



平台：论坛 博客 Club168 精华 文库 自测 访谈录 | 频道：操作系统 开发 数据库 存储 服务器 网络 IT新闻 Linux 下载 Power用户组

·买域名送域名 海外主机免备案 ·挑战思科交换机配置·赢手机充值卡 ·CU币兑换活动获奖名单已公布 ·如何从Linux菜鸟进化成老鸟

论坛 操作系统 Linux论坛 程序开发 [原创] 用 C 语言编写一个网络蜘蛛来搜索网上出现的电子 ...

发表主题

版块跳转最近访问板块12345678下一页

查看: 14799 | 回复: 72

zhoulifa

丰衣足食
帖子 165
主题 29
精华 8
可用积分 740
专家积分 0
在线时间 1 小时
注册时间 2005-09-06
最后登录 2009-10-25
串门 好友
博客 消息
论坛徽章: 0

[原创] 用 C 语言编写一个网络蜘蛛来搜索网上出现的电子邮件地址 [复制链接]

0 0

发表于 2006-09-01 09:53:56 | 只看该作者 | 倒序浏览 [报告] [收藏(0)] 1楼 电梯直达

可能大家经常要去互联网上搜索特定的内容，比如收集大量邮件地址，如果用 google 之类的搜索引擎是没法实现这种特定功能的，所以用 C 语言来写一个吧。它的功能就是不断去取得网络上的页面，然后分析出网页上出现的邮件地址保存下来。象个蜘蛛一样，从网络上一个网页爬向另一个网页，不停地搜索邮件地址。

当然这只是一个原理展示程序，并没有进行优化。

这个程序的 main 函数流程图如下：

即：分析程序运行时的参数，把各网页地址作为根节点加入到链表，然后从链表头开始处理各节点

对整个链表的处理是先处理兄弟节点，流程图如下：

然后再处理各节点的子节点，流程图如下：

当然，这里采用了递归调用方法，处理子节点的数据时和处理整个链表一样循环处理就是了。

/******关于本文档*****
*filename: 用 C 语言编写一个网络蜘蛛来搜索网上出现的电子邮件地址
*purpose: 一个邮址搜索程序的雏形
*wrote by: zhoulifa(zhoulifa@163.com) 周立发(<http://zhoulifa.bokee.com>)
Linux爱好者 Linux知识传播者 SOHO族 开发者 最擅长C语言
*date time:2006-08-31 21:00:00
*Note: 任何人可以任意复制代码并运用这些文档，当然包括你的商业用途
* 但请遵循GPL
*Hope:希望越来越多的人贡献自己的力量，为科学技术发展出力
*****/

程序在运行的过程中要建立一个树形链表结构，结构图如下：


程序启动时分析所带参数，把各参数加入到根网页节点，如果有多个参数则这个根网页有兄弟节点。
然后从根节点开始处理这一级上各节点，把各节点网页上出现的网页链接加到该节点的子节点上，处理完当前这一级后处理子节点这一级。

源代码如下：
[code]
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <netdb.h>
#include <errno.h>
#include <locale.h>

#define USERAGENT "Wget/1.10.2"
#define ACCEPT "*/*"
#define ACCEPTLANGUAGE "zh-cn,zh;q=0.5"
#define ACCEPTENCODING "gzip,deflate"
#define ACCEPTCHARSET "gb2312,utf-8;q=0.7,*;q=0.7"
#define KEEPALIVE "300"

```

#define CONNECTION "keep-alive"
#define CONTENTTYPE "application/x-www-form-urlencoded"

#define MAXFILENAME 14
#define DEBUG 1

typedef struct webnode {
    char * host;          /* 网页所在的主机 */
    int port;             /* 网络服务器所使用的端口 */
    char * dir;           /* 网页所在的目录 */
    char * page;          /* 网页文件名 */
    char * file;          /* 本地保存的文件名 */
    char IsHandled;       /* 是否处理过 */
    struct webnode * brother; /* 兄弟节点链表指针 */
    struct webnode * child; /* 子节点链表指针 */
} WEBNODE;

struct sockaddr_in server_addr;
int sockfd = 0, dsend = 0, totalsend = 0, nbytes = 0, reqn = 0, i = 0, j = 0, ret = 0;
struct hostent *host;
char request[409600] = "", buffