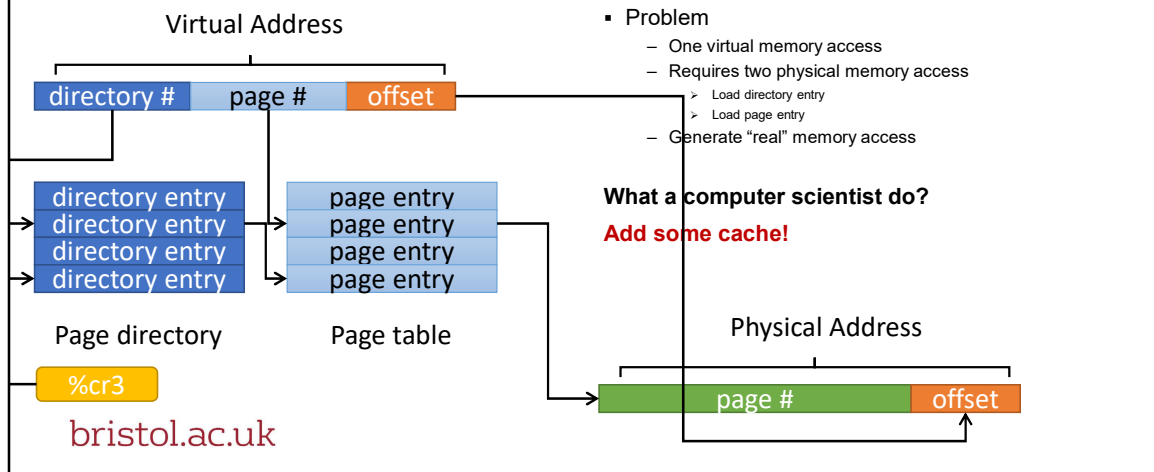
## Previously



## Previously

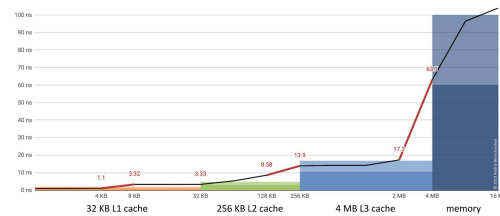


## Previously



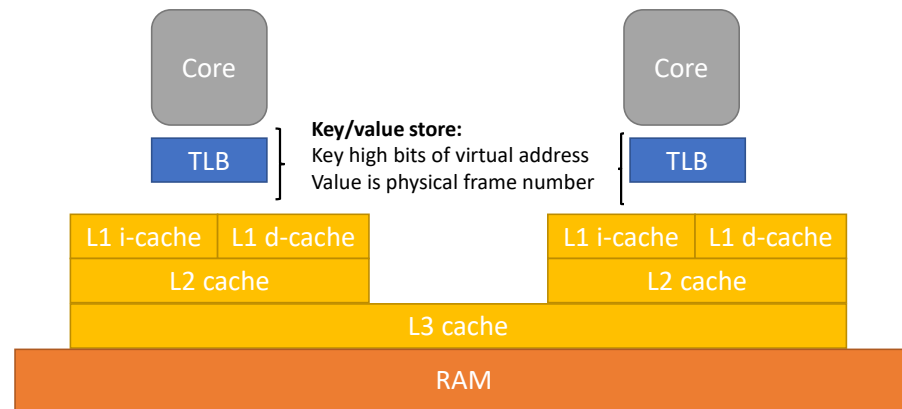
## Translation Lookaside Buffer (TLB)

- Cache some PTE in hardware buffer
- No need to go to physical memory to fetch PTE
- Hardware memory is way faster than main memory!
- We can also be clever about caching!



[bristol.ac.uk](http://bristol.ac.uk)

## Translation Lookaside Buffer (TLB)



bristol.ac.uk

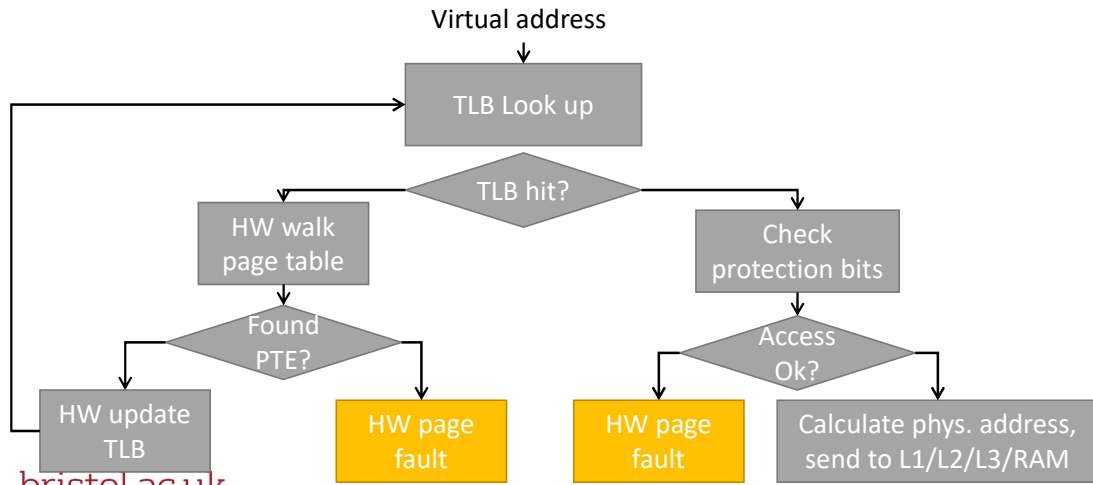
## Why does this work?

