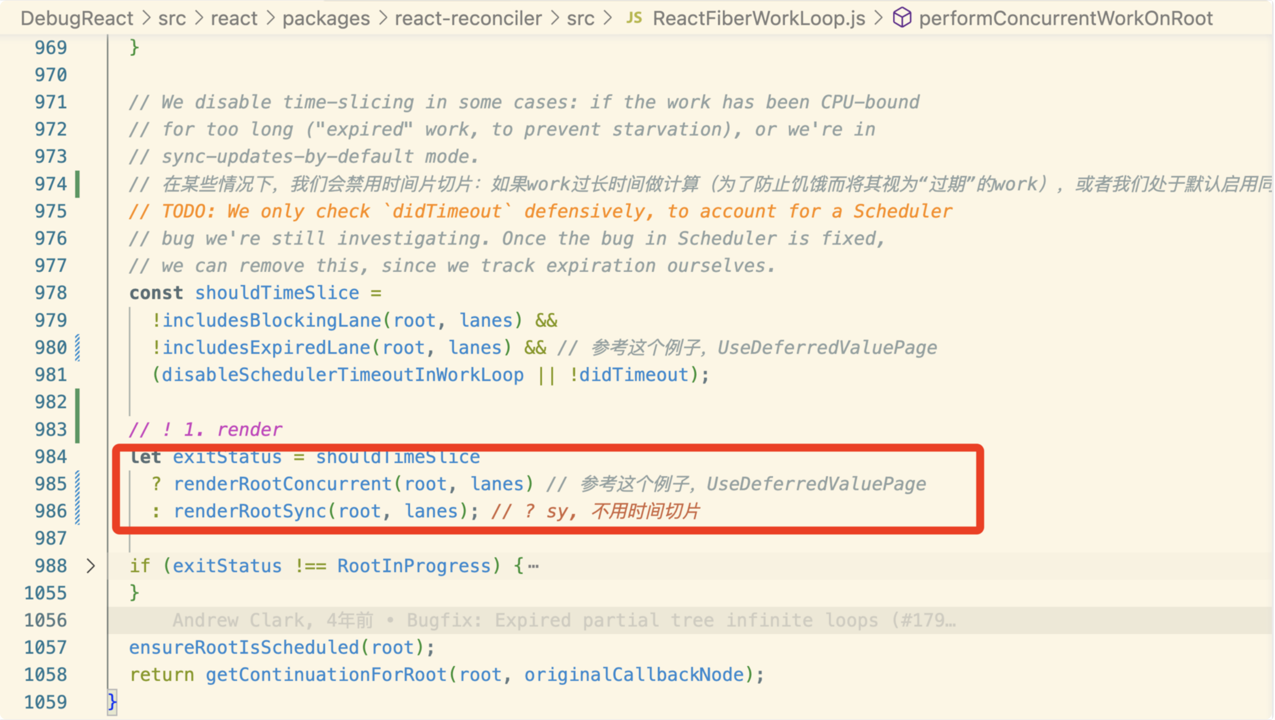
8-8 render阶段-renderRootConcurrent


DebugReact > src > react > packages > react-reconciler > src > JS ReactFiberWorkLoop.js > performConcurrentWorkOnRoot

