Systems Engineering Interview Prep

CODING

- Coding Interview Environment

The coding is done 'live' via CoderPad (http://coderpad.io/), a collaborative document editor specifically designed for coding phone interviews.

- Language Selection

As there are many different programming languages used at Facebook to solve infrastructure challenges, we don't have a preference for the language (or languages) you use during the interview. We think that if you can show us solid programming skills in one language, you'll be able to transfer them to other languages. At least one of the problems we ask you will require that you use something other than shell scripting (bash/zsh/etc.). We've ended up hiring candidates who have done their phone screens in virtually every modern programming language, including Python, PHP, Perl, Ruby, Java, C++, Go, C#, and Haskell.

- Types of Questions Asked

The questions we ask require skills used every day by Engineers (Systems / Operations / Infrastructure Engineers), 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.

- Interview Tips

Your primary goal in the interview is to obtain a working solution to each problem in a reasonable amount of time.

Make sure you are in a quiet place with a reliable internet connection. Headphones are very handy. A speakerphone also usually provides acceptable sound quality. Holding the phone in your neck for 45 minutes while you try and type is going to be a pain. Close your email and chat apps to avoid interruptions.

We will always paste in the text of any coding question we ask you. Take some time after reading the question to ask questions and plan out your solution, rather than jumping right into its 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.

Don’t get hung up on syntax. If you can’t remember the order or arguments to a function or its name, just say so, leave a placeholder and move on.

While you don’t need to provide a play by play of your thought process throughout the interview, it’s best to let the interviewer know why you are making certain decisions, and do most of your work in the CoderPad window. This will also help with any course corrections that may be needed while you are solving the problem.

Take hints from the interviewer and be open to other solutions as you go. It's totally fine to present a rough solution in the beginning and iterate as you go along.

Use the language you are strongest in. Don't use a language you know less well because it's trendy or you think it will please the interviewer. If the interviewer doesn't know your strongest language, they will figure it out later.

Don’t be afraid to change your mind. If you think you’ve started your solution in the wrong way, or even in the wrong language, it’s OK to admit it and change tack (and the sooner you do it the better).

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. We want to see you write code and think on your feet, not search the web.

- Preparation

The best way to prepare for coding interviews is to practice under similar circumstances by yourself or with a friend, using sample questions. A coding phone screen is an unnatural event, even if you are used to coding regularly for your job. The problems are different, the environment is different, and you are under time pressure.

Choose at least one language and know the basics solidly. It is better to know one language really well than multiple languages poorly. This includes creating classes and methods, conditional constructs, loops, built-in data structures, input / output, and interfacing with external processes / systems. Be prepared to use this language to solve any type of problem you may get.

Consider practicing by trying coding exercises here: https://www.interviewbit.com/

Facebook Engineering

https://code.facebook.com/

https://code.facebook.com/posts/

https://www.facebook.com/Engineering

Crush Your Coding Interview https://www.facebook.com/Engineering/posts/10152735857917200

SYSTEMS

- Types of Questions Asked

The Systems interview generally covers at least one the following topics:

Linux Internals

Troubleshooting

Some interviewers may spend the entirety of the interview focused on one topic while others may incorporate both. They are described in detail below.

- Linux Internals

This portion of the Systems interview is designed to evaluate your knowledge of Linux internals and topics can include, but are not limited to:

Process Execution and/or Threads

Memory Usage

RAID Levels

The kernel and how it interacts with other system components

System Calls

Signals and Signal Handlers

Modern Web Architectures and Webservers

- Tips

No one tool or topic is critical, but a broad familiarity with each of these spaces and the ability to apply that knowledge to new situations is important.

Everything from the kernel to the user space is fair game, and we look for both breadth and depth of knowledge.

- Preparation

Review Linux system fundamentals and the topics mentioned above.

Review userspace / Kernel space boundaries and interactions.

Examples might include: ioctls, sysctls, context switches.

- Troubleshooting

The Engineer will lay out a system related issue or problem that you are seeing in the infrastructure and ask you to troubleshoot it. As you take each step, the Engineer will provide the hypothetical system output to prompt you to the next step.

Here is an example exchange:

Interviewer: 'You're seeing this in the infrastructure. How would you troubleshoot it?'

Interviewee: 'I'd check this first'

Interviewer: 'OK, you get this in response. What next?'

Interviewee: 'I'd run this command next'

A second scenario might involve having you analyze Linux / Unix system performance using various tools. The engineer will share a copy of a report via coderpad.io and the expectation would be for you to identify what the potential issues might be and your approach for resolving them. While this is a less common scenario, please be sure to have access to a computer / internet connection during the interview.

- Tips

We’re looking for a structured, methodical approach. Narrow down potential issues systematically as you zero in on the root cause.

- Preparation

Review troubleshooting tools for system-level performance issues.

Review troubleshooting tools for debugging application-level performance issues or bugs.

- General Interview Tips

Be open about what you know and don’t know. One of the goals is to find the limits of your knowledge and see how you react outside of that so always attempt to answer the question at hand. The interviewer will provide hints along the way if you get stuck.

Ask clarifying questions whenever necessary.

Think through your solutions out loud, and share as much relevant knowledge as possible.

Take hints from the interviewer and don’t be afraid to change your mind as you go.

Again, please make sure you are in a quiet place with a reliable internet connection.