TASK

Interview Preparation: Concurrency

Visit our website

Introduction

WELCOME TO THE INTERVIEW PREPARATION 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 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 **www.hyperiondev.com/portal** to start a chat with a code reviewer. You can also schedule a call or get support via email.

Our expert code reviewers are 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!

SOFTWARE ENGINEERING INTERVIEWS

Interviews for a Software Engineer position are a little different from other, more “traditional”, interviews. Software engineering interviews generally consist of a practical component where the applicant is asked to do some programming. These interviews evaluate your ability to solve problems on the spot, without any references at your disposal and with someone watching you the whole time. It is, therefore, extremely important for you to be prepared.

The best way to prepare for software engineering 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. Therefore, getting used to answering programming questions in this way can be a huge advantage.

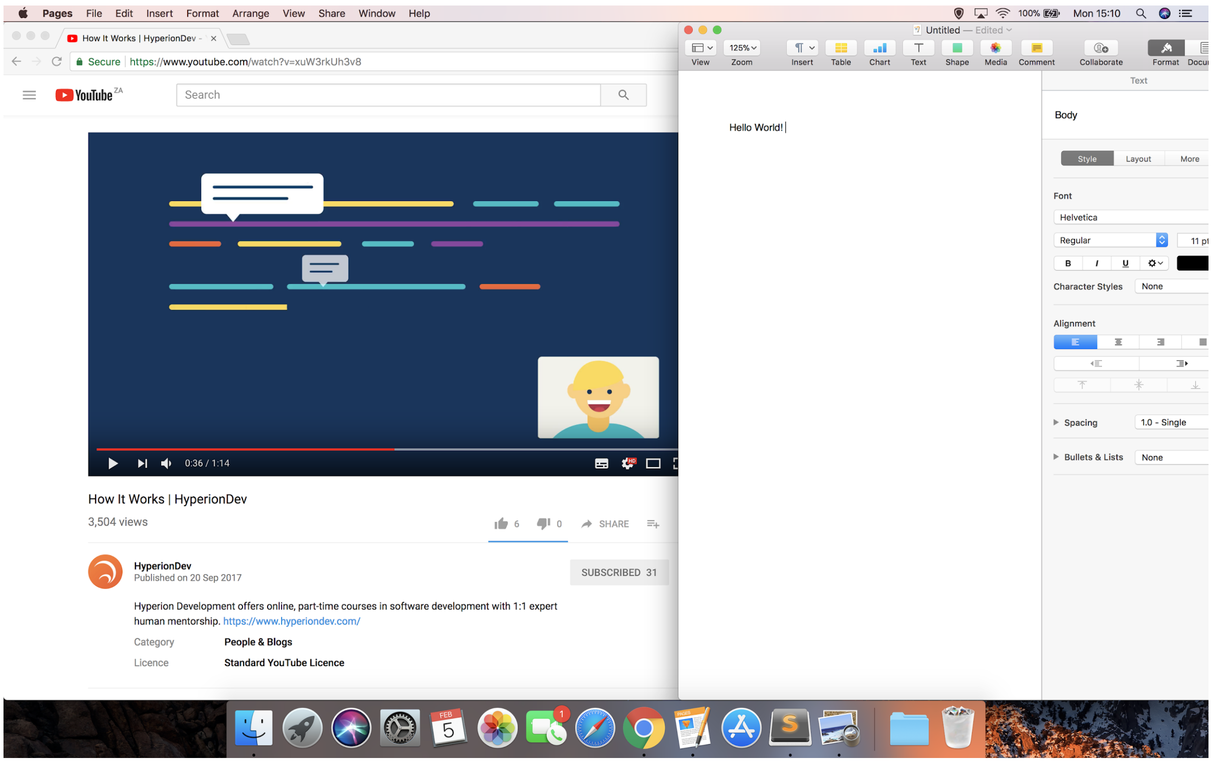
You might be wondering which programming topics are the most important and you should concentrate on. The bad news is, you are equally likely to be asked questions on all programming concepts. Give yourself plenty of time before the interview to prepare. Pick a particular area and try to solve 2 to 3 coding questions per day on that topic. Increase the difficulty of the questions as you go.

