CS 4410: Operating Systems
Homework 6

- Homework may be done in pairs, or individually. If doing in pairs, **one** of you should upload to gradescope and add your partner to the group assignment in the upper right corner of the screen. (Do **not** just upload the assignment twice or it will be graded twice, which means grading will take longer.)
- The deadline is Tues, Nov 15 at 11:59AM.
- No late submissions will be accepted.
- You must attribute every source used to complete this homework.
- For some of the problems, you will need two integers. Here is the algorithm for computing these integers:
  - If you are working with a partner, let `var` be the lexicographically smaller of the two NetIDs.
  - Let `varInt` be the integral part of `var`. That is, if `var = rst12`, then `varInt = 12`.
  - If `varInt` is a single digit integer, let `varInt = 13 × varInt`
  - Let `Int1` be the first digit of `varInt`
  - Let `Int2` be the second digit of `varInt`
- Assume the storage unit convention: $1\text{G} = 2^{10} \times \text{M} = 2^{20} \times \text{K} = 2^{30}$ (bytes).
- **For all problems that use `Int1` or `Int2`, please write down the parameters (related variables and settings calculated from your NetID) before answering each question.**

1 VWMare

A hot new startup, VWMare, has asked you to analyze its latest system. The machine has a 40-bit physical address space and a 32-bit virtual address space. The operating system uses a two-level page table scheme that partitions the virtual address space into three pieces as follows:

P_1 bits	P_2 bits	N bits

where

$$P_1 = 9 + (\texttt{Int1} \mod 3)$$
$$P_2 = 9 + (\texttt{Int2} \mod 3)$$
$$N = 32 - P_1 - P_2$$

The leftmost bits are used to index into the top-level page table, the middle bits index into the second-level page table, and the rightmost bits are the page offset. Each page table entry is 4 bytes.

1. In this system, how large is a single physical frame?
2. For a process with a 64K virtual address space starting at address 0:
   (a) How much memory is consumed by the first and second level page tables?
   (b) When storing the virtual pages required by the process into physical frames (*not* including the overhead of the page table), how much space is wasted by internal fragmentation?
3. For a process with 48K of code starting at address 0x1000000, 800K of data starting at address 0x80000000, and a 64K heap starting at address 0xf0000000 and growing upward (towards higher addresses):
   (a) How much memory is consumed by the first and second level page tables?
   (b) When storing the virtual pages required by the process into physical frames (*not* including the overhead of the page table), how much space is wasted by internal fragmentation?

2 Performance Estimation

In a second consulting task for VWMare, you are asked to give a performance assessment of a four prototype systems. In all of the systems, it takes the CPU $(\texttt{Int1}+1)\times 100$ ns to access memory and the access time of a TLB is considered negligible. The page table is stored in memory.

1. For system with a single-level page table and no TLB, how long does it take to access a paged [1] memory reference?
2. For system with a two-level page table and no TLB, how long does it take to access a paged memory reference?
3. For system with a single-level page table and a TLB with a $(\texttt{Int2}+70)\%$ hit rate, how long does it take on average to access a paged memory reference?
4. For system with a two-level page table and a TLB with a $(\texttt{Int2}+80)\%$ hit rate, how long does it take to access a paged memory reference?

[1] meaning the data is currently in a page in memory

3 Ted's Workspace

The surface of Ted's desk is big enough to hold 4 research papers at a time. Ideally, whenever he has to read a paper, it's already sitting on his desk. Sometimes the current paper isn't one of the four on his desk. In that case, he prints out the paper, and puts it on one of the 4 spots on his desk (removing one paper if there were already four there).

Ted needs to read all the papers starting at paper k in the following list (rotating to the beginning when he reaches the end and reading until the entire list has been read). Each letter identifies a paper to read. At first, Ted's desk is empty (no papers) with open slots p_0, p_1, p_2 and p_3 respectively. Assume that Ted prefers placing papers onto the empty slot with a lower subscript if there is one.

$$A, B, C, B, B, \mathbf{A}, D, D, D, E, A, C, E$$

For example, when $k = 6$, Ted starts with the sixth paper in the list (bolded above) and reads the papers in the order:

$$A, D, D, E, A, C, E, A, B, C, B, B$$

1. For ($k = (\texttt{Int1} \mod 12) + 1$), what papers are in slots p_0, p_1, p_2 and p_3 after reading all the papers in the list if he uses each of the following replacement policies? (Break any ties by the lowest subscript of p.)
   (a) FIFO
   (b) LRU
   (c) LFU
2. Assume that Ted's desk begins with papers A, B, C, D in spots p_0, p_1, p_2 and p_3 that were placed there in alphabetical order. Construct a sequence of 6 more papers to read in which
   (a) FIFO outperforms LRU
   (b) LRU outperforms LFU
   (c) LFU outperforms LRU