CS 240 ASSIGNMENT 2:

Bitmap Implementation of XV6's Kernel Memory Allocator

NAME: Xiaopeng Xu KAUST ID: 129052

Goal:

Implement a bitmap as kernel's memory allocator to keep track of free pages.

Results:

Modified *kalloc.c* file, implemented the bitmap memory allocator, and passed `usertests`.

```
$ usertests
usertests starting
bigarg test
...
exec test
ALL TESTS PASSED
```

Methods: (All modifications are in *kalloc.c* file)

1. Define physical memory size `QMPHYMEM` and bitmap size `BITMAPSIZE`. `QMPHYMEM` is the physical memory available for qemu virtual machine. `BITMAPSIZE` equals to number of page frames divided by 8.
2. Modify `freelist` to `bitmap` pointer in `kmem` struct.
3. Locate `bitmap` after `end` by `kmem.bitmap = end`, and initiate all bitmap bits to 1 in `kinit1` function.
4. In kinit1, modify the `freerange` arguments to not include the bitmap area in freeing pages.
5. Change `kfree` function and use `bitmap` bit operations to record the freed pages through:
   ```
   kmem.bitmap[byteId] &= ~(1<<(7-bitId));
   ```
   If a page is used the corresponding bit is 1, 0 otherwise.
6. Change `kalloc` function to search the bitmap and find empty pages. Once a page is found (bit equals 0 with certain `byteId` and `bitId`), change the corresponding bit to 1, and return the virtual address through:
   ```
   kmem.bitmap[byteId] |= (1<<(7-bitId));
   r = p2v((uint)(byteId*8 +bitId)*PGSIZE);
   ```
7. After previous modifications, run `usertests` in qemu virtual machine.