<https://www.cnblogs.com/liunianshiwei/p/6086577.html>

今天我们来简单介绍一下，关于xml的一些内容，包括自己编写一些程序进行生成和解析。

首先我们我们还是从xml的演化历史来了解一下它吧。

历史演化

　　GML：

　　　　　在20世纪60年代为了促进数据交换和操作，通过IBM公司研究人员的杰出工作，得出了重要的结论：要提高系统的移植性，必须采用一种通用的文档格式，这种文档的格式必须遵守特定的规则。这也就是创建通用标记语言（外语全称：Generalized Markup Language、外语缩写：GML）的指导原则，从人们所产生的将文件结构化为标准的格式的动机出发，IBM创建了GML。GML是一种IBM格式化文档语言，用于就其组织结构、各部件及其之间的关系进行文档描述。GML将这些描述标记为章节、重要小节和次重要小节（通过标题的级来区分）、段落、列表、表等。GML在文档具体格式方面，为文档员提供了一些方便，他们不必再为IBM得打印机格式化语言SCRIPT要求的字体规范、行距以及页面设计等浪费精力。GML是标准通用标记语言的先驱和基础，SGML是当今创建结构化文档描述语言规则的战略集合。很多网页是用HTML标记表示出的，就是使用GML概念创建文档的例子。扩展标志语言（XML）也根源于GML。   
　　SGML ：

　　　　　　它源于1969年IBM公司开发的文档描述语言GML，GML主要用来解决不同系统中文档格式不同的问题。后经过多年发展，1986年经ISO批准为国际标准ISO8897，并被称为SGML。制定SGML的基本思想是把文档的内容与样式分开。在SGML中，标记分两种：一种用来描述文档显示的样式，称为程序标记；另一种用来描述文档中语句的用途，称为描述标记。一个SGML文件通常分三个层次：结构、内容和样式。结构为组织文档的元素提供框架，内容是信息本身，样式控制内容的显示。SGML的平台无关性、结构化、可扩展等特性，使得它使用范围很广，被许多大型公司开始用来创建和发布信息。

　　HTML ：

　　　　　　超文本标记语言(HTML)起源于标准通用标记语言(SGML)，由世界上最大的粒子物理研究实验室欧洲核子研究中心CERN(the European Organization for Nuclear Research)于1991年首先提出，是推动Web迅速发展的原动力。在互联网发展的早期，为了在各种网络环境之间、不同文件格式之间进行交流，在SGML基础上，CERN提出了超文本标记语言(Hyper Text Markup Language，HTML)的概念。HTML是一种用来制作超文本文档的简单标记语言，它定义了一组标记符号(tag)，对文件的内容进行标注，指出内容的输出格式，如字体大小、颜色、背景颜色、表格形式、各部分之间逻辑上的组织等，从而实现了文件格式的标准化。简单地说，HTMI。文件包含了文档数据和显示样式两部分，其中文档数据是显示在Web浏览器中的数据内容，显示样式则规定了这些内容在浏览器中以何种格式、样子呈现给用户。通过统一使用支持HTMI。的浏览  
软件，用户可以在任意异构的网络环境中阅读同一个文件，得到相同的显示结果，并可以对文件进行跳跃式阅读，展现了很强的表现力。

　　XML：

　　　　　XML是由互联网联盟(World Wide Web Consortium，W3C)的XML工作组定义的。 “扩展标记语言(XML)是SGML的子集，其目标是允许普通的SGML在Web上以目前HTML的方式被服务、接收和处理。XML被设计成易于实现，且可在SGML和HTML之间互相操作。” HTML的出现极大地推动了世界范围内的互联网的发展，万维网就是一个主要成果。然而，HTML在某些方面是一种通用编码的倒退。 第一，HTML为了获得精简编码方式的有效性，而抛弃了通用编码的一些基本原则。例如，通用编码要求一个文档类型能用于任何目的，要求用户编码时应重载标签而不是重新定义特殊目的的标签。   
　　XML特点和作用 :  
　　　　　　XML 指可扩展标识语言（ eXtensible Markup Language）  
　　　　　　XML 的设计宗旨是传输数据， 而非显示数据  
　　　　　　XML 标签没有被预定义。 您需要自行定义标签。  
　　　　　　作为一种通用的数据存储和通信格式被广泛应用。  
　　　　　　描述的数据作为一棵树型的结构而存在。  
第一个简单的xml文件 :

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="utf-8"?>

2 <note>

3 <to>Tove</to>

4 <from>Jani</from>

5 <heading>Reminder</heading>

6 <body>Don't forget me this weekend!</body>

7 </note>

[复制代码](javascript:void(0);)

规则说明:它定义 XML 的版本（1.0）和所使用的编码 UTF-8，version不可省略，encoding可以省略;

