

全部开发者教程

行为

22: 源码阅读: renderer 渲染器下, Comment 节点的挂载行为

23: 框架实现: renderer 渲染器下, Comment 节点的挂载行为

24: 源码阅读: renderer渲染器下, Fragment 节点的挂载、更新行为

25: 框架实现: renderer渲染器下, Fragment 节点的挂载、更新行为

26: 总结

第十一章: runtime 运行时 - 组件的设计原理与渲染方案

01: 前言

02: 源码阅读: 无状态基础组件挂载逻辑

03: 框架实现: 完成无状态基



Sunday • 更新于 2022-10-19

上一节 23: 框架实现: ... 25: 框架实现: ... 下一节

24: 源码阅读: renderer渲染器下, Fragment 节点的挂载、更新行为

最后我们来看一下 Fragment 节点的挂载和更新行为。

对于 Fragment 它是一个包裹性质的容器, 本身并不渲染, 只渲染子节点。

那么基于此, 我们创建对应的测试实例 packages/vue/examples/imooc/runtime/renderer-fragment.html :

<> 代码块

```
1 <script>
2   const { h, render, Fragment } = Vue
3
4   const vnode = h(Fragment, 'hello world')
5   // 挂载
6   render(vnode, document.querySelector('#app'))
7
8   setTimeout(() => {
9     const vnode2 = h(Fragment, '你好, 世界')
10    // 挂载
11    render(vnode2, document.querySelector('#app'))
12  }, 2000);
13 </script>
```

根据之前的经验, 我们直接在 patch 方法中进行 debugger , 运行代码进入 debugger :

1. 第一次进入 patch 方法, 执行 挂载 操作:
1. 触发 switch , 执行 processFragment 方法
1. 进入 processFragment 方法, 该方法的代码相比较多, 但是我们忽略掉无用的之后, 逻辑和其他的渲染并无其别, 同样是直接 mountXXX 方法进行挂载
2. 只不过需要注意: 因为 Fragment 本身并不渲染, 所以它的渲染 仅渲染子节点
3. 所以此时触发的渲染方法为: mountChildren
1. 进入 mountChildren , 此时的参数为
- ```
const mountChildren: MountChildrenFn = (
 children, children = "hello world"
 container, container = div#app {align: '', title: '', lang
 anchor, anchor = #text {wholeText: '', assignedSlot: null,
 parentComponent, parentComponent = null
 parentSuspense, parentSuspense = null
 isSVG, isSVG = false
 slotScopeIds, slotScopeIds = null
 optimized, optimized = false
 start = 0 start = 0
) => {
```
2. 该方法内部的逻辑就比较简单了
1. 循环 children , 分别触发 patch 方法进行挂载
2. 因为此时的 children 是一个字符串, 所以循环之后的每一个 child 都是一个字符
3. 把字符通过 normalizeVNode 加工之后, 将得到一个 Text 节点 类型的 VNode
4. 交给 patch 进行渲染即可



2. 至此，**挂载** 完成

3. 第二次进入 `patch`，执行 **更新操作**：

1. 同样通过 `switch`，触发 `processFragment` 方法

1. 进入 `processFragment`

2. 因为是 **更新**，所以会触发 `patchChildren` 方法

1. 进入 `patchChildren`，此时我们知道：

1. 挂载时：渲染的是 `hello world` 文本

2. 更新时：渲染的是 `你好，世界` 文本

2. 所以这次的更新本质上是一个 **文本替换文本** 的操作。

3. 所以执行代码会发现，代码进入 `hostSetElementText(container, c2 as string)` 方法

1. 对于该方法我们是熟悉的，它本质上就是一个 `el.textContent = text` 的方法执行

4. 至此，**更新** 操作完成

由以上代码可知：

1. `Fragment` 只是一个包裹的容器，挂载和更新时，都只会渲染 `children`

2. 根据 `children` 的不同，`Fragment` 的渲染也会存在很多的不同

23: 框架实现: renderer 渲染器下, Comm... ◀ 上一节

下一节 ▶ 25: 框架实现: renderer渲染器下, Fram...

 我要提出意见反馈

