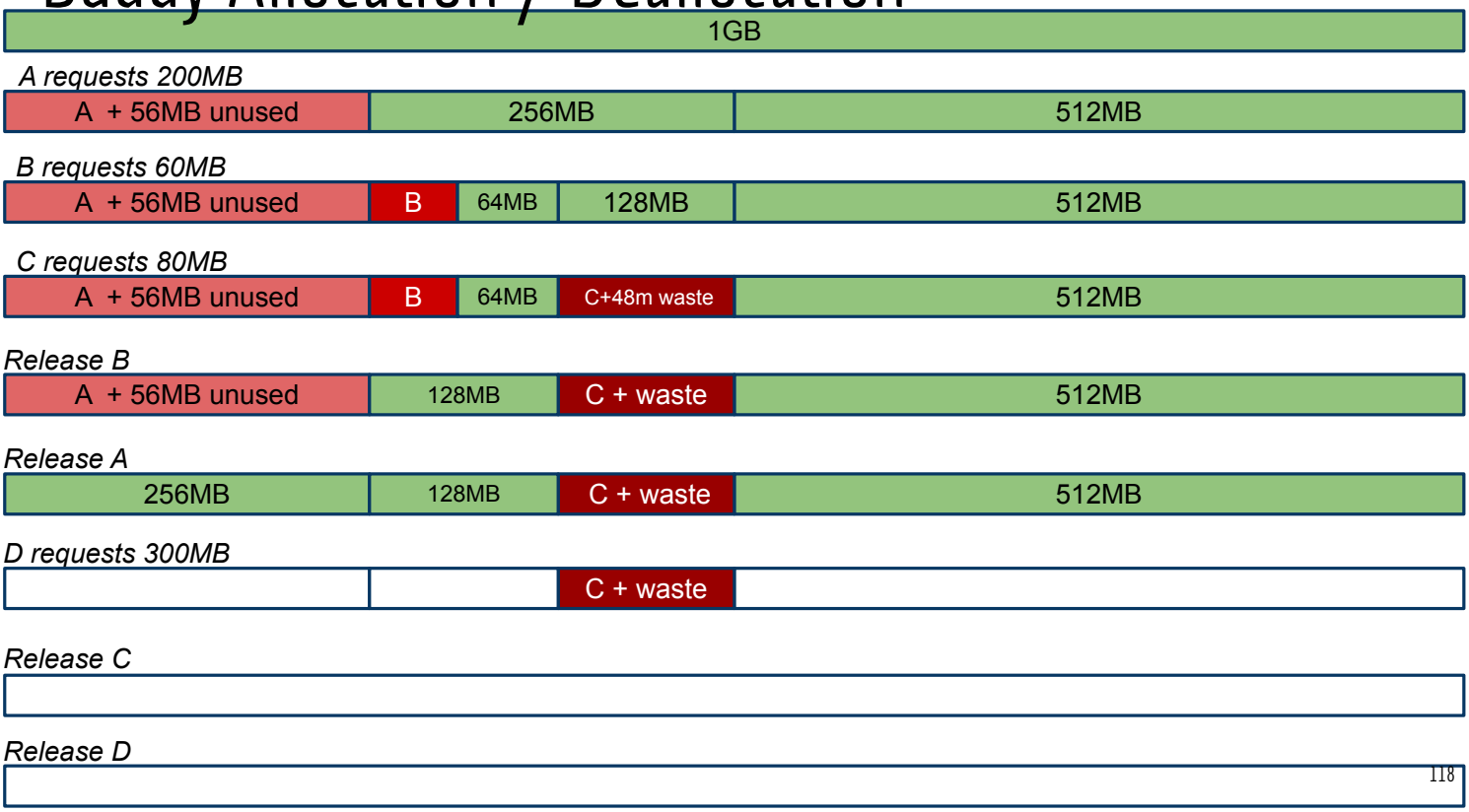
Memory Allocation for Kernel

- Kernel memory must be allocated from a separate pool, different from the pool used user processes
- Most kernel data structures must stay **resident** in RAM all the time (should not be paged out)
- Most kernel data structures must be allocated **contiguously** in RAM (must be accessed without going through the virtual memory interface)
  - Pages = frames

Strategies for Kernel Mem Allocation

- Buddy System (power-of-two allocator)
  - Always allocate to the nearest (larger) **power of two** memory size
    - Request 15 MB => Allocate 16MB
    - Request 17MB => Allocate 32MB (too much wasted space)
    - May suffer large internal fragmentation
- Slab Allocation (better than Buddy)

