5-4 如何实现任务调度函数入口

callback、task、work

callback 是任务的初始值，task 是 scheduler 封装的 task，work 是指一个时间切片内的工作单元。

scheduleCallback

TypeScript

```
//标记task的唯一性
let taskIdCounter = 1;

// 主线程是否在调度
let isHostCallbackScheduled = false;


function scheduleCallback(priorityLevel: PriorityLevel, callback: Call
  // todo
  //任务进入调度器的时间
  const startTime = getCurrentTime();
  let timeout: number;
```

```
  switch (priorityLevel) {
    case ImmediatePriority:
      // 立即超时
      timeout = -1;
      break;
    case UserBlockingPriority:
      // 最终超时
      timeout = userBlockingPriorityTimeout;
      break;
    case IdlePriority:
      // 永不超时
      timeout = maxSigned31BitInt;
      break;
    case LowPriority:
      // 最终超时
      timeout = lowPriorityTimeout;
      break;
    case NormalPriority:
    default:
      // 最终超时
      timeout = normalPriorityTimeout;
      break;
  }

  const expirationTime = startTime + timeout;

  const newTask: Task = {
    id: taskIdCounter++,
    callback,
    priorityLevel,
    startTime,
    expirationTime,
    sortIndex: -1,
  };

  newTask.sortIndex = expirationTime;
  push(taskQueue, newTask);

  // Schedule a host callback, if needed. If we're already performing
  // wait until the next time we yield.
  if (!isHostCallbackScheduled && !isPerformingWork) {
```

```
    isHostCallbackScheduled = true;
    requestHostCallback();
  }
}
```