Yu Xuan (Shawn) Wang
301227972

CMPT 300: Assignment #4

Virtual Memory & the UNIX ls Command

Part I: Virtual Memory Questions

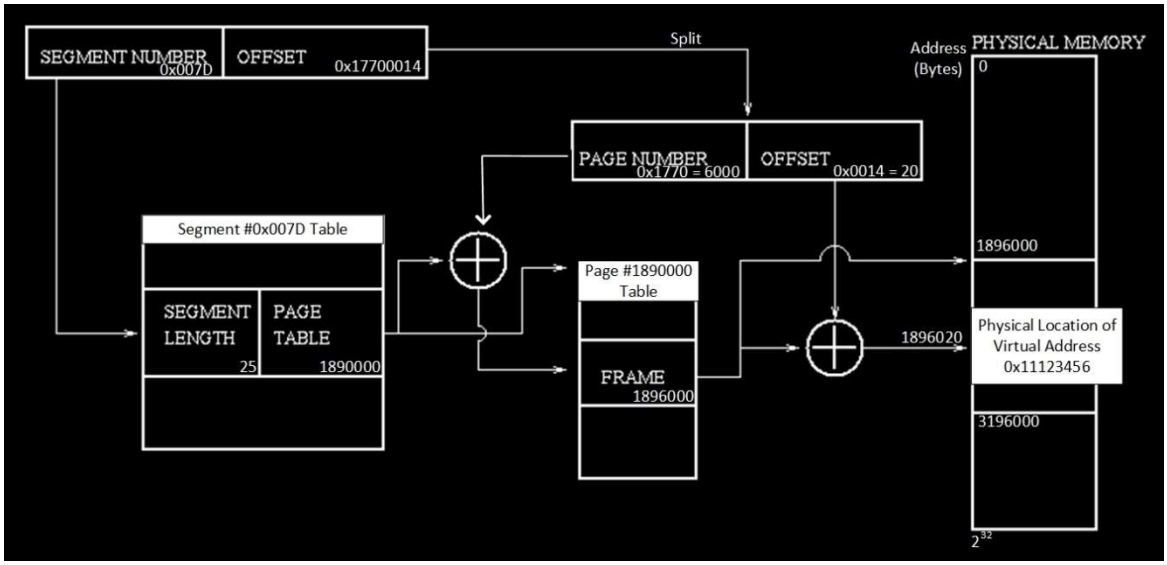
1. A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456 (this is in hexadecimal (base 16)). Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations. Feel free to use a diagram (simple ASCII is fine) if you wish, but that's not required.

   Ans: Since the question stated that the virtualization is implemented by paging, we can deduce that the virtual memory is implemented by the hardware approach, demand paging, instead of software approaches like two-pass assembler or dynamic loading.

   The premise of virtual memory is to free processes from memory constraints bounded by the physical memory size that is present. Demand paging achieves this by partially loading a large process into memory, and loading the other parts when needed. I.e., pages would only be swapped in when needed, and pages that haven't been used for a while would be swapped out to make room when needed.

   In contrast to a software approach, doing so has the advantage of reduced complexity (for example, unnecessary to distinguish main routine and subroutines) and consequently less runtime overhead, at the cost of requiring hardware support. Specifically what type and to what extent of hardware support is required, depends on the Page Replacing Algorithm (PRA) the designer chooses.

   Suppose the PRA deduces that now the page with virtual address at 0x11123456 will be loaded into memory. A lookup in a table generated for this process returns that the page has string '0x007D17700014'. Parsing this yields segment number = 0x07D and offset = 0x17700014, the graph below shows how the physical location is established through virtualization. We see that the physical memory address of 0x11123456 will be 1896020 = 0x001CEE54



Note [1]: For the numbers in the diagram, unless prefixed with 0x to represent hexadecimal, they all represent decimal by default.
Note [2]: The appropriate boundary trapping is not demonstrated but necessary in practice.

2. Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or if the replaced page is not modified, but 20 milliseconds if the replaced page was modified. Memory access time is 100 nanoseconds.

   Assume that the page to be replaced is modified 70% of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds? *Show your work.*

Ans:

∵ Effective Access Time =

$$P_{\text{Page Fault}} \times \text{Page Fault Service Time} + (1 - P_{\text{Page Fault}}) \times \text{Memory Access Time}$$

∴ Effective Access Time =

$$P_{\text{Page Fault}} \times (P_{\text{Page Fault\_Modified}} \times \text{Page Fault Service Time}_{\text{Modified}} + (1 - P_{\text{Page Fault Modified}}) \times \text{Page Fault Service Time}_{\text{Unmodified}}) + (1 - P_{\text{Page Fault}}) \times \text{Memory Access Time}$$

(Continued next page)

$\therefore$

$$eff.\ acc.\ t \le 200ns$$

$$200ns \ge P_{PageFault}(20ms \times 70\% + 8ms \times (1-70\%)) + (1-P_{PageFault})100ns$$

$$2\times10^{-7} \ge (0.0164 - 10^{-7})P_{PageFault} + 10^{-7}$$

$$P_{PageFault} \le 6.0976\times10^{-6} \approx 6.098\times10^{-4}\%$$