Buddy Allocation / Deallocation



Slab Allocation

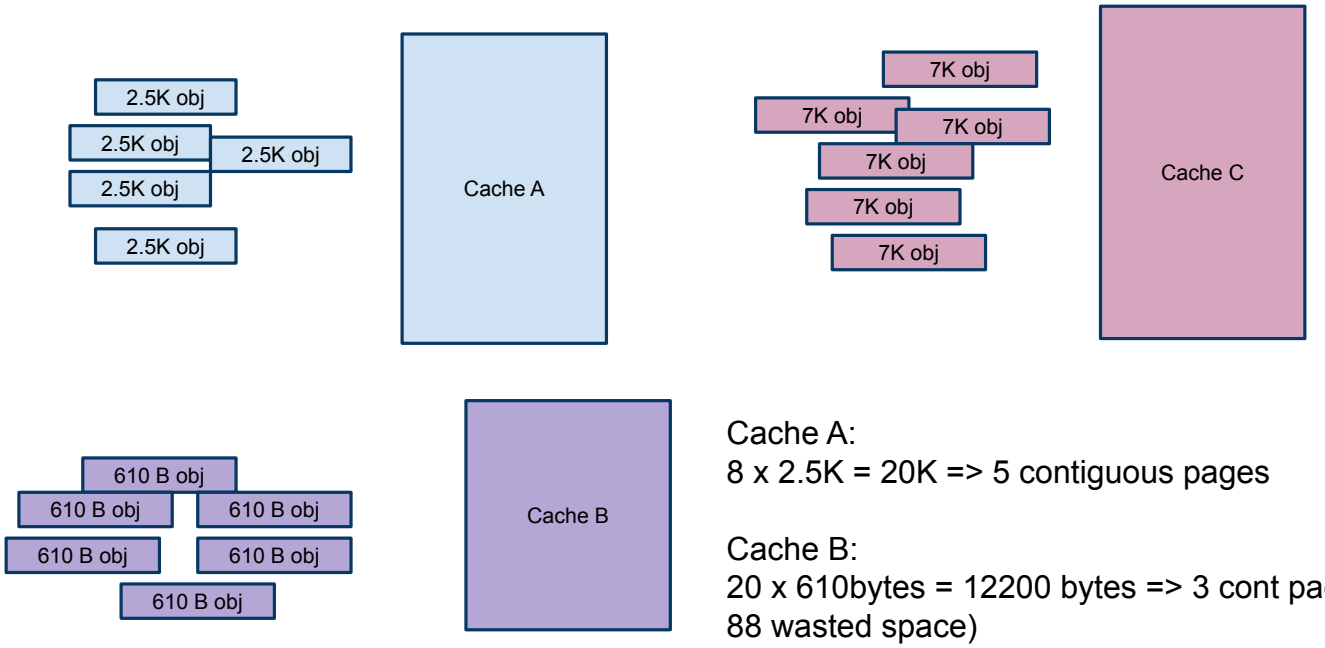
- One slab = N physically contiguous pages/frames
- One cache = M slabs
- One slab holds K kernel objects (kernel data structures) of **the same type**
  - Caches for file descriptors, caches for semaphores, caches for process X, …
  - Number of KOs per slab is determined from the size of KO: multiple of page size and minimizes wasted space
- Allocations/deallocation are handled in units of slab
  - One “slab_alloc” call pre-allocates K kernel objects
  - Used by Linux, FreeBSD

Slab States

- Full: all of the kernel objects within the slab are assigned,
- Empty: all of the kernel objects are free
- Partial: some free, some assigned

Slab Allocation: One Cache per Kernel Object Type

Assume page size 4K

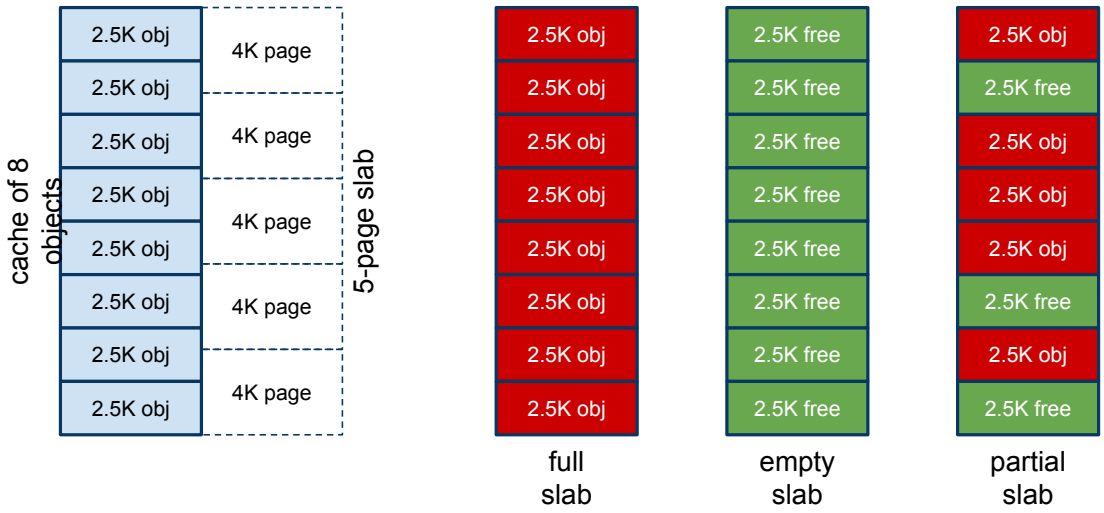


Cache A:
8 x 2.5K = 20K => 5 contiguous pages

Cache B:
20 x 610bytes = 12200 bytes => 3 cont pages (and 88 wasted space)

Cache C:
4 x 7K = 28K => 7 contiguous pages

Slab Allocation into Contiguous Pages



8 x 2.5K = 20K = 5 x 4K

For the above example: cache grows/shrinks 5 pages at a time

OS Examples

Linux Page Replacement

- LRU 2Q (LRU clock approximation with two queues)
  - Inactive_list: pages which are not referenced (ref-bit ZERO)
  - Active_list: pages which are referenced (ref-bit ONE)
- Working Set
- Global replacement strategy

Windows Page Replacement

- Working Set
- LRU with local replacement strategy