Assignment 2: Page Table Replacement

- Operation of Page Tables
  - Implement virtual-to-physical address translation and demand paging using a two-level page table
  - Edit methods *allocate_frame* and **find_physpage* in `pagetable.c`
  - **Pagetable**: used to map virtual address and physical addresses. Virtual addresses used by program executed, while physical addresses are used by hardware.
  - **Frame**: chop up space in physical memory in fixed-sized pieces. Each of these frames can contain a single virtual-memory page.
  - **Page**: chop up space in virtual memory in fixed-sized pieces
- Page Replacement: what page to evict to make room
  - Algo
    - **FIFO(First In First Out)**
    - **Clock**(with one ref-bit): evict the oldest page that has not been used
      - One reference bit because one bit is used to keep track if the page is used or not
    - **Exact LRU(Least Recently Used)**: time stamp every reference, evict page with the oldest time stamp
    - **OPT(Optimal Page Replacement)**: also called Belady's algo. Optimal page replacement because lowest fault rate
  - Table columns
    - Hit Rate: displayed after sim.c is executed
    - Hit Count: displayed after sim.c is executed
    - Miss Count: displayed after sim.c is executed
    - Overall Eviction Count: clean eviction count + dirty eviction count
    - Clean Eviction Count: displayed after sim.c is executed
    - Dirty Eviction Count: displayed after sim.c is executed
  - Code inside respective files and may edit struct frame in `pagetable.h`
- Write up
  - Write in file `README.pdf`
  - Four tables prepared in Task 2 and answer the questions