Linux experience

Description of parts of a linux system (bootloader, kernel/os, recovery)  
  
Ways to run code/action on system boot up (rcS, init.d, systemd)  
  
Name a couple common linux utilities and what purpose they serve (ifconfig for network)  
  
Ways to troubleshoot code on linux (gdb, valgrind, etc)  
  
What kind of things do you need to be mindful of when building for a target platform? (architecture, toolchain)

Experience with linux build systems (buildroot, yocto) Why use one over the other?

What purpose do device trees serve in a linux system?  
  
How would you go about troubleshooting a linux system that wont boot?  
  
C experience  
  
What kind of things do you need to be mindful of when writing in C? (memory management, concurrency, threads)  
  
How would you go about troubleshooting a memory leak in a C program?  
  
When would you use multi-threading over single-threading? (do x 100 times)

What are some potential issues we run into when multi-threading? (race conditions, use locks)

C Code Sample Live Test  
  
Automated builds/Automated tests  
  
What experience do you have with automating build processes? (shell scripts, jenkins, build server)  
  
Have you ever done automated testing on the code you’ve written? (pytest, jenkins)  
  
Overall

Working in a team, code decisions  
  
How would you explain a code feature to a non-technical person?  
  
How would you handle a situation where there is a conflict over a code design?  
  
Peer review of C ( review PRs, code design)