1. C
2. a.

b. Password is encrypted so wouldn’t be able to understand it

c. A large computing base will create more bugs because the larger the computing system, more and more holes and bugs.

d. Shared secret keys are expensive, public keys are still widely used to encrypt for communication.

e. Principals are people not numbers.

3. a. If you’re opening a file, you’re opening same file

b. Don’t need generation number to avoid race conditions. What if file was deleted, then reused inode, then would point to different value. That’s need generation number.

c. Dumb server and smart client.

d. In practice, most NFS operations are idempotent; this keeps the protocol simple. Idempotent: If do same operation, nothing changes. (if do operation 10 times or one time, nothing changes.)

e. We don’t validate caches for performance reasons.

4. a. Cache affinity: multiple process running at same time, for each process, schedule to same CPU core. When stick same process to same CPU, every time run will have the same cache.

b. It’s about sys calls

c. causes too many problems

d.

e. Want to choose clean pages as victims, have to write dirty pages to disk first

5. a. fsck is supposed to be reliable

b.

c. totally wrong

d. fsync and fdatasync, fdatasync syncs data, the min metadata for executing file, they go to cache

e. true

6. a. Wrong, filesystem doesn’t have inode.

b. Filetype and inode value aren’t change after creation

9

c. batching decreases fairness

d. flash reading is faster than writing to it.

e. RT-11 before create file, have to tell what the filesize is.

10

1. any lock can lock the spinlock
2. anyone can call semlock.
3. If accessing same resource at same time, can have race conditions
4. Same as a and b, totally wrong
5. Doesn’t make sense at all.

11. Question is not about SSL or TLS, key is if you’re just working with single computer, do you need to use encryption. If you’re reading a sector from one disk, do you need to encrypt the reads? There are 2 different answers.

Yes: If it’s sensitive data, want to protect from other programs running in machine. Kernel should already be protecting data from other processes. Kernel already provides good isolation. If you have a hardwall that sniffs the bus, it could read whatever is going through the bus, could read the bytes that are running through the bus. In that case, would want to use encryption.

No: Systems already have good isolation and good security, don’t need encryption.

Questions asks about whether or not use encryption, not about what SSL or TSL does.

12. Need to print man pages of everything seen before.

‘clri’ reads meta data given inode number. Reads inode data for inode 2. Would write to direction 2. Couldn’t access root directory.

If this is the only damage done to disk, can recover by looking at different links and bitmap and what datablock go to the root directory and see where it would’ve been.

For each data block, needs to have a pointer from datablokc.

You would lose metadata for the directoy though .

13.

Both trying to access a resource. If critical section is very short, or long. Discuss different situations what would happen. Dpeneds on numbers of writes and execs.

14.

Who should excecute command? User or kernel. Gives you a security question. If run in kernel mode, could cause more problems.

Short answers 9:

Priority inversion problem: Doesn’t work at all, doesn’t solve the problem.