VMCOREINFO

What is it?

VMCOREINFO is a special ELF note section. It contains various information from the kernel like structure size, page size, symbol values, field offsets, etc. These data are packed into an ELF note section and used by user-space tools like crash and makedumpfile to analyze a kernel's memory layout.

Common variables

init_uts_ns.name.release

The version of the Linux kernel. Used to find the corresponding source code from which the kernel has been built. For example, crash uses it to find the corresponding vmlinux in order to process vmcore.

PAGE_SIZE

The size of a page. It is the smallest unit of data used by the memory management facilities. It is usually 4096 bytes of size and a page is aligned on 4096 bytes. Used for computing page addresses.

init_uts_ns

The UTS namespace which is used to isolate two specific elements of the system that relate to the uname(2) system call. It is named after the data structure used to store information returned by the uname(2) system call.

User-space tools can get the kernel name, host name, kernel release number, kernel version, architecture name and OS type from it.

(uts_namespace, name)

Offset of the name's member. Crash Utility and Makedumpfile get the start address of the init_uts_ns.name from this.

node_online_map

An array node_states[N_ONLINE] which represents the set of online nodes in a system, one bit position per node number. Used to keep track of which nodes are in the system and online.

swapper_pg_dir

The global page directory pointer of the kernel. Used to translate virtual to physical addresses.

_stext

Defines the beginning of the text section. In general, _stext indicates the kernel start address. Used to convert a virtual address from the direct kernel map to a physical address.

vmap_area_list

Stores the virtual area list. makedumpfile gets the vmalloc start value from this variable and its value is necessary for vmalloc translation.

mem_map

Physical addresses are translated to struct pages by treating them as an index into the mem_map array. Right-shifting a physical address PAGE_SHIFT bits converts it into a page frame number which is an index into that mem_map array.

Used to map an address to the corresponding struct page.

contig_page_data

Makedumpfile gets the pglist_data structure from this symbol, which is used to describe the memory layout.

User-space tools use this to exclude free pages when dumping memory.

mem_section|(mem_section, NR_SECTION_ROOTS)|(mem_section, section_mem_map)

The address of the mem_section array, its length, structure size, and the section_mem_map offset.

It exists in the sparse memory mapping model, and it is also somewhat similar to the mem_map variable, both of them are used to translate an address.

MAX_PHYSMEM_BITS

Defines the maximum supported physical address space memory.

page

The size of a page structure. struct page is an important data structure and it is widely used to compute contiguous memory.

pglist_data

The size of a pglist_data structure. This value is used to check if the pglist_data structure is valid. It is also used for checking the

memory type.

zone

The size of a zone structure. This value is used to check if the zone structure has been found. It is also used for excluding free pages.

free_area

The size of a free_area structure. It indicates whether the free_area structure is valid or not. Useful when excluding free pages.

list_head

The size of a list_head structure. Used when iterating lists in a post-mortem analysis session.

nodemask_t

The size of a nodemask_t type. Used to compute the number of online nodes.

(page, flags|_refcount|mapping|lru|_mapcount|private|compound_dtor|compound_order|compound_head)

User-space tools compute their values based on the offset of these variables. The variables are used when excluding unnecessary pages.

(pglist_data, node_zones|nr_zones|node_mem_map|node_start_pfn|node_spanned_pages|node_id)

On NUMA machines, each NUMA node has a pg_data_t to describe its memory layout. On UMA machines there is a single pglist_data which describes the whole memory.

These values are used to check the memory type and to compute the virtual address for memory map.

(zone, free_area|vm_stat|spanned_pages)

Each node is divided into a number of blocks called zones which represent ranges within memory. A zone is described by a structure zone.

User-space tools compute required values based on the offset of these variables.

(free_area, free_list)

Offset of the free_list's member. This value is used to compute the number of free pages.

Each zone has a free_area structure array called free_area[MAX_ORDER]. The free_list represents a linked list of free page blocks.

(list_head, next|prev)

Offsets of the list_head's members. list_head is used to define a circular linked list. User-space tools need these in order to traverse lists.

(vmap_area, va_start|list)

Offsets of the vmap_area's members. They carry vmalloc-specific information. Makedumpfile gets the start address of the vmalloc region from this.