　　　　<note>描述文档的根元素，有且只有一个根元素。

　　　　所有 XML 元素都须有关闭标签

　　　　XML标签对大小写敏感。

XML元素 :

　　　　XML元素指的是开始标签直到结束标签的部分，可以包含文本、属性、其他元素 中的一种或者多种。 比如下面这个xml文件

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="utf-8"?>

2 <bookstore><bookstore>

3 <book category="CHILDREN">

4 <title>Harry Potter</title>

5 <author>J K. Rowling</author>

6 <year>2005</year>

7 <price>29.99</price>

8 </book>

9 <book category="WEB">

10 <title>Learning XML</title>

11 <author>Erik T. Ray</author>

12 <year>2003</year>

13 <price>39.95</price>

14 </book>

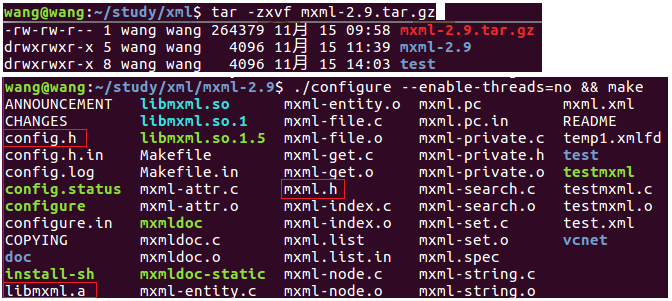
15 </bookstore>

[复制代码](javascript:void(0);)

上面的示例中<bookstore> <book>都有元素内容 因为他们包含其他元素。 <book>元素也有属性category="WEB"， <title> <author> <year> <price>有文本内容。

 下面正式介绍在linux下安装以及代码实现（我们下面介绍的依赖XML解析库是minixml），首先下载mxml的压缩包，并进行解压，解压后进入目录执行如图所示命令进行编译，编译后我们进行代码测试需要的三个文件分别是 libmxml.a mxml.h config.h,将其拷贝到代码编写目录。

附mxml解析库下载地址：http://www.msweet.org/downloads.php?L+Z3



我们还是从mxml的函数开始介绍，才能事半功倍

minixml常用函数接口：

　　均需要包含头文件 #include <mxml.h>   
　　  
　　　　创建一个新xml文件 ：mxml\_node\_t \*mxmlNewXML(const char \*version);  
　　　　　　　　参数1:version默认传入"1.0"即可  
　　　　　　　　返回值:返回新创建的xml文件节点  
　　　　　　　　提示： 虽然xml中可以指定字符编码， 但是在mxml中默认的编码就是UTF8编码。   
　　　　其中一个节点新增节点 ：mxml\_node\_t \*mxmlNewElement(mxml\_node\_t \*parent, const char \*name);   
　　　　　　　　参数1:父节点指针  
　　　　　　　　参数2:新节点名称  
　　　　　　　　返回值:返回新创建的子节点

　　　　设置节点属性名和值 ：void mxmlElementSetAttr(mxml\_node\_t \*node, const char \*name,const char \*value);

　　　　　　　　参数1:被设置的节点的地址  
　　　　　　　　参数2:设置的属性名  
　　　　　　　　参数3:设置的属性值

　　　　创建结点的文本 ：mxml\_node\_t \*mxmlNewText ( mxml\_node\_t \*parent, int whitespace, const char\*string);

　　　　　　　　参数1： 被设置的结点的地址  
　　　　　　　　参数2： 1 = leading whitespace, 0 = no whitespace  
　　　　　　　　参数3： 文本

　　　　保存节点到xml文件 ：int mxmlSaveFile(mxml\_node\_t \*node, FILE \*fp,mxml\_save\_cb\_t cb);   
　　　　　　　　参数1:node表示希望被保存的xml节点树的根节点， 指向xml结构的节点指针  
　　　　　　　　参数2:fp为C中使用fopen函数打开文件所返回的FILE类型的指针  
　　　　　　　　参数3:默认情况下使用MXML\_NO\_CALLBACK即可

　　　　删除节点内存 ：mxmlDelete(mxml\_node\_t \*node);   
　　　　　　　　参数1:node为指向节点树的指针。  
　　　　　　　　该函数将释放node节点指针所指向的整棵节点树， 而不用我们去一个一个节点释放。 如果该节点还有父节点， 会从节点树中先使用mxmlRemove()移除该节点。

　　　　从文件中加载xml ：mxml\_node\_t \*mxmlLoadFile(mxml\_node\_t \*top, FILE \*fp,mxml\_type\_t (\*cb)(mxml\_node\_t \*));

　　　　　　　　参数1:top为加载的节点的父节点 如果是文档节点则填NULL即可  
　　　　　　　　参数2:fopen函数返回的文件的指针  
　　　　　　　　参数3:默认情况下使用MXML\_NO\_CALLBACK即可

