C語言實現url解析小實例

腳本之家 2022-03-14 17:01

以下文章來源於一口Linux，作者土豆居士

出品| 一口Linux （ID：yikoulinux）

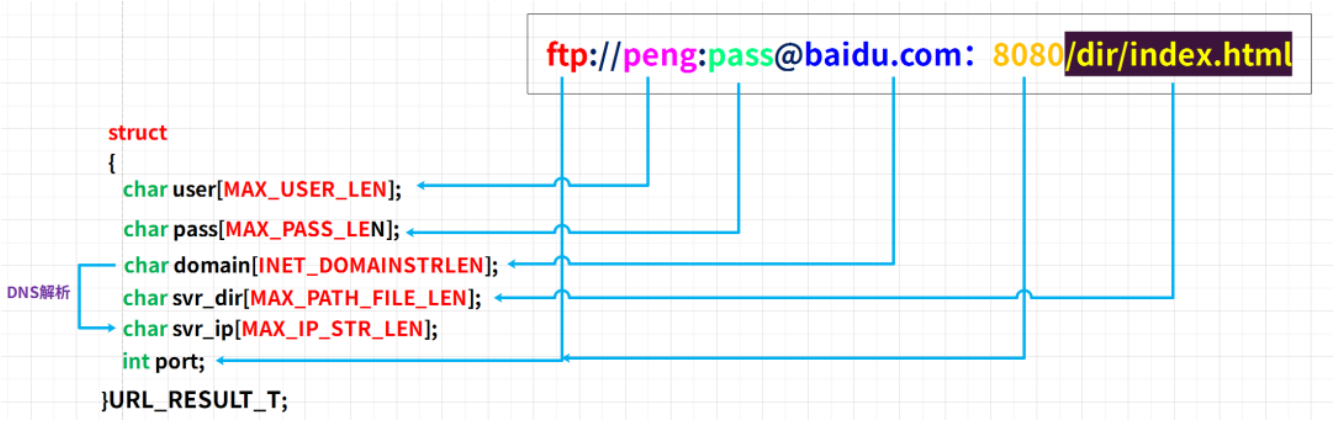
已獲得原公眾號的授權轉載

一、前言

前面一口君寫了一篇關於url的文章：《一文帶你理解URI 和URL 有什麼區別？》

本篇在此基礎上，編寫一個簡單的用於解析url的小例子，

最終目標是解析出URL中所有的數據信息。



二、庫函數

用到的幾個庫函數如下:

1. strncasecmp

頭文件

```
#include<string.h>
```

函數定義

```
int strncasecmp(const char *s1,const char *s2,size_t n);
```

函數說明

```
用来比较参数s1和s2字符串前n个字符，比较时会自动忽略大小写的差异。
```

返回值

```
若参数s1和s2 字符串相同则返回0。
s1 若大于s2则返回大于0的值，
s1若小于s2则返回小于0 的值。
```

2. strstr

頭文件

```
#include<string.h>
```

函數定義

```
char *strstr( const char* str, const char* substr );
```

函數說明

```
查找 substr 所指的空终止字节字符串在 str 所指的空终止字节字符串中的首次出现。不比较空终止字符。

若 str 或 substr 不是指向空终止字节字符串的指针，则行为未定义。
```

參數

```
str :指向要检验的空终止字节字符串的指针
substr :指向要查找的空终止字节字符串的指针
```

返回值

```
指向于 str 中找到的子串首字符的指针，或若找不到该子串则为空指针。若 substr 指向空字符串，则返回 str 。
```

3. strtok

函數定義

```
char *strtok(char *str, const char *delim)
```

功能

分解字符串 str 为一组字符串，delim 为分隔符

参數

str -- 要被分解成一组小字符串的字符串。
delim -- 包含分隔符的 C 字符串。

返回值

该函数返回被分解的第一个子字符串，如果没有可检索的字符串，则返回一个空指针。

4. strncpy

函數說明

```
char *strncpy(char *dest, const char *src, size_t n)
```

功能

将src指向的字符串拷贝到dest执行的内存中，最多拷贝n个字符

參數

dest -- 指向用于存储复制内容的目标数组。
src -- 要复制的字符串。
n -- 要从源中复制的字符数。

返回值

该函数返回最终复制的字符串。

5. inet_pton/inet_ntop

頭文件

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

函數聲明

```
#include <arpe/inet.h>
int inet_pton(int family, const char *strptr, void *addrptr);
```

功能：

```
将点分十进制的ip地址转化为用于网络传输的数值格式
对于IPv4地址和IPv6地址都适用
```

参數

```
family：协议类型既可以是AF_INET（ipv4）也可以是AF_INET6（ipv6）。如果，以不被支持的地址族作为family参数，

strptr：指向点分十进制的IP地址字符串，比如"192.168.1.1"

addrptr：转换结果存放在addrptr中,比如"192.168.1.1"转换为：0xC0A80101

addrptr类型为：struct in_addr
typedef uint32_t in_addr_t;
struct in_addr {
    in_addr_t s_addr;
};
```

