INTERVIEW PREP

Embedded Systems Design

Amazon – a place where builders can build. We hire the worlds brightest minds and offer them an environment in which they can invent and innovate to improve the experience of our customers. We want employees who will help share and shape our mission to be Earth's most customer-centric company. Amazon's evolution from Web site, to e-commerce partner, to development platform, is driven by the spirit of invention that is part of our DNA. We do this every day by solving complex technical and business problems with ingenuity and simplicity. We're making history, and the good news is that we've only just begun.

Thank you for taking the time to speak with us. These tips are intended to enhance your candidate experience.

Our employees tackle some of the most complex challenges in large-scale computing. Software development engineers, technical program engineers, test engineers, technical program managers and user-interface experts work in small teams across the company to create experiences that our customers will be thrilled with.

amazon

Answering the Systems Design Question

When interviewing for a Software Development Engineer, a Software Development Manager, or a Technical Program Manager position at Amazon, you will have at least one interview focused on software systems design. This is a very important interview, so it's critical to prepare for it thoroughly. You'll know when you're being asked the systems design question because you'll be asked to design a software system.

Answering this question will be very interactive; the interviewer will ask you lots of questions related to the design and you are encouraged to ask the interviewer any necessary questions to complete your design. If you are suggesting a technology to solve a problem, please make sure you understand how that technology works; it is more important that you understand how your solution solves the problem than specific technology solutions. It helps to think out loud and take hints from the interviewer.

You will most likely be diagramming your design on a white board or virtual white boarding tool, so if you have access to a white board at home (or even just a pen and paper), writing these sorts of designs out by hand can be great practice.

Topics to Review

Often times, embedded system software is decomposed into different components, for example, something to store data to temporary/permanent storage devices, make decisions (such as business logic), process internal/external events, and flow data between components. Reviewing software design diagrams for embedded systems can be helpful for preparation.

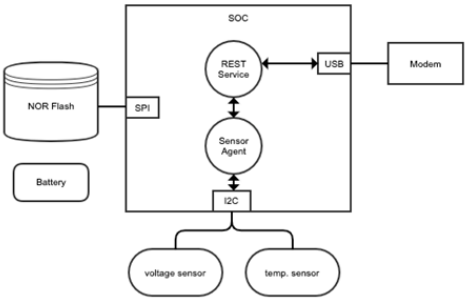
Embedded systems are typically resource constrained and subject to be optimized from multiple different angles. It's important to consider system-level optimization when diagramming and designing your software system. Prior to your interview be sure to research resource optimization concepts. For example, what are the goal for optimization (throughput, latency, power, memory footprint, cost, time to market, etc.)? What are the bottlenecks when the system is under heavy loads? What are potential optimization techniques (caching, non-blocking data structures, zero-copy, etc.) and their trade-offs?

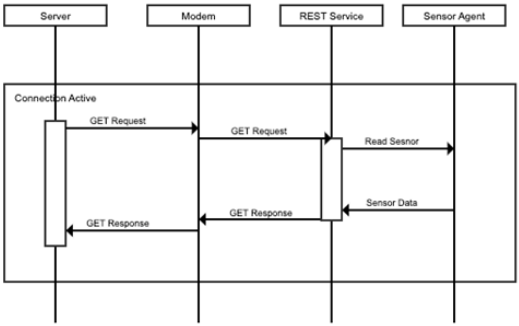
Knowledge of designing against stringent resource constraints, dividing a system into smaller components, APIs, multi-core heterogeneous systems, and managing scalable number of distributed devices (e.g., IOT) are very important in answering system design questions. If you don't work with these concepts regularly, be sure to review them prior to your interview.

Complex embedded software system designs often need to trade off availability, consistency, and other desirable performance characteristics. Be prepared to discuss these and other types of tradeoffs.

Steps in the Systems Design Interview

1) Ask clarifying questions; while the interviewer won't try to trick you, they might be intentionally vague. It's important to know what sort of design the interviewer is looking for, so ask questions. When asking your questions, start with the customer in mind. who is the customer and what problems are you solving for them?
2) As you ask clarifying questions, begin writing a list of requirements on the board. This should typically be the first thing you add to the white board.
3) Once you have a good idea on the sort of problem the system you are designing is supposed to solve, begin drawing a diagram on the whiteboard to express your ideas. A great way to do this is to draw shapes to represent different software components and data sources, and then arrows connecting them to show software/hardware components, APIs, and interactions between components.





4) Be prepared to discuss trade-offs in your design. With any software system there are multiple ways to design it. What advantages would yours have? Disadvantages? What if you were to change a component or process? Be prepared to discuss these questions.
5) Operational performance of your design is important as well. Be prepared to answer the following: how will you ensure this system is working an acceptable level of performance? If a problem occurs, what will be involved in troubleshooting and resolving it quickly? What are the possible points of failure and how can they be made more robust against failure?

"Many of problems we face have no textbook solution, and so we happily invent new ones"

Jeff Bezos, 2010 Shareholder letter

amazon