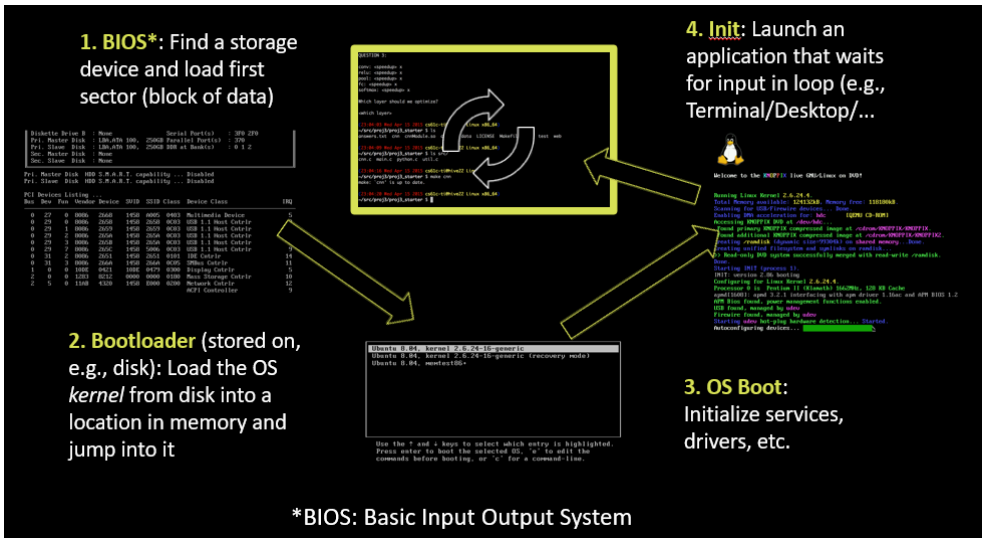
Problem 1 - 50 points - Describe the 4-step process of what the computer architecture does at "boot".

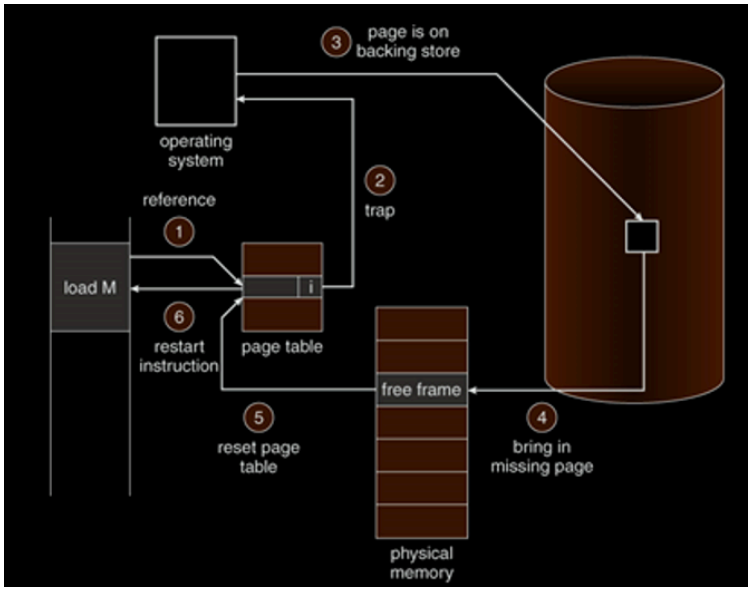


Problem 2 - 50 points - Describe **demand paging** and its benefits in the context of Virtual Memory.

Demand paging provides the ability to run programs larger than the primary memory (DRAM).

- OS can share memory and protect programs from each other.
- Hides differences between machine configurations.
- Today, more important for protection than space management.

Problem 3 - 100 points - Draw and describe the 6-step process for how the operating system resolves a **Page Miss**.



