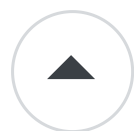


Which concurrent programming concepts do hiring managers expect developers to understand?

Asked 16 years ago Modified 12 years, 11 months ago Viewed 1k times



6



When I hire developers for general mid-to-senior web app development positions, I generally expect them to understand core concurrent programming concepts such as liveness vs. safety, race conditions, thread synchronization and deadlocks. I'm not sure whether to consider topics like fork/join, wait/notify, lock ordering, memory model basics (just the basics) and so forth to be part of what every reasonably seasoned developer ought to know, or whether these are topics that are more for semi-specialists (i.e. developers who have made a conscious decision to know more than the average developer about concurrent programming).

I'd be curious to hear your thoughts.

concurrency

Share

Improve this question

Follow

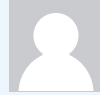
edited Dec 28, 2011 at 20:36



Joel Coehoorn

415k ● 114 ● 577 ● 813

asked Dec 17, 2008 at 6:27



user41871

4 Answers

Sorted by:

Highest score (default)



9



I tend to think that at this point in time concurrent programming at any serious level of depth is still a specialist skill. Many will claim to know about it through study, but many will also make an almighty mess of it when they come to apply it.



In addition to the considerations listed, I would also look at resource implications and the various overheads of using processes, threads and fibers. In some contexts, e.g. mobile devices, excessive multithreading can have serious performance implications. This can lead to portability issues with multithreaded code.

I guess if I was interviewing a candidate in this situation, I would work with a real world example rather than hitting on more general topics which can be quoted back verbatim from a text book. I say this having done a fair bit of multithreaded work myself and remembering how badly I screwed up the first couple of times. Many can talk the talk... ;)

[Share](#) [Improve this answer](#)

[Follow](#)

answered Dec 17, 2008 at 8:54



[SmacL](#)

22.8k ● 15 ● 99 ● 151



4



I know all these topics, but I studied them. I also know many competent senior programmers that don't know these. So unless you expect these programmers to be using those concepts actively, there is no reason to turn down a perfectly good candidate because they don't understand every aspect of concurrency

Share Improve this answer

answered Dec 17, 2008 at 6:38

Follow



Robert Gould

69.7k ● 61 ● 191 ● 275



2



The real question is:

In what ways does it matter to the code they will be developing?

You should know which concepts the development position you're hiring for needs to know to be able to work on the projects that they will be responsible for.

As with anything in the programming world.. The devil is in the details, and you can't know everything. Would you expect them to know Perl if you were hiring for a Java position?

Also, concurrency, at this stage, while well described in generalized theory, is heavily implementation and platform dependent. Concurrency in Perl on an AIX box is not the same game as concurrency in a C++ Winforms app. They can have all the theory in the world under their

belts, but if it's required for the job, then they should have intimate knowledge of the platform they are expected to use it on as well.

Share Improve this answer

answered Dec 18, 2008 at 1:02

Follow



Troy Howard

2,622 ● 21 ● 26



I interview folks for concurrency-related positions frequently and I look for three general aspects:

2



1. General understanding of core concepts like the ones you list (language-independent)
2. Specific understanding of Java concurrency libraries and primitives (specific to the work they'd be doing)
3. Ability to design the solution to a concurrent problem in a reasonable way.



I consider #1 a requirement (for my positions). I consider #2 a nice to have. If they understand it and can describe it in terms of pthreads or whatever other library, it's no biggie to learn the latest Java concurrency libraries (the concepts are the hard part). And #3 tends to separate the hires from the maybe-hires.

Per your question, I wouldn't consider fork/join to be known by almost anyone, esp someone applying for a web app developer position. I would look for developers to have experience with some (but not all) of those topics. Most developers I've interviewed have not used the Java

5+ concurrency libs at all but they can typically describe things like data race or deadlock.

Share Improve this answer

answered Dec 23, 2008 at 5:27

Follow



Alex Miller

70.1k ● 25 ● 124 ● 168