返回值

若成功则为1，若输入不是有效的表达式则为0，
若出错则为-1

```
const char * inet_ntop(int family, const void *addrptr, char *strptr, size_t len);
```

功能

将数值格式转化为点分十进制的ip地址格式，从数值格式（addrptr）转换到表达式（strptr），

返回值

若成功则为指向结构的指针，若出错则为NULL

6. gethostbyname

函數的定義

```
#include <netdb.h>
struct hostent * gethostbyname(const char * hostname);
```

功能

解析hostname指向的域名，该函数会将该域名封装到DNS协议包中，发送给DNS服务器，DNS服务器会将该域名对应的地址返

参數

hostname ：存储域名对应的字符串。

返回值

若成功则为非空指针，若出错则为NULL且设置h_errno

返回的指针类型为：

```
struct hostent{
    char *h_name;  //official name
    char **h_aliases;  //alias list
    int  h_addrtype;  //host address type
    int  h_length;  //address lenght
    char **h_addr_list;  //address list
}
```

DNS服务器返回的地址就存储在该结构体中

三、自定義結構

結構體用於存放需要解析的協議和端口號

```
struct pro_port{
 char pro_s[32];
 unsigned short port;
};
```

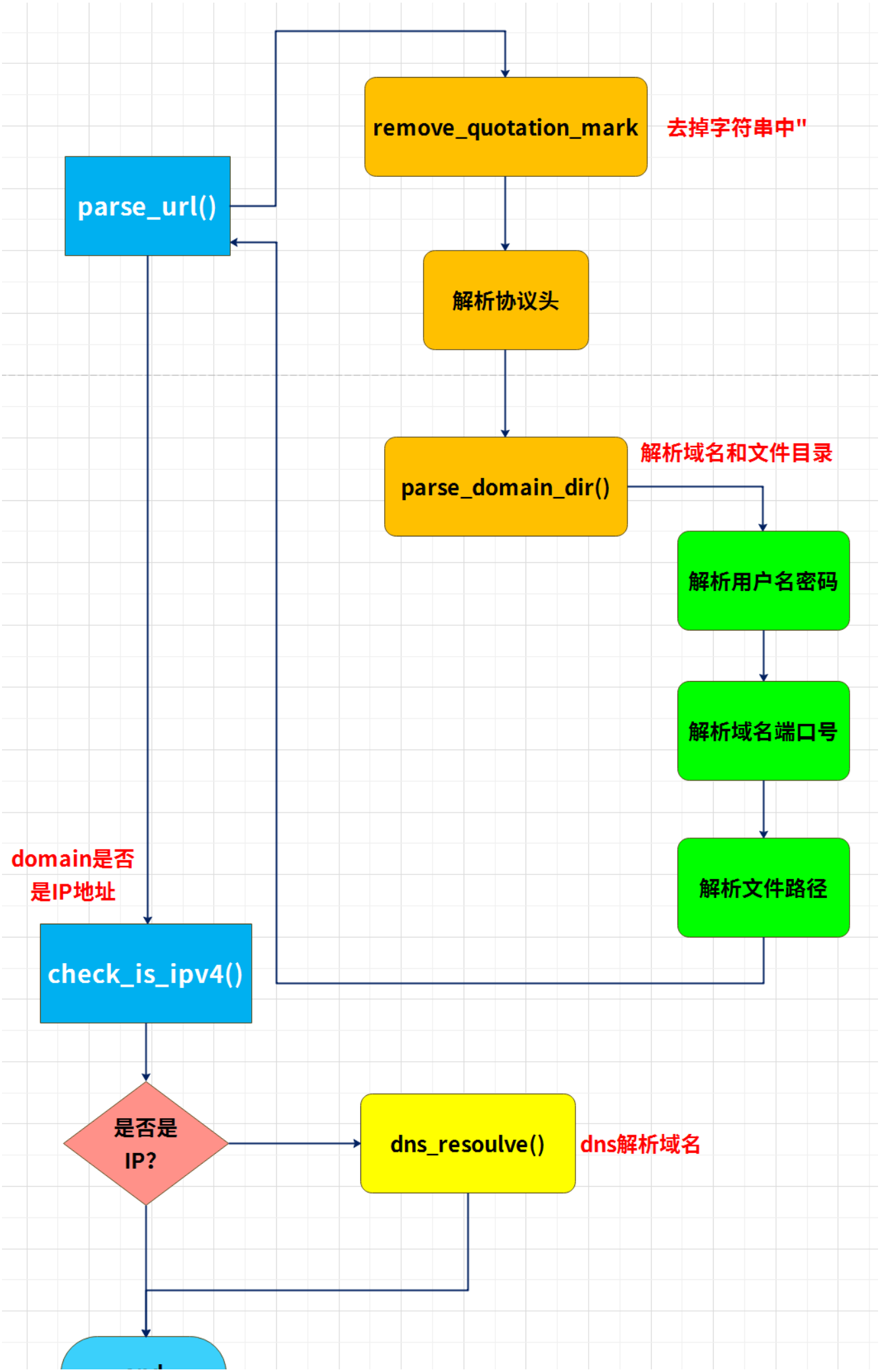
目前本例子只解析以下集中協議，讀者需要支持其他協議可以按照該格式增加對應信息即可

