

[illegible]

Coding Interview

This interview will assess the coding skills you'll use every day, including text manipulation, handling input/output, automating tasks, interfacing with external systems/processes, etc. The questions can be a real problem, or something contrived to use these skills.

- **Coding environment:** The coding will be on a white board.
- **Language selection:** We've hired candidates who have done their phone screens in virtually every modern programming language, including Python, PHP, Perl, Ruby, Java, C++, Go, C#, and Haskell. Choose the one you're most comfortable with and be prepared to use it to solve any type of problem (for at least one of the problems, we ask that you use something other than shell scripting like bash/zsh etc).

What we're looking for

As you're coding, among the things we're looking for is to see how you:

- Obtain an overall working solution to each problem in a reasonable amount of time.
- Ask questions and plan your solution rather than jumping right into implementation.
- Defensive coding is important, but don't focus on details (such as error handling and corner cases) to the detriment of the overall solution. If you're not sure if a given error handling or edge case is important, ask the interviewer.
- If you can't remember the order or arguments to a function or its name, just say so, leave a placeholder and move on. Don't get hung up on syntax.
- Explain your decisions to the interviewer and be open to feedback. It's totally fine to present a rough solution in the beginning and iterate as you go along.
- Be open to changing your mind if you think you've started your solution in the wrong way, or even in the wrong language.
- Resist the temptation to look things up on the internet during your interview. We'd rather have you get a function name incorrect or the order of arguments wrong than have you looking things up during the interview.

How successful candidates prepare

The best way to prepare for coding interviews is to practice under similar circumstances by yourself or with a friend, using sample questions.

Production Engineering Interview Preparation

(Password: ProductionEng2016)

Practice Coding Questions - FB Lab

Systems Interview

Systems interview may be done on a white board or in a conversation. This interview will assess your knowledge of UNIX/Linux theory and practical troubleshooting of Linux systems.

What we're looking for

On the theoretical side, we want to know you can explain the core building blocks of how a Linux system operates. For example:

- How systems boot and load Linux
- The shell, and how it interacts with the underlying operating system
- UNIX filesystems and storage
- The Linux virtual memory model
- Techniques for resource control
- Common system troubleshooting tools and techniques

Examples

On the practical side, you will likely get one or more troubleshooting problems. These problems might have you debugging an issue with one server, or multiple servers in a distributed system. We want to know how you apply your knowledge to solve problems that you will likely encounter in the real world.

How successful candidates prepare

Our systems interview is not a “trivia” interview, but a test of how well you understand fundamental concepts thoroughly. At Facebook, we want engineers with both a solid base of knowledge and a healthy curiosity about how the systems they are responsible for actually work.

For the practical portion, most successful candidates run through a number of debugging scenarios, and familiarize themselves with the tools for each. Know the various tools to identify different aspects of system performance and resource utilization, and when/where to use them. Concentrate on working through scenarios with a defined, methodical workflow.

Practice is also very useful, especially if you aren't working with debugging Linux systems on a regular basis. For example, you or a friend can create some kind of troubleshooting/debugging scenario on an actual UNIX system, and you can practice troubleshooting the issue.

Networking Interview

This interview will assess your knowledge of theoretical and practical networking between UNIX hosts, potentially at every level of the networking stack.

What we're looking for

On the theoretical side, we want to make sure that you can explain the fundamental building blocks of host-level networking, such as:

- Basic network protocols, such as ICMP, TCP, and UDP
- How packets are constructed
- TCP/IP connection establishment and teardown
- How common internet application protocols such as DNS, HTTP, and DHCP are designed, how they work in practice, and common pitfalls with them.
- How to design your own application networking protocol, or changes you might recommend to existing application networking protocols.
- IPv4/IPv6 differences
- Host and application level networking options in Linux/UNIX, and where and how they are used
- How load balancing works, different load balancing methods, and pros and cons of each.
- Basic network design concepts

On the practical side, we want to see how you apply this theoretical knowledge to solving actual problems, such as:

- Common host-level networking troubleshooting scenarios
- How — and where — to use network troubleshooting tools such as tcpdump/Wireshark, ping, and traceroute.