　　　　获取节点属性 ：const char \*mxmlElementGetAttr(mxml\_node\_t \*node, const char \*name);   
　　　　　　　　参数1:node为指向结点树的指针  
　　　　　　　　参数2:name为获取的属性名  
　　　　　　　　返回值:指定属性名的属性

　　　　获取指定节点的文本内容： const char \*mxmlGetText(mxml\_node\_t \*node, int \*whitespace);   
　　　　跳转到下一个节点 ：mxml\_node\_t \*mxmlWalkNext(mxml\_node\_t \*node, mxml\_node\_t \*top,int descend);  
　　　　　　　　遍历到XML树中的下一个逻辑节点。  
　　　　　　　　node 当前节点  
　　　　　　　　top 顶级节点  
　　　　　　　　descend参数有三个备选项  
　　　　　　　　MXML\_NO\_DESCEND含义是不查看任何的子节点在XML元素层次中， 仅查看同层级的节点或者父节点直到到达根节点或者top节点.  
　　　　　　　　MXML\_DESCEND\_FIRST含义是向下搜索到一个节点的第一个匹配子节点， 但不再继续向下搜索。 你一般使用于遍历一个父节点的直接的子节点。  
　　　　　　　　MXML\_DESCEND含义是可以一直向下搜索.

　　　　查找节点 ：mxml\_node\_t \*mxmlFindElement(mxml\_node\_t \*node, mxml\_node\_t \*top,const char \*name, const char \*attr,const char \*value, int descend);

　　　　　　　　node节点为被查找的结点  
　　　　　　　　top为顶层结点  
　　　　　　　　name， attrr， value为NULL时表示任意匹配； 否则为精确匹配。  
　　　　　　　　descend同上个接口使用方式。

介绍了那么多了，下面终于要进入到具体的代码编程实现当中了

首先我们实现一个生成一个简单的xml文件的功能，代码1如下：

[复制代码](javascript:void(0);)

1 #include"mxml.h"

2

3 int main()

4 {

5 //创建一个xml文件

6 mxml\_node\_t \*xml = mxmlNewXML("1.0");

7 //新增一个节点html

8 mxml\_node\_t \*html = mxmlNewElement(xml,"html");

9 //在html节点下新增一个head节点

10 mxml\_node\_t \*head = mxmlNewElement(html,"head");

11 //在head节点下新增一个title节点

12 mxml\_node\_t \*title = mxmlNewElement(head,"title");

13 //title节点添加内容"Docment"

14 mxmlNewText(title,0,"Docment");

15

16 //在html节点下新增body节点

17 mxml\_node\_t \*body = mxmlNewElement(html,"body");

18 //在body节点下新增p节点

19 mxml\_node\_t \*p = mxmlNewElement(body,"p");

20 //给p节点增加属性style，属性内容为 color:red

21 mxmlElementSetAttr(p,"style","color:red");

22 //给p节点增加内容 Helloworld.cpp

23 mxmlNewText(p,0,"Helloworld.cpp");

24

25 //打开文件index.xml

26 FILE \*fp = fopen("index.xml","wb");

27 //保存节点到xml文件

28 mxmlSaveFile(xml,fp,MXML\_NO\_CALLBACK);

29 fclose(fp);

30 //删除节点内存

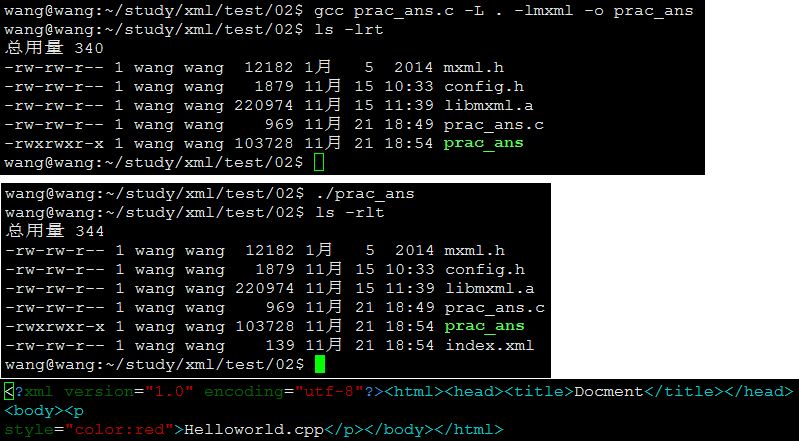
31 mxmlDelete(xml);

32 return 0;

33 }

[复制代码](javascript:void(0);)

首先执行如上图所示命令进行编译，生成可执行程序，然后执行程序，看到生成index.xml文件，vim打开查看该xml文件



下面我们实现一个生成另一个xml文件的功能，代码2如下：

[复制代码](javascript:void(0);)

1 #include<stdio.h>

2 #include"mxml.h"

3

4 int main(void)

