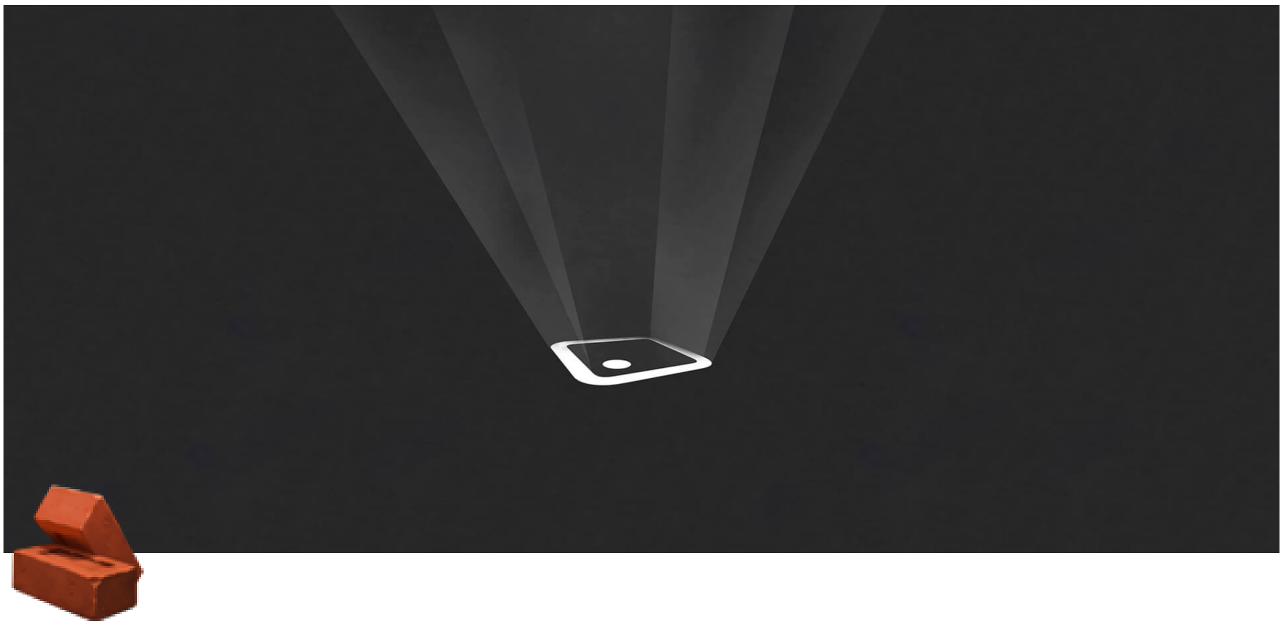

8-5 render 阶段

如果以下代码没有特殊标记路径的源码，那么路径都是 react/packages/react-reconciler/src/ReactFiberWorkLoop.js。

`performConcurrentWorkOnRoot`

这个函数中当中两个重要的阶段，render 和 commit。本节课讲 render~

`renderRootSync`

进入 render 阶段。

1. render 阶段开始

```JavaScript
executionContext |= RenderContex
```

2. workInProgressTransitions 赋值

JavaScript

```javascript
workInProgressTransitions = getTransitionsForLanes(root, lanes);
```

3. `prepareFreshStack` 初始化

初始化 `workInProgressRoot`、`workInProgress`、`workInProgressRootRenderLanes` 等值。

JavaScript

```javascript
function prepareFreshStack(root: FiberRoot, lanes: Lanes): Fiber {
  root.finishedWork = null;
  root.finishedLanes = NoLanes;

  // ...

  workInProgressRoot = root; // FiberRoot
  const rootWorkInProgress = createWorkInProgress(root.current, null);
  workInProgress = rootWorkInProgress; // Fiber
  workInProgressRootRenderLanes = lanes;

  // ...

  // 把concurrentQueues的内容添加到fiber的queue中，即给fiber的lanes、childL
  finishQueueingConcurrentUpdates();

  return rootWorkInProgress;
}
```

4. `workLoopSync`

JavaScript

```
function workLoopSync() {
  while (workInProgress !== null) {
    performUnitOfWork(workInProgress);
  }
}
```

`performUnitOfWork`

```javascript
function performUnitOfWork(unitOfWork: Fiber): void {
  const current = unitOfWork.alternate;

  let next = beginWork(current, unitOfWork, entangledRenderLanes);

  // ! 把pendingProps更新到memoizedProps
  unitOfWork.memoizedProps = unitOfWork.pendingProps;
  if (next === null) {
    // 如果不再产生新的work，那么当前work结束
    completeUnitOfWork(unitOfWork);
  } else {
    workInProgress = next;
  }

  // 当前 Fiber 为null
  ReactCurrentOwner.current = null;
}
```

`beginWork`

1. 更新当前 fiber，比如 props/state 更新，生命周期函数执行、Hooks 函数执行等。
2. 返回一个下一个 fiber。

详情参考下节 `beginWork` 。

`completeUnitOfWork`

深度优先遍历。

`completeWork` 详情参考 `completeWork` 章节。

```JavaScript
function completeUnitOfWork(unitOfWork: Fiber): void {
  let completedWork: Fiber = unitOfWork;
  do {

    const current = completedWork.alternate;
    const returnFiber = completedWork.return;
    let next = completeWork(current, completedWork, entangledRenderLan

    if (next !== null) {
      workInProgress = next;
      return;
    }

    const siblingFiber = completedWork.sibling;
    if (siblingFiber !== null) {
      workInProgress = siblingFiber;
      return;
    }

    completedWork = returnFiber;
    workInProgress = completedWork;
  } while (completedWork !== null);

  if (workInProgressRootExitStatus === RootInProgress) {
    workInProgressRootExitStatus = RootCompleted;
  }
}
```

5. 重置 workInProgressX

这里没有重置 `workInProgress`，因为 `workInProgress` 已经在 `performUnitOfWork` 阶段更新。

```JavaScript
  // 重置Context的相关值，如currentlyRenderingFiber等
  resetContextDependencies();
```

```
// 重置 executionContext
executionContext = prevExecutionContext;

workInProgressRoot = null;
workInProgressRootRenderLanes = NoLanes;
```

6. 再遍历一遍更新队列

JavaScript
```javascript
finishQueueingConcurrentUpdates();
```