

I DSP,Xilinx zynq FPGA,MCU 및  
Xilinx  
zynq FPGA 프로그래밍 전문가 과정

강사-INNOVA LEE(이상훈)

[Gccompil3r@gmail.com](mailto:Gccompil3r@gmail.com)

학생-윤지원

[Yoonjw789 @naver.com](mailto:Yoonjw789@naver.com)

가상 메모리의 역할:큰 용량을 커버 하기 위해 사용한다. 가상 메모리는 물리 메모리가 아니다(허상)

컴파일러는 무조건 가상 메모리에 데이터를 넣기만 하고 배치는 상관을 하지 않는다.물리 메모리에 넣는것은 exec\_struct 가 한다.이것은 나중에 구조를 까볼 것이다.

UMA 는 버스 한개 이기 때문에 CPU 의 정보가 한쪽으로 몰리고 결국 병목현상이 발생 하게 되고 뜨거워진다.그래서 발열이 심해지는것을 방지하기위해 CPU 를 몇개의 그룹으로 만들고 버스가 1 개가 더 생기기 때문에 CPU 의 데이터가 한쪽으로 쏠릴 수가 없기때문에 병목현상이 사라지고 발열이 UMA 보다 적어진다는 장점이 있는것이 NUMA 이다.

뱅크:접근 속도가 같은 메모리으 집합

UMA 뱅크:1 개

NUMA 뱅크:2 개이상

```
struct bootmem_data;
typedef struct pglist_data {
    struct zone node_zones[MAX_NR_ZONES];
    struct zonelist node_zonelists[MAX_ZONELISTS];
    int nr_zones;
#ifdef CONFIG_FLAT_NODE_MEM_MAP /* means !SPARSEMEM */
    struct page *node_mem_map;
#endif
#ifdef CONFIG_PAGE_EXTENSION
    struct page_ext *node_page_ext;
#endif
#endif
#ifdef CONFIG_NO_BOOTMEM
    struct bootmem_data *bdata;
#endif
#ifdef CONFIG_MEMORY_HOTPLUG
    /*
     * Must be held any time you expect node_start_pfn, node_present_pages
     * or node_spanned_pages stay constant. Holding this will also
     * guarantee that any pfn_valid() stays that way.
     *
     * pgdat_resize_lock() and pgdat_resize_unlock() are provided to
     * manipulate node_size_lock without checking for CONFIG_MEMORY_HOTPLUG.
     *
     * Nests above zone->lock and zone->span_seqlock
     */
    spinlock_t node_size_lock;
#endif
    unsigned long node_start_pfn;
    unsigned long node_present_pages; /* total number of physical pages */
    unsigned long node_spanned_pages; /* total size of physical page
                                         range, including holes */
}
640,1-8 52%
```

```

#ifdef CONFIG_NUMA_BALANCING
    /* Lock serializing the migrate rate limiting window */
    spinlock_t numabalancing_migrate_lock;

    /* Rate limiting time interval */
    unsigned long numabalancing_migrate_next_window;

    /* Number of pages migrated during the rate limiting time interval */
    unsigned long numabalancing_migrate_nr_pages;
#endif

#ifdef CONFIG_DEFERRED_STRUCT_PAGE_INIT
    /*
     * If memory initialisation on large machines is deferred then this
     * is the first PFN that needs to be initialised.
     */
    unsigned long first_deferred_pfn;
#endif /* CONFIG_DEFERRED_STRUCT_PAGE_INIT */
} pg_data_t;

```

UMA 를 관리하는 pglist\_data 의 구조이다.

unsigned long node\_present\_pages;;페이징의 갯수.

unsigned long node\_spanned\_pages;;최종 물리 메모리를 가리킨다.

int nr\_zones;;존이 몇개 있는지 확인

만약 물리 메모리의 할당 요청을 받게 되면, 되도록 할당을 요청한 데스크가 수행 되고 있는 cpu 와 가까운 노드에서 메모리를 할당하려 한다. 리눅스는 데스크 가 되도록 이전에 수행되었던 cpu 에서 다시 수행 되도록 하기 때문에 이러한 정책을 통해 보다 높은 성능을 얻을 수 있게 된다

->위에 말을 한마디로 하자면 캐쉬를 사용하겠다.

## Zone

노드 안에 존재하고

zone:cat /proc/zoneinfo,특정 메모리 영역.

Vi -t pglist\_data 에 zone 이 특정 영역을 관리한다.

zone\_dma:데이터양이 방대한 것들(비디오,영상정보,음성정보,네트워크)처리한다.

->오디오코덱을 만든다(알사에서 만든다)

zone\_normal:dma 에 포함하지 않은 영역.

zone\_highmem:가상 메모리를 페이징해서 물리 메모리와 연결하고 896MB 가 되면 가상 메모리와 물리 메모리가 간접 참조가 가능하다.