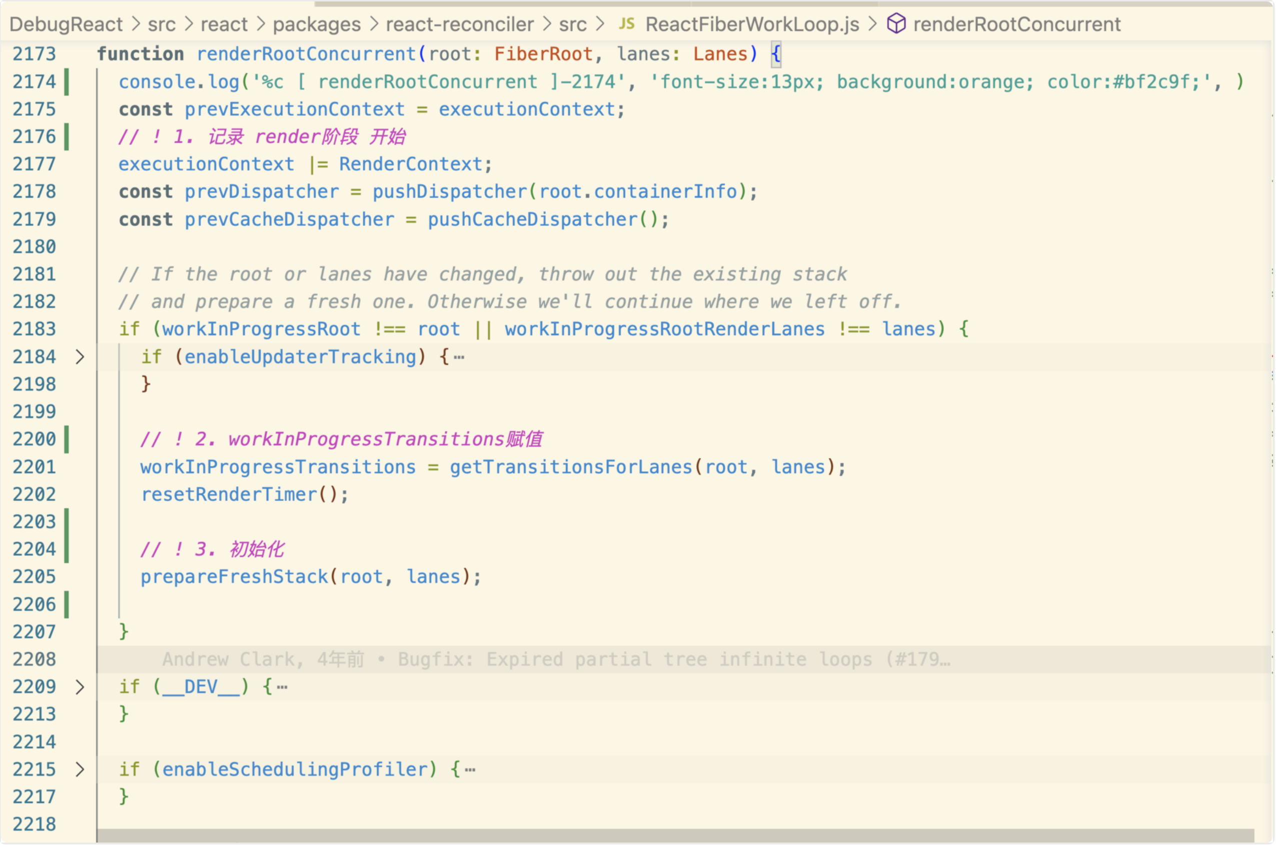
```
  }

  // We disable time-slicing in some cases: if the work has been CPU-bound
  // for too long ("expired" work, to prevent starvation), or we're in
  // sync-updates-by-default mode.
  // 在某些情况下，我们会禁用时间片切片：如果work过长时间做计算（为了防止饥饿而将其视为“过期”的work），或者我们处于默认启用
  // TODO: We only check `didTimeout` defensively, to account for a Scheduler
  // bug we're still investigating. Once the bug in Scheduler is fixed,
  // we can remove this, since we track expiration ourselves.
  const shouldTimeSlice =
    !includesBlockingLane(root, lanes) &&
    !includesExpiredLane(root, lanes) && // 参考这个例子，UseDeferredValuePage
    (disableSchedulerTimeoutInWorkLoop || !didTimeout);

  // ! 1. render
  let exitStatus = shouldTimeSlice
    ? renderRootConcurrent(root, lanes) // 参考这个例子，UseDeferredValuePage
    : renderRootSync(root, lanes); // ? sy, 不用时间切片

  if (exitStatus !== RootInProgress) {…
  }
      Andrew Clark, 4年前 • Bugfix: Expired partial tree infinite loops (#179…
  ensureRootIsScheduled(root);
  return getContinuationForRoot(root, originalCallbackNode);
}
```

renderRootConcurrent



DebugReact > src > react > packages > react-reconciler > src > JS ReactFiberWorkLoop.js > renderRootConcurrent

```
function renderRootConcurrent(root: FiberRoot, lanes: Lanes) {
  console.log('%c [ renderRootConcurrent ]-2174', 'font-size:13px; background:orange; color:#bf2c9f;', )
  const prevExecutionContext = executionContext;
  // ! 1. 记录 render阶段 开始
  executionContext |= RenderContext;
  const prevDispatcher = pushDispatcher(root.containerInfo);
  const prevCacheDispatcher = pushCacheDispatcher();

  // If the root or lanes have changed, throw out the existing stack
  // and prepare a fresh one. Otherwise we'll continue where we left off.
  if (workInProgressRoot !== root || workInProgressRootRenderLanes !== lanes) {
    if (enableUpdaterTracking) {…
    }

    // ! 2. workInProgressTransitions赋值
    workInProgressTransitions = getTransitionsForLanes(root, lanes);
    resetRenderTimer();

    // ! 3. 初始化
    prepareFreshStack(root, lanes);

  }
      Andrew Clark, 4年前 • Bugfix: Expired partial tree infinite loops (#179…
  if (__DEV__) {…
  }

  if (enableSchedulingProfiler) {…
  }

```

workLoopConcurrent

JavaScript

```javascript
function workLoopConcurrent() {
  // 执行work，直到Scheduler要求让出
  while (workInProgress !== null && !shouldYield()) {
    performUnitOfWork(workInProgress);
  }
}
```