5 {

6 FILE \*fp = fopen("new.xml","w");

7

8 //生成xml文件的头 <?xml version="1.0" encoding="utf-8"?>

9 mxml\_node\_t \*xml = mxmlNewXML("1.0");

10 //添加一个新节点note

11 mxml\_node\_t \*note = mxmlNewElement(xml,"note");

12 //往note节点添加子节点to

13 mxml\_node\_t \*to = mxmlNewElement(note,"to");

14 //给to节点添加文本内容

15 mxmlNewText(to,0,"科比");

16 //给to节点设置属性 time 属性内容为 1115

17 mxmlElementSetAttr(to,"time","1115");

18

19 //给note节点添加子节点from

20 mxml\_node\_t \*from = mxmlNewElement(note,"from");

21 //给from节点添加文本内容

22 mxmlNewText(from,0,"瓦妮莎");

23 //给note节点添加子节点heading

24 mxml\_node\_t \*heading = mxmlNewElement(note,"heading");

25 //给heading节点添加文本内容

26 mxmlNewText(heading,0,"Reminder");

27

28 //给note节点添加子节点body

29 mxml\_node\_t \*body = mxmlNewElement(note,"body");

30 //给body节点添加文本内容

31 mxmlNewText(body,0,"Don't froget NBA programe this weekend!");

32 //保存节点内容

33 mxmlSaveFile(xml,fp,MXML\_NO\_CALLBACK);

34

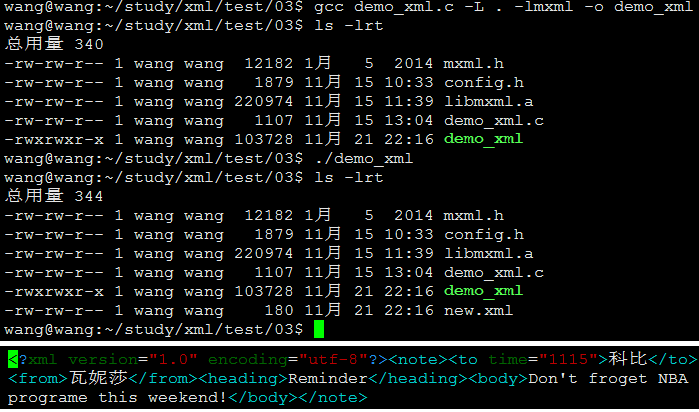
35 fclose(fp);

36 return 0;

37 }

[复制代码](javascript:void(0);)

使用下面的命令进行编译，生成可执行程序，然后执行程序，生成xml文件，可以用vim打开查看



 前面两个都是介绍生成xml文件的案例，下面我们看几个解析已知的xml文件的例子

解析已有的xml文件，提取出信息代码3：

[复制代码](javascript:void(0);)

1 #include<stdio.h>

2 #include"mxml.h"

3

4 int main(void)

5 {

6 FILE\* fp = fopen("prac.xml","r");

7 //从prac.xml文件中加载xml

8 mxml\_node\_t \*xml = mxmlLoadFile(NULL,fp,MXML\_NO\_CALLBACK);

9 //定义两个空节点

10 mxml\_node\_t \*book = NULL;

11 mxml\_node\_t \*title = NULL;

12 //从xml开始向下查找 name=book attrr=category

13 book = mxmlFindElement(xml,xml,"book","category",NULL,MXML\_DESCEND);

14

15 while(book)

16 {

17 //获取title子元素的文本 book元素的属性

18 title = mxmlFindElement(book,xml,"title",NULL,NULL,MXML\_DESCEND);

19 if(title == NULL)

20 {

21 printf("title not found\n");

22 continue;

23 }

24 else

25 {

26 printf("book'titele is %s\n",mxmlGetText(title,NULL));

27 printf("book'category:%s\n",mxmlElementGetAttr(book,"category"));

28 book = mxmlFindElement(title,xml,"book","category",NULL,MXML\_DESCEND);

29 }

30 }

31 mxmlDelete(xml);

32 fclose(fp);

33 return 0;

34 }

[复制代码](javascript:void(0);)

已知的xml文件如下所示：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="utf-8"?>

2 <bookstore>

3 <book category="CHILDREN">

4 <title>Harry.Potter</title>

5 <author>JK.Rowling</author>

6 <year>2005</year>

7 <price>29.99</price>

8 </book>

9 <book category="WEB">

10 <title>LearningXML</title>

11 <author>ErikT.Ray</author>

12 <year>2003</year>

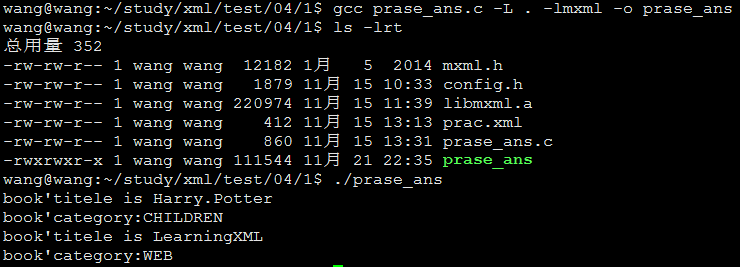
13 <price>39.95</price>

14 </book>

15 </bookstore>

[复制代码](javascript:void(0);)

