



(12) 发明专利

(10) 授权公告号 CN 113254023 B

(45) 授权公告日 2023.08.11

(21) 申请号 202110531800.6

(22) 申请日 2021.05.14

(65) 同一申请的已公布的文献号
申请公布号 CN 113254023 A

(43) 申请公布日 2021.08.13

(73) 专利权人 网易(杭州)网络有限公司
地址 310052 浙江省杭州市滨江区长河街
道网商路599号4幢7层

(72) 发明人 杨泽伟

(74) 专利代理机构 北京超凡宏宇专利代理事务
所(特殊普通合伙) 11463
专利代理师 钟扬飞

(51) Int.Cl.
G06F 8/41 (2018.01)

(56) 对比文件
US 2009172653 A1,2009.07.02
US 6757362 B1,2004.06.29

US 2013247017 A1,2013.09.19

CN 1842767 A,2006.10.04

CN 102521008 A,2012.06.27

CN 111722951 A,2020.09.29

CN 112379917 A,2021.02.19

CN 108647029 A,2018.10.12

US 2013139134 A1,2013.05.30

CN 110196790 A,2019.09.03

JP 2016177691 A,2016.10.06

CN 108268332 A,2018.07.10

US 2002059568 A1,2002.05.16

CN 106919370 A,2017.07.04

CN 111209004 A,2020.05.29

US 2019310835 A1,2019.10.10

「已注销」.前端练习04 取对象的深层属性.
《<https://blog.csdn.net/duola8789/article/details/84238242>》.2018,全文.

审查员 冯泽

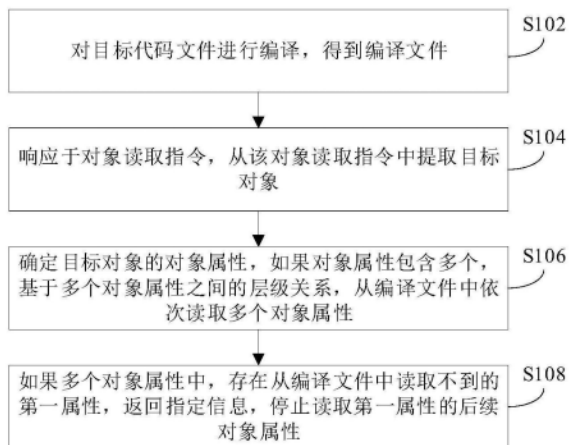
权利要求书3页 说明书13页 附图4页

(54) 发明名称

对象读取方法、装置和电子设备

(57) 摘要

本发明提供了一种对象读取方法、装置和电子设备,首先编译目标代码文件得到编译文件;响应于对象读取指令,基于对象读取指令确定需读取的目标对象的对象属性,如果对象属性包含多个,基于多个对象属性间的层级关系,从编译文件中依次读取对象属性;如果多个对象属性中,存在从编译文件中读取不到的属性,返回指定信息,停止读取该属性的后续对象属性。该方式在读取属性包含有多个层级关系的对象时,如果读不到某一属性,返回指定信息并跳出读取进程,避免因读取进程报错而产生的页面白屏;同时,相对于使用if-else语句对属性进行多层判断赋值的方式,该方式中编译文件的代码更加简单,降低了代码的冗余度和维护难度,提高了代码开发效率。



1. 一种对象读取方法,其特征在于,所述方法包括:
 - 对目标代码文件进行编译,得到编译文件;
 - 响应于对象读取指令,从所述对象读取指令中提取目标对象;
 - 确定所述目标对象的对象属性,如果所述对象属性包含多个,基于多个对象属性之间的层级关系,从所述编译文件中依次读取所述多个对象属性;
 - 如果所述多个对象属性中,存在从所述编译文件中读取不到的第一属性,返回指定信息,停止读取所述第一属性的后续对象属性;
 - 所述对目标代码文件进行编译,得到编译文件的步骤,包括:
 - 从所述目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的目标节点;其中,所述代码语法结构包括多个节点,每个节点包括所述目标代码文件中的一部分代码片段对应的对象的对象属性,以及每个所述对象属性之间的层级关系;
 - 基于所述目标节点中每个所述对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组;
 - 响应针对于所述属性数组的读取规则的插入操作,将所述读取规则与所述目标节点合并,得到最终的目标节点;其中,所述读取规则包括:基于所述属性数组依次读取每个所述对象属性,当未读取到所述属性数组中的第二属性时,返回指定信息,停止读取所述第二属性的后续对象属性;
 - 将所述最终的目标节点替换为所述代码语法结构中的所述目标节点,得到替换后的代码语法结构,将所述替换后的代码结构转换为编译文件。
2. 根据权利要求1所述的方法,其特征在于,所述基于所述目标节点中每个所述对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组的步骤,包括:
 - 从所述目标节点中提取对象的各个对象属性,以及每个所述对象属性之间的层级关系;
 - 通过预设的分割函数或者正则表达式,基于所述每个所述对象属性之间的层级关系,得到用于表征属性读取顺序的链表结构;将所述链表结构组合成属性数组。
3. 根据权利要求2所述的方法,其特征在于,从所述目标节点中提取对象的各个对象属性,以及每个所述对象属性之间的层级关系的步骤,包括:
 - 根据预设的所述对象属性之间的链接方式,从所述目标节点中确定所述对象的各个对象属性,以及每个所述对象属性之间的层级关系。
4. 根据权利要求1所述的方法,其特征在于,所述代码语法结构中还包含有每个所述节点对应的类型信息;所述从所述目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的目标节点的步骤,包括:
 - 从所述代码语法结构中,确定与对象链式调用的类型信息相匹配的节点;其中,所述对象链式调用的方式用于读取对象属性包含有多个层级关系的对象;
 - 将确定的节点作为所述目标节点。
5. 根据权利要求1或4所述的方法,其特征在于,所述目标代码文件的代码语法结构,通过下述方式建立:
 - 通过babel的词法分析扫描所述目标代码文件中的每一个代码字符,生成所述目标代码文件中每个代码片段对应的标识信息;

组合每个所述标识信息,得到标识数组;

根据预设的语法规则,通过babel的语法分析算法将所述标识数组中的每个标识信息转化为包含有对象的每个对象属性之间的层级关系的节点,基于所述节点生成代码语法结构。

6. 根据权利要求5所述的方法,其特征在于,所述标识信息中包含有所述代码片段对应的类型信息和数值;其中,所述类型信息中包括所述代码片段在所述目标代码文件中的位置和对象状态;所述对象状态用于指示代码片段对应的对象的对象属性,以及每个所述对象属性之间的层级关系。

