University of BRISTOL

Computer Systems B

COMS20012

Introduction to Operating Systems and Security

bristol.ac.uk

University of BRISTOL

Swapping

bristol.ac.uk

Swapping pages out

- Physical RAM may be oversubscribed
  - Total virtual pages greater than the number of physical pages
- Swapping is moving virtual pages from physical RAM to a swap device
  - SSD
  - Hard Drive
  - etc.

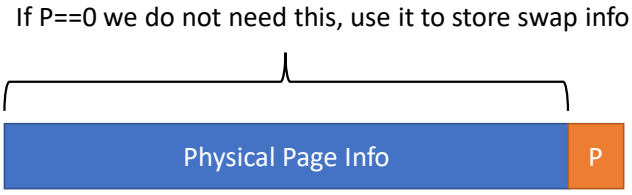
What's in a page table entry (Linux)

- One bit to state if the memory is present in memory or not

bristol.ac.uk

What's in a page table entry (Linux)

- One bit to state if the memory is present in memory or not



bristol.ac.uk

What's in a page table entry (Linux)

- One bit to state if the memory is present in memory or not
- Kernel maintain a list of swap file
- Each file contains several map

bristol.ac.uk

What's in a page table entry (Linux)

- One bit to state if the memory is present in memory or not
- Kernel maintain a list of swap file
- Each file contains several map
- This is greatly simplified, but sufficient
  - Details for interested students: https://www.kernel.org/doc/gorman/html/understand/understand014.html

bristol.ac.uk

Page Faults

- When process try to access page not in memory, problem detected because the presence bit is set to 0
  - With **hardware TLB**, the **MMU** detect this when checking the PTE and raise an exception
  - With **software TLB**, the kernel detects the problem on TLB miss, the **TLB should not contain entry for page not present in memory**!
- Attempting to access a page not in RAM is a **page fault**
- The kernel job on page fault is to:
  - Swap the page from secondary storage to memory, evicting another page if necessary
  - Update the PTE (set physical address + presence bit)
  - Return from the exception so the application can try again

Page Faults are slow!

- Accessing secondary storage is slow
  - Millisecond for harddrive
  - Microsecond for SSD
  - … comparing to nanoseconds for RAM
- Suppose secondary storage is 1000 time slower
  - 1 in 10 access results in page fault -> Average access 100 times slower
  - 1 in 100 access results in page fault -> Average access 10 times slower
  - 1 in 1000 access results in page fault -> Average access 2 times slower
- Goal is to reduce occurrence of page faults
  - Limit the number of processes, so that there is enough RAM
  - Hide latencies by prefetching a page before a process needs them
  - Be clever about which page is kept in physical memory and which page is evicted

Page Faults are slow!

- Accessing secondary storage is slow
  - Millisecond for harddrive
  - Microsecond for SSD
  - … comparing to nanoseconds for RAM
- Suppose secondary storage is 1000 time slower
  - 1 in 10 access results in page fault -> Average access 100 times slower
  - 1 in 100 access results in page fault -> Average access 10 times slower
  - 1 in 1000 access results in page fault -> Average access 2 times slower
- Goal is to reduce occurrence of page faults
  - Limit the number of processes, so that there is enough RAM
  - Hide latencies by prefetching a page before a process needs them
  - **Be clever about which page is kept in physical memory and which page is evicted**

Simplest replacement policy: FIFO

- What page to evict?
- FIFO: remove the page that has been in memory the longest

Num	1	2	3	4	5	6	7	8	9
Refs	a	b	c	d	a	b	e	a	b
PP1	a	a	a	d	d	d	e	e	e
PP2		b	b	b	a	a	a	a	a
PP3			c	c	c	b	b	b	b
Fault?	x	x	x	x	x	x	x		

bristol.ac.uk

Optimum replacement policy: MIN

- What page to evict?
- MIN: replace the page that won't be referenced for the longest

Num	1	2	3	4	5	6	7	8	9
Refs	a	b	c	d	a	b	e	a	b
PP1	a	a	a	a	a	a	a	a	a
PP2		b	b	b	b	b	b	b	b
PP3			c	d	d	d	e	e	e
Fault?	x	x	x	x			x		

bristol.ac.uk

University of BRISTOL

Problem?

bristol.ac.uk

University of BRISTOL

Problem?

Need to know the future…

bristol.ac.uk

Least recently used (LRU) replacement policy

- What page to evict?
- LRU : remove the page that has been used the least recently (temporal locality)

Num	1	2	3	4	5	6	7	8	9
Refs	a	b	c	d	a	b	e	a	b
PP1	a	a	a	d	d	d	e	e	e
PP2		b	b	b	a	a	a	a	a
PP3			c	c	c	b	b	b	b
Fault?	x	x	x	x	x	x	x		

bristol.ac.uk

Practical replacement policy: Clock

- What page to evict?
- Add a “used” bit to PTE
  - Set by MMU when page accessed
  - Can be cleared by kernel

victim = 0

while use bit of victim is set

clear use bit of victim

victim = (victim + 1) % num_frames

evict victim

bristol.ac.uk

University of BRISTOL

Thank you

bristol.ac.uk