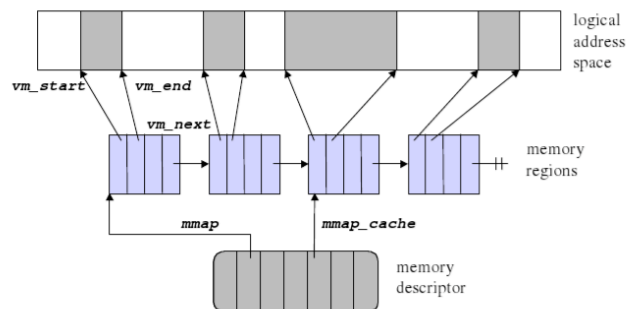


## Dynamic Memory Management

- Allocating memory: The Interface
  - Buddy System
  - Slab Allocation
- 
- *Reading: Doeppner, 3.3*

## Memory Areas

Memory areas (regions) are intervals of legal addresses.



## Allocation at Different Levels

---

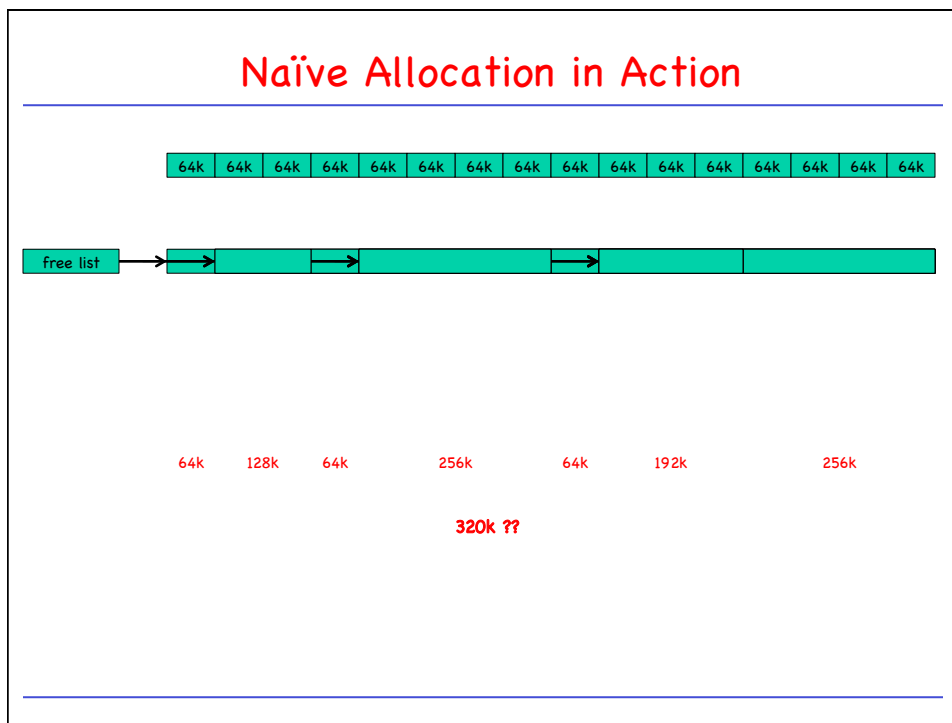
- `alloc_pages()` and `__get_free_pages()`
    - allocate **pages** or **frames**, at low level
    - useful to allocate contiguous pages/frames.
  - **byte-sized** allocations:
    - `kmalloc(size, gfp_mask)`
      - allocate physically contiguous sequence of bytes
    - `vmalloc(size, gfp_mask)`
      - allocate virtually contiguous sequence of bytes
  - explicit **user-level** allocation:
    - `malloc(size)`
      - allocate virtually contiguous sequence of bytes at user level
- 