- Program exhibit nice locality property
- **Temporal locality:** when a process accesses virtual address  $x$ , it is likely to access it again in the future (e.g., variable on the stack)
- **Spatial locality:** when a process accesses a virtual address  $x$ , the process is likely to address other addresses close to  $x$  (e.g., reading elements of an array on the heap)

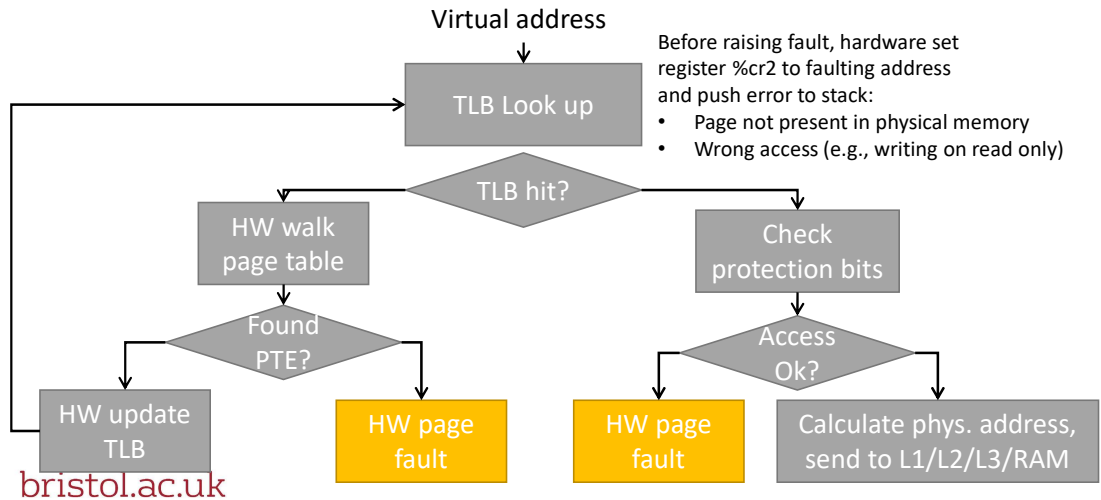
[bristol.ac.uk](http://bristol.ac.uk)



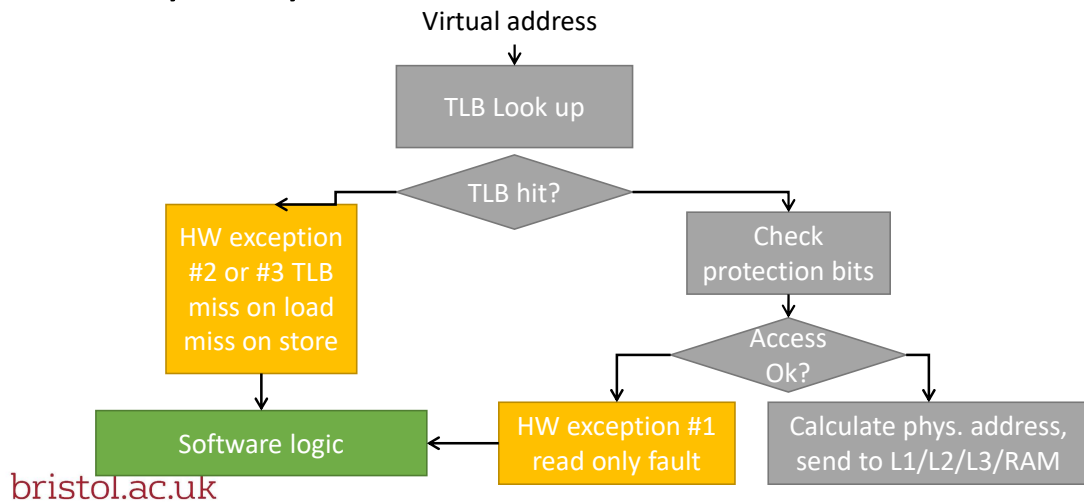
## Memory lifecycle on x86



## Memory lifecycle on x86



## Memory lifecycle on MIPS



## TLB design trade-offs

- **Software TLB**

- Good: freedom to design page directory, page tables and other structures as needed
- Good: OS can implement TLB eviction policies (i.e., deciding which entry to remove when full)
- Bad: slower than hardware

- **Hardware TLB**

- Good: faster!
- Bad: OS cannot change the design of page directory, page table etc.

[bristol.ac.uk](http://bristol.ac.uk)

Thank you

[bristol.ac.uk](http://bristol.ac.uk)

