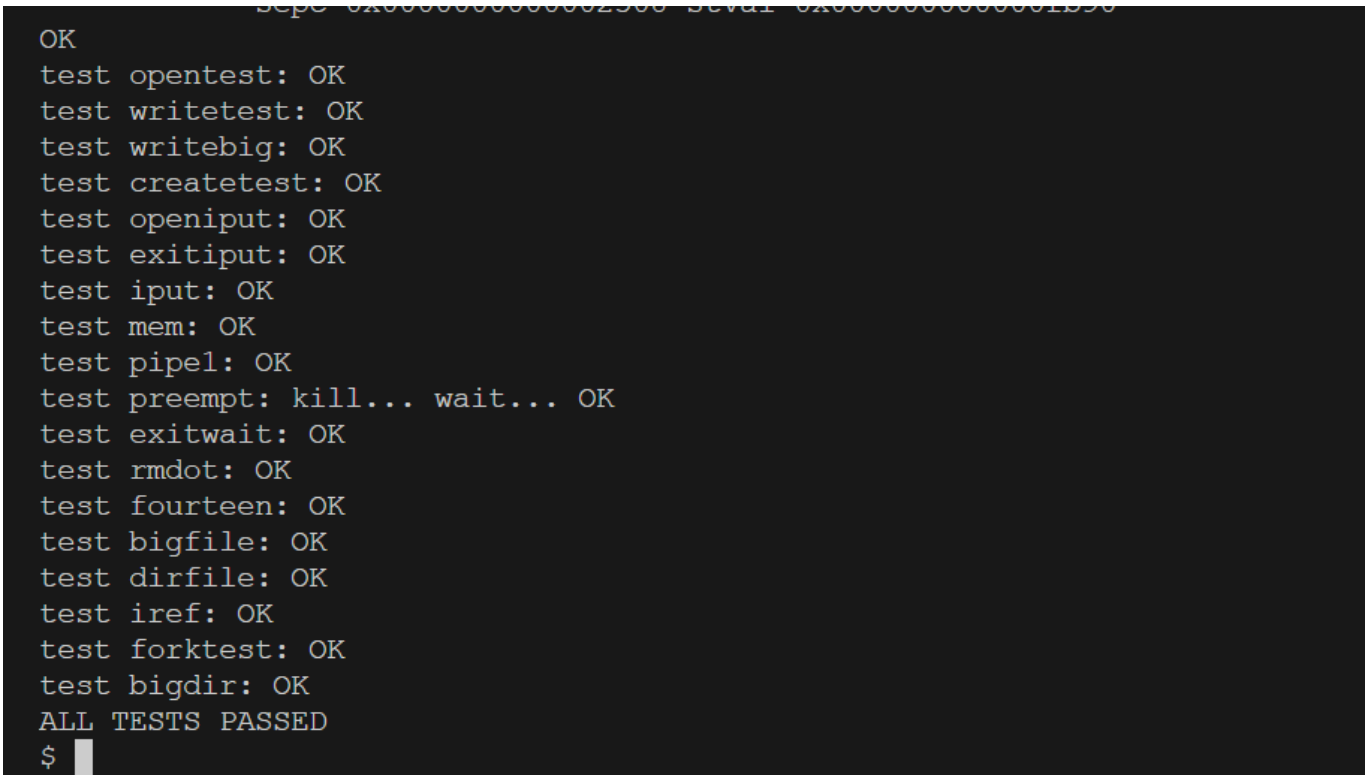
OK
test opentest: OK
test writetest: OK
test writebig: OK
test createtest: OK
test openiput: OK
test exitiput: OK
test iput: OK
test mem: OK
test pipe1: OK
test preempt: kill... wait... OK
test exitwait: OK
test rmdot: OK
test fourteen: OK
test bigfile: OK
test dirfile: OK
test iref: OK
test forktest: OK
test bigdir: OK
ALL TESTS PASSED
$

```
root@2cea5af01129:~/github/fudan-os/xv6-oslab24# make qemu
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/kalloc.o kernel/kalloc.c
riscv64-unknown-elf-ld -z max-page-size=4096 -T kernel/kernel.ld -o kernel/kernel kernel/entry.o kernel/start.o kernel/console.o
 kernel/printf.o kernel/uart.o kernel/kalloc.o kernel/spinlock.o kernel/string.o kernel/main.o kernel/vm.o kernel/proc.o kernel/
swtch.o kernel/trampoline.o kernel/trap.o kernel/syscall.o kernel/sysproc.o kernel/bio.o kernel/fs.o kernel/log.o kernel/sleeplo
ck.o kernel/file.o kernel/pipe.o kernel/exec.o kernel/sysfile.o kernel/kernelvec.o kernel/plic.o kernel/virtio_disk.o kernel/sta
ts.o kernel/sprintf.o
riscv64-unknown-elf-ld: warning: cannot find entry symbol _entry; defaulting to 0000000080000000
riscv64-unknown-elf-ld: warning: kernel/kernel has a LOAD segment with RWX permissions
riscv64-unknown-elf-objdump -S kernel/kernel > kernel/kernel.asm
riscv64-unknown-elf-objdump -t kernel/kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > kernel/kernel.sym
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -drive file=fs.img,if=none,format=r
aw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ kalloctest
start test1
test1 results:
--- lock kmem/bcache stats
lock: kmem: #fetch-and-add 0 #acquire() 78731
lock: kmem: #fetch-and-add 0 #acquire() 180888
lock: kmem: #fetch-and-add 0 #acquire() 173399
lock: bcache: #fetch-and-add 0 #acquire() 340
--- top 5 contended locks:
lock: proc: #fetch-and-add 29838 #acquire() 264577
lock: proc: #fetch-and-add 14234 #acquire() 264623
lock: proc: #fetch-and-add 7157 #acquire() 264627
lock: virtio_disk: #fetch-and-add 5270 #acquire() 57
lock: pr: #fetch-and-add 2725 #acquire() 5
tot= 0
test1 OK
start test2
total free number of pages: 32499 (out of 32768)
.....
test2 OK
```

```
$ bcachetest
start test0
test0 results:
--- lock kmem/bcache stats
lock: kmem: #fetch-and-add 0 #acquire() 361001
lock: kmem: #fetch-and-add 0 #acquire() 717444
lock: kmem: #fetch-and-add 0 #acquire() 868896
lock: kmem: #fetch-and-add 0 #acquire() 39
lock: kmem: #fetch-and-add 0 #acquire() 39
lock: kmem: #fetch-and-add 0 #acquire() 39
lock: kmem: #fetch-and-add 0 #acquire() 39
lock: kmem: #fetch-and-add 0 #acquire() 39
lock: bcache: #fetch-and-add 0 #acquire() 123260
lock: bcache: #fetch-and-add 7 #acquire() 93890
lock: bcache: #fetch-and-add 0 #acquire() 82454
lock: bcache: #fetch-and-add 0 #acquire() 57570
lock: bcache: #fetch-and-add 0 #acquire() 34462
lock: bcache: #fetch-and-add 0 #acquire() 20290
lock: bcache: #fetch-and-add 439 #acquire() 57156
lock: bcache: #fetch-and-add 155 #acquire() 36178
lock: bcache: #fetch-and-add 290 #acquire() 602278
lock: bcache: #fetch-and-add 0 #acquire() 19244
lock: bcache: #fetch-and-add 27 #acquire() 337846
lock: bcache: #fetch-and-add 0 #acquire() 178176
lock: bcache: #fetch-and-add 0 #acquire() 148912
```

```
--- top 5 contended locks:
lock: proc: #fetch-and-add 77359410 #acquire() 6510638
lock: proc: #fetch-and-add 55154449 #acquire() 6512960
lock: proc: #fetch-and-add 47765513 #acquire() 6503739
lock: proc: #fetch-and-add 43905714 #acquire() 6509091
lock: log: #fetch-and-add 30811130 #acquire() 74133
tot= 918
test0: OK
start test1
test1 OK
$
```

```
OK
test opentest: OK
test writetest: OK
test writebig: OK
test createtest: OK
test openiput: OK
test exitiput: OK
test iput: OK
test mem: OK
test pipe1: OK
test preempt: kill... wait... OK
test exitwait: OK
test rmdot: OK
test fourteen: OK
test bigfile: OK
test dirfile: OK
test iref: OK
test forktest: OK
test bigdir: OK
ALL TESTS PASSED
$
```

```
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/proc.o kernel/proc.c
riscv64-unknown-elf-gcc    -c -o kernel/swtch.o kernel/swtch.S
riscv64-unknown-elf-gcc    -c -o kernel/trampoline.o kernel/trampoline.S
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/trap.o kernel/trap.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/syscall.o kernel/syscall.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/sysproc.o kernel/sysproc.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/bio.o kernel/bio.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/fs.o kernel/fs.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/log.o kernel/log.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/sleeplock.o kernel/sleeplock.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/file.o kernel/file.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/pipe.o kernel/pipe.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/exec.o kernel/exec.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/sysfile.o kernel/sysfile.c
riscv64-unknown-elf-gcc    -c -o kernel/kernelvec.o kernel/kernelvec.S
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/plic.o kernel/plic.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/virtio_disk.o kernel/virtio_disk.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/stats.o kernel/stats.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie   -c -o kernel/sprintf.o kernel/sprintf.c
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -DSOL_LOCK -DLAB_LOCK -MD -mcmodel=medany -ffreestanding
-fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -march=rv64g -nostdinc -I. -Ikernel -c user/initcode.
S -o user/initcode.o
riscv64-unknown-elf-ld -z max-page-size=4096 -N -e start -Ttext 0 -o user/initcode.out user/initcode.o
riscv64-unknown-elf-ld: warning: user/initcode.out has a LOAD segment with RWX permissions
riscv64-unknown-elf-objcopy -S -O binary user/initcode.out user/initcode
riscv64-unknown-elf-objdump -S user/initcode.o > user/initcode.asm
riscv64-unknown-elf-ld -z max-page-size=4096 -T kernel/kernel.ld -o kernel/kernel kernel/entry.o kernel/start.o kernel/console.o
 kernel/printf.o kernel/uart.o kernel/kalloc.o kernel/spinlock.o kernel/string.o kernel/main.o kernel/vm.o kernel/proc.o kernel/
swtch.o kernel/trampoline.o kernel/trap.o kernel/syscall.o kernel/sysproc.o kernel/bio.o kernel/fs.o kernel/log.o kernel/sleeplo
ck.o kernel/file.o kernel/pipe.o kernel/exec.o kernel/sysfile.o kernel/kernelvec.o kernel/plic.o kernel/virtio_disk.o kernel/sta
ts.o kernel/sprintf.o
riscv64-unknown-elf-ld: warning: cannot find entry symbol _entry; defaulting to 0000000080000000
riscv64-unknown-elf-ld: warning: kernel/kernel has a LOAD segment with RWX permissions
riscv64-unknown-elf-objdump -S kernel/kernel > kernel/kernel.asm
riscv64-unknown-elf-objdump -t kernel/kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > kernel/kernel.sym
make[1]: Leaving directory '/root/github/fudan-os/xv6-oslab24'
== Test running kalloctest ==
$ make qemu-gdb
(84.5s)
== Test   kalloctest: test1 ==
  kalloctest: test1: OK
== Test   kalloctest: test2 ==
  kalloctest: test2: OK
== Test kalloctest: sbrkmuch ==
$ make qemu-gdb
kalloctest: sbrkmuch: OK (6.5s)
== Test running bcachetest ==
$ make qemu-gdb
(2.3s)
== Test   bcachetest: test0 ==
  bcachetest: test0: OK
== Test   bcachetest: test1 ==
  bcachetest: test1: OK
== Test usertests ==
$ make qemu-gdb
usertests: OK (75.0s)
== Test time ==
time: OK
Score: 70/70
root@2cea5af01129:~/github/fudan-os/xv6-oslab24#
```

