Operating Systems
COMS W4118
Lecture 25

Alexander Roth

2015 – 04 – 30

1 Memory Management 2 Electric Boogaloo: Virtual Memory

- If multiple processes use a lot of memory and use more memory than you have, what do you do?
- What if you run out of all physical memory because you were running too many processes?
- To be able to load more processes and have them use more memory physically available.
- Memory access tends to be temporal and spatial.
- Chances are we need to access memory that is adjacent to each other.
- A program tends to access a memory address locally for some time and then jump to different locations.
- A virtual memory system tries to have a disk as the backing storage for memory to give the illusion that accessing memory is handled quickly.
- Page fault refers to the condition where there is a legitimate access to a page, but the page has been swapped out to a disk before it was accessed.
- If a page fault occurs, there is a page fault hanlder that halts the running process, loads the page to memory, and then restart the process that caused the page fault.
- From the process point of view, it never knew what happened. It was just interrupted.
- When we look for the page, we have to translate the virtual mapping to the physical mapping.

- The operating system must know when to bring pages from disk to memory.
- If you are bringing back something from disk drive, then the chances are you don't have a free frame.
- Thus, one of the frames in the table needs to be removed and saved to disk.
- This is known as page replacement.

1.1 Page Selection

- Load pages when they must absolutely be in memory.
- Also used when not necessarily need to save memory.
- The idea is to be as lazy as possible.
- The user can specify which pages are needed.
- This is known as request paging.
- The operating system tends to discount whatever the user asks for.
- Pre-paging tries to smooth out bursts of page faults within the page select.
- A working set refers to all the pages that the processes touches within a given time.
- The point of pre-paging, adaptive paging, is to smooth out some of the page fault rate.
- If we read pages in at a predictive fashion, we won't have the issue of and delay of page faults.
- Thrashing refers to the case where you have a limited amount of RAM but you are running a large amount of processes.
- You don't have enough memory for all the processes so the operating system has to keep saving to disk to make room for the processes.

1.2 Page Replacement Algorithms

- Throw out the pages that won't be used for the longest time in the future.
- The problem is we don't know how long a page will not be used for.
- So we just have to make an approximation.
- Every process algorithm has FIFO, so we could use it for the replacement algorithm.

- The theory is that the past will produce the future.
- When we make a decision to choice a frame to save to disk, LRU will look at the page that was last used the longest.
- Chances are the algorithm will not be used in the future as well.
- However, we need to keep track of when everything was last used, which is expensive.
- The biggest problem about FIFO is that it looks at which page got loaded first, which is not a fair statement.
- What if the item being kicked out is a global variable that is being accessed all the time?
- FIFO ignores access patterns.
- LRU stands for Least-Recently-Used.
- In a normal program where memory access is localized, LRU is pretty standard.

1.3 Least Recently Used Algorithm

1.3.1 Implementation 1

- Hard to implement
- Need to maintain what was the last item accessed.
- Every time page is referenced, save system clock into the counter of the page.
- Page replacement: scan through pages to find the one with the oldest clock.
- Linear scan through all the frames.
- Every time there is a memory access, the clock field in the entry has to be updated.

1.3.2 Implementation 2

- Every time a frame is referenced, move it to the front of the list that is containing the pages.
- The front of the list has the pages that were not accessed in quiet some time.
- Manipulating a link list everytime there is a memory access can be pretty heavy.

1.4 Linux LRU

- Linux approximates LRU by not finding the least recently accessed page, but by finding a not recently accessed page.
- The page being selected may not be the least recently accessed page, but it's close.
- This is known as the clock algorthim or the second-chance algorithm.
- There is a reference bit within the page entry.
- Everytime the page is accessed, the bit is turned on.
- Whenever you read or write something from this page, the reference bit is turned on.
- The hardware sets the bit from 0 to 1.
- The OS tries to find a page with the reference bit cleared.
- The OS traverse all pages, clearing bits over time.
- The hardware will keep turning on the reference bit whenever it is accessed.
- This system will act like a FIFO but only kick out a page if its reference bit is not set.
- The operating system goes through the circularly linked array of pages looking for whichever have their referenced bits set. It then turns the bit off.