Step 0 - Cache Memory does not contain the address

Step 1: Look up the Reference in Page Table

Step 2) Trap: Inform the OS that we need to perform a page swap

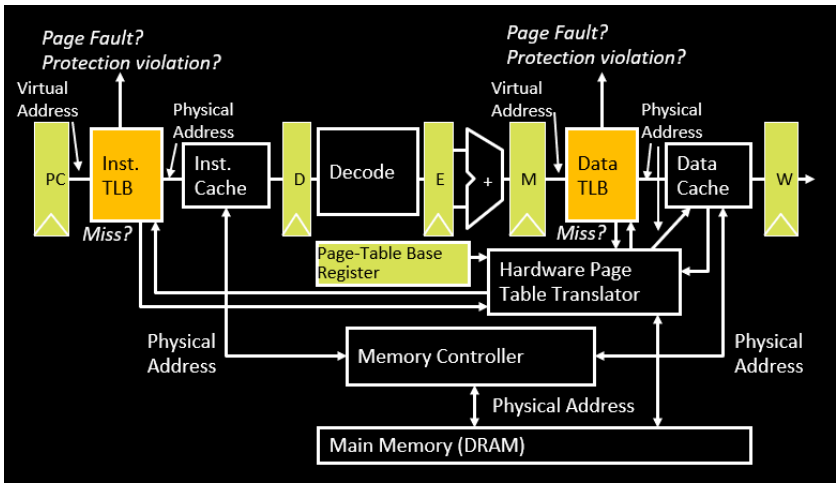
3) Find the page in the non-volatile memory

4) Bring the missing page into the Physical Memory

5) Update the page table with the new page address

6) Restart the instruction

Problem 4 - 100 points - Draw the **Page-Based Virtual-Memory Machine** stages



Problem 5 - 200 points - Given a 64-bit x86_64 Architecture where the page size is set at 4096, the number of physical address size in memory is 2^{52}, and the virtual memory address size is 2^{44}, derive the number of pages in the disk and draw the representation of the Virtual-to-Physical address translation.

Virtual Address - Single Level

51 VPN 12 | 11 Page Offset 0 |

Address translation

Physical Address
43 PPN 12 | 11 Page Offset 0 |

For a single level Page Table in a 64-Bit architecture, define the size of a single Page Table size.

$2^{64} / 2^{12} = 2^{52}$ => $64/8 * 2^{52} = 2^{55}$

Then, what is the smallest amount of page table data we need to keep in memory for n-64 bit applications in a two-level Hierarchical Page Table? Draw the representation of the Virtual-to-Physical address translation.

Virtual Address - Two Level
51 VPN1 40| 39 VPN2 12 | 11 Page Offset 0 |

Address translation

Physical Address
43 PPN 12 | 11 Page Offset 0 |

Finally, how many applications can be implemented in this two-level Hierarchical Page Table in the equivalent memory space?

52 - 12 (Page Offset) - 12 (Level 1) = 28
$2^{55} = 8*(4096 + n*4906)$
$2^{40} = n-1$
$n = 2^{40} - 1$

- Note, the computer won't realistically run this many programs. The 64-bit architecture was designed to account for mind-boggling, ridiculous situations that we will never actually encounter in our lifetimes.

Problem 6 - 150 points - Given the following parameters for memory for a machine with a 2GHz clock.

- L1 cache hit = 2 clock cycles, hit 95% of accesses
- L2 cache hit = 6 clock cycles, hit 85% of L1 misses
- L3 cache hit = 10 clock cycles, hit 60% of L1 misses
- DRAM = 100 clock cycles
- Disk = 10,000,000 clock cycles

a) Derive the Average Memory Access Time without paging

Solution: (0.95*2 + 0.05(0.85*(2+6) + 0.15*(0.6*(2+6+10) + 0.4*(2+6+10+100))) / 2 = 1.3375ns

b) Derive the Average Memory Access Time with paging for a Memory Hit Rate of 98%?