执行如下命令进行编译，执行程序，即可看到解析出的xml信息



上面都是练手的，下面我们来点实际的，解析一下QQ数据试试

代码4如下：

[复制代码](javascript:void(0);)

1 #include<stdio.h>

2 #include"mxml.h"

3

4 int main(void)

5 {

6 FILE \*fp = fopen("QQ.xml","r");

7 //加载文件

8 mxml\_node\_t \*xml = mxmlLoadFile(NULL,fp,MXML\_NO\_CALLBACK);

9 //创建三个空节点

10 mxml\_node\_t \*pic = NULL;

11 mxml\_node\_t \*resurl = NULL;

12 mxml\_node\_t \*thumbnail = NULL;

13 //查找Pic元素

14 pic = mxmlFindElement(xml,xml,"Pic",NULL,NULL,MXML\_DESCEND);

15

16 while(pic)

17 {

18 resurl = mxmlFindElement(pic,xml,"resurl","md5",NULL,MXML\_DESCEND);

19 if(resurl == NULL)

20 {

21 printf("resurl == NULL");

22 }

23 else

24 {

25 printf("md5 == %s\n",mxmlElementGetAttr(resurl,"md5"));

26 printf("text:%s\n",mxmlGetText(resurl,NULL));

27 }

28 thumbnail = mxmlFindElement(pic,xml,"thumbnail","md5",NULL,MXML\_DESCEND);

29 if(resurl == NULL)

30 {

31 printf("thumbnail == NULL");

32 }

33 else

34 {

35 printf("md5 == %s\n",mxmlElementGetAttr(thumbnail,"md5"));

36 printf("text:%s\n",mxmlGetText(thumbnail,NULL));

37 }

38 pic = mxmlFindElement(pic,xml,"Pic",NULL,NULL,MXML\_DESCEND);

39 }

40 mxmlDelete(xml);

41 fclose(fp);

42 return 0;

43 }

[复制代码](javascript:void(0);)

QQ数据的xml文件

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="utf-8"?>

2

3 <!--pkgName对应GetResourceRespInfo的strPkgName字段-->

4 <!--version对应GetResourceRespInfo的uiNewVer字段-->

5 <PicArray version="1" pkgName="QQCHATPic\_hdpi" >

6 <Pic>

7 <!-- 原图地址-->

8 <resurl md5 = "qq2013\_chatbg101\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_101\_720.jpg</resurl>

9 <!-- 缩略图地址-->

10 <thumbnail md5 = "qq2013\_chatbg\_thumb\_101">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_101..png</thumbnail>

11 </Pic>

12 <Pic>

13 <!-- 原图地址-->

14 <resurl md5 = "qq2013\_chatbg102\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_102\_720.jpg</resurl>

15 <!-- 缩略图地址-->

16 <thumbnail md5 = "qq2013\_chatbg\_thumb\_102">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_102..png</thumbnail>

17 </Pic>

18 <Pic>

19 <!-- 原图地址-->

20 <resurl md5 = "qq2013\_chatbg103\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_103\_720.jpg</resurl>

21 <!-- 缩略图地址-->

22 <thumbnail md5 = "qq2013\_chatbg\_thumb\_103">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_103..png</thumbnail>

23 </Pic>

24

25 <Pic>

26 <!-- 原图地址-->

27 <resurl md5 = "qq2013\_chatbg104\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_104\_720.jpg</resurl>

28 <!-- 缩略图地址-->

29 <thumbnail md5 = "qq2013\_chatbg\_thumb\_104">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_104..png</thumbnail>

30 </Pic>

31

32 <Pic>

33 <!-- 原图地址-->

34 <resurl md5 = "qq2013\_chatbg105\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_105\_720.jpg</resurl>

35 <!-- 缩略图地址-->

36 <thumbnail md5 = "qq2013\_chatbg\_thumb\_105">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_105..png</thumbnail>

37 </Pic>

38

39 <Pic>

40 <!-- 原图地址-->

41 <resurl md5 = "qq2013\_chatbg106\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_106\_720.jpg</resurl>

42 <!-- 缩略图地址-->

43 <thumbnail md5 = "qq2013\_chatbg\_thumb\_106">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_106..png</thumbnail>

44 </Pic>

45

46 <Pic>

47 <!-- 原图地址-->

48 <resurl md5 = "qq2013\_chatbg107\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_107\_720.jpg</resurl>

