Lab-1 report

□□□-21307130254

2023-10-1

task 1

□□□□

- □□□□□□□□□□□kernel/kalloc.c□□□□□□□□□□□□□□□□(races)□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

```
r = kmem.freelist;
if (r)
    kmem.freelist = r->next;
```

`r`□□□□□□□□□□□□□□`r = kmem.freelist`□□□□□□`kmem.freelist`□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□kernel/kalloc.c□□□□(contention)□□
  - □□□□□□□□□□□□□□□□□□□□□
  - □□□□□□□□□□□□□□□□□□□□□□□□cpu□□□□□□□□□□□□□□□□□□□□□
- □□acquire□release□push_off/pop_off□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□cpuID□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□cpuID□□ □□□□□□□□

□□□□

- □□NCPU□kmem□□□□□□□□□□□NCPU□□□□CPU□□□□□□□CPU□□□□□□□□□□□
- □□kinit□□□□□□□CPU□□□□□freerange□□□□□kfreeID□□□□□□□□cpuID□□□□□□□□□□□□□free□ freerange□□□□□□□□□□NCPU□□□□□□□□□□kfreeID□□□□□ID□CPU□□□□□
- □□kalloc□kfree□□□□□□□□□□cpuID□□□`kmem[cpuID]`□□□□□□□□cpuID□□□□push_off□□□□□pop_off
- □□□CPU□□□□□□□□□□□CPU□□□□□□□□□□□□□□□□□□□□□□□□□□□CPU□□□kalloc□□□□□□□□□□

□□□□

```
$ kalloctest
start test1
test1 results:
--- lock kmem/bcache stats
lock: kmem: #test-and-set 0 #acquire() 16686
lock: kmem: #test-and-set 0 #acquire() 192570
lock: kmem: #test-and-set 0 #acquire() 203321
lock: kmem: #test-and-set 0 #acquire() 4091
lock: kmem: #test-and-set 0 #acquire() 4091
lock: kmem: #test-and-set 0 #acquire() 4092
lock: kmem: #test-and-set 0 #acquire() 4091
lock: kmem: #test-and-set 0 #acquire() 4092
lock: bcache: #test-and-set 0 #acquire() 356
--- top 5 contended locks:
lock: proc: #test-and-set 87610 #acquire() 793643
lock: proc: #test-and-set 78181 #acquire() 793639
lock: proc: #test-and-set 32615 #acquire() 393552
lock: proc: #test-and-set 20254 #acquire() 393514
lock: proc: #test-and-set 19070 #acquire() 393514
tot= 0
test1 OK
start test2
total free number of pages: 32497 (out of 32768)
.....
test2 OK
start test3
child done 1
child done 100000
test3 OK
$
```

```
$ usertests sbrkmuch
usertests starting
test sbrkmuch: OK
ALL TESTS PASSED
```

task 2

[illegible]

- [illegible]
- 5[illegible]
  - [illegible]fork[illegible]

```
struct Basket
{
    int type;
} basket;
queue<int> jdgb,jbgz;
```

```
int producedJdgb,producedJbgz;
lock lock1;
lock lock2;
lock lock3;
```

- [illegible]

```
while(1)
{
    if(produced < queue.length())
    {
        produce();
        produced++;
    }
    if(basket.type == empty)
    {
        basket.type = jdgb;
    }
}
```

- [illegible]

```
while(1)
{
    if(someoneCome)
    {
        jdgb.push();
    }
    if(!jdgb.empty() && basket.type == jdgb)
    {
        jdgb.pop();
        basket.type = empty;
        producedJdgb--;
    }
}
```

- [illegible]

[illegible]

[illegible]

- [illegible]

```
make clean && make DiningPhilosopher
```

- □□

```
./DiningPhilosopher
```

□□□□

- 5□□□□5□□□□□□□□□□□LEFT(x)□RIGHT(x)□□□□□□□□□
- pickup()□□□□□□□□□□□
- putdown()□□□□□□□□□□□
- □□0□□□□□□□□□□□□□□□□□□

□□□□

```
⊗ ➜ lab1_synchronization ./DiningPhilosopher
  Philosopher 0 will think for 2 seconds
  Philosopher 1 will think for 1 seconds
  Philosopher 4 will think for 1 seconds
  Philosopher 3 will think for 3 seconds
  Philosopher 2 will think for 1 seconds
  Philosopher 1 will eat for 1 seconds
  Philosopher 4 will eat for 3 seconds
  Philosopher 1 will think for 1 seconds
  Philosopher 2 will eat for 2 seconds
  Philosopher 4 will think for 2 seconds
  Philosopher 2 will think for 2 seconds
  Philosopher 0 will eat for 1 seconds
  Philosopher 3 will eat for 2 seconds
  Philosopher 1 will eat for 3 seconds
  Philosopher 0 will think for 1 seconds
  Philosopher 3 will think for 1 seconds
  Philosopher 4 will eat for 3 seconds
  Philosopher 1 will think for 1 seconds
  Philosopher 4 will think for 1 seconds
  Philosopher 0 will eat for 2 seconds
  Philosopher 3 will eat for 1 seconds
  Philosopher 2 will eat for 3 seconds
  Philosopher 3 will think for 3 seconds
  Philosopher 0 will think for 3 seconds
  Philosopher 4 will eat for 2 seconds
  Philosopher 4 will think for 3 seconds
  Philosopher 2 will think for 3 seconds
  Philosopher 1 will eat for 1 seconds
  Philosopher 3 will eat for 3 seconds
  Philosopher 1 will think for 2 seconds
  Philosopher 0 will eat for 2 seconds
  Philosopher 2 will eat for 3 seconds
  Philosopher 3 will think for 2 seconds
  Philosopher 0 will think for 2 seconds
  Philosopher 4 will eat for 2 seconds
  Philosopher 4 will think for 1 seconds
  ^C
```