PA3 MID SUBMISSION

Kabirsingh Karamjeetsingh Bhatia - kbhatia
Abhishek Arun Sheth - asheth
Krishna Chaithanya Marripati – kmarrip

4.1.1 Paging/Framing (frame.c)

The `pd_t` structure defines an entry in a **page directory** used in a virtual memory system. Each entry contains flags and metadata that control various aspects of memory management, such as whether the page table is present, writable, or accessible by users. It also includes flags that determine caching behavior, whether the page table uses large 4MB pages, and a field (`pd_base`) that holds the base address of the page table. These entries help manage the mapping between virtual memory and physical memory, and the flags are used by the operating system to control access and optimize memory usage.

The `virt_addr_t` structure represents a **virtual address** in the system, which is broken down into three distinct parts: the `pg_offset` (12 bits), `pt_offset` (10 bits), and `pd_offset` (10 bits). These fields allow the operating system to efficiently navigate a multi-level page table system, where each part of the address corresponds to an index into a page table or page directory. The `pg_offset` specifies the exact byte within a page, while the `pt_offset` and `pd_offset` are used to locate the correct page table and page directory entry. This breakdown enables the system to translate virtual addresses to physical memory locations.

4.1.3 Backing store memory layout (i386.c and the main.c for verifying)

The following components of the PA3 Demand Paging assignment have been successfully implemented:

1. Memory Layout Configuration:
    - Correctly set up the 16MB physical memory (4096 pages)
    - Properly divided memory into required sections:
        - Pages 0-24: Xinu text, data, bss
        - Pages 25-1023: Kernel memory (with hole from 640K to 1024K+600K)
        - Pages 1024-2047: Frame pool (1024 frames)
        - Pages 2048-4095: Backing stores (16 backing stores)

2. Memory Boundaries:
    - Set maxaddr to 0x007fffff (8MB - 1), which correctly marks the end of usable memory
    - Properly initialized initsp to 0x007ffffc (just below 8MB)
3. Backing Store Configuration:
    - Correctly "stolen" physical memory frames 2048-4095 for backing stores
    - Set up the backing store base at 0x00800000
    - Each backing store can hold up to 128 pages
4. Global Memory Mapping:
    - Properly set up the first 16MB of memory to be mapped to physical memory

The test program (main.c) successfully verifies these memory configurations by displaying:

- Memory boundaries and page size
- Memory region allocations
- Backing store configuration
- Frame pool information
- Memory hole configuration
- Stack initialization