49 <!-- 缩略图地址-->

50 <thumbnail md5 = "qq2013\_chatbg\_thumb\_107">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_107..png</thumbnail>

51 </Pic>

52

53 <Pic>

54 <!-- 原图地址-->

55 <resurl md5 = "qq2013\_chatbg108\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_108\_720.jpg</resurl>

56 <!-- 缩略图地址-->

57 <thumbnail md5 = "qq2013\_chatbg\_thumb\_108">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_108..png</thumbnail>

58 </Pic>

59 <Pic>

60 <!-- 原图地址-->

61 <resurl md5 = "qq2013\_chatbg109\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_109\_720.jpg</resurl>

62 <!-- 缩略图地址-->

63 <thumbnail md5 = "qq2013\_chatbg\_thumb\_109">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_109..png</thumbnail>

64 </Pic>

65 <Pic>

66 <!-- 原图地址-->

67 <resurl md5 = "qq2013\_chatbg1101\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_110\_720.jpg</resurl>

68 <!-- 缩略图地址-->

69 <thumbnail md5 = "qq2013\_chatbg\_thumb\_1101">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chat\_bg\_thumb\_110..png</thumbnail>

70 </Pic>

71 <Pic>

72 <!-- 原图地址-->

73 <resurl md5 = "qq2013\_chatbg1112\_hdpi">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/720V2/chat\_bg\_111\_720.jpg</resurl>

74 <!-- 缩略图地址-->

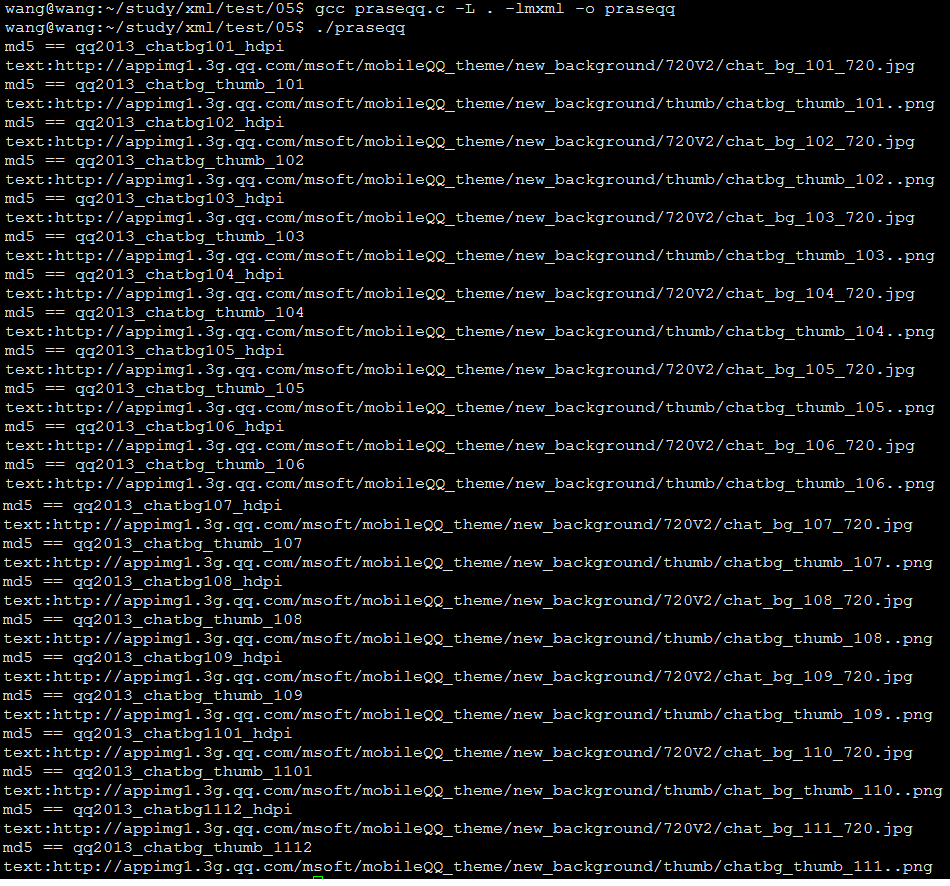
75 <thumbnail md5 = "qq2013\_chatbg\_thumb\_1112">http://appimg1.3g.qq.com/msoft/mobileQQ\_theme/new\_background/thumb/chatbg\_thumb\_111..png</thumbnail>

76 </Pic>

77 </PicArray>

[复制代码](javascript:void(0);)

编译程序，执行可执行程序，得到如下图所示结果：



下面我们再来解析一个天气情况的xml文件来熟悉一下函数使用，代码5如下：

[复制代码](javascript:void(0);)

1 #include <stdio.h>

2 #include "mxml.h"

3

4 int main(void)

5 {

6 FILE \*fp = fopen("weather.xml","rb");

7 mxml\_node\_t \*xml = mxmlLoadFile(NULL,fp,MXML\_NO\_CALLBACK);