(zone.free_area, MAX_ORDER)

Free areas descriptor. User-space tools use this value to iterate the free_area ranges. MAX_ORDER is used by the zone buddy allocator.

prb

A pointer to the printk ringbuffer (struct printk_ringbuffer). This may be pointing to the static boot ringbuffer or the dynamically allocated ringbuffer, depending on when the the core dump occurred. Used by user-space tools to read the active kernel log buffer.

printk_rb_static

A pointer to the static boot printk ringbuffer. If @prb has a different value, this is useful for viewing the initial boot messages, which may have been overwritten in the dynamically allocated ringbuffer.

clear_seq

The sequence number of the printk() record after the last clear command. It indicates the first record after the last SYSLOG_ACTION_CLEAR, like issued by 'dmesg -c'. Used by user-space tools to dump a subset of the dmesg log.

printk_ringbuffer

The size of a printk_ringbuffer structure. This structure contains all information required for accessing the various components of the kernel log buffer.

(printk_ringbuffer, desc_ring|text_data_ring|dict_data_ring|fail)

Offsets for the various components of the printk ringbuffer. Used by user-space tools to view the kernel log buffer without requiring the declaration of the structure.

prb_desc_ring

The size of the prb_desc_ring structure. This structure contains information about the set of record descriptors.

(prb_desc_ring, count_bits|descs|head_id|tail_id)

Offsets for the fields describing the set of record descriptors. Used by user-space tools to be able to traverse the descriptors without requiring the declaration of the structure.

prb_desc

The size of the prb_desc structure. This structure contains information about a single record descriptor.

(prb_desc, info|state_var|text_blk_lpos|dict_blk_lpos)

Offsets for the fields describing a record descriptors. Used by user-space tools to be able to read descriptors without requiring the declaration of the structure.

prb_data_blk_lpos

The size of the prb_data_blk_lpos structure. This structure contains information about where the text or dictionary data (data block) is located within the respective data ring.

(prb_data_blk_lpos, begin|next)

Offsets for the fields describing the location of a data block. Used by user-space tools to be able to locate data blocks without requiring the declaration of the structure.

printk_info

The size of the printk_info structure. This structure contains all the meta-data for a record.

(printk_info, seq|ts_nsec|text_len|dict_len|caller_id)

Offsets for the fields providing the meta-data for a record. Used by user-space tools to be able to read the information without requiring the declaration of the structure.

prb_data_ring

The size of the prb_data_ring structure. This structure contains information about a set of data blocks.

(prb_data_ring, size_bits|data|head_lpos|tail_lpos)

Offsets for the fields describing a set of data blocks. Used by user-space tools to be able to access the data blocks without requiring the declaration of the structure.

atomic_long_t

The size of the atomic_long_t structure. Used by user-space tools to be able to copy the full structure, regardless of its architecture-specific implementation.

(atomic_long_t, counter)

Offset for the long value of an atomic_long_t variable. Used by user-space tools to access the long value without requiring the architecture-specific declaration.

(free_area.free_list, MIGRATE_TYPES)

The number of migrate types for pages. The free_list is described by the array. Used by tools to compute the number of free pages.

NR_FREE_PAGES

On linux-2.6.21 or later, the number of free pages is in vm_stat[NR_FREE_PAGES]. Used to get the number of free pages.

PG_lru|PG_private|PG_swapcache|PG_swapbacked|PG_slab|PG_hwpoision|PG_head_mask

Page attributes. These flags are used to filter various unnecessary for dumping pages.

PAGE_BUDDY_MAPCOUNT_VALUE(~PG_buddy)|PAGE_OFFLINE_MAPCOUNT_VALUE(~PG_offline)

More page attributes. These flags are used to filter various unnecessary for dumping pages.

HUGETLB_PAGE_DTOR

The HUGETLB_PAGE_DTOR flag denotes hugetlbfs pages. Makedumpfile excludes these pages.

x86_64

phys_base

Used to convert the virtual address of an exported kernel symbol to its corresponding physical address.

init_top_pgt

Used to walk through the whole page table and convert virtual addresses to physical addresses. The init_top_pgt is somewhat similar to swapper_pg_dir, but it is only used in x86_64.

pgtable_l5_enabled

User-space tools need to know whether the crash kernel was in 5-level paging mode.

node_data

This is a struct pglist_data array and stores all NUMA nodes information. Makedumpfile gets the pglist_data structure from it.

(node_data, MAX_NUMNODES)

The maximum number of nodes in system.

KERNELOFFSET

The kernel randomization offset. Used to compute the page offset. If KASLR is disabled, this value is zero.

KERNEL_IMAGE_SIZE

Currently unused by Makedumpfile. Used to compute the module virtual address by Crash.

sme_mask

AMD-specific with SME support: it indicates the secure memory encryption mask. Makedumpfile tools need to know whether the crash kernel was encrypted. If SME is enabled in the first kernel, the crash kernel's page table entries (pgd/pud/pmd/pte) contain the memory encryption mask. This is used to remove the SME mask and obtain the true physical address.

