TASK

Interview Preparation: Concurrency

Visit our website

Introduction

WELCOME TO THE INTERVIEW PREPARATION: CONCURRENCY TASK!

The most important step in landing your dream job is acing your interview. Interviews allow you to impress your future employer by demonstrating your knowledge and skills. However, an interview can be a nerve-wracking process. Taking the time to prepare for your interview can ensure that the process runs smoothly and is less stressful. In this task, we will be discussing the common interview topic, concurrency.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to Discord at **https://discord.com/invite/hyperdev** where our specialist team is ready to support you.

Our team is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

FULL STACK WEB DEVELOPER INTERVIEWS

Let's face it, an interview for any job can be terrifying and not knowing what to expect from an interview can make the whole process even more daunting. Although there are very few (if any) of us that will be completely confident for an interview, being better prepared is definitely a key to a successful interview and helps with confidence. This is your first of four interview preparation tasks. These tasks will not only help you to prepare for interviews in the real world but will also be used as a basis for the mock interview that will be conducted with you as the very final task of this Bootcamp!

Interviews for full stack web developer positions are a little different from other, more "traditional" interviews. Besides being able to answer questions, it is common for interviews for software developers to include a practical component where the applicant is asked to do some programming or debug some existing code. These technical interviews evaluate your knowledge of a programming language and your ability to solve problems on the spot when under pressure.

The best way to prepare for technical interviews is to build confidence by researching common programming concepts and answering possible interview questions based on these concepts beforehand as practice. When answering questions, try using a pencil and paper or whiteboard rather than an IDE. Generally, during a technical interview, you will not be allowed to use an IDE. You will instead be required to write down your answer on a whiteboard. Getting used to answering programming questions in this way can, therefore, be a huge advantage.

You might be wondering which web development topics are most likely to come up in an interview. The bad news is that it is really hard to tell exactly since web development is such a broad topic. The good news is that there will definitely be clues as to what they are likely to want to know in the job description of the position for which you have applied. Look at the "key skills", "core competencies" or "experience using the following web technologies" section of the job description of the post you have applied for, and be sure that you know the tools listed. Give yourself plenty of time before the interview to prepare.

The other good news is that in this Bootcamp, we have covered some essential web development concepts. This has given you a good foundation for your journey as a professional web developer. However, as a web developer, you must do your own research outside of your studies to extend your knowledge. In this task, we will ask you to do some research on one of the more complex and advanced topics brought up during web development interviews: concurrency.

INTRODUCTION TO CONCURRENCY

Concurrency basically means being able to do more than one thing at the same time (concurrently). It should be obvious that web servers must support concurrency - imagine the number of requests that web servers for Google, Youtube, or Airbnb have to support at once.

Concurrency can be implemented in various ways - multiprocessing and multithreading or an approach that uses both are common methods of implementation. Node.js has another approach to concurrency. It uses the I/O operations model that uses asynchronous non-blocking I/O calls to implement concurrency.

As a web developer, you must understand and be able to explain various approaches to implementing concurrency and the pros and cons of each approach.

PROCESSES AND THREADS

The two basic units of concurrency are processes and threads. Programs are created using a programming language and stored on a disk or in non-volatile memory. In order to execute, a program needs memory and various other system resources.

A **process** is a program that has been loaded into memory along with all the resources it needs to run. Every program needs certain essential resources, such as registers, a program counter, and a stack. A *register* is part of the CPU and is used to hold data. It can hold an instruction, a storage address or any other kind of data needed by the process. The *program counter* keeps track of where a computer is in its program sequence and the *stack* is a data structure that stores information about the active subroutines of a computer program. It is used as scratch space for the process.

Each instance of a running program is a process and there can be multiple instances of a single program. Each process runs independently and is isolated from other processes since they all have a separate memory address space.

A **thread** is the unit of execution within a process. A process can have one or more threads but a thread cannot contain a process. As mentioned above, a process is assigned memory and resources when it starts. If a process contains only one thread, it is known as a single-threaded process and, in such a case, the process

and the thread are the same thing. This process only accomplishes a single thing at a time. A process that contains more than one thread is known as a multi-threaded process. Multi-threaded processes accomplish several things at (almost) the same time. Threads are known as *lightweight processes* because they have their own stack but can access shared data in the heap. The operational cost of communication between the threads is low because threads share the same address space as the process and other threads within the process. However, unlike with processes which don't affect each other, a problem with one thread in a process will affect other threads in that process and the viability of the process as a whole.

