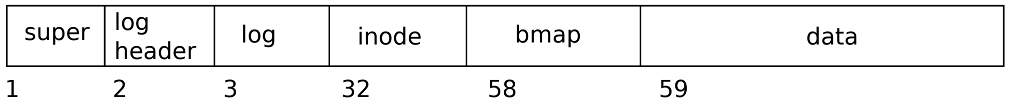
Principles of Operating Systems **Name (Print):** ______________________
Winter 2017
Final
03/22/2017
Time Limit: 8:00am - 10:00am

- **Don't forget to write your name on this exam.**
- **This is an open book, open notes exam. But no online or in-class chatting.**
- **Ask me if something is not clear in the questions.**
- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by explanation will receive no credit; an incorrect answer supported by substantially correct explanations might still receive partial credit.
- If you need more space, use the back of the pages; clearly indicate when you have done this.

Problem	Points	Score
1	20	
2	10	
3	20	
4	20	
5	15	
6	15	
7	5	
Total:	105	

1. File system

   Xv6 lays out the file system on disk as follows:



   Block 1 contains the super block. Blocks 2 through 31 contain the log header and the log. Blocks 32 through 57 contain inodes. Block 58 contains the bitmap of free blocks. Blocks 59 through the end of the disk contain data blocks.

   Ben modifies the function bwrite in bio.c to print the block number of each block written.

   Ben boots xv6 with a fresh fs.img and types in the command ln cat cat2. This command creates a symbolic link cat2 to file cat. This command produces the following trace:

```
$ ln cat cat2
write 3
write 4
write 2
write 32
write 59
write 2
$
```

   (a) (5 points) Why is block 2 written twice?

   (b) (5 points) Briefly explain what block 32 contains in the above trace. Why is it written?

(c) (5 points) What does block 3 contain?

(d) (5 points) If writes to 32 and 59 are reordered like below, will it violate correctness of the file system, explain why?

```
$ ln cat cat2
write 3
write 4
write 2
write 59
write 32
write 2
$
```

2. Memory management.

   (a) (5 points) Explain organization of the xv6 memory allocator.

   (b) (5 points) Why do you think xv6 does not have buddy or slab allocators? Under what conditions you would have to add these allocators to the xv6 kernel?

3. Synchronization

   (a) (10 points) The code below is the xv6's sleep() function. Remember the whole idea of passing a lock inside sleep() is to make sure it is released before the process goes to sleep (otherwise it will never be woken up). However, it looks like that if the lock passed inside sleep is ptable.lock (i.e., lk == &ptable.lock) the lock remains acquired and is never released. But xv6 does call sleep with ptable.lock as an argument and it works, can you explain why?

```
// Atomically release lock and sleep on chan.
// Reacquires lock when awakened.
void
sleep(void *chan, struct spinlock *lk)
{
  if(proc == 0)
    panic("sleep");

  if(lk == 0)
    panic("sleep without lk");

  // Must acquire ptable.lock in order to 2818   // change p>state and
  // change p>state and then call sched.
  // Once we hold ptable.lock, we can be
  // guaranteed that we wont miss any wakeup
  // (wakeup runs with ptable.lock locked),
  // so its okay to release lk.
  if(lk != &ptable.lock){
    acquire(&ptable.lock);
    release(lk);
  }

  // Go to sleep.
  proc>chan = chan;
  proc>state = SLEEPING;
  sched();

  // Tidy up.
  proc>chan = 0;

  // Reacquire original lock.
  if(lk != &ptable.lock){
    release(&ptable.lock);
    acquire(lk);
  }
}
```

(b) (10 points) Alyssa runs xv6 on a machine with 8 processors and 8 processes. Each process calls uptime() (3738) system call continuously, reading the number of ticks passed since boot. Alyssa measures the number of uptime() system calls per second and notices that 8 processes achieve the same total throughput as 1 process, even though each process runs on a different processor. Why is the throughput of 8 processes the same as that of 1 process?

4. Scheduling

   (a) (10 points) You would like to extend xv6 with priority based scheduler, i.e., each process has a priority, and processes with a higher priority are scheduled first. Write the code for your implementation below (which xv6 functions need to be changed?)

(b) (10 points) Now you would like to extend your priority scheduler with support for interactive tasks, e.g., a task that spends a lot of time waiting, should run first (i.e., receive priority boost). Provide code that handles waiting tasks and implements priority boost (again, just change related xv6 functions).

5. Page tables.

   Xv6 uses 4MB page table during boot. It is defined as:

```
// The boot page table used in entry.S and entryother.S.
// Page directories (and page tables) must start on page boundaries,
// hence the __aligned__ attribute.
// PTE_PS in a page directory entry enables 4Mbyte pages.

__attribute__((__aligned__(PGSIZE)))
pde_t entrypgdir[NPDENTRIES] = {
  // Map VAs [0, 4MB) to PAs [0, 4MB)
  [0] = (0) | PTE_P | PTE_W | PTE_PS,
  // Map VAs [KERNBASE, KERNBASE+4MB) to PAs [0, 4MB)
  [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,
};
```

   (a) (5 points) What virtual addresses (and to what physical addresses) does this page table map?

   (b) (10 points) Imagine now that 4MB pages are not available, and you have to use regular 4KB pages. How do you need to change the definition of entrypgdir for xv6 to work correctly (provide code and short explanation).

6. Process creation

   While editing the xv6 code, Jimmy accidentally erases the below section of code under fork() function on proc.c

```
for(i = 0; i < NOFILE; i++)
    if(proc>ofile[i])
        np>ofile[i] = filedup(proc>ofile[i]);
```

   (a) (5 points) Explain what the above section of code does?

   (b) (10 points) Explain where things can go wrong without this code. Quote a concrete example.

7. ics143A. I would like to hear your opinions about 6.828, so please answer the following questions. (Any answer, except no answer, will receive full credit.)

   (a) (1 point) Grade ics143A on a scale of 0 (worst) to 10 (best)?

   (b) (2 points) Any suggestions for how to improve ics143A?

   (c) (1 point) What is the best aspect of ics143A?

   (d) (1 point) What is the worst aspect of ics143A?