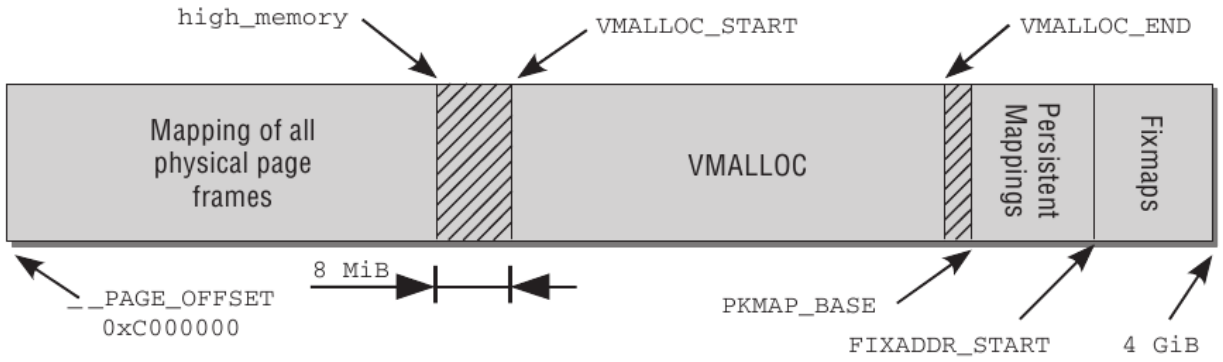
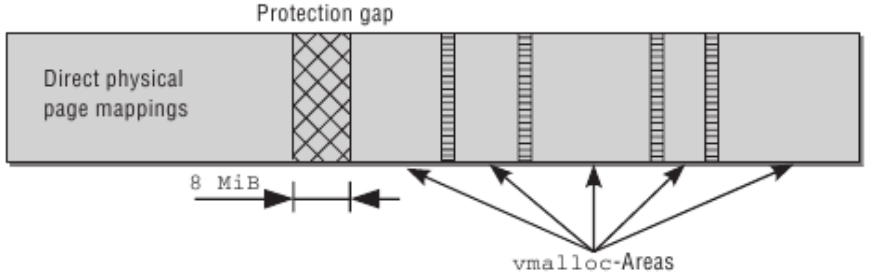
Figure 1.1 - kernel virtual memory map

Note: more during driver architecture



The figure shows the structure of the page table entries used to manage the fourth gigabyte of virtual address space. It indicates the *purpose* of each area of *virtual* address space, and this has nothing to do with the assignment of physical RAM.



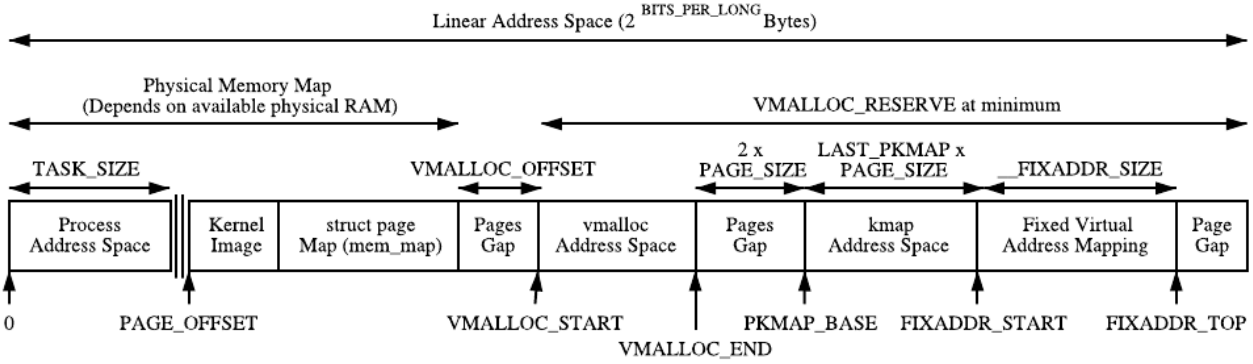
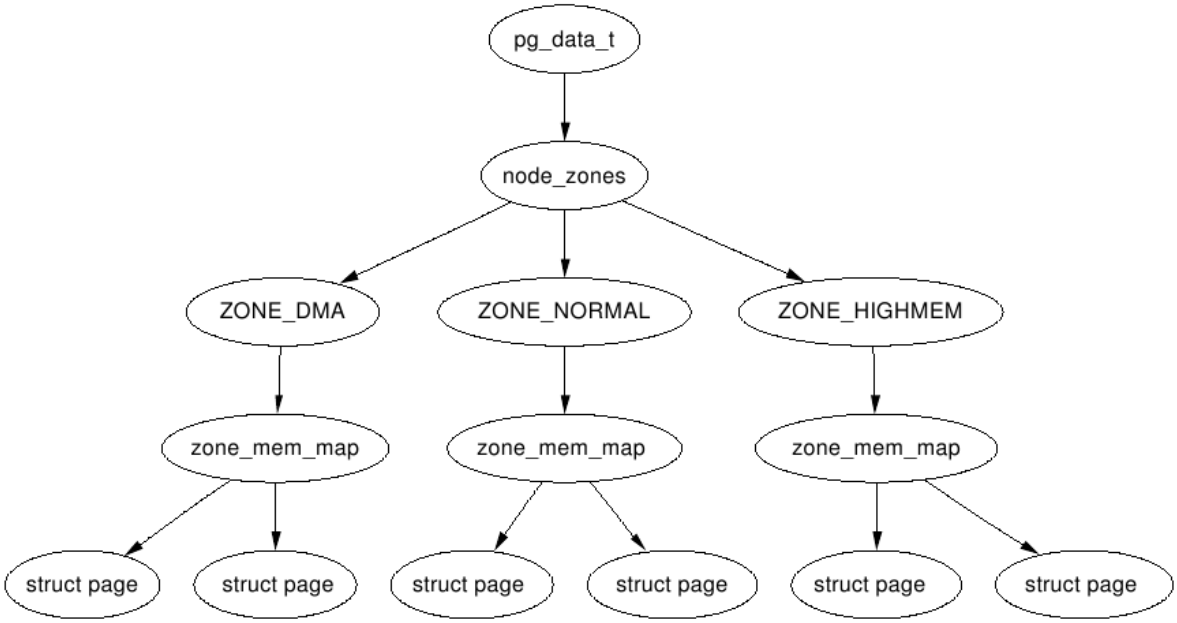


