22 什么是 current 树和 workInProgress 树

更新时间：2020-09-16 09:55:30

> 老骥伏枥，志在千里； 烈士暮年，壮心不已。 ——曹操

前言

上一节介绍了应用程序首次渲染时 React 在 prerender 阶段所做的一些具体工作。从本节开始将要介绍的就是应用程序首次渲染时 React 在 render 阶段所做的工作。在前面文章中多次提到过 workInProgress 树，React 在 render 阶段的其中一项重要工作就是构建 workInProgress 树。那么，current 树和 workInProgress 树到底是什么呢？

current 树和 workInProgress 树均是 Fiber 树

current 树是 fiberRoot 对象上面 current 属性指向的内存空间，实际上就是一个对象，该对象由众多 Fiber 结点连接而成。 current 树描述的是应用程序渲染完成后最终的 Fiber 结构，它反映了用于渲染 UI 的状态。在应用程序首次渲染过程 prerender 阶段执行结束后，fiberRoot 对象的结构如图 5.3.1。

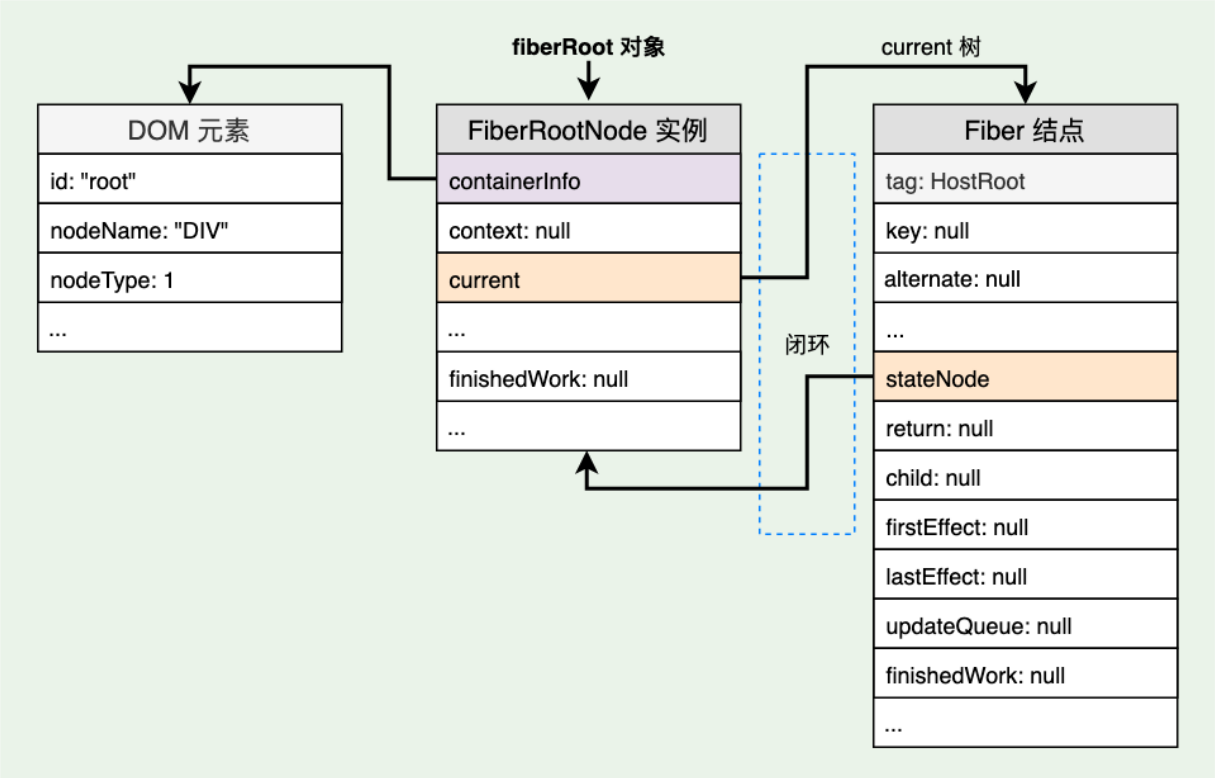

图 5.3.1 fiberRoot 对象初始化完成后的结构

由于是应用程序的首次渲染，在渲染未完成之前 current 树中只有一个 Fiber 结点，该结点也就是 Fiber 树的根结点。那么 workInProgress 树又是什么呢？见图 5.3.2。

