Xv6 Book, Chapter 1. KERNBASE limits the amount of memory a single process can use, which might be irritating on a machine with a full 4 GB of RAM. Would raising KERNBASE allow a process to use more memory?
Xv6: physical page allocator

Protected Mode

CS : 0x8        EIP: main
SS : 0x10       ESP: stack
GDT: 0x7c78     TSS: 0x0
IDT: 0x0        CR3: entrypgdir

Virtual

kmem.freelist
end (0x801126fc)

2GB + 4MB
0x80400000

0x80000000
2GB
Physical memory
We need a smaller array to describe physical pages, e.g., `mem_map[]` in Linux.
Memory allocation
Simplest memory allocator

- Bitmap of all pages
  - Bootmem allocator in Linux
- Allocation searches for an unused page
  - Multiple sub-page allocations can be served from the same page by advancing a pointer

- Works ok, but what is the problem?
Boot memory allocator

- Bitmap of all pages
  - Bootmem allocator in Linux
- Allocation searches for an unused page
  - Multiple sub-page allocations can be served from the same page by advancing a pointer

- Works ok, but what is the problem?
  - Linear scan of the bitmap
    - Too long
Buddy:
Physical Memory Allocator
Buddy memory allocator
Buddy allocator
What's wrong with buddy?
What's wrong with buddy?

- Buddy allocator is ok for large allocations
  - E.g. 1 page or more
- But what about small allocations?
  - Buddy uses the whole page for a 4 bytes allocation
    - Wasteful
  - Buddy is still slow for short-lived objects
Slab:
Allocator for object of a fixed size
Slab

- A 2 page slab with 6 objects
Keeping track of free objects

- `kmem_bufctl` array is effectively a linked list

- First free object: 3
- Next free object: 1
A cache is formed out of slabs
Slab is fine, but what's wrong?
Slab is fine, but what's wrong?

• We can only allocate objects of one size
Kmalloc(): variable size objects

- A table of caches
  - Size: 32, 64, 128, etc.
NUMA
Non-uniform memory access
Uniform and non-uniform memory access

● Parts of memory can be faster than others
Uniform memory access (UMA)
Nonuniform memory access (NUMA)
Nodes

- Attempt to allocate memory from the current node
  - Fall back to the next node in list
    - If ran out of local memory

```
Node1
struct pgliist_data

node_next → node_next → node_next
```
Nodes

Node1

struct pglist_data

node_next

node_start_pfn

Node2

node_next

node_start_pfn

Node3

node_next

node_start_pfn

Physical memory
Linux memory management
Linux memory management
Thank you!
Zones

32bit

16MB 896MB 6GB

- ISA devices can DMA only in 0 - 16MB
- Normal zone
- High memory (above 896MB on 32 bit machines)

64bit

16MB 4GB 6GB

- ISA devices can DMA only in 0 - 16MB
- 32bit devices can DMA into 0 - 4GB
- Normal zone
struct pglst_data

zones

dnode_next

zones

dnode_next

zones

dnode_next

zones

ZONE_DMA
ZONE_NORMAL
ZONE_HIGMEM
ZONE_HIGMEM
ZONE_HIGMEM