Figure 1.2 - page fault scenarios

Bit	Set (1)	Not set (0)
0	No page present in RAM	Protection fault (insufficient access permission)
1	Read access	Write access
2	Privileged kernel mode	User mode

Figure 1.3 - core page-frame management



ZONE_DMA **DMA-able pages** **< 16MB**

ZONE_NORMAL **Normally addressable pages 16–896MB**

ZONE_HIGHMEM **Dynamically mapped pages > 896MB**

Figure 1.4 – buddy allocator per zone

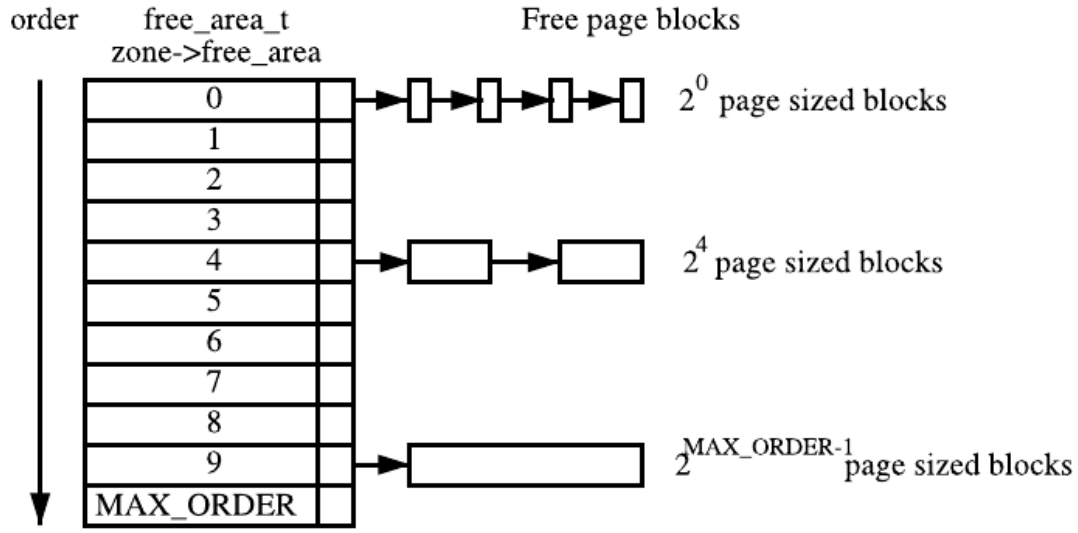


Figure 1.5 - buddy allocator in action

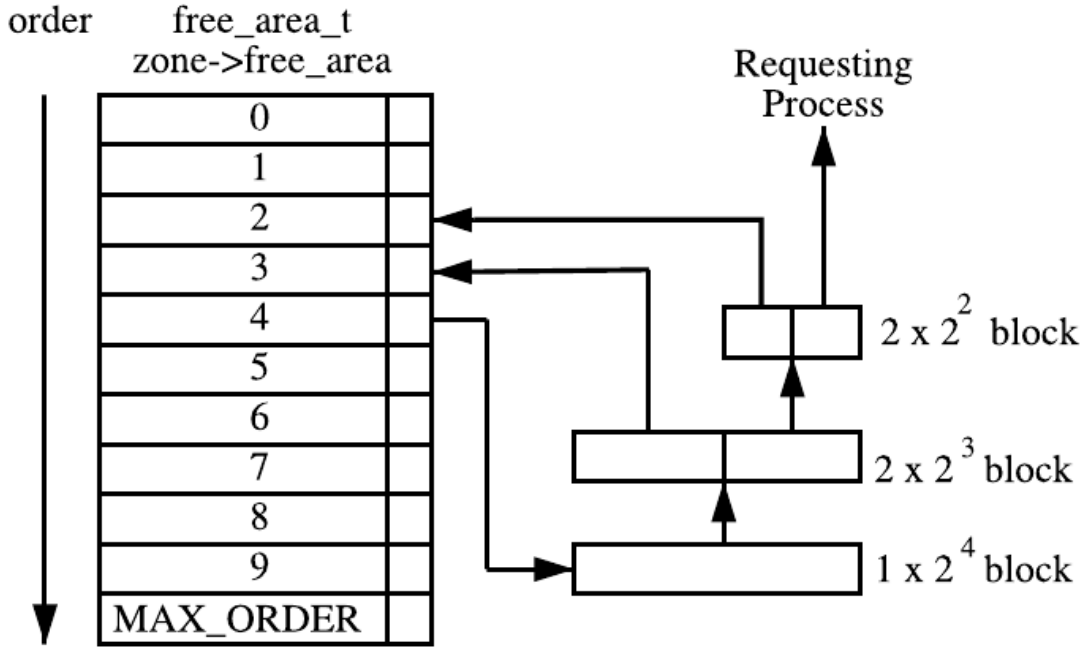


Figure 1.6 - buddy allocator system APIs

struct page * alloc_page(unsigned int gfp_mask)
Allocate a single page and return a struct address

struct page * alloc_pages(unsigned int gfp_mask, unsigned int order)
Allocate 2-power-of_order number of pages and returns a struct page

unsigned long get_free_page(unsigned int gfp_mask)
Allocate a single page, zero it and return a virtual address

unsigned long __get_free_page(unsigned int gfp_mask)