7. 根据权利要求1所述的方法,其特征在于,所述将所述替换后的代码结构转换为编译文件的步骤,包括:

通过预设的编译器,将所述替换后的代码结构翻译成最终的目标代码文件,将所述最终的目标代码文件作为所述编译文件。

8. 根据权利要求1所述的方法,其特征在于,所述停止读取所述第一属性的后续对象属性的步骤之后,所述方法还包括:

停止读取所述多个对象属性在所述编译文件中对应的代码片段,继续执行所述编译文件中停止读取的所述代码片段的下一个代码片段。

9. 一种对象读取装置,其特征在于,所述装置包括:

代码编译模块,用于对目标代码文件进行编译,得到编译文件;

对象提取模块,用于响应于对象读取指令,从所述对象读取指令中提取目标对象;

对象读取模块,用于确定所述目标对象的对象属性,如果所述对象属性包含多个,基于多个对象属性之间的层级关系,从所述编译文件中依次读取所述多个对象属性;

读取停止模块,用于如果所述多个对象属性中,存在从所述编译文件中读取不到的第一属性,返回指定信息,停止读取所述第一属性的后续对象属性;

所述代码编译模块,还用于从所述目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的目标节点;其中,所述代码语法结构包括多个节点,每个节点包括所述目标代码文件中的一部分代码片段对应的对象的对象属性,以及每个所述对象属性之间的层级关系;

基于所述目标节点中每个所述对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组;

响应针对于所述属性数组的读取规则的插入操作,将所述读取规则与所述目标节点合并,得到最终的目标节点;其中,所述读取规则包括:基于所述属性数组依次读取每个所述对象属性,当未读取到所述属性数组中的第二属性时,返回指定信息,停止读取所述第二属性的后续对象属性;

将所述最终的目标节点替换为所述代码语法结构中的所述目标节点,得到替换后的代码语法结构,将所述替换后的代码结构转换为编译文件。

10. 一种电子设备,其特征在于,包括处理器和存储器,所述存储器存储有能够被所述处理器执行的机器可执行指令,所述处理器执行所述机器可执行指令以实现权利要求1至8任一项所述的对象读取方法。

11. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机可

执行指令,所述计算机可执行指令在被处理器调用和执行时,所述计算机可执行指令促使所述处理器实现权利要求1至8任一项所述的对象读取方法。

对象读取方法、装置和电子设备

技术领域

[0001] 本发明涉及代码编译执行技术领域,尤其是涉及一种对象读取方法、装置和电子设备。

背景技术

[0002] 在JavaScript中,前端请求接口通常会返回一份JSON格式的JavaScript对象,该对象通常包含多个层级的属性。由于该JavaScript对象的多层级结构的未知性可能引发对象读取错误,又由于JavaScript的执行栈是单线程模式,当一个代码片段中的JavaScript对象出现读取报错就会导致后续JavaScript代码都无法运行,造成页面功能大面积瘫痪,在用户角度来看就是页面白屏。页面白屏是一种非常不友好的用户体验,且会导致网站的跳出率急剧上升,可能再短时间内造成大量用户流失。

[0003] 为了避免读取JavaScript对象的多层级属性而导致的页面白屏现象,相关技术中通常使用if-else语句对JavaScript对象的属性进行多层判断赋值,若判断某个属性已存在则继续进入下一个if-else语句,再次判断子属性是否存在,若存在则再次进入下一个if-else语句,否则停止执行其他if-else语句。但是多层判断赋值会让整体的JavaScript代码变得非常臃肿和难以维护,若JavaScript对象被修改可能需要重构全部或部分if-else语句,影响代码开发效率。

发明内容

[0004] 本发明的目的在于提供一种对象读取方法、装置和电子设备,以避免读取JavaScript对象的多层级属性而导致的页面白屏现象。

[0005] 第一方面,本发明提供了一种对象读取方法,该方法包括:对目标代码文件进行编译,得到编译文件;响应于对象读取指令,从对象读取指令中提取目标对象;确定目标对象的对象属性,如果对象属性包含多个,基于多个对象属性之间的层级关系,从编译文件中依次读取多个对象属性;如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。

[0006] 在可选的实施方式中,上述对目标代码文件进行编译,得到编译文件的步骤,包括:从目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的目标节点;其中,该代码语法结构包括多个节点,每个节点包括目标代码文件中的一部分代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系;基于目标节点中每个对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组;响应针对于属性数组的读取规则的插入操作,将读取规则与目标节点合并,得到最终的目标节点;其中,该读取规则包括:基于属性数组依次读取每个对象属性,当未读取到属性数组中的第二属性时,返回指定信息,停止读取第二属性的后续对象属性;将最终的目标节点替换为代码语法结构中的目标节点,得到替换后的代码语法结构,将替换后的代码结构转换为编译文件。

[0007] 在可选的实施方式中,上述基于目标节点中每个对象属性之间的层级关系,得到

用于表征对象属性的读取顺序的属性数组的步骤,包括:从目标节点中提取对象的各个对象属性,以及每个对象属性之间的层级关系;通过预设的分割函数或者正则表达式,基于每个对象属性之间的层级关系,得到用于表征属性读取顺序的链表结构;将链表结构组合成属性数组。

[0008] 在可选的实施方式中,上述从目标节点中提取对象的各个对象属性,以及每个对象属性之间的层级关系的步骤,包括:根据预设的所述对象属性之间的链接方式,从目标节点中确定对象的各个对象属性,以及每个对象属性之间的层级关系。

[0009] 在可选的实施方式中,上述代码语法结构中还包含有每个节点对应的类型信息;上述从目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的目标节点的步骤,包括:从代码语法结构中,确定与对象链式调用的类型信息相匹配的节点;其中,对象链式调用的方式用于读取对象属性包含有多个层级关系的对象;将确定的节点作为目标节点。

[0010] 在可选的实施方式中,上述目标代码文件的代码语法结构,通过下述方式建立:通过babel的词法分析扫描目标代码文件中的每一个代码字符,生成目标代码文件中每个代码片段对应的标识信息;组合每个标识信息,得到标识数组;根据预设的语法规则,通过babel的语法分析算法将标识数组中的每个标识信息转化为包含有对象的每个对象属性之间的层级关系的节点,基于节点生成代码语法结构。

