Intel 64-Bit Architecture (x86-x64) Paging

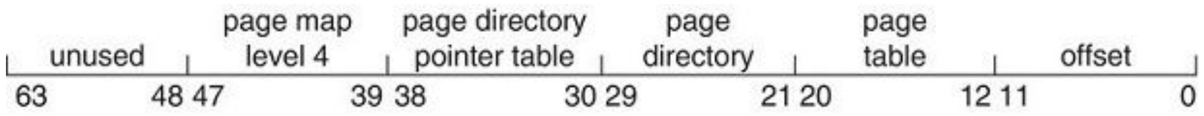
By: Gabriel Tirado

x86-x64 Background

- Initial Intel 64-bit architecture was named IA-64 or Itanium
- AMD later created x86-x64 expanding upon Intel’s IA-64 architecture
- Intel decided to adopt AMD’s x84-x64 architecture
    - Historically it was AMD that would make developments based on Intel architecture. However, the roles reversed this time around.

x86-x64 Background

- Able to hold 2^{64} bytes of memory, or over 16 exabytes
- In practice, only 48 bits are actually used
- 48-bit virtual processor supports 4 KB, 2 MB, or 1 GB page sizes
    - Done through a 4 level page hierarchy visualized by image below



x86-x64 Background

- The 48-bit address scheme allows the use of page address extension (PAE)
- PAE was technology introduced for 32-bit architecture to allow processors to access physical space larger than it normally could.
- Due to PAE, virtual addresses are 48 bits in size but support 52-bit address spaces
  - Around 4,096 terabytes

Linux Background

- Open source operating system initially created by Linus Torvalds in 1991 with the first version releasing September of 1991
- Was a personal project during his time as a student in the University of Helsinki, Finland
- Inspired from Minix, an operating system developed in the 1980s, and based upon Unix design philosophies.
- Initially begun as a command line operating system

Memory Paging

- Modern systems use a technique called paging for managing memory
  - Paging splits physical memory into fixed sized blocks called frames and logical blocks called pages that are allocated to processes upon request
- Logical addresses are split into multiple parts for handling pages
  - The rightmost part of the address is the page offset
  - Subsequent sections are the various levels of page tables that make up the locations of physical space
- Page sizes can range from 4 KB up to 1 GB
  - The typical page size is either 4 KB or 8 KB

Memory Paging

- Linux supports two different page table sizes
  - Default size is around 4 Kilobytes
  - Architecture dependent larger page size called huge pages
- Most processes in Linux use the libc library. Paging allows multiple processes to share similar code, saving memory that would be used calling the same code multiple times
  - Code that is shared must be reentrant or code that supports multiple threads
  - Operating systems must enforce the read-only nature of the code

Page Table

- x86-x64 processors opt to use a four level paging scheme over a hash map table
- x86-x64 though support the ability to use 64 bits, only around 48 of the bits are actually used for paging
    - The remaining 15 bits are currently unused but may find use later on

Swapping

- Swapping is a technique where either parts or entire processes will be moved between main memory and a backing storage
    - This would allow systems to run more processes than there is physical memory to actually hold them
- Traditional Unix systems would swap entire processes
    - All data related to the entire process must be written into backing storage as well as the process
- Linux/Windows OS will swap pages of a process rather than an entire process
    - This is due to the fact that swapping an entire process takes much more time than swapping only parts needed

Demand Paging

- Technique where only pages of a process that are currently needed are loaded into main memory
- If a page is currently not in memory when called, a page fault will occur
    - A page fault will have the operating system go through the process that will move the page into memory
- Linux approach to increase demand paging performance is demand from a file system initially but write pages to a swap space as they're swapped out

Page Replacement

- When no free frames are available, the system will look for an unused frame and free it to be allocated
- For effective demand paging, systems must have a good frame allocation algorithm and page replacement algorithm
- Most common page replacement algorithm used is the least recently used (LRU) algorithm
    - Pages are associated with a timestamp and when it comes to replacing a page, the page that was unused for the longest period is swapped out

Frame Allocation

- Modern systems use proportional allocation which allocates memory to processes based on their size
- If there is too little free memory in a system, a kernel reaper routine will come into play.
    - The routine forcefully reclaims pages until the free memory reaches its max threshold
- Linux has a routine known as the out-of-memory killer (OOM) if memory is too low
    - The routine will choose a process based on an OOM score and terminate that process to free memory

Copy-On-Write

- A common technique used on many operating systems like Linux, Windows, Mac, etc.
- Used mainly for when creating a process through the fork() function
- Initially allows parent and child processes to share pages
    - If either the parent or child attempts to modify the page data, a copy will then be made to accomodate

References

- Canonical Ltd. “About the Ubuntu Project | Ubuntu.” *Ubuntu*, 2019, ubuntu.com/about.
- GeeksforGeeks. “History of Linux.” *GeeksforGeeks*, 12 Feb. 2023, www.geeksforgeeks.org/linux-history/.
- “ZyBooks.” *Zybooks.com*, 2024, learn.zybooks.com/zybook/CSUNIVCSCI431HendersonFall2024. Accessed 6 Nov. 2024.