CONCURRENCY WITH DATABASE INTERACTION

Database interactions also need to support concurrency. As you may well imagine, if several users are using a data-driven web app at the same time, your web app is going to have to be able to modify the data in your database for each user. A potential problem is that data in your database becomes 'unreliable' if attempts are made to modify a specific piece of data at the same time.

Oracle describes three potential problems with ensuring data reliability with a database that must support concurrent interaction:

- Dirty reads: A transaction reads data that has been written by another transaction that has not been committed yet.

- Nonrepeatable (fuzzy) reads: A transaction rereads data it has previously read and finds that another committed transaction has modified or deleted the data.

- Phantom reads (or phantoms): A transaction re-runs a query returning a set of rows that satisfy a search condition and finds that another committed transaction has inserted additional rows that satisfy the condition.

One way of solving potential problems is by locking data when it is being accessed by one user so that it cannot be accessed or changed by another user.

Concurrency is a very important topic for web developers who must be able to create apps that can be used by many (sometimes millions) of users concurrently. It is, therefore, very likely that your prospective employer would expect you to understand the problems related to concurrency and approaches used to address them. The compulsory task can be used to prepare you for this discussion.

BEFORE YOU START THE COMPULSORY TASK

For the next few tasks, you will be asked to create a blog to answer the questions. The reason for this is twofold:

- Blogs are usually "conversational" in the sense that the blogger usually writes in such a way that they are explaining a topic to the reader. Since these interview preparation tasks are meant to help you explain your knowledge of a topic to a prospective employer, a blog is a good format to help you get used to explaining technical concepts to others.
- Your blog post could, in itself, be a useful tool that you could use to impress your potential employer. Imagine this: your employer asks you about concurrency, maybe even something you don't completely understand about the way a specific tool/language supports concurrency, but you can say, "It's interesting that you ask that. I haven't specifically worked with ... but I actually have written a blog that explores concurrency with JavaScript and Python..." Impressive!!

If you don't yet have a blog, you can learn how to create one **here**. If you would like to make your blog private at this stage, please make sure that at least your reviewer has access to your blog by following these instructions **here**.

In this and the next few interview preparation tasks, you are also going to be asked to do some research. Why do you need to do research when you have paid for a Bootcamp to teach you web development? Because we want to live up to our promise to you. This Bootcamp is designed to help you learn, not only how to write the code needed to become a web developer but also to be able to start in your career as a web developer.

You have learned how to use popular and powerful web development languages and tools to become a full stack web developer in this Bootcamp. We have helped you develop core skills and competencies you need to start off on your career. However, web development is a massive and ever-growing and changing field. Any Bootcamp, course, or degree that claims that you know everything you need regarding web development when you are finished is lying to you. As technologies evolve and new tools and techniques for web development become popular, you are going to have to be able to keep up. When you see something unfamiliar in a job description, you will need to be able to research and understand that topic to prepare for the interview. As a web developer, you are going to need to be able to continue learning and researching. We want to help you develop this skill and that is why you are asked to do some research in these interview preparation tasks. Let

us help you hone your research skills and be fully prepared as you get ready to go out into the real world and start working as a web developer.

Compulsory Task 1

Follow these steps:

- Create a blog post called “Concurrency in Web Development” that includes the following:
    - Describes what concurrency is and briefly describes various methods of implementing concurrency. **Recommended reading**.
    - Describes how concurrency is implemented with Node.js. **Recommended reading**.
        - What role does asynchronous programming play in concurrency?
        - How are web APIs related to implementing concurrency?
        - What is the event-loop in JavaScript and how does it relate to concurrency?
    - Compare the way Oracle and MongoDB support database concurrency. Recommended reading: **resource 1** and **resource 2**.

If you are having any difficulties, please feel free to contact our specialist team **on Discord** for support.

Completed the task(s)?

Ask an expert to review your work!

Review work

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.