[0011] 在可选的实施方式中,上述标识信息中包含有代码片段对应的类型信息和数值;其中,类型信息中包括代码片段在目标代码文件中的位置和对象状态;该对象状态用于指示代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系。

[0012] 在可选的实施方式中,上述将替换后的代码结构转换为编译文件的步骤,包括:通过预设的编译器,将替换后的代码结构翻译成最终的目标代码文件,将最终的目标代码文件作为编译文件。

[0013] 在可选的实施方式中,上述停止读取第一属性的后续对象属性的步骤之后,该方法还包括:停止读取多个对象属性在编译文件中对应的代码片段,继续执行编译文件中停止读取的代码片段的下一个代码片段。

[0014] 第二方面,本发明提供了一种对象读取装置,该装置包括:代码编译模块,用于对目标代码文件进行编译,得到编译文件;对象提取模块,用于响应于对象读取指令,从对象读取指令中提取目标对象;对象读取模块,用于确定目标对象的对象属性,如果对象属性包含多个,基于多个对象属性之间的层级关系,从编译文件中依次读取多个对象属性;读取停止模块,用于如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。

[0015] 第三方面,本发明提供了一种电子设备,该电子设备包括处理器和存储器,该存储器存储有能够被处理器执行的机器可执行指令,该处理器执行机器可执行指令以实现前述实施方式任一项所述的对象读取方法。

[0016] 第四方面,本发明提供了一种计算机可读存储介质,该计算机可读存储介质存储有计算机可执行指令,该计算机可执行指令在被处理器调用和执行时,该计算机可执行指令促使处理器实现前述实施方式任一项所述的对象读取方法。

[0017] 本发明实施例带来了以下有益效果:

[0018] 本发明提供了一种对象读取方法、装置和电子设备,首先对目标代码文件进行编译得到编译文件;然后响应于对象读取指令,基于对象读取指令确定需读取的目标对象的对象属性,如果对象属性包含多个,基于多个对象属性间的层级关系,从编译文件中依次读取多个对象属性;如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。该方式在读取属性包含有多个层级关系的对象时,如果读取不到某一属性,直接返回指定信息并跳出读取进程,避免了因读取进程报错而产生的页面白屏现象;同时,相对于使用if-else语句对属性进行多层判断赋值的方式,该方式中编译文件的代码更加简单,降低了代码的冗余度和维护难度,提高了代码开发效率。

[0019] 本发明的其他特征和优点将在随后的说明书中阐述,或者,部分特征和优点可以从说明书推知或毫无疑义地确定,或者通过实施本发明的上述技术即可得知。

[0020] 为使本发明的上述目的、特征和优点能更明显易懂,下文特举较佳实施方式,并配合所附附图,作详细说明如下。

附图说明

[0021] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0022] 图1为本发明实施例提供了一种对象读取方法的流程图;

[0023] 图2为本发明实施例提供的另一种对象读取方法的流程图;

[0024] 图3为本发明实施例提供了一种代码编译的流程图;

[0025] 图4为本发明实施例提供了一种对象读取装置的结构示意图;

[0026] 图5为本发明实施例提供了一种电子设备的结构示意图。

具体实施方式

[0027] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。

[0028] 因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0029] 在JavaScript中,前端请求接口通常会返回一份JSON(JavaScript Object Notation,JS对象简谱)格式的JavaScript对象,该对象通常包含多个层级的属性。如下代码:

```
constobj={
  code: 200,
  data: {
    educationalInfo: {
      course: [ “性能优化”,
                “设计模式”,
                “数据算法” ],
      grade: 2011,
      major: “计算机系统”,
      school: “中山大学” },
      personalInfo: {
        age: 28,
        name: “Lu way” ,
        sex: “male” }
    },
    msg: “接口请求成功” };
```

[0031] 基于上述代码,如果想要获取该同学的第三门课程“数据算法”,则需要通过obj.data.educationalInfo.course[2]的方式,读取对象的多层属性,这种方式也可以称为链式调用的方式。其中,obj.data.educationalInfo.course中包含有对象obj的多个层级的属性,字段data、educationalInfo和course分别作为对象的一个属性,在代码中该属性的冒号后面对应的值为该属性的属性信息(也可称为属性值),例如,属性course的属性信息包括性能优化、设计模式和数据算法。在具体实现时,如果代码中的对象因某些原因不包含有course字段,如下述代码,那么在读取对象时将出现报错,也即是通过链式调用读取“数据算法”这门课程时就会在浏览器里报错,报错的信息可以为:Uncaught TypeError: Cannot read property '2' of undefined。代码如下:


```
constobj={
  code: 200,
  data: {
    educationalInfo: {
      grade: 2011,
      major: “计算机系统”,
[0032] school: “中山大学” },
      personalInfo: {
        age: 28,
        name: “Lu way” ,
        sex: “male” }
    },
    msg: “接口请求成功”};
```

[0033] 在实际应用中,由于JavaScript的执行栈是单线程模式,当一个代码片段报错就会导致该代码片段的后续JavaScript代码都无法运行,造成页面功能大面积瘫痪,在用户角度来看就是页面白屏。而且,不管是接口返回的JavaScript对象,还是定义变量的JavaScript对象,由于其属性的多层结构(相当于多层级关系)的未知性都可能引发上述读取报错的问题,而报错问题的到会,将引起页面功能大面积瘫痪,导致页面局部空白或全部空白。

[0034] 为了避免读取JavaScript对象的多层级属性而导致的页面白屏现象,相关技术中通常使用if-else语句对对象的多个属性进行多层判断赋值,若判断某个属性已存在则继续进入下一个if-else语句,再次判断子属性是否存在,若存在则再次进入下一个if-else语句,否则停止执行其他if-else语句。

[0035] 但是使用该方式存在以下缺点:

[0036] 1、原本通过链式调用多层级属性“obj.data.educationalInfo.course[2]”的简单形式,为了兼容报错问题而硬性使用if-else语句进行多层判断赋值,导致代码量大大增加,使最终输出的JavaScript代码体积增大,间接导致页面首屏渲染时间加长,从而影响用户体验。而且增加的if-else语句越多,这种现象越明显,从而又会引发新一轮的性能优化问题。

[0037] 2、大量使用if-else语句进行多层判断赋值,会让源码变得非常臃肿并难以维护。而且,当新增一个获取某个属性的需求时,又需在众多的if-else语句中寻找,大大降低了开发效率。