Unless you know that you are interviewing for a network engineering role, this interview won't cover WANs, routing protocol details (e.g., BGP, OSPF), or dedicated networking hardware (e.g., routers, switches).

How successful candidates prepare

Most successful candidates study for the theoretical portion of the interview, and practice for the practical portion. You can find what works best for you, but reading RFC's (<https://www.ietf.org/rfc.html>) is also an excellent way to educate yourself about the designs and tradeoffs in core networking protocols such as DNS and HTTP.

For the practical portion of the interview, successful candidates do some troubleshooting exercises involving networking problems on systems. For example: host is unreachable, host is "slow," or an application protocol is not working properly. Practice using some of the common troubleshooting tools, such as ping and traceroute. Take some tcpdumps from live systems and learn how to get information from those dumps by loading them into Wireshark.

System Design Interview

This interview will require you to design a large-scale product or infrastructure component. For example: “Design a URL shortening service like bit.ly”.

What we’re looking for

As you’re designing the system, among the things we’re looking for is to see how you:

- Ask clarifying questions to determine what the goals and requirements of the system are.
- Determine which parts of the problem are most important and will affect the overall design.
- Describe the system at a high level, explaining the overall architecture.
- Draw diagrams that clearly describe the relationship among different system components.
- Identify trade-offs in your design (such as consistency, availability, partitioning, performance), and describe how you make decisions around them.
- Calculate back-of-the-envelope resources necessary (storage, CPU, RAM, etc.).
- Adjust the design of your system when requirements or constraints change.
- Determine how your system will perform at scale, and identify any bottlenecks and limitations in your design.
- Explain how your system handles both success and failure cases.

How successful candidates prepare

Most successful candidates prepare by coming up with designs for large scale products and then justifying them to a friend (who plays the role of interviewer). They focus on technology they really understand and/or have personally used, and tell us about how they would solve a problem rather than how they think Facebook actually solves the problem.

This Quora post has some helpful suggestions that you can look through:

<https://www.quora.com/How-do-I-prepare-to-answer-design-questions-in-a-technical-interview>

This resource walks you through solutions for the following examples:

<https://www.hiredintech.com/classrooms/system-design/>

- Design a URL shortening service like bit.ly.
- How would you implement the Google search?
- Design a client-server application which allows people to play chess with one another.
- How would you store the relations in a social network like Facebook and implement a feature where one user receives notifications when their friends like the same things as they do?

Soft Skills Interview

Production Engineers partner with their teammates, software engineers, project managers, and other Facebookers every day. During this interview, we want to learn how you have worked in a team, deal with conflict, and overcome challenges. For recent graduates, we'll discuss any internships, as well as interesting academic and side projects. Unlike the technical interviews for coding, systems and design, this interview focuses on your unique experiences and approaches to solving non-technical problems. For this interview, you'll be meeting with a leader in Production Engineering.

What we're looking for

- Quick “elevator pitch”, high-level descriptions of completed or in progress projects.
- Clear descriptions of your role in the projects, who you worked with, and the challenges you faced, rather than what the specific work was.
- Things don't always go as planned, so think through failures you've encountered in the past. Share what could have been done differently and what you learned from the experience.
- Your reasons and motivations for why you changed roles in the past and why you find PE exciting. If you're currently a software engineer, or other type of engineer, think through why Production Engineering is a good fit for you. [Pedro's SREcon talk](#) is a great resource for understanding Production Engineering.

Sample questions

- Tell me about a recent project you are proud of.
- Tell me about a design decision you made that you later regretted.
- Tell me about a disagreement or conflict between you and a coworker.
- What motivates you to leave your current role?
- Describe your ideal day at work.
- What is your understanding of Production Engineering?