Systems Interview Preparation-What to expect?

The purpose of the systems interview is to understand how well you understand the inner workings of a Linux system.

At Facebook scale, a lot of the problems we encounter don't really occur at smaller scale. It's really important that a Facebook engineer can really deep-dive a systems level problem and uncover the root cause of an issue that has never existed before. The interview process is designed to discover if the interviewee has the required skills to do this.

It's important to remember that the interviewer's job is to find the limits of your knowledge. The interviewer will probably keep asking more and more detailed questions until you run out of answers. If this happens, don't be afraid to say you don't know something. You're not expected to ace the entire interview. It's much better to admit the limits of your knowledge and then perhaps take an educated guess as to what the answer might be, rather than answer a question incorrectly while pretending that you really know what you are talking about.

The Systems interview generally covers at least one of 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 Unix/Linux internals. This will be an assessment if you truly understand what is happening under the hood of a Unix/Linux system. It's very rare that you will be asked questions about specific software packages like how apache, MySQL or chef work. Lower-level systems knowledge is more generic and more transferrable. If you understand how the system itself works, it's much easier to troubleshoot an application running on top of it.

Topics can include, but are not limited to:

- Process creation, execution and destruction
- Differences between processes and threads
- Memory management
- System calls
- Signals and signal handlers
- Unix filesystem structure

Top Preparation Tips

Review Linux system fundamentals and the topics mentioned above. A book like “Modern Operating Systems” by Andrew Tanenbaum is a great stating point.

- Review userspace / Kernel space boundaries and interactions
- Examples might include: ioctls, sysctls, context switches

Troubleshooting

Understanding how a Unix/Linux system works is one set of skills. Being able to effectively use that knowledge to troubleshoot a running system is actually a slightly different set of skills. For a troubleshooting question, we are looking at your ability to do two things:

- The first is that you can demonstrate knowledge of the tools that are available to troubleshoot a system and that you understand the data that these tools give you
- The second thing we are looking for is if you have a good approach to troubleshooting a problem. This is demonstrating that you can follow a path of investigation where you are progressively ruling out potential problems until you arrive at the root cause of the problem

In most of these questions the Interviewer will lay out a system-related issue or problem that you are seeing in the infrastructure and ask you to troubleshoot it. Typically, the description of the problem will be pretty vague like a customer complains that “the system is slow”. The Interviewer will ask you how you would like to investigate this problem. You’ll propose a course of action like running a specific command looking for something in the output.

As you take each step, the Interviewer will provide the hypothetical system output to prompt you to the next step.

Here is an example exchange:

Interviewer: Someone says their application is not running. How would you troubleshoot it?
Interviewee: I'd log into the machine and run df –h to see if the disks were full'
Interviewer: 'OK, none of the disks are over 50% full. What next?'
Interviewee: 'I'd run this command next'

A second scenario might involve having you analyze Linux/Unix system performance using various tools.

- 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

The interviewer may share a copy of a report or command output via coderpad.io and the expectation would be for you to understand the data and what it means for a system.

(While this is a less common scenario, please be sure to have access to a computer / internet connection during the interview.)

Additional Tips

- 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. 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

Best of luck