2.6 MEMORY ALLOCATION

Linux memory allocation is a complex system that handles both physical and virtual memory, with different methods for kernel and user space. Key techniques include dynamic allocation via system calls like `malloc` for user space, and specialized kernel functions like kmalloc and vmalloc for the kernel.

User space memory allocation

- **Dynamic allocation:** Processes allocate memory during runtime using functions like `malloc()`, which uses the `sbrk()` or `mmap()` system calls under the hood.
- **Memory mapping:** The `mmap()` system call allows a process to map a file into its virtual address space, enabling efficient access to large files as if they were in memory.
- **Anonymous memory:** This is dynamically allocated memory not tied to a specific file, used for things like the program's heap and stack.

Kernel space memory allocation

- **Page allocator:** The fundamental allocator for physical memory, often used via functions like alloc_pages which requests pages directly from the memory manager.
- **`kmalloc()` :** The most common function for allocating smaller, physically contiguous chunks of memory for the kernel.
- **vmalloc()`:** Used to allocate large, virtually contiguous blocks of memory that may not be physically contiguous.
- **Slab and Buddy allocators:** The kernel uses the buddy system for allocating physical pages and the slab allocator for managing frequently used objects, which helps reduce fragmentation and improve performance.

Key concepts

- **Virtual memory:** Linux uses virtual memory to give each process its own private address space, translating virtual addresses to physical RAM addresses through a hardware-based address translation mechanism.

- **GFP flags:** Functions like `kmalloc()` use get free pages (GFP) flags to specify the allocation context (e.g., for user space or kernel space, for blocking or non-blocking).
- **Memory management subsystem:** The Linux kernel's memory management subsystem is responsible for managing all memory, including virtual memory, demand paging, and memory allocation for both kernel and user space.

Linux uses a virtual memory system where processes access memory indirectly through page tables and a Memory Management Unit (MMU). Memory allocation in Linux can be broadly categorized into physical memory allocation (managing physical pages) and virtual memory management (handling virtual address spaces).

In Linux, memory is divided into several types:

- **Physical RAM**: The actual memory installed on your machine.
- **Swap Space**: Virtual memory on disk, used when RAM is exhausted.
- **Page Cache**: The cache for files that are accessed frequently.
- **Kernel Space**: The part of memory that is reserved for the kernel.
- **User Space**: Memory allocated to user processes.

Memory Allocation Mechanisms:

There are various kernel functions for allocating memory, like

  - kmalloc for small chunks.
  - vmalloc for larger, non-contiguous blocks.

- **malloc()**: Memory allocation in user space is usually done using the malloc() function.
- **brk() and mmap()**: Low-level system calls for memory allocation. Memory Management and System Monitoring in Operating Systems are slab allocator, /proc, top, vmstat