8 mxml\_node\_t \*city = NULL;

9 city = mxmlFindElement(xml,xml,"city",NULL,NULL,MXML\_DESCEND);

10 while(city)

11 {

12 printf("省市名称:%s \n",mxmlElementGetAttr(city,"quName"));

13 printf("省会名称:%s \n",mxmlElementGetAttr(city,"cityname"));

14

15 printf("最低气温:%s \n",mxmlElementGetAttr(city,"tem1"));

16 printf("最高气温:%s \n",mxmlElementGetAttr(city,"tem2"));

17

18 printf("天气情况:%s \n",mxmlElementGetAttr(city,"stateDetailed"));

19 printf("风力:%s \n\n",mxmlElementGetAttr(city,"windState"));

20 city = mxmlFindElement(city,xml,"city",NULL,NULL,MXML\_DESCEND);

21 }

22

23 return 0;

24 }

[复制代码](javascript:void(0);)

天气的xml文件如下所示：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <china dn="nay">

3 <city quName="黑龙江" pyName="heilongjiang" cityname="哈尔滨" state1="1" state2="1" stateDetailed="多云" tem1="17" tem2="21" windState="东南风小于3级"></city>

4 <city quName="吉林" pyName="jilin" cityname="长春" state1="21" state2="4" stateDetailed="小到中雨转雷阵雨" tem1="16" tem2="20" windState="东南风转西北风小于3级"></city>

5 <city quName="辽宁" pyName="liaoning" cityname="沈阳" state1="1" state2="4" stateDetailed="多云转雷阵雨" tem1="15" tem2="26" windState="西南风小于3级"></city>

6 <city quName="海南" pyName="hainan" cityname="海口" state1="1" state2="4" stateDetailed="多云转雷阵雨" tem1="26" tem2="33" windState="西南风3-4级"></city>

7 <city quName="内蒙古" pyName="neimenggu" cityname="呼和浩特" state1="1" state2="1" stateDetailed="多云" tem1="7" tem2="23" windState="西北风小于3级"></city>

8 <city quName="新疆" pyName="xinjiang" cityname="乌鲁木齐" state1="0" state2="0" stateDetailed="晴" tem1="18" tem2="29" windState="微风"></city>

9 <city quName="西藏" pyName="xizang" cityname="拉萨" state1="4" state2="1" stateDetailed="雷阵雨转多云" tem1="10" tem2="23" windState="微风"></city>

10 <city quName="青海" pyName="qinghai" cityname="西宁" state1="8" state2="3" stateDetailed="中雨转阵雨" tem1="7" tem2="20" windState="东南风小于3级"></city>

11 <city quName="宁夏" pyName="ningxia" cityname="银川" state1="2" state2="2" stateDetailed="阴" tem1="15" tem2="25" windState="微风"></city>

12 <city quName="甘肃" pyName="gansu" cityname="兰州" state1="7" state2="7" stateDetailed="小雨" tem1="14" tem2="24" windState="微风"></city>

13 <city quName="河北" pyName="hebei" cityname="石家庄" state1="1" state2="1" stateDetailed="多云" tem1="19" tem2="31" windState="北风转南风小于3级"></city>

14 <city quName="河南" pyName="henan" cityname="郑州" state1="0" state2="0" stateDetailed="晴" tem1="20" tem2="33" windState="微风"></city>

15 <city quName="湖北" pyName="hubei" cityname="武汉" state1="2" state2="7" stateDetailed="阴转小雨" tem1="21" tem2="29" windState="微风"></city>

16 <city quName="湖南" pyName="hunan" cityname="长沙" state1="3" state2="7" stateDetailed="阵雨转小雨" tem1="24" tem2="29" windState="北风小于3级"></city>

17 <city quName="山东" pyName="shandong" cityname="济南" state1="0" state2="1" stateDetailed="晴转多云" tem1="20" tem2="32" windState="南风小于3级"></city>

18 <city quName="江苏" pyName="jiangsu" cityname="南京" state1="1" state2="1" stateDetailed="多云" tem1="21" tem2="32" windState="东南风转东风小于3级"></city>

19 <city quName="安徽" pyName="anhui" cityname="合肥" state1="0" state2="1" stateDetailed="晴转多云" tem1="22" tem2="32" windState="北风小于3级"></city>

20 <city quName="山西" pyName="shanxi" cityname="太原" state1="1" state2="1" stateDetailed="多云" tem1="12" tem2="27" windState="微风"></city>

21 <city quName="陕西" pyName="sanxi" cityname="西安" state1="1" state2="2" stateDetailed="多云转阴" tem1="19" tem2="26" windState="东风小于3级"></city>

22 <city quName="四川" pyName="sichuan" cityname="成都" state1="7" state2="1" stateDetailed="小雨转多云" tem1="19" tem2="29" windState="微风"></city>