Currently, sme_mask stores the value of the C-bit position. If needed, additional SME-relevant info can be placed in that variable.

For example:

```
[ misc               ][ enc bit  ][ other misc SME info       ]
0000_0000_0000_0000_1000_0000_0000_0000_0000_0000_..._0000
63   59   55   51   47   43   39   35   31   27   ... 3
```

x86_32

X86_PAE

Denotes whether physical address extensions are enabled. It has the cost of a higher page table lookup overhead, and also consumes more page table space per process. Used to check whether PAE was enabled in the crash kernel when converting virtual addresses to physical addresses.

ia64

pgdat_list|(pgdat_list, MAX_NUMNODES)

pg_data_t array storing all NUMA nodes information. MAX_NUMNODES indicates the number of the nodes.

node_memblk|(node_memblk, NR_NODE_MEMBLKS)

List of node memory chunks. Filled when parsing the SRAT table to obtain information about memory nodes.
NR_NODE_MEMBLKS indicates the number of node memory chunks.

These values are used to compute the number of nodes the crashed kernel used.

node_memblk_s|(node_memblk_s, start_paddr)|(node_memblk_s, size)

The size of a struct node_memblk_s and the offsets of the node_memblk_s's members. Used to compute the number of nodes.

PGTABLE_3|PGTABLE_4

User-space tools need to know whether the crash kernel was in 3-level or 4-level paging mode. Used to distinguish the page table.

ARM64

VA_BITS

The maximum number of bits for virtual addresses. Used to compute the virtual memory ranges.

kimage_voffset

The offset between the kernel virtual and physical mappings. Used to translate virtual to physical addresses.

PHYS_OFFSET

Indicates the physical address of the start of memory. Similar to kimage_voffset, which is used to translate virtual to physical addresses.

KERNELOFFSET

The kernel randomization offset. Used to compute the page offset. If KASLR is disabled, this value is zero.

KERNELPACMASK

The mask to extract the Pointer Authentication Code from a kernel virtual address.

TCR_EL1.T1SZ

Indicates the size offset of the memory region addressed by TTBR1_EL1. The region size is 2^(64-T1SZ) bytes.

TTBR1_EL1 is the table base address register specified by ARMv8-A architecture which is used to lookup the page-tables for the Virtual addresses in the higher VA range (refer to ARMv8 ARM document for more details).

MODULES_VADDR|MODULES_END|VMALLOC_START|VMALLOC_END|VMEMMAP_START|VMEMMA

Used to get the correct ranges:

> MODULES_VADDR ~ MODULES_END-1 : Kernel module space. VMALLOC_START ~ VMALLOC_END-1 : vmalloc() / ioremap() space. VMEMMAP_START ~ VMEMMAP_END-1 : vmemmap region, used for struct page array.

arm

ARM_LPAE

It indicates whether the crash kernel supports large physical address extensions. Used to translate virtual to physical addresses.

s390

lowcore_ptr

An array with a pointer to the lowcore of every CPU. Used to print the psw and all registers information.

high_memory

Used to get the vmalloc_start address from the high_memory symbol.

(lowcore_ptr, NR_CPUS)

The maximum number of CPUs.

powerpc

node_data|(node_data, MAX_NUMNODES)

See above.

contig_page_data

See above.

vmemmap_list

The vmemmap_list maintains the entire vmemmap physical mapping. Used to get vmemmap list count and populated vmemmap regions info. If the vmemmap address translation information is stored in the crash kernel, it is used to translate vmemmap kernel virtual addresses.

mmu_vmemmap_psize

The size of a page. Used to translate virtual to physical addresses.

mmu_psize_defs

Page size definitions, i.e. 4k, 64k, or 16M.

Used to make vtop translations.

vmemmap_backing|(vmemmap_backing, list)|(vmemmap_backing, phys)|(vmemmap_backing, virt_addr)

The vmemmap virtual address space management does not have a traditional page table to track which virtual struct pages are backed by a physical mapping. The virtual to physical mappings are tracked in a simple linked list format.

User-space tools need to know the offset of list, phys and virt_addr when computing the count of vmemmap regions.

mmu_psize_def|(mmu_psize_def, shift)

The size of a struct mmu_psize_def and the offset of mmu_psize_def's member.

Used in vtop translations.

sh

node_data|(node_data, MAX_NUMNODES)

See above.

X2TLB

Indicates whether the crashed kernel enabled SH extended mode.