[0038] 3、若后端修改了接口返回的对象的多层级属性而没有告知前端的情况下,原本正常运行的页面在用户刷新后立马变成页面白屏,这种问题产生的根本原因是后端接口的多层级属性修改了,前端之前编写的if-else判断逻辑已局部失效,导致JavaScript对象多层读取产生问题。

[0039] 4、大量使用if-else语句会造成代码块多层嵌套,会引发编译性能降低的问题。

[0040] 基于上述问题,本发明实施例提供了一种对象读取方法、装置和电子设备,该技术可以应用在链式调用读取对象的场景中。为便于对本实施例进行理解,首先对本发明实施例所公开的一种对象读取方法进行详细介绍,如图1所示,该方法包括如下步骤:

[0041] 步骤S102,对目标代码文件进行编译,得到编译文件。

[0042] 前端项目开发都以工程化为主,开发者需要先将开发代码打包成最终的生产代码,也即是目标代码文件编译为编译文件,其中,该目标代码文件中包含的是开发代码,编译文件中包含的是生产代码。

[0043] 在一些实施例中,可以使用Webpack脚本对目标代码文件中的开发代码进行打包,生成最终的生产代码。在对开发代码进行打包的过程中,需要构建代码以生成最终的生产代码,在构建代码时对生产代码中对象包含有多个对象属性,且对象属性间存在多层级关系的代码片段进行预处理,以使在后续读取该代码片段对应的对象的对象属性时,如果某一属性不存在,则返回指定信息,并停止读取该属性的后续对象属性;对代码片段的预处理也可以理解为在该代码片段中插入读取该代码片段对应的对象的对象属性的读取规则,将插入的读取规则与开发代码进行整合,即可得到生产代码。

[0044] 在具体实现时,上述代码通常为JavaScript代码,在JavaScript代码中对象通常是拥有属性和方法的数据集合,该对象可以定义为以“key:value”对的形式出现的相关数据的无序集合,包括基本类型或引用类型,这些key可以是变量或函数,在对象上下文中分别称为属性和方法。可以使用带有可选属性列表的花括号{...}创建对象。属性是“key:value”对,其中键key是字符串(也称为属性名称),值value(也可以称为属性值或者属性信息等)可以是任何值。例如,如下代码中,对象school的属性包含有name、location和established,“ABC School”、“Delhi”和“1971”分别为各个属性对应的属性值。代码如下:

```
letschool={  
    name: “ABCSchool”,  
    location: “Delhi”,  
    established: “1971”  
};
```

[0045] 步骤S104,响应于对象读取指令,从该对象读取指令中提取目标对象。

[0047] 上述对象读取指令可以是在执行编译文件时,用户通过电子设备触发的指令,也可以是前端请求接口返回的指令。根据该对象读取指令可以确定编译文件中当前需要读取的某一代码片段对应的目标对象,该目标对象可以包括一个或者多个对象属性(相当于上述属性),当包含有多个对象属性时,多个对象属性间可以存在多个层级关系,也可以仅存在在一个层级关系,具体可依据代码的实际编写确定。

[0048] 步骤S106,确定目标对象的对象属性,如果对象属性包含多个,基于多个对象属性之间的层级关系,从编译文件中依次读取多个对象属性。

[0049] 上述对象读取指令中通常携带有目标对象的对象属性,如果对象属性包含有多个,相邻的两个对象属性之间通过预设的链接方式链接,具体的链接方式可以根据研发需求设置,例如,可以“.”或者“[]”作为链接方式。具体地,如果对象读取指令中携带的目标对象的对象属性的链接方式为“obj.data.educationalInfo.course[2]”,其中,字段data、educationalInfo和course均为对象obj的对象属性,而且对象属性之间的连接顺序也可以表征多个对象属性之间的层级关系,多个对象属性之间的层级关系通常指示了多个对象属性的读取顺序,上述目标对象对应的对象属性的读取顺序依次为:首先读取obj,进而读取data,再读取educationalInfo,然后读取course。

[0050] 步骤S108,如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。

[0051] 在从编译文件中,根据目标对象的多个对象属性之间的层级关系依次读取多个对象属性时,如果未读取到某一对象属性(相当于上述第一属性),则直接返回指定信息,并停止读取多个对象属性中该对象属性的后续对象属性。其中,上述指定信息可以是编译文件中插入的读取规则中的字符,该字符可以根据研发需求设定,例如,可以设置为undefined或者null等。上述第一属性通常是指在从编译文件中依次读取多个对象属性时,从编译文件中第一个读取不到的对象属性,该第一属性可以是多个对象属性中的任意一个。

[0052] 现有技术中在读取目标对象的多个对象属性时,如果某一对象属性未读取到将进行报错,从而导致页面出现白屏现象;而本发明在未读取到某一属性时,仍会返回指定信息,并不会进行报错,而是可以停止读取该目标对象,继续执行该目标对象的下一个代码片段,从而可以避免页面白屏现象。

[0053] 本发明实施例提供的一种对象读取方法,首先对目标代码文件进行编译得到编译文件;然后响应于对象读取指令,基于对象读取指令确定需读取的目标对象的对象属性,如果对象属性包含多个,基于多个对象属性间的层级关系,从编译文件中依次读取多个对象属性;如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。该方式在读取属性包含有多个层级关系的对象时,如果读取不到某一属性,直接返回指定信息并跳出读取进程,避免了因读取进程报错而产生的页面白屏现象;同时,相对于使用if-else语句对属性进行多层判断赋值的方式,该方式中编译文件的代码更加简单,降低了代码的冗余度和维护难度,提高了代码开发效率。

[0054] 本发明实施例还提供了另一种对象读取方法,该方法在上述实施例方法的基础上实现;该方法重点描述对目标代码文件进行编译,得到编译文件的具体过程(通过下述步骤S202-S208实现),如图2所示,该方法包括如下具体步骤:

[0055] 步骤S202,从目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的节点;其中,该代码语法结构包括多个节点,每个节点包括目标代码文件中的一部分代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系。

[0056] 上述目标代码文件的代码语法结构是通过目标代码文件中的代码进行解析后得到的,解析方法可以根据研发需求设定,例如,可以选用抽象语法树或者其他解析方法。在具体实现时,上述目标代码文件的代码语法结构,可以通过下述步骤10-12建立:

[0057] 步骤10,通过babel的词法分析扫描目标代码文件中的每一个代码字符,生成目标代码文件中每个代码片段对应的标识信息。

[0058] 上述babel通常是一个Javascript编译器,它可以解析高版本Javascript语法代码,生成向后兼容的低版本Javascript代码。在具体实现时,可以使用babel解析目标代码文件中的所有JavaScript代码,并通过babel的词法分析扫描代码中的每一个字符,依据目标代码文件中的代码结构将代码分割成多个代码片段,并依据分词法则生成每个代码片段对应的标识信息(也可以称为Token)和去除代码中注释的空格,该Token可以描述代码片段在整个代码中的位置和记录当前值的一些信息,例如,某一个对象属性的属性值等。

[0059] 具体地,上述标识信息中包含有代码片段对应的类型信息(可以称为type值)和数值;其中,类型信息中包括代码片段在目标代码文件中的位置和对象状态;该对象状态用于指示代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系。因而,在一些实施例中,可以通过类型信息从代码语法结构中寻找目标节点。

[0060] 步骤11,组合每个标识信息,得到标识数组。

[0061] 在具体实现时,可以按照预定的规则合并目标代码文件中每个代码片段对应的标识信息,得到标识数组,也即是该标识数组中包括目标代码文件所对应的所有的标识信息。

[0062] 步骤12,根据预设的语法规则,通过babel的语法分析算法将标识数组中的每个标识信息转化为包含有对象的每个对象属性之间的层级关系的节点,基于节点生成代码语法结构。

[0063] 上述语法规则通常是JavaScript语法规则,通过babel的语法分析可以解析标识数组,并依据JavaScript语法规则生成代码语法结构。该代码语法结构中包含了目标代码文件中每个代码片段对应的标识信息对应的节点。

[0064] 在一些实施例中,上述代码语法结构还可以用抽象语法树(Abstract Syntax Tree,简称AST)表示,该抽象语法树可以以树状的形式表现编程语言的语法结构。其中,上述节点为抽象语法树上的节点,这个节点也可以称为AST片段,而且该抽象语法树上的每个节点都表示源代码中的一种结构。

[0065] 在具体实现时,由于代码语法结构中还包含有每个节点对应的类型信息;因而,在确定目标节点时,可以从代码语法结构中,确定与对象链式调用的类型信息相匹配的节点;其中,对象链式调用的方式用于读取对象属性包含有多个层级关系的对象;将确定的节点作为目标节点。该确定方式可以快速便捷地定位目标节点。

[0066] 例如,像obj.data.educationalInfo.course[2]这种对象链式调用的代码换成代码语法结构的节点后,该节点的类型信息为Identifier,因此可从代码语法结构的节点中查找类型信息为Identifier的节点,并将查询到的节点作为目标节点。

[0067] 步骤S204,基于目标节点中每个对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组。

[0068] 多个对象属性之间的层级关系表征了对象属性的读取顺序,例如,对象属性1是对象属性2的下一个层级,那么在读取对象属性时,需要先读取对象属性1,再读取对象属性2。上述属性数组可以用一维向量的形式表示,该一维向量中的每一个元素都代表了一个对象属性,并且元素的排序顺序与对象的读取顺序一致。在具体实现时,上述步骤S204可以通过下述步骤20-21实现:

[0069] 步骤20,从目标节点中提取对象的各个对象属性,以及每个对象属性之间的层级关系。

[0070] 在目标节点中,代码片段对应的对象的各个对象属性是通过预设的链接方式进行连接的,通过预设的链接方式可以表征各个对象属性之间的层级关系,也可以理解为根据预设的对象属性之间的链接方式,从目标节点中确定对象的各个对象属性,以及每个对象属性之间的层级关系。例如,可以将“.”或者“[]”作为预设的链接方式,此时,从目标节点中提取出的对象的各个对象属性,以及每个对象属性之间的层级关系可以为obj.data.educationalInfo.course[2]这段字符串,其中,obj、data、educationalInfo和course均表示对象的对象属性,各个属性之间的连接顺序表示对象属性之间的层级关系。

[0071] 步骤21,通过预设的分割函数或者正则表达式,基于每个对象属性之间的层级关系,得到用于表征属性读取顺序的链表结构;将链表结构组合成属性数组。

[0072] 在具体实现时,可以通过分割函数split()或者正则表达式从每个对象属性之间的层级关系“obj.data.educationalInfo.course”中提取出链表结构“obj=>data=>educationalInfo=>course=>2”,并组合成[obj,data,educationalInfo,course,2]这样的一维属性数组。此时,属性数组中的首个成员是目标对象的整体,然后基于该属性数组可以逐渐往下递归读取对象属性,也即是根据属性数组中每个属性对象的元素位置依次读取对象属性。

[0073] 步骤S206,响应针对于属性数组的读取规则的插入操作,将读取规则与目标节点合并,得到最终的目标节点;其中,该读取规则包括:基于属性数组依次读取每个对象属性,当未读取到属性数组中的第二属性时,返回指定信息,停止读取第二属性的后续对象属性。

[0074] 上述第二属性通常是指在从预设代码中依次读取属性数组中的每个对象属性的过程中,第一个未读取到的对象属性,该第二属性可以是属性数组中的任意一个对象属性。在具体实现时,上述读取规则通常是以代码的形式进行插入的,然后将该读取规则与目标节点进行合并,得到最终的目标节点,也可以理解为将读取规则对应的语法结构与目标节点进行合并处理,形成新的目标节点,从而可以完成读取规则对应的代码与目标节点对应的原始代码片段的整合。其中,新的目标节点中包含有原始代码片段对应的语法结构,以及读取规则对应的代码片段的语法结构。

[0075] 步骤S208,将最终的目标节点替换为代码语法结构中的目标节点,得到替换后的代码语法结构,将替换后的代码结构转换为编译文件。

[0076] 在具体实现时,将最终的目标节点替换为代码语法结构中的目标节点后,可以得到替换后的代码语法结构,然后通过预设的编译器(例如,可以是babel),将替换后的代码结构翻译成最终的目标代码文件,将最终的目标代码文件作为编译文件。

[0077] 步骤S210,响应于对象读取指令,从该对象读取指令中提取目标对象。

[0078] 目标代码文件编译完成后,如果接收到对象读取指令,需要基于对象读取指令,从编译文件中确定需要读取的目标对象,以读取该目标对象。

[0079] 步骤S212,确定目标对象的对象属性,如果对象属性包含多个,基于多个对象属性之间的层级关系,从编译文件中依次读取多个对象属性。

[0080] 步骤S214,如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。

[0081] 在对象目标对象的过程中,如果对象属性包含多个,多个对象属性之间存在多层次关系,在读取多个对象属性时,如果存在从编译文件中读取不到的第一属性,需要执行上述步骤S206插入的读取规则,也即是如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。该第一属性通常是指在从编译文件中依次读取多个对象属性时,从编译文件中第一个读取不到的对象属性。

