Sunday • 更新于 2022-10-19

上一节 07：源码阅读：... 09：源码阅读：... 下一节

08：框架实现：场景三：新节点多余旧节点时的 diff 比对

根据上一小节的分析，我们可以直接在 `packages/runtime-core/src/renderer.ts` 中的 `patchKeyedChildren` 方法下，实现如下代码：

代码块

```
// 3. 新节点多余旧节点时的 diff 比对。
if (i > oldChildrenEnd) {
    if (i <= newChildrenEnd) {
        const nextPos = newChildrenEnd + 1
        const anchor =
            nextPos < newChildrenLength ? newChildren[nextPos].el : parentAnchor
        while (i <= newChildrenEnd) {
            patch(null, normalizeVNode(newChildren[i]), container, anchor)
            i++
        }
    }
}
```

创建对应测试实例 `packages/vue/examples/runtime/render-element-diff-3.html`：

代码块

```
<script>
  const { h, render } = Vue

  const vnode = h('ul', [
    h('li', {
      key: 1
    }, 'a'),
    h('li', {
      key: 2
    }, 'b'),
  ])
  // 挂载
  render(vnode, document.querySelector('#app'))

  // 延迟两秒，生成新的 vnode，进行更新操作
  setTimeout(() => {
    const vnode2 = h('ul', [
      h('li', {
        key: 3
      }, 'c'),
      h('li', {
        key: 1
      }, 'a'),
      h('li', {
        key: 2
      }, 'b'),
    ])
    render(vnode2, document.querySelector('#app'))
  }, 2000);
</script>
```

测试成功

07：源码阅读：场景三：新节点多余旧节点... 上一节 下一节 09：源码阅读：场景四：旧节点多于新节点...