COL 331 HW-4 2015CS10233

AKHILESH KARRA

Q1 SPIN LOCKING

i) Explain in a few sentences why kernel panicked. You may find it useful to stack trace.

A) Invariant of holding one lock is violated, so the kernel panicked.

Other interrupts can come in and take our lock, if we allow interrupts by calling sti() after acquiring the lock. This can lead to data being in an inconsistent state or violate invariants.

ii) Explain in few sentences why kernel didn't panic. . . . . . .

why do file-table-lock and ide-lock have diff behaviour?

A) File-table-locks critical sections are less likely to trigger intterrupts when compared to ~~these file lock~~'s those involving IDE. ~~Maybe~~ perhaps they are shorter; so having less chance for external interrupts to occur.

iii) Why does release ( ) clear lk->pcs[0] and lk->cpu before lk->locked? Why not wait until after?

A) Other process can acquire the lock and have access to the information, if we clear lk->locked. We want the lock to be in a consistent state. If an interrupt occurs and we have a stack trace before setting the the old pcs and cpu after clearing locked, we now have an inconsistent looking lock.

Uniprocessor locking

*) Does this implementation of locks work on a Uniprocessor?

A) No, While loop will keep spinning forever, since thread which held the lock will not run.

*) Does this implementation of locks work on a uniprocessor?

A) Yes, Every time loop fails to obtain the lock, interrupts are enabled. ∴ the thread holding the lock run and can release the lock.

Sleep and Wake up

Q) Both producer and consumer are sleeping on the same channel q. Is this correct.

A) Yes. When wake up is called all waiting threads wake up, one gets the lock and rest sleep.

XV6 file system

i) echo > a :

A) log_write 34
log_write 34
log_write 59

File named "a" is created.

disk writes : creating & writing to a's INODE
Updating the data block of a's parent.

echo x > a :

content of a is updated.

disk corites :

allocation of data block to a

updating bitmap of that block to not free

writing x to data block

Updating a's INODE.

rm a :

File a is removed.

disk corites :

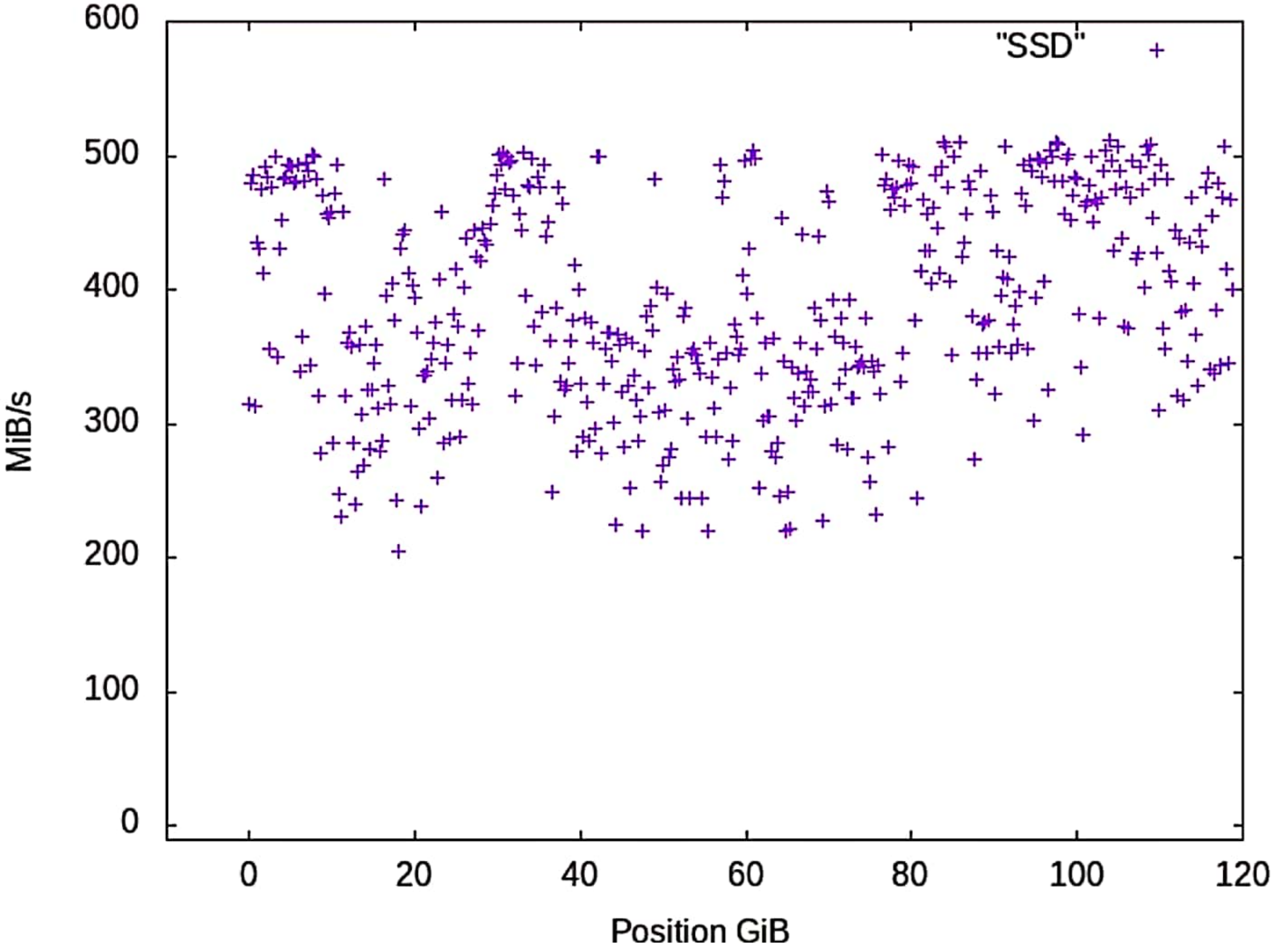
"a" is removed from its parent's directory entry