그럼 여기서 간접 참조는 무엇인가

간접 참조:한마디로 적은 비트수로 큰 메모리를 맵핑한다는 것이다.

가상 메모리 1GB 로 물리 메모리 64GB 를 1 대 1 로 맵핑 할 수 없기때문에 간접 참조를 이용하면 맵핑이 가능하다. 가상 메모리랑 물리 메모리가 맵핑이 될라면 무조건

페이징을 해야한다. 저번에 페이징을 할때 가상 메모리를 10bit 10bit 로 나눠서 2048 개로 모든것 표현했었다.

struct free\_area :현재 사용되는 페이지,사용 되지 않는 페이지,비어있는 페이지를 모두다 관리한다.

물리 메모리 최소단위:page frame

구조체 PAGE 는 PAGE FRAME 보다 용량이 작아야 한다. 만약 용량이 같다면 메모리의 용량이 부족하다.

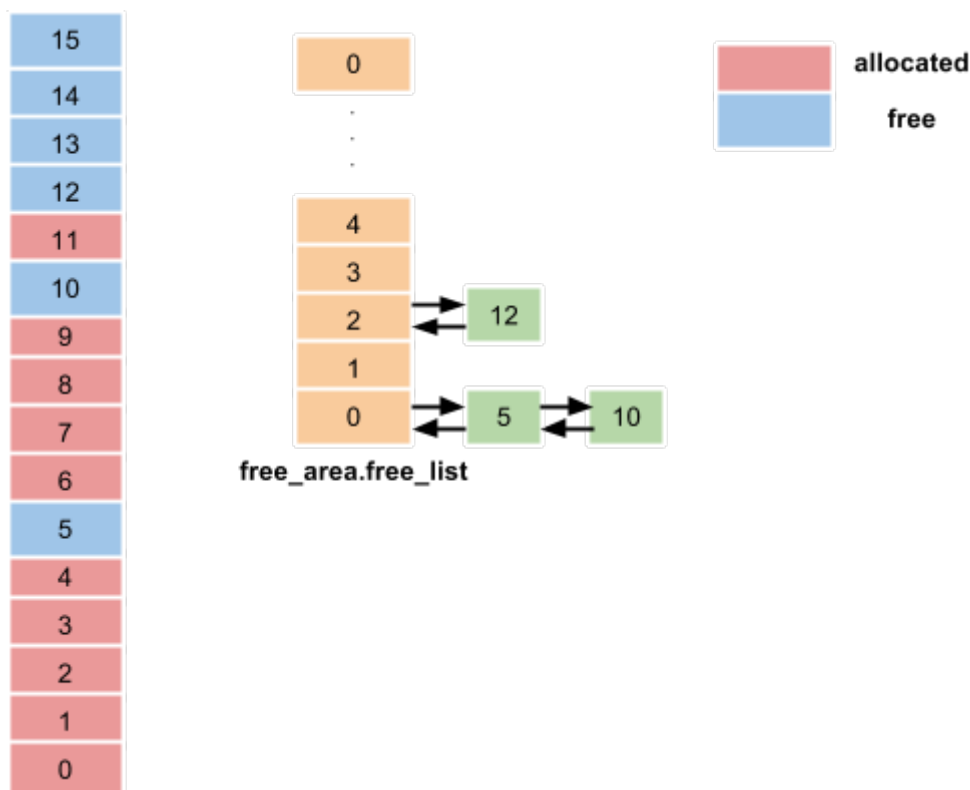
pglist\_data:node

버디=4kbyte \* 2 의 n 으로 동작을 한다 그래서 만약 10kb 를 요청하면 8kb 는 부족 하니까 2 의 2 승인 16kb 으로 할당을 해준다.

nr\_free:현재 free 한 page 가 몇개가 있나 확인.

버디:외부 단편화를 줄일 수 있다.

→ page 에 할당하는 속도가 빨라진다.



버디 할당자의 표이다. 오른쪽은 page 이며 빨간색은 채워져있는거고,파란색은 비어 있는거다. 버디는 4kb\*2 의 n 승으로 처리하기 때문에

처음 4kb 에서 보면 00 부터 4 까지는 다 채워져 있기때문에 0 이고,5 와 6 은 서로 다르기때문에 order 가 1 로 된다 page 각각은 4kb 이다. 그 다음 order2 는 8kb 로 채워 지기 때문에 0 부터 3 까지는 다 채워져 있으니 8kb 넣는것이 불가능 하고 4 부터 7 까지는 보면 6,7 은 다 채워져 있어서 불가능하고 4,5 를 보면 4 는 채워져 있고 5 는 비어

있지만 8kb 입장에선 채울 수가 없으니 불가능 상태이다 그래서 order 는 0 이다 12 부터 15 까지는 다 비어있으니 가능이런식으로 버디 할당자가 채워나간다.

**unsigned long vm\_pgoff; :offset**

**struct vm\_area\_struct \*vm\_next, \*vm\_prev;:연결 리스트**