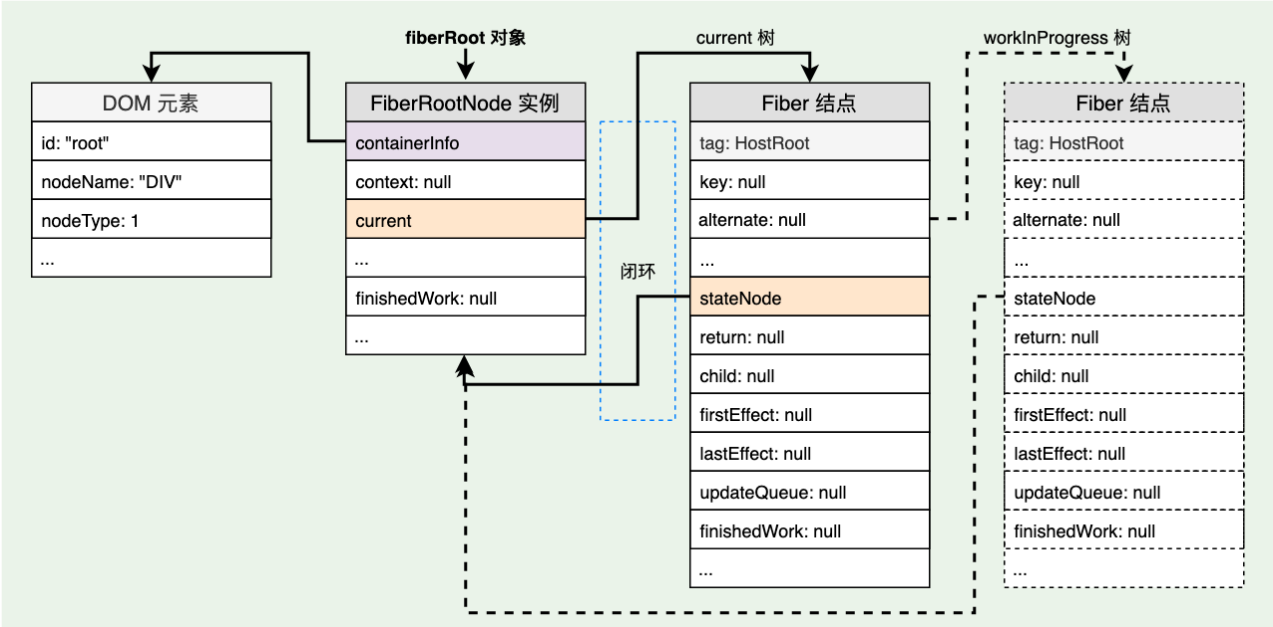

图 5.3.2 fiberRoot 对象初始化完成后 workInProgress 树尚未开始构建

workInProgress 树是 **fiberRoot** 对象上面 current 属性指向的对象上面 alternate 属性指向的内存空间，也是一个对象，在渲染结束后该对象被赋值给 **current** 。也就是说 **workInProgress** 树是 **current** 树的中间形态。React 的核心原则之一是一致性，React 总是一次性更新 DOM （不会显示部分中间结果）。因此，workInProgress 树就充当用户不可见的「草稿」，这样 React 可以先处理所有组件，然后将更新统一刷新到屏幕。下面，我们详细看一下 current 树和 workInProgress 树之间的关系。

current 树与 workInProgress 树之间的关系

所有更新计算相关的工作都在 workInProgress 树的 Fiber 结点上执行。处理完更新并完成所有相关工作后，会得到一个带有更新标识的 Fiber 结点链表，也就是副作用链表，它是 workInProgress 树的子集。React 将副作用链表映射到屏幕上后，workInProgress就会变成 current 树，其中赋值逻辑见代码示例 5.3.1。

```
// fiberRoot.current.alternate为workInProgress树
// 将workInProgress树的引入存入到finishedWork变量中
var finishedWork = fiberRoot.current.alternate;
// 将fiberRoot.current.alternate指向的内存制为null
fiberRoot.current.alternate = null;
// 将workInProgress树赋值给current树
fiberRoot.current = finishedWork;
```