## How does this all work?

---

- `alloc_pages()` and `__get_free_pages()`
    - allocate pages, at low level
    - useful to allocate contiguous pages/frames.
  - **byte-sized** allocations:
    - `kmalloc(size, gfp_mask)`
      - allocate physically contiguous sequence of bytes
    - `vmalloc(size, gfp_mask)`
      - allocate virtually contiguous sequence of bytes
  - explicit user-level allocation:
    - `malloc(size)`
      - allocate virtually contiguous sequence of bytes at user level
- Buddy System!**
- Slab Allocator (+ caching)**
-

## Naïve Allocation in Action



## Buddy System Allocation



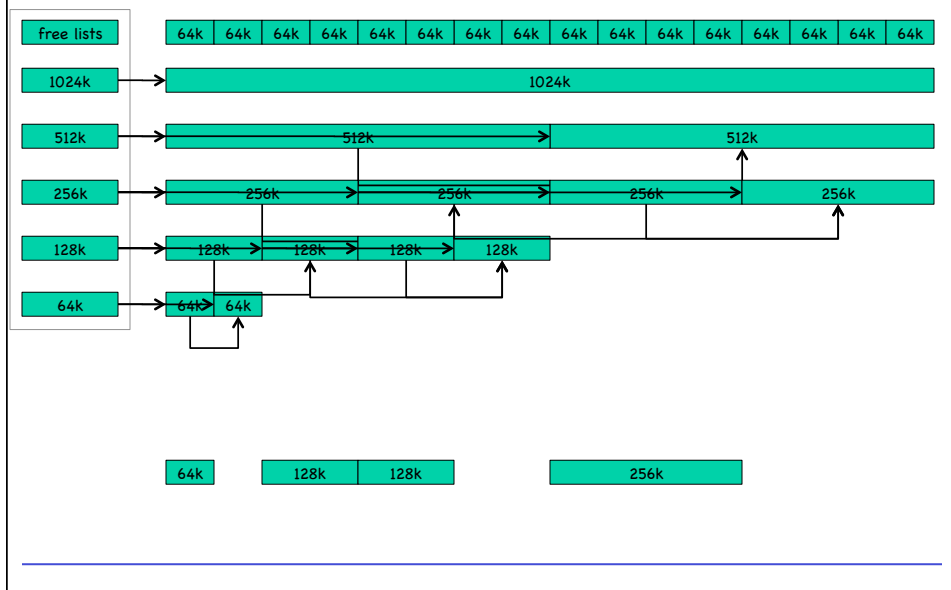
Harry Markowitz  
1927-  
1990 Nobel Memorial  
Prize in Economics

- Allocation:
  - Increase size of request to next power of  $2^*$ .
  - Look up block in free lists.
  - If exists, allocate.
  - If none exists, split next larger block in half, put first half (the "buddy") on free list, and return second half.
- De-Allocation:
  - Return segment to free list.
  - Check if buddy is free. If so, coalesce.
- For details, see lecture.

(\* ) For case of binary buddy system.

References: Donald Knuth: The Art of Computer Programming Volume 1: Fundamental Algorithms. Second Edition (Reading, Massachusetts: Addison-Wesley, 1997), pp. 435-455. ISBN 0-201-89683-4

## Buddy System in Action



## Slab Allocation

- First described by Jeff Bonwick for the SunOS kernel.
- Currently used in Linux and other kernels.
- Key observations:
  - Kernel memory often used for allocated for a **finite set of objects**, such as file descriptors and other common structures.
  - Amount of time required to initialize a regular object in the kernel exceeds the amount of time required to allocate and de-allocate it.
- Conclusion:
  - Instead of freeing the memory back to a global pool, have the memory remain initialized for its intended purpose.
- References: "The Slab Allocator: An Object-Caching Kernel Memory Allocator (1994)"

## Slab Allocation (II)

- Set of objects pre-allocated
- Marked as free
- When needed, assign a free one and mark as used
- No free ones available?
  - allocate a new slab
  - slab states (full, empty, partial)
  - fill partial slab first
- Advantages:
  - no fragmentation
  - memory requests satisfied quickly