Solution: 2.675 + 0.05*0.15*0.4*10,000,000*(1-0.98) = 301.34ns

c) Derive the Average Memory Access Time with paging for a Memory Hit Rate of 99.9999%?

Solution: 2.675 + 0.05*0.15*0.4*10,000,000*(1-0.999999) = 1.3525ns

Problem 7 - 200 points

a) Assuming that the I/O requests are distributed evenly in time, the average time that a device will have to wait for the processor to poll is half the time between polling attempts. What should be the polling frequency for an I/O device if the average delay between the time when the device wants to make a request and the time when it is polled, is to be at most 10 ms?

Solution: The average time that a device will have to wait for the processor to poll is half the time between polling attempts. Therefore, to provide an average delay of 10 ms, the processor will have to poll every 20 ms, or 50 times per second.

b) If it takes 10,000 cycles to poll the I/O device, and the processor operates at 100MHz, what % of the CPU time is spent polling?

Solution: If each polling attempt takes 10,000 cycles, then the processor will spend 500,000 cycles polling each second. The % of CPU time spent in polling is then $(0.5x10^6)/(100x10^6)=0.5\%$

c) What % of the CPU time is spent polling if an average delay of 1msec is added to the system?

Solution: To provide an average delay of 1ms, the polling frequency must be increased. The processor will have to poll every 2ms, or 500 times per second. This will consume 5,000,000 cycles for polling. The % of CPU time spent polling then becomes 5/100=5%.

Note: In the event a student interpreted adding 1ms to make 11ms instead of 1ms, here is the answer they would have. We will award full credit for this response:

The polling frequency must be decreased. The processor will have to poll every 22ms, or ~45.45 times per second. This will consume 454,545 cycles for polling. The % of CPU time spent polling then becomes $(0.454545x10^6)/(100x10^6)=0.454545\%$

Problem 8 - 150 points - In this last problem, you will do a little research to investigate the virtual and physical memory characteristics of different Notre Dame servers. You will compare the results on:

- `student04.cse.nd.edu`
- ~~`student06.cse.nd.edu`~~
- `student13.cse.nd.edu`

Answer the following questions for each of the three Notre Dame student machines and make sure for each to 1) cite any sources used for your research; 2) include screenshots showing where you found any information you gathered from the machine itself; 3) explain your reasoning.

a) What is the model name for each of these systems?
Hint: Use the command `lscpu`
student04: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
student13: Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz

b) How much L1, L2, and L3 cache are on each of these systems?
student04:
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 20480K

student13:
L1d cache: 32K
L1i cache: 32K
L2 cache: 1024K
L3 cache: 11264K

c) Look in `man proc`: What is the `proc` filesystem?
Note: This will be the same in all the servers, so only answer one.

The proc file system is a pseudo-file system which is used as an interface to kernel data structures. It is commonly mounted at /proc. Most of it is read-only, but some files allow kernel variables to be changed.

d) How much total physical memory (RAM) exists on the systems?
Hint: look at `/proc/meminfo` for each

student04: 32654732 kB

student13: 32163068 KB

e) Look in `man proc` for information on “swap”. The physical pages that are not resident in RAM will be on the hard drive (secondary storage) in a “swap file.” How much space is set aside for the swap file, how much is currently in use, and where is the swap located?

student04:
SwapTotal: 134217720 kB

-bash-4.2$ cat /proc/swaps

Filename	Type	Size	Used	Priority
/dev/sdb3	partition	67108860	0	-2
/dev/sda3	partition	67108860	0	-3

student13:
SwapTotal: 67108856 kB

cat /proc/swaps

Filename	Type	Size	Used	Priority
/dev/sda3	partition	33554428	0	-2
/dev/sdb3	partition	33554428	0	-3

f) How many bits are the virtual and physical addresses on each syste,?
Hint: look at `/proc/cpuinfo` for each

student04 : 46 bits physical, 48 bits virtual
student13: 46 bits physical, 48 bits virtual