In this Bootcamp, we have covered some of the most essential software engineering concepts. However, as a Software Engineer, it is important that you 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 most complex and advanced topics brought up during technical interviews: concurrency.

INTRODUCTION TO CONCURRENCY

Concurrency simply means being able to do more than one thing at the same time. Believe it or not, early computers were only able to do one single task at a time. Now, however, you don’t even think twice about doing two, three or more tasks simultaneously when using a computer. For example, while you work on a document using a word processor, you might also find yourself streaming audio or downloading files at the same time.

The following screenshot shows an example of concurrency. In this screenshot, the user is simultaneously working on a document using the word processor *Pages* and playing a video on *YouTube*.


Pages File Edit Insert Format Arrange View Share Window Help
100% Mon 15:10
How It Works | HyperionDev
Secure https://www.youtube.com/watch?v=xuW3rkUh3v8
YouTube ZA
Search
0:36 / 1:14
6 0 SHARE
SUBSCRIBED 31
Untitled — Edited
125%
View Zoom Insert Table Chart Text Shape Media Comment Collaborate Format Docu
Text
Body
Style Layout More
B I U
Alignment
FEB 5

How It Works | HyperionDev

3,504 views

HyperionDev
Published on 20 Sep 2017

Hyperion Development offers online, part-time courses in software development with 1:1 expert human mentorship. https://www.hyperiondev.com/

Category: People & Blogs

Licence: Standard YouTube Licence

Hello World!

Font: Helvetica, Regular, 11 pt

Character Styles: None

Spacing: 1.0 - Single

Bullets & Lists: None

Even a single application might need to do more than one thing at a time. The word processor that you are using, for example, needs to be ready to respond to keyboard and mouse events, even if it is busy reformatting text or updating the display. Software that can do more than one thing at a time (like your word processor) is known as concurrent software.

So how are computers able to run multiple tasks at the same time? You might be tempted to assume that it is due to a computer having multiple processors. However, this is not the case. Concurrency is possible on a system with only a single processor as well. Computers can also execute more tasks than the number of processors available. How then, can multiple tasks execute at the same time even on a single CPU? It turns out they actually don’t execute at the same physical instant. **Concurrency allows multiple tasks not to execute at the same time, but to make progress during the same period of time.** The tasks are executed in an interleaved manner, with the system switching between the tasks so frequently that it appears as if they are being executed simultaneously.

PROCESSES AND THREADS

The two basic units of concurrency are processes and threads. Before we can define processes and threads, however, we need to understand programs.

You are probably more than familiar with programs at this point. A program is code that is stored on your computer that is intended to perform a specific task. Task-specific programs are also known as applications. Examples of applications include word processors and web browsers. 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 in order to run. Every program needs certain essential resources, namely, registers, a program counter, and a stack. A register is part of the CPU and used to hold data. It can hold an instruction, a storage address, or other kinds 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 a 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. This isolation of processes is a good thing because if there is a problem with one process, the other processes are not affected. You would probably appreciate this process independence if one of your programs freezes unexpectedly and you are able to code without affecting other programs.

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; in this instance, the process and the thread are the same things. 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. These are accomplishing a number of 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 a process, a problem with one thread in a process will affect other threads and the viability of the process as a whole.

This was a brief introduction to concurrency. **Here** is a link to the official Oracle concurrency tutorial. Using this tutorial as well as any other references you might like, answer the questions in the compulsory task below.

BEFORE YOU START THE COMPULSORY TASK

We advise that you create a blog to answer the questions in the compulsory task. 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 Java 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, please wait for your task to be reviewed. Once this is done you can set your blog to private. If you want more information on blog access please look **here**.

Compulsory Task

Answer the following questions:

- What is the difference between process and thread?
- How do you create a thread in Java and run it?
- What are the different states of a thread and when do the state transitions occur?
- What is a daemon thread and what are its use cases?
- How do you create a daemon thread?

- What is Java Memory Model (JMM)?
- What are deadlock, livelock, and starvation? What causes these conditions?
- What happens if you don't override the thread class run() method?
- What is atomic operation and what are atomic classes in the Java Concurrency API?
- What are Executor and ExecutorService and what are the differences between them?
- What are Concurrent Collection Classes?

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.