Segmentation and Paging

- **Segmentation:**
  - Divide process memory into *segments* based on their functionality.
  - *Segment Table* holds *base* (segment start address) and *limit* (segment size) registers.
  - Logical memory request has format <*segment, offset*>
    - Request successful if *offset* < *limit,* otherwise *segmentation fault.*
- **Paging:**
  - Divide *main memory* into fixed-size *Frames* (*generally* powers of 2).
  - Divide processes' memory into *frame*-sized chunks called *Pages* (*generally* ~4KB).
  - Logical memory request has format <*page, offset*>
  - *Process Page Table*s stored by a process track in which *Frames* processes' *Pages* are stored.
  - *Process Table* stored in main memory containing [*page, frame*] entries.
    - *Page Tables* are *very* long; require optimizations.
    - **Shared Pages:**
      - Multiple processes of same program often run together.
      - Possible if program code *non-reentrant* (i.e. stateless because of no *static* or *global* variable usage).
      - Processes storing common-code can share *Pages*.
  - **Page Table Structure**
    - **Hierarchical:**
      - Table contains *outer-page* entries, which contains *inner-page* table
      - If *p* bits used for *page* address, then *p-k* used to *outer-page, k* used for *inner-page*
    - **Hashed (with Linked-Lists):**
      - *page address* used as hash-value
      - *Page* is searched for in the Linked-list using *page address*
    - **Inverted:**
      - Replace *page table* with *frame table***.**
        - *frame table* maps *frame number* to *page number* (and *Process ID*)
      - Required to search *entire* frame table (*very*, *very* slow)
        - Operation can be *optimized* using hardware optimizations
  - **Hardware Support using Translation Lookaside Buffer**:
    - Simply a *cache* for *page table* containing recently used entries
    - *Cache hit* means [*page, frame*] (key-value pair) entry is in cache
    - *Cache miss* means entry not in cache; slower main-memory access required to *page table.*
      - If *page* is not in main-memory, *page-fault* occurrs
      - *page-fault* is a *trap;* process will initiate reading from hard-disk.