Linux Memory Management

Thadeu Lima de Souza Cascardo

August 29, 2025

Agenda

- Virtual Memory
- Linux memory space
- Zones and nodes
- kmalloc and slabs
- get_free_pages and page order
- GFP flags and memory reserves
- page cache and swap
- kswapd and memory reclaim
- LRUs

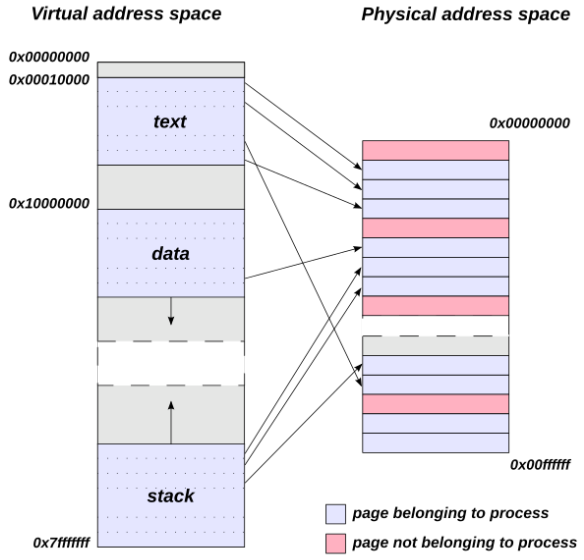
igalia

Virtual memory

- Processor addresses memory by physical address
- Software may use a virtual address instead: memory is “segmented” in pages
- OS as a virtual machine: programs may see a single contiguous address space
- OS as a resource manager: memory may not even be available and be “paged”

igalia

Virtual memory



igalia

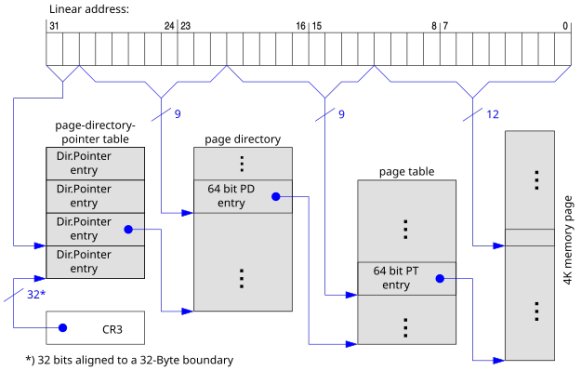
Copyright

igalia

Linux memory space

- Kernel has a contiguous visibility of the physical memory, in what is called a linear mapping
- Other kernel mappings may require large virtual contiguous memory, which use vmalloc
- User space mappings are not contiguous in physical memory, but there may be large (huge) pages

Page tables



Attribution: RokerHRO

License:

https://creativecommons.org/licenses/by-sa/3.0/deed.en

igalia

Zones and nodes

- Non-Uniform Memory Access
  - In multiple socket systems, different sockets may have different access times to a given memory “node”
- Memory is also split in zones, mostly to support DMA and High Memory
- High Memory was typical in the past on 32-bit systems when more than (or even close to) 4GiB of memory meant physical addresses past the 32-bit addressing space.

igalia

kmalloc and slabs

- Though not the focus of this presentation, let's do a quick review
- slabs allow fixed-size pieces carved out from pages to be allocated, reducing internal fragmentation
- Linux SLUB implementation works scalably as it uses per-cpu structures
- kmalloc is built around the next order-2 sized slabs
- When there are no free slabs, new pages are allocated

igalia

get_free_pages and page order

- get_free_pages is the main memory allocation API
- pages are allocated as order-2 multiples of pages
- buddy system

igalia

GFP flags and memory reserve

- GFP flags prefix comes from get_free_pages
- GFP_ATOMIC allows memory reserves to be used. It respects that the code may not sleep/schedule. It does not reclaim (explained later).
- GFP_KERNEL respects memory reserves and will trigger reclaim, potentially scheduling and waiting for IO.
- GFP_NOIO and GFP_NOFS can be used to avoid recursing into the filesystem and IO paths.
- __GFP_KSWAPD_RECLAIM and __GFP_DIRECT_RECLAIM can be masked off

igalia

Memory reserves

- min_free_kbytes sysctl
- System tries not to go below a minimal
- A low threshold is set, when reclaim is started
- A high threshold is set, when reclaim should stop

More GFP flags

- __GFP_HIGH
- __GFP_NORETRY
- __GFP_NOMEMALLOC
- __GFP_RETRY_MAYFAIL
- __GFP_NOFAIL
- __GFP_NOWARN

igalia

Page cache and swap

- User space memory is either mapped from files or anonymous (no files)
- Some files (from tmpfs or other shared memory) are actually anonymous
- When a real filesystem is used, with backing storage (though that could be RAM-based), file data may be put into what is called the page cache
- Memory with no backing filesystem can only go to swap space

kswapd and memory reclaim

- When the system is under memory pressure, that is, below its low threshold, it must reclaim memory.
- Memory may be reclaimed from slabs
- But mostly, they are reclaimed from page cache and anonymous memory.
- If file data is clean, pages can be released. If it is dirty, it must be written back and system must wait for IO
- Anonymous pages may have been written to swap before and still be clean
- Otherwise, anonymous pages go to swap, if available
- Pages may be mlocked

LRUs

- Linux classic LRU system has two levels: active and inactive
- Pages in the inactive list can be reclaimed
- Pages in the active list can be moved to the inactive list

A little bit of history

- From Linux 2.0 to 2.4.9, pages had an age and were promoted and demoted according to their access.
- This was simplified and pages are not aged anymore, but simply move from/to the active/inactive list since then.
- Processes page tables were also scanned until reverse-mappings were added around 2002.

igalia

MGLRU

- Multi generational LRU: pages are promoted or demoted to younger or older generations. Older generations are equivalent to the inactive list and may be reclaimed.
- Page tables are scanned as the rmaps were taking too long. It uses a bloom filter and feedback from rmaps.

Discussion

Join us!

https://www.igalia.com/jobs

igalia