Allocate a single page and return a virtual address

unsigned long __get_free_pages(unsigned int gfp_mask, unsigned int order)
Allocate 2-power-of -order number of pages and return a virtual address

void __free_pages(struct page *page, unsigned int order)
Free an order number of pages from the given page

void __free_page(struct page *page)
Free a single page

void free_page(void *addr)
Free a page from the given virtual address

Figure 1.7 - buddy allocator flags

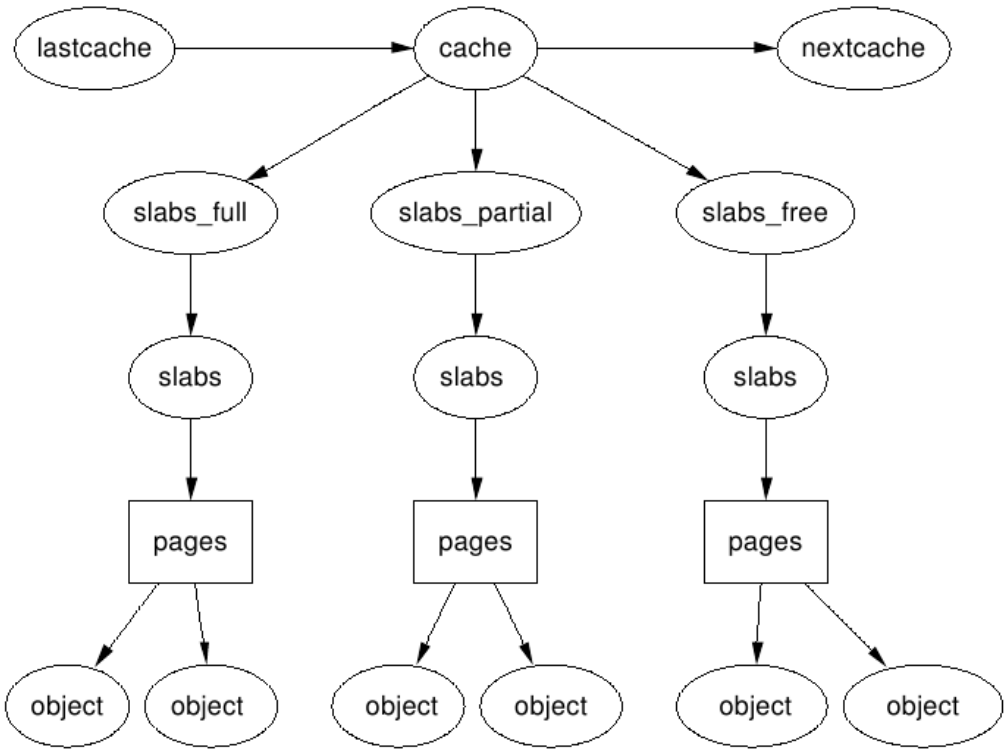
ZONE_DMA	**DMA-able pages**	**< 16MB**
ZONE_NORMAL	**Normally addressable pages**	**16–896MB**
ZONE_HIGHMEM	**Dynamically mapped pages**	**> 896MB**

GFP_KERNEL	**(__GFP_WAIT | __GFP_IO | __GFP_FS)**
GFP_DMA	**__GFP_DMA**
GFP_ATOMIC	**__GFP_HIGH**
	__GFP_HIGHMEM

Allocates from ZONE_HIGHMEM or ZONE_NORMAL

Note: more during device drivers

Figure 1.8 - slab allocator subsystem architecture



Note: we will see more during driver architecture.