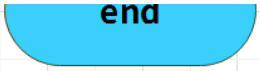
```
#define HEAD_FTP_P "ftp://"
#define HEAD_FTPS_P "ftps://"
#define HEAD_FTPES_P "ftpes://"
#define HEAD_HTTP_P "http://"
#define HEAD_HTTPS_P "https://"

#define PORT_FTP  21
#define PORT_FTPS_I  990    //implicit
#define PORT_FTPS_E  21     //explicit
#define PORT_HTTP 80
#define PORT_HTTPS 443
struct pro_port g_pro_port[]={
 {HEAD_FTP_P,PORT_FTP},
 {HEAD_FTPS_P,PORT_FTPS_I},
 {HEAD_FTPES_P,PORT_FTPS_E},
```

```
  {HEAD_HTTP_P,PORT_HTTP},
  {HEAD_HTTPS_P,PORT_HTTPS},
};
```

四、程序流程圖


remove_quotation_mark
去掉字符串中"
parse_url()
解析协议头
parse_domain_dir()
解析域名和文件目录
解析用户名密码
解析域名端口号
解析文件路径
domain是否
是IP地址
check_is_ipv4()
是否是
IP?
dns_resoulve()
dns解析域名



程序流程相對來說，比較簡單，主函數功能說明如下：

1. parse_url()

```
int parse_url(char *raw_url,URL_RESULT_T *result)
```

參數：

```
raw_url：指向一个url字符串，比如：ftp://peng:pass@baidu.com/dir/index.html
result ：url解析后的结果存放在该结构体中

结构体类型定义如下：
typedef struct
{
 char user[MAX_USER_LEN];
 char pass[MAX_PASS_LEN];
 char domain[INET_DOMAINSTRLEN];//域名
 char svr_dir[MAX_PATH_FILE_LEN]; //文件路径
 char svr_ip[MAX_IP_STR_LEN];
 int port;
}URL_RESULT_T;
```

功能:

```
解析url字符串，并将解析结果存放在result中
```

返回值;

```
成功返回 URL_OK
失败返回 URL_ERROR
```

2. void remove_quotation_mark()

```
void remove_quotation_mark(char *input)
```

参數

```
input：字符串
```

功能

```
去掉字符串中的双引号 \"
```

返回值

```
无
```

3. parse_domain_dir

```
int  parse_domain_dir(char *url,URL_RESULT_T *result)
```

参數

```
url：执行去掉协议头的url字符串，比如：peng:pass@baidu.com/dir/index.html
result  ：url解析后的结果存放在该结构体中
```

功能

```
解析出url中用户名、密码、域名/ip、文件路径等信息
```

返回值

```
成功：URL_OK
失败：URL_ERROR
```

失败：URL_ERROR

4. check_is_ipv4()

```
int check_is_ipv4(char *domain)
```

参數

domain：指向一个域名或者IP地址点分十进制字符串，最大长度为：MAX_URL_LEN

功能

判断domain中存放的是不是合法的IP地址

返回值

1：是IP地址

-1：不是IP地址

5、dns_resoulve()

```
int dns_resoulve(char *svr_ip,const char *domain)
```

参數

svr_ip：存放DNS协议解析过的域名对应的IP地址点分十进制字符串
domain：域名字符串

功能

将domain中的域名，通过DNS协议解析成对应的IP地址

返回值

```
成功：URL_OK
失败：URL_ERROR
```

五、運行

測試程序

```
void main(void)
{
 int ret;

 char url_str[256]="ftp://peng:pass@baidu.com/dir/index.html";
 parse_url(url_str,&url_result_t);

 ret = check_is_ipv4(url_result_t.domain);
 if(ret != 1)
 {
  //dns
  dns_resoulve(url_result_t.svr_ip,url_result_t.domain);
 }
 printf("\n-------------result--------------\n");

 printf("user:%s\n",url_result_t.user);
 printf("pass:%s\n",url_result_t.pass);
 printf("port:%d\n",url_result_t.port);
 printf("domain:%s\n",url_result_t.domain);
 printf("svr_dir:%s\n",url_result_t.svr_dir);
 printf("svr_ip:%s\n",url_result_t.svr_ip);

 printf("-------------end--------------\n");
}
```

執行結果

```
peng@ubuntu:/mnt/hgfs/code/url$ ./a.out

------------result--------------
user:peng
pass:pass
port:21
domain:baidu.com
svr_dir:/dir/index.html
svr_ip:220.181.38.148
-------------end---------------
```

六、代碼獲取

完整代碼可以進入我的倉庫獲取

```
https://gitee.com/yikoulinux/url
```

小程序

19個C語言必殺技，宏定義的常用方法總結~

嵌入式資訊精選