23 <city quName="云南" pyName="yunnan" cityname="昆明" state1="3" state2="3" stateDetailed="阵雨" tem1="17" tem2="24" windState="微风"></city>

24 <city quName="贵州" pyName="guizhou" cityname="贵阳" state1="2" state2="21" stateDetailed="阴转小到中雨" tem1="18" tem2="23" windState="东北风小于3级"></city>

25 <city quName="浙江" pyName="zhejiang" cityname="杭州" state1="1" state2="1" stateDetailed="多云" tem1="22" tem2="29" windState="东风小于3级转3-4级"></city>

26 <city quName="福建" pyName="fujian" cityname="福州" state1="3" state2="2" stateDetailed="阵雨转阴" tem1="25" tem2="31" windState="微风转东风小于3级"></city>

27 <city quName="江西" pyName="jiangxi" cityname="南昌" state1="1" state2="1" stateDetailed="多云" tem1="24" tem2="32" windState="微风"></city>

28 <city quName="广东" pyName="guangdong" cityname="广州" state1="4" state2="4" stateDetailed="雷阵雨" tem1="26" tem2="33" windState="微风"></city>

29 <city quName="广西" pyName="guangxi" cityname="南宁" state1="1" state2="3" stateDetailed="多云转阵雨" tem1="25" tem2="33" windState="东南风小于3级"></city>

30 <city quName="北京" pyName="beijing" cityname="北京" state1="0" state2="0" stateDetailed="晴" tem1="18" tem2="31" windState="微风"></city>

31 <city quName="天津" pyName="tianjin" cityname="天津" state1="0" state2="1" stateDetailed="晴转多云" tem1="20" tem2="31" windState="西南风转南风小于3级"></city>

32 <city quName="上海" pyName="shanghai" cityname="上海" state1="0" state2="1" stateDetailed="晴转多云" tem1="23" tem2="31" windState="南风转东北风小于3级"></city>

33 <city quName="重庆" pyName="chongqing" cityname="重庆" state1="8" state2="8" stateDetailed="中雨" tem1="20" tem2="23" windState="微风"></city>

34 <city quName="香港" pyName="xianggang" cityname="香港" state1="3" state2="3" stateDetailed="阵雨" tem1="26" tem2="30" windState="微风"></city>

35 <city quName="澳门" pyName="aomen" cityname="澳门" state1="3" state2="3" stateDetailed="阵雨" tem1="26" tem2="31" windState="微风"></city>

36 <city quName="台湾" pyName="taiwan" cityname="台北" state1="9" state2="10" stateDetailed="大雨转暴雨" tem1="26" tem2="30" windState="微风"></city>

37 <city quName="西沙" pyName="xisha" cityname="西沙" state1="1" state2="1" stateDetailed="多云" tem1="29" tem2="32" windState="西南风4-5级"></city>

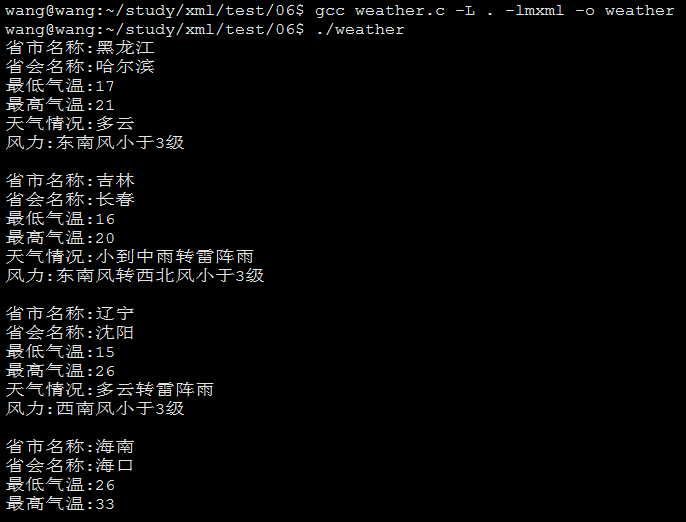
38 <city quName="南沙" pyName="nanshadao" cityname="南沙" state1="4" state2="4" stateDetailed="雷阵雨" tem1="28" tem2="31" windState="西风5-6级"></city>

39 <city quName="钓鱼岛" pyName="diaoyudao" cityname="钓鱼岛" state1="7" state2="7" stateDetailed="小雨" tem1="26" tem2="28" windState="北风转东北风3-4级"></city>

40 </china>

[复制代码](javascript:void(0);)

编译程序，执行程序，如下图所示，由于内容较多，所以截图只有前半部分，结果已经出来了



以上就是基于mxml库的对于xml一些简单介绍，xml是很强大的，如果还想继续深入学习可以继续深入研究libmxml库，这里就介绍到这里。