[0082] 在具体实现时,在根据对象读取指令从编译文件中,读取对象属性包含有多个层级关系的对象时,可以根据插入的上述读取规则进行读取,以使在从编译文件中读取不到某一个对象属性时,直接返回指定信息,并同时停止读取该对象属性后续的对象属性,也即是停止该读取进程。

[0083] 步骤S216,停止读取多个对象属性在编译文件中对应的代码片段,继续执行编译文件中停止读取的代码片段的下一个代码片段。

[0084] 本发明的方式中,当停止读取多个对象属性在编译文件中对应的代码片段时,并不会影响后续代码片段的读取,从而该方式可以避免页面白屏现象。

[0085] 为了便于对本发明实施例进行理解,下面以目标代码文件的代码语法结构用抽象语法树进行表示,对本发明实施例中的代码编译进行详细介绍,如图3所示为一种代码编译的流程图。图3中所示的代码编译过程(也可以称为打包过程)是基于Webpack脚本完成的,首先启动Webpack脚本,然后构建代码,代码构建完成后再关闭Webpack脚本,从而得到编译后的代码。其中,本发明是在代码构建过程中插入的读取规则,首先进行步骤P001,启动本发明提供的脚手架,也即是初始化本发明技术方案的实例,然后解析目标代码文件中的Javascript代码,即实现步骤P002:通过babel的词法分析扫描目标代码文件中的Javascript代码的每一个字符,依据分词法则去除代码中的空格和注释,并生成Token(相当于上述标识信息),也可以理解为分割出代码对应的词法单元,将Token组合成Token数组(相当于上述标识数组),将Token数组标记为A。

[0086] 然后执行步骤P003:通过babel的语法分析解析Token数组,依据JavaScript语法规则生成抽象语法树,标记为B。再执行步骤P004:找到符合对象链式调用属性的AST片段(相当于上述目标节点)。将符合条件的AST片段放到链式递归读取流程中处理,待处理结果返回后将其合并到原来的AST片段中,生成新的抽象语法树,标记为C。其中,上述链式递归读取流程包括:从AST片段中提取对象的各个对象属性,以及每个对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组;然后响应针对于属性数组的读取规则的插入操作,插入读取规则。进而执行步骤P005:babel依据抽象语法树结构翻译成最终的JavaScript代码,得到编译文件,从而完成代码构建过程。最后执行步骤P006:关闭本发明提供的脚手架,也即是进入Webpack进程的其他构建阶段(若存在其他构建阶段)。

[0087] 上述对象读取方法,在读取属性不存在时直接返回执行信息并跳出读取进程,不会因进程报错而产生页面白屏现象,让用户看到的问题页面还能保证大部分功能的正常使用,在保留用户体验的同时为开发者争取了一定的时间修复Bug;而且该方式减少了大量的if-else语句,使代码变得更加整洁干净,优化了代码的冗余和臃肿,使最终的生产代码质量更小加载更快;同时,该方式在底层编译方面,可以修复对象读取的底层编译缺陷,使开发人员在编码方面继续保持原先的对象读取写法。另外,该方式可以更加安全优雅高效地读取JavaScript对象的多层属性,不会因为数据结构修改或不存在而导致页面白屏,使得

编码时更加优雅地书写对象链式调用。

[0088] 对应于上述方法实施例,本发明实施例还提供了一种对象读取装置,如图4所示,该装置包括:

[0089] 代码编译模块40,用于对目标代码文件进行编译,得到编译文件。

[0090] 对象提取模块41,用于响应于对象读取指令,从该对象读取指令中提取目标对象。

[0091] 对象读取模块42,用于确定目标对象的对象属性,如果该对象属性包含多个,基于多个对象属性之间的层级关系,从编译文件中依次读取多个对象属性。

[0092] 读取停止模块43,用于如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。

[0093] 上述对象读取装置,首先对目标代码文件进行编译得到编译文件;然后响应于对象读取指令,基于对象读取指令确定需读取的目标对象的对象属性,如果对象属性包含多个,基于多个对象属性间的层级关系,从编译文件中依次读取多个对象属性;如果多个对象属性中,存在从编译文件中读取不到的第一属性,返回指定信息,停止读取第一属性的后续对象属性。该方式在读取属性包含有多个层级关系的对象时,如果读取不到某一属性,直接返回指定信息并跳出读取进程,避免了因读取进程报错而产生页面白屏;同时,相对于使用if-else语句对属性进行多层判断赋值的方式,该方式中编译文件的代码更加简单,降低了代码的冗余度和维护难度,提高了代码开发效率。

[0094] 具体地,上述代码编译模块,包括:节点确定模块,用于从目标代码文件的代码语法结构中,确定对象属性包含有多个层级关系的节点;其中,该代码语法结构包括多个节点,每个节点包括目标代码文件中的一部分代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系;数组生成模块,用于基于目标节点中每个对象属性之间的层级关系,得到用于表征对象属性的读取顺序的属性数组;规则插入模块,用于响应针对于属性数组的读取规则的插入操作,将读取插入规则与目标节点合并,得到最终的目标节点;其中,该读取规则包括:基于属性数组依次读取每个对象属性,当未读取到属性数组中的第二属性时,返回指定信息,停止读取第二属性的后续对象属性;代码转换模块,用于将最终的目标节点替换为代码语法结构中的目标节点,得到替换后的代码语法结构,将替换后的代码结构转换为编译文件。

[0095] 进一步地,上述数组生成模块,用于:从目标节点中提取对象的各个对象属性,以及每个对象属性之间的层级关系;通过预设的分割函数或者正则表达式,基于每个对象属性之间的层级关系,得到用于表征属性读取顺序的链表结构;将该链表结构组合成属性数组。

[0096] 在具体实现时,上述数组生成模块,还用于:根据预设的对象属性之间的链接方式,从目标节点中确定对象的各个对象属性,以及每个对象属性之间的层级关系。

[0097] 进一步地,上述代码语法结构中还包括有每个节点对应的类型信息;上述节点确定模块,还用于:从代码语法结构中,确定与对象链式调用的类型信息相匹配的节点;其中,该对象链式调用的方式用于读取对象属性包含有多个层级关系的对象;将确定的节点作为目标节点。

[0098] 进一步地,上述装置还包括代码语法结构建立模块,用于:通过babel的词法分析扫描目标代码文件中的每一个代码字符,生成目标代码文件中每个代码片段对应的标识信

息;组合每个标识信息,得到标识数组;根据预设的语法规则,通过babel的语法分析算法将标识数组中的每个标识信息转化为包含有对象的每个对象属性之间的层级关系的节点,基于节点生成代码语法结构。

[0099] 在具体实现时,上述标识信息中包含有代码片段对应的类型信息和数值;其中,该类型信息中包含代码片段在目标代码文件中的位置和对象状态;该对象状态用于指示代码片段对应的对象的对象属性,以及每个对象属性之间的层级关系。

[0100] 进一步地,上述代码转换模块,还用于:通过预设的编译器,将替换后的代码结构翻译成最终的目标代码文件,将最终的目标代码文件作为编译文件。

[0101] 进一步地,上述装置还包括代码跳转模块,用于:在停止读取第一属性的后续对象属性之后,停止读取多个对象属性在编译文件中对应的代码片段,继续执行编译文件中停止读取的代码片段的下一个代码片段。

[0102] 本发明实施例所提供的对象读取装置,其实现原理及产生的技术效果和前述对象读取方法实施例相同,为简要描述,装置实施例部分未提及之处,可参考前述方法实施例中相应内容。

[0103] 本发明实施例还提供了一种电子设备,参见图5所示,该电子设备包括处理器101和存储器100,该存储器100存储有能够被处理器101执行的机器可执行指令,该处理器执行机器可执行指令以实现上述对象读取方法。

[0104] 进一步地,图5所示的电子设备还包括总线102和通信接口103,处理器101、通信接口103和存储器100通过总线102连接。

[0105] 其中,存储器100可能包含高速随机存取存储器(RAM,Random Access Memory),也可能还包括非不稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。通过至少一个通信接口103(可以是有线或者无线)实现该系统网元与至少一个其他网元之间的通信连接,可以使用互联网,广域网,本地网,城域网等。总线102可以是ISA总线、PCI总线或EISA总线等。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示,图5中仅用一个双向箭头表示,但并不表示仅有一根总线或一种类型的总线。

[0106] 处理器101可能是一种集成电路芯片,具有信号的处理能力。在实现过程中,上述方法的各步骤可以通过处理器101中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器101可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(Digital Signal Processor,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)、现场可编程门阵列(Field-Programmable Gate Array,简称FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0107] 结合本发明实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器,闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器100,处理器101读取存储器100中的信息,结合其硬件完成前述实施例的方法的步骤。

[0108] 本发明实施例还提供一种机器可读存储介质,该机器可读存储介质存储有机器可执行指令,该机器可执行指令在被处理器调用和执行时,该机器可执行指令促使处理器实现上述对象读取方法,具体实现可参见方法实施例,在此不再赘述。

[0109] 本发明实施例所提供的对象读取方法、装置和电子设备的计算机程序产品,包括存储了程序代码的计算机可读存储介质,所述程序代码包括的指令可用于执行前面方法实施例中所述的方法,具体实现可参见方法实施例,在此不再赘述。

[0110] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,电子设备,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0111] 最后应说明的是:以上所述实施例,仅为本发明的具体实施方式,用以说明本发明的技术方案,而非对其限制,本发明的保护范围并不局限于此,尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化,或者对其中部分技术特征进行等同替换;而这些修改、变化或者替换,并不使相应技术方案的本质脱离本发明实施例技术方案的精神和范围,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应所述以权利要求的保护范围为准。

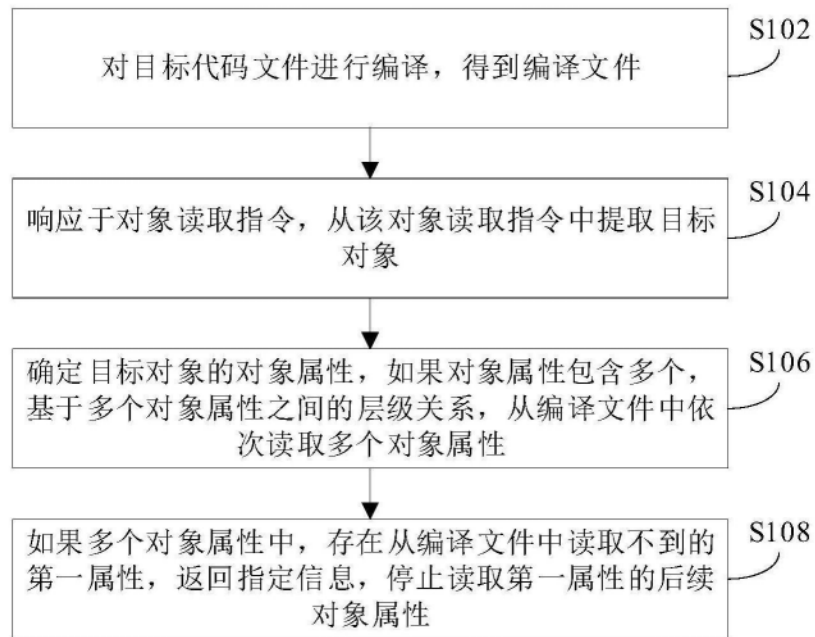


图1

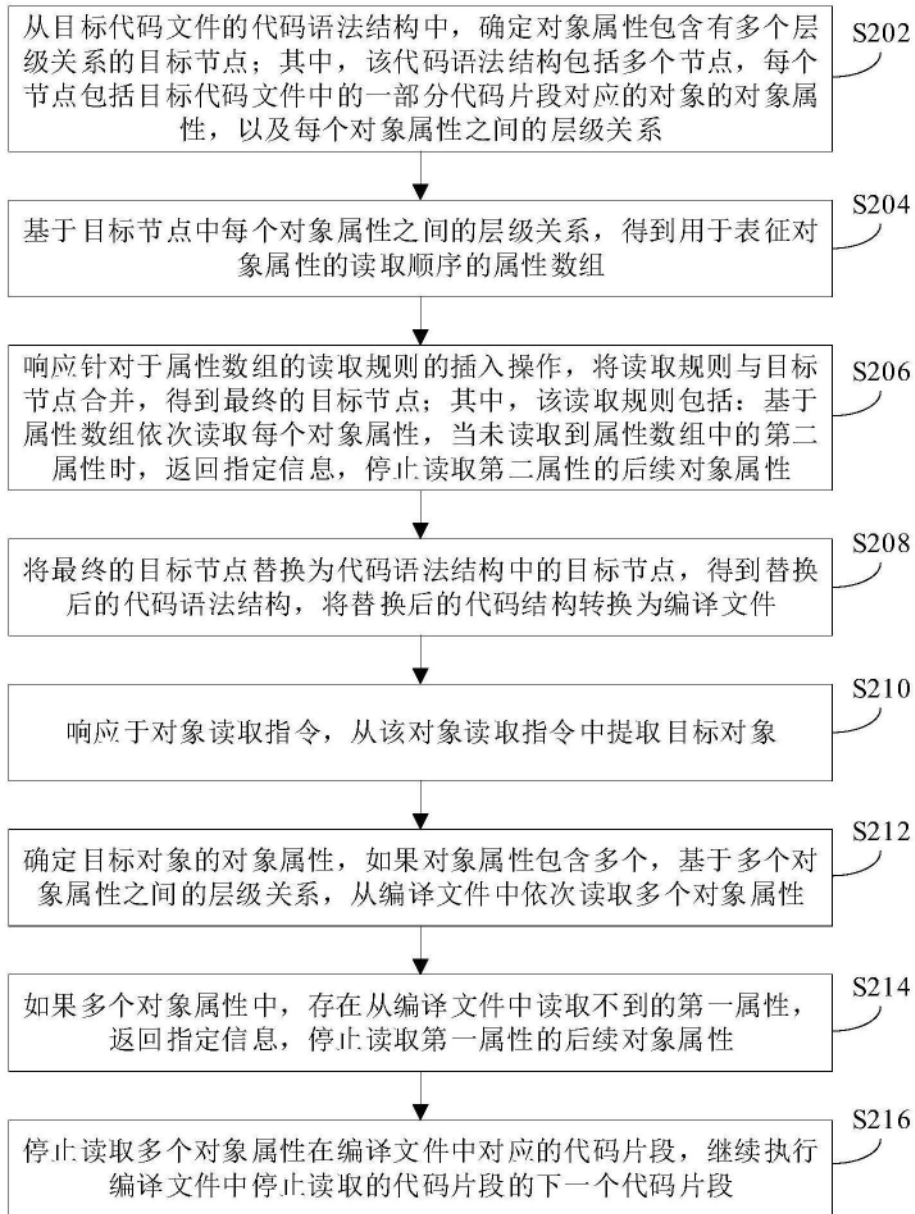


图2

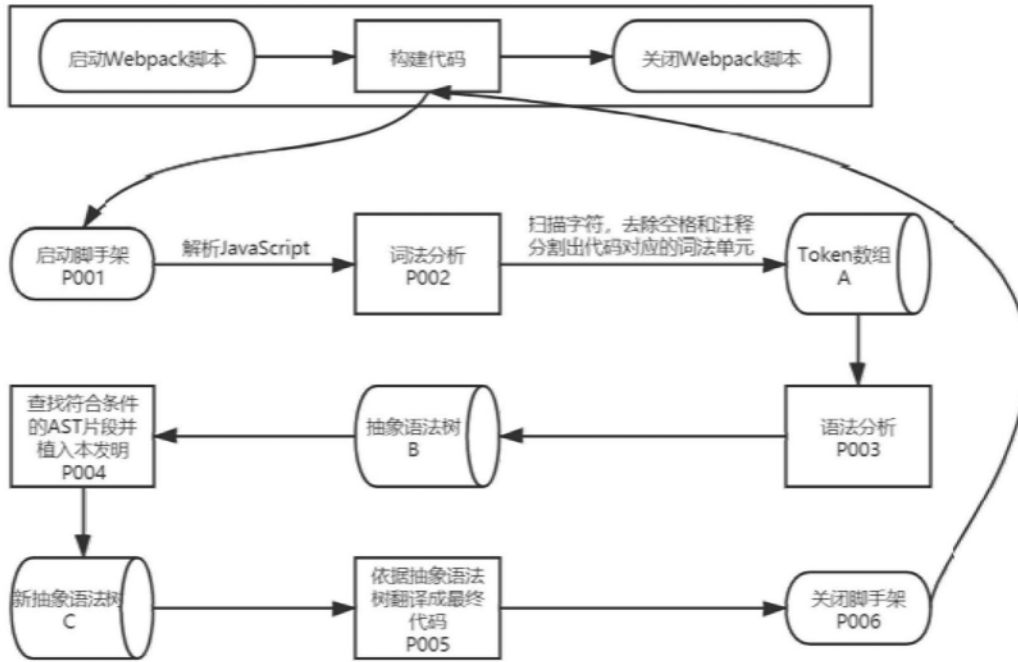


图3



图4

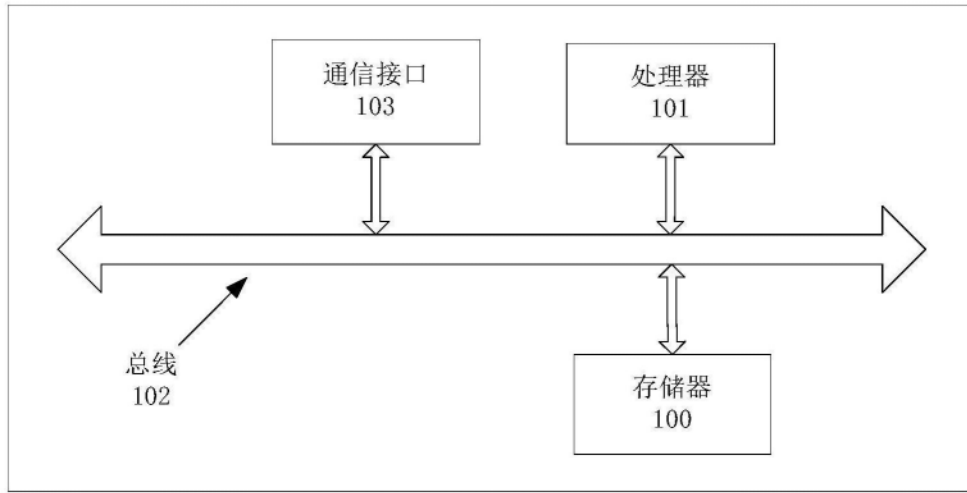


图5