代码示例 5.3.1 渲染完成后 current 树与 workInProgress 树赋值逻辑

workInProgress 树构建完成后 fiberRoot 对象的简化结构见图 5.3.3。

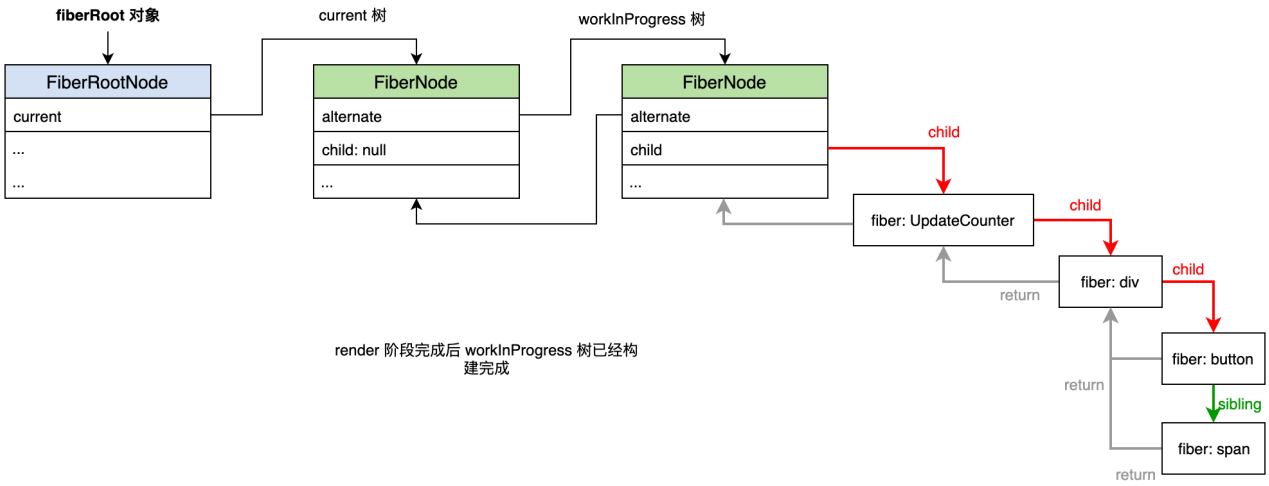

图 5.3.3 render 阶段完成后 fiberRoot 对象后的结构简图

React 应用程序首次渲染时构建 workInProgress 树的过程也是将 React 元素树转化为 Fiber 树的过程，workInProgress 树构建完成后会在该「树」上面完成更新计算、调用生命周期函数以及收集副作用列表等工作。收集好的副作用列表会在 commit 阶段统一映射到屏幕上。至此 workInProgress 树在此次更新历程中已经完成了它的使命，React 会将当前的 current 树设置为 workInProgress 树，将 workInProgress 树置为 null，应用程序首次首次渲染 render 阶段结束后的 fiberRoot 对象结构如图 5.3.4。

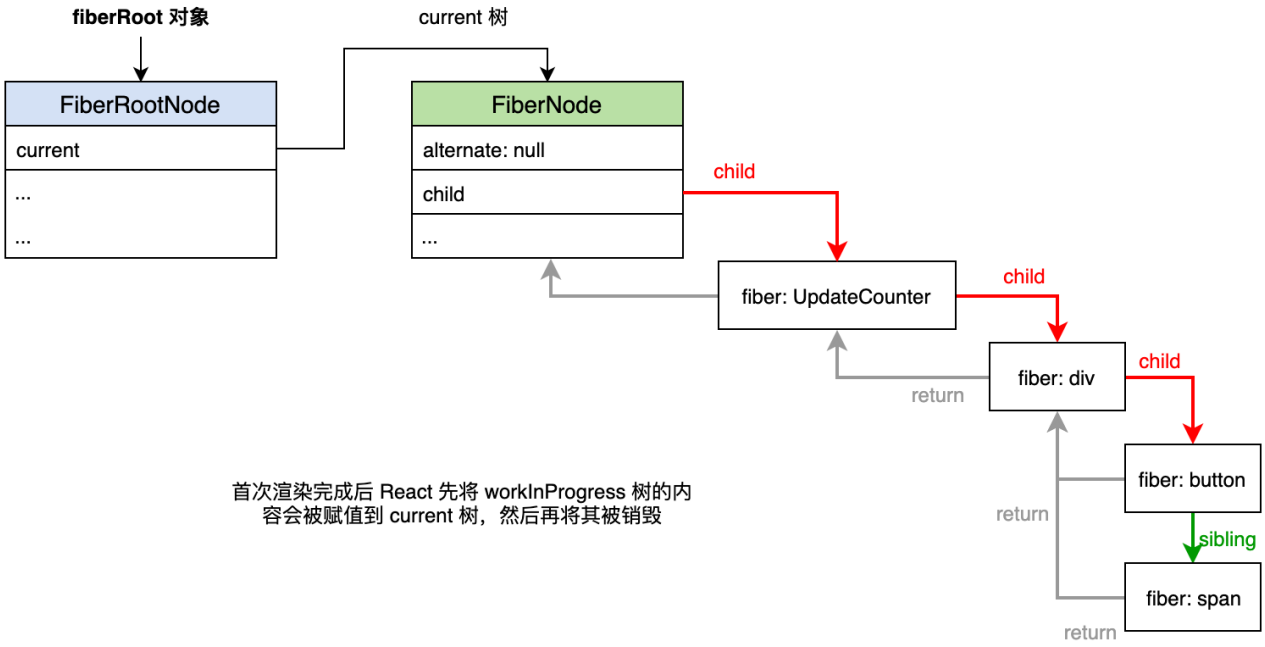

图 5.3.4 应用程序首次渲染完成后 fiberRoot 对象后的结构简图

图 5.3.3 和图 5.3.4 展示了 workInProgress 树从构建完成到销毁这个过程中 fiberRoot 对象数据结构的变化。

小结

本文简单介绍了 React Fiber 架构中的 current 树和 workInProgress 树以及它们之间的关系。应用程序渲染时在 render 阶段的一个重要工作就是构建 workInProgress 树，构建 workInProgress 树的过程中 React 也会完成更新计算、调用生命周期函数以及收集副作用列表等工作。那么，这个过程是什么样的呢？下一节将会介绍 **React 如何构建 workInProgress 树** 。

}

21 React 应用程序首次渲染时 prerender 阶段的工作

23 React 如何构建 workInProgress 树