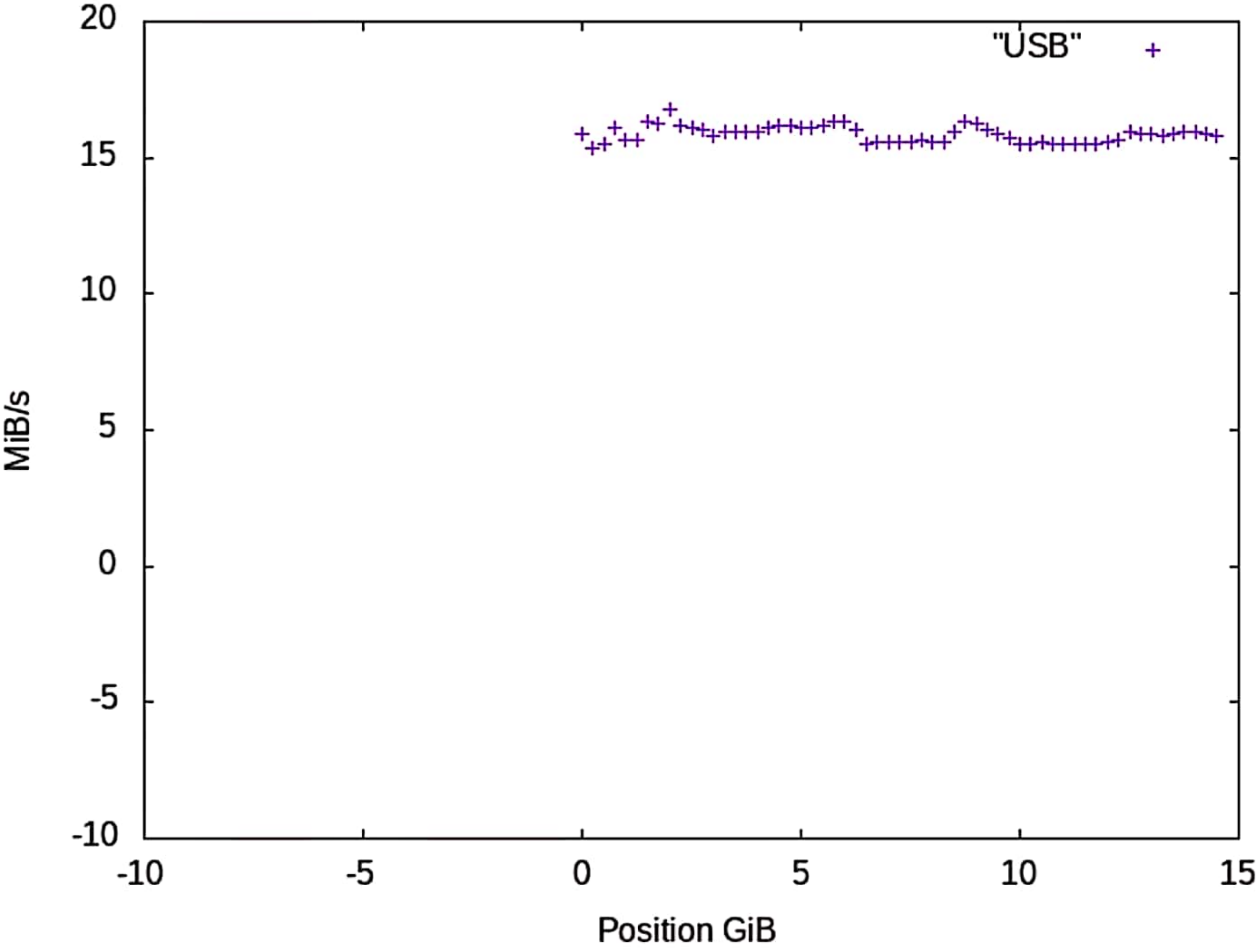
set a's INODE Value to free

- update a's bitmaps.

Q4 ZCAV

My laptop has SSD. ~~[illegible]~~


600
500
400
300
200
100
0
MiB/s
0
20
40
60
80
100
120
Position GiB
"SSD" +


"USB" +
MiB/s
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
Position GiB