1. Experiment Objective

The goal of the experiment is to redesign the memory allocator and disk buffer cache in the xv6 kernel by utilizing the provided locking mechanisms. This redesign aims to reduce contention in CPU memory allocation and disk cache management to improve overall system efficiency.

---

2. Experiment Process and Notes

(1) Understanding the Memory Allocator (`kernel/kalloc.c`)

- **Structure Overview:**
  The `kmem` structure manages a single global free list for memory pages of 4096 bytes, protected by a spinlock.
  - When allocating a page, it is removed from the free list's head.
  - When freeing a page, it is added back to the free list's head.
  - The spinlock ensures there are no race conditions during these operations.
- **Function Analysis:**
  - **`kinit`:** Initializes the spinlock for the free list and calls `freerange` to prepare memory pages between `end` and `PHYSTOP`.
  - **`freerange`:** Iteratively splits memory into pages and adds them to the free list using `kfree`.
  - **`kfree`:** Validates memory addresses, locks the free list, and inserts the page as a node into the list's head.

---

(2) Spinlock Implementation (`kernel/spinlock.c`)

- **Structure Overview:**
  The `spinlock` structure includes:
  - A `locked` flag indicating if the lock is held.
  - Debugging fields such as `name` (lock name) and `cpu` (holding CPU).
- **Key Functions:**
  - **`initlock`:** Initializes the lock and sets the `name` field.
  - **`acquire`:** Implements the spinlock mechanism using the atomic operation `__sync_lock_test_and_set` to acquire the lock.
    - Disables interrupts to avoid deadlocks.
    - Uses memory barriers (`__sync_synchronize`) to prevent instruction reordering.

---

(3) Disk Buffer Cache (`kernel/bio.c`)

- **Structure Overview:**
  - The `bcache` structure contains a spinlock, an array of buffers, and a doubly linked list for managing buffers with an LRU (Least Recently Used) strategy.
  - The `struct buf` structure holds metadata for the buffers, including references to the actual data.
- **Key Functions:**
  - **`binit`:** Initializes the buffer cache by constructing a circular doubly linked list of buffers.
  - **`bread`:** Fetches a block into the cache, reading from disk if necessary.
  - **`bget`:** Finds or allocates a buffer using the LRU strategy if a cache miss occurs.

---

(4) Sleeplock Implementation (`kernel/sleeplock.c`)

- **Structure Overview:**
  - `sleeplock` contains a spinlock (`lk`), a `locked` flag, and fields for debugging such as `pid` (process ID).
- **Key Functions:**
  - **`acquiresleep`:** Acquires the lock by sleeping on a channel until it becomes available.
  - **`sleep`:** Atomically releases the spinlock, puts the process to sleep, and reacquires the lock after being awakened.

---

(5) Redesigning the Memory Allocator

- **Enhancements:**
  - Replaced the global free list with per-CPU free lists to reduce contention.
  - Implemented a `steal_pages_from` function to allow CPUs to borrow pages from others when their free list is empty.
  - Optimized locking to ensure safe, efficient operations across multiple CPUs.

---

(6) Redesigning the Disk Buffer Cache

- **Enhancements:**
  - Added a hashing mechanism to divide the buffer cache into buckets, each protected by a separate spinlock.
  - Allocated a fixed number of buffers to each bucket, simplifying buffer management and reducing contention.
  - Adjusted functions such as `binit`, `bget`, and `brelse` to integrate the new bucket structure.