

# 基于 Linux 的嵌入式浏览器的研究

石功恒, 陈建勋

(武汉科技大学 计算机科学与技术学院, 湖北 武汉 430065)

**摘要:** 对 Linux 操作系统的嵌入式浏览器相关技术进行了研究与分析。分析并阐述嵌入式浏览器开发的技术, 主要包括嵌入式浏览器的结构、HTML 词法分析、JavaScript 支持等。总结了嵌入式浏览器开发的技术要点和策略。

**关键词:** 嵌入式浏览器; 词法分析; JavaScript 支持

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2011)10-0095-03

## The research of the embedded browser based on Linux

Shi Gongheng, Chen Jianxun

(College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)

**Abstract:** This paper explains mainly the research and analysis of the embedded browser based on Linux operating system. It is focused on the technology of embedded browser development. Embedded browser structure, HTML lexical analysis, JavaScript support are analyzed in detail. And it summarizes the important technology and the policy of embedded browser development.

**Key words:** embedded browser; lexical analysis; JavaScript support

随着信息技术与网络技术的快速发展, 以及人们日常生活对网络信息的需求的不断增加。掌上电脑、上网本、手机、导航仪器、智能电视、智能家电等对嵌入式系统提出了更多更高的要求。在嵌入式系统中利用嵌入式浏览器获取网络信息是必不可少的, 因此嵌入式浏览器技术成为现如今的技术热点之一。而且 Linux 操作系统完全开源, 可以降低产品开发成本, 因此对基于 Linux 操作系统的嵌入式浏览器开发技术的研究有着重要意义。

### 1 嵌入式浏览器的结构

根据嵌入式浏览器的<sup>[1-2]</sup>特点, 分析嵌入式浏览器的各个部分功能。下面介绍嵌入式浏览器的主要功能: (1)词法分析, 主要是对 HTML 语言进行词法分析, 并且要求 HTML 词法分析可以兼容 XML 等其他标签语言, 达到只需更换语言的元素名称就可以进行词法分析的效果; (2)语法分析与布局, 在非嵌入式浏览器中语法分析是独立的一个模块, 但是在嵌入式浏览器中不作为一个独立模块, 简化系统结构, 即实现语法分析与布局同步化, 节省了语法分析数据存储的内存空间; (3)嵌入式浏览器的用户界面, 设计高效快捷的用户界面是应用软件的重要部分; (4)嵌入式浏览器 JavaScript 支持,

JavaScript 在浏览器里主要是用于操纵界面元素并与浏览器内部命令进行结合; (5)I/O 端, 主要是通过 HTTP 协议和文本系统来读取页面信息; (6)控制机制, 负责命令响应和消息处理。嵌入式浏览器功能结构如图 1 所示。

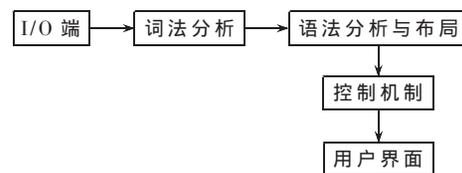


图 1 嵌入式浏览器功能结构图

### 2 HTML 词法分析

#### 2.1 组成及功能

HTML 词法分析是嵌入式浏览器开发<sup>[3]</sup>的基础, 是整个嵌入式浏览器整体设计的最前端部分, HTML 词法分析的数据结构是语法分析与布局的重要依据。HTML 词法分析的效率、准确率以及容错率的高低在一定程度上决定了整个嵌入式浏览器的工作效率。下面对 HTML 词法分析的组成和功能进行解析。

HTML 词法分析的功能主要是负责接收从 I/O 端读取到的 HTML 代码并进行词法分析, 提取网页中 HTML 代码中的各元素信息, 按照定义的数据结构保存元素的

## 应用奇葩

Example of Application

属性和名称等,实现数据的结构化。

根据 HTML 词法分析的功能可以解析为几个组成部分:(1)初始化。完成对数据结构的初始化,主要是分配内存,变量赋初值;(2)主体的数据流分析,逐字符地进行判断,确定数据的归属类型;(3)元素的分析,提取元素的名称、属性和值域;(4)释放,主要是对数据内存空间的释放。

## 2.2 算法

根据 HTML 词法分析的功能以及组成部分,可以很清晰地得到一个简单的算法。其算法过程具体如下:

- (1)从 I/O 端读取的字符串流中,顺序读取一个字符;
- (2)如果遇到<,表示将遇到元素,处理该元素(使用元素处理函数来处理该元素),元素处理结束后,指针指向该元素的尾部;
- (3)如果遇到空格或回车,则跳过;
- (4)如果遇到>,则跳过(因为不可能存在<>这种不包含任何元素的情况,只是为了增强程序的容错性);
- (5)如果遇到的不是上述几种情况,则遇到的是文字(使用文字处理函数来处理该文字),文字处理结束后,指针指向下一个元素的首部;
- (6)循环(1)~(5)步骤,直到 I/O 端读取的字符串流全部分析完。

根据上述词法分析算法过程可以得到相应的流程图,如图 2 所示。

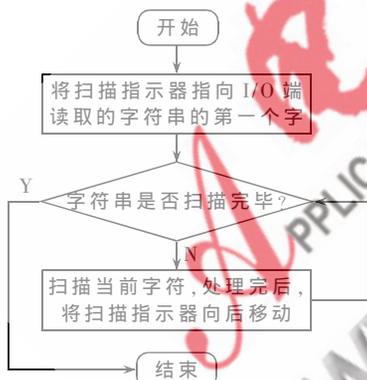


图 2 词法分析算法流程图

## 2.3 词法分析的设计

根据上述提出的词法分析的算法,通过扫描 I/O 端读取的字符串,来检测字符是否为标签元素、提取标签元素属性等。设计词法分析的基础要定义数据结构,为词法分析算法的实现提供基础。根据功能需要设计基本的数据结构,定义如下:

```

char token[]; //标签元素表
char * tokenPointer; //指向标签元素表的指针
char chText[]; //字符表
char *chPointer; //字符表指针
char text[]; //字符串表
char *textPointer; //字符串表指针
  
```

```

char textRead[]; //存放待识别的源程序字符串
char *textReadPointer; //读取字符串指针
char ch; //存放最读取的源程序字符
char strToken[]; //存放构成单词符号的字符串
bool flag=false; //是否扫描过一次
  
```

## 3 浏览器 JavaScript 支持

## 3.1 JavaScript 简介

JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。它的一个重要功能就是面向对象的功能,通过基于对象的程序设计,可以用更直观、模块化和可重复使用的方式进行程序开发。一组包含数据的属性和对属性中包含数据进行操作的方法,称为对象。用户与网页交互时产生的操作,称为事件。事件可以由用户引发,也可能是页面发生改变,甚至还有看不见的事件(如 Ajax 的交互进度改变)。绝大部分事件都由用户的动作所引发,例如用户按鼠标的按键,就产生 click 事件,若鼠标的指针在链接上移动,就产生 MouseOver 事件等。在 JavaScript 中,事件与事件处理程序配套使用。而对事件的处理用 addEventListener() 函数,它有 3 个参数:事件、引发的函数以及是否使用事件捕捉。为了保障安全性,将第三个参数始终设置为 false。

JavaScript 具有几个特性:能使网页增加互动性;能使有规律地重复的 HTML 源代码简化,减少下载时间;能及时响应用户的操作,对提交表单做即时的检查,无需浪费额外的验证时间。

## 3.2 JavaScript 引擎

JavaScript 引擎一般作为共享库使用,应用程序调用引擎提供的 API 函数。引擎 API 函数大致分为以下几种:数据类型操作、RunTime 控制、类与对象的创建和维护、函数与脚本执行、字符串操作、错误处理、安全控制、Debug 支持。一般情况下,在应用程序中只需使用某几类函数。例如,在进行 JavaScript 调用之前必须调用 JS\_NewRuntime 函数来创建并初始化 JavaScript 引擎。有些类型的函数(像安全控制类)提供可选择的特征。

JavaScript 引擎是系统上的一个共享资源。通过将引擎 API 调用嵌入到应用程序中,可以请求 JavaScript 引擎进行操作。引擎处理请求,并将结果或状态信息返回给应用程序。例如,假定使用 JavaScript 引擎自动化应用程序,脚本应用程序鉴别用户并设置权限。首先,应用程序创建 JavaScript 对象,该对象描述用户信息,包括姓名、ID、权限和可用的函数列表。在这种情况下,应用程序首先调用 JS\_NewObject 创建对象。当 JavaScript 引擎创建对象后,返回一个指针给应用程序。应用程序再调用 JavaScript 引擎执行脚本。在创建用户对象后,应用程序即刻传递脚本给 JS\_EvaluateScript 以便编译和运行。脚本获取得并校验用户信息,然后建立用户存取的权利。

JavaScript 引擎收到初始化请求后,给 JavaScript Run-Time 分配内存,应用程序使用的变量、对象和上下文都

保存在 RunTime 中。一个上下文是脚本的执行状态 (JavaScript 引擎使用的)。每个同时存在的脚本或线程都必须有自己的上下文。单个的 JavaScript RunTime 可以包含多个上下文、对象和变量。几乎所有的 JavaScript 引擎调用都需要一个上下文变量,应用程序在创建 RunTime 后,首先应调用至少一次 JS\_NewContext 来创建一个上下文。上下文的实际数量依赖于程序中同时使用的脚本数。程序中每个同时存在的脚本都需要一个上下文。另一方面,如果某个时刻只有一个脚本编译和运行,则只需一个上下文给每个脚本重复使用即可。

### 3.3 JavaScript 与浏览器结合

#### 3.3.1 直接加入方式

绝大部分含有 JavaScript 代码的网页都采用直接加入的方式。

```
<script type="test/javascript">
document.write("This is Javascript! ");
</script>
```

其中, <script> </script> 是 JavaScript 的标签。 <script type="test/javascript"> 用来提示浏览器该程序采用 JavaScript 语言编写,需要调动相应的解释程序进行解释。

除此之外,还有一个更高版本的嵌入脚本,使用 HTML 中的 CDATA 语法,就是把 CDATA 中的文本全部当作纯文本处理,当遇到 CDATA 时结束。

```
<script language="javascript" type="test/javascript">
<! CDATA[[>
<! ]]>
</script>
```

#### 3.3.2 引用方式

如果已经存在一个 JavaScript 源文件(以 js 为扩展名),则可以采用引用的方式提高程序代码的利用率。其基本格式如下:

```
<script src="url" type="test/javascript">
</script>
```

其中 url 就是程序文件的地址。

同样地,这样的 JavaScript 程序语句可以放在 HTML 文档头部或主体的任何部分。如果要实现直接加入 JavaScript 程序代码,可以首先创建一个 JavaScript 源代码文件“javascript.js”,其代码为 document.write("This is Javascript! ")。在网页中调用程序的方法,如:

```
<script src="javascript.js" type="test/javascript">
</script>
```

也可以同时在导入文件时指定 JavaScript 的版本,如:

```
<script src="javascript.js" type="test/javascript; version=
2.0">
</script>
```

本文对 Linux 操作系统的嵌入式浏览器进行了研究与分析。分析了嵌入式浏览器的结构,并对嵌入式浏览器的各部分功能进行阐述,其中包括词法分析、语法分析与布局、浏览器 JavaScript 支持、I/O 端等部分。其中,在嵌入式浏览器的研究过程中,词法分析的算法是开发技术的核心。在后续研究工作中,可以对词法分析算法做进一步的研究,提高词法分析的效率,完善嵌入式浏览器功能实现的效果。

#### 参考文献

- [1] 李永,熊光泽,杨玉平.一种嵌入式浏览器设计技术[J]. 微型机及应用,2001,20(12):57-59.
- [2] 钟遥,罗克露.基于压缩 XML 嵌入式浏览器的实现技术[C].中国西部嵌入式系统与单片机技术论坛 2005 学术年会,成都,2005.
- [3] 刘峥嵘,张智超,许振山.嵌入式 Linux 应用开发详解[M]. 北京:机械工业出版社,2004:53-166.

(收稿日期:2010-12-16)

#### 作者简介:

石功恒,男,1985年生,硕士研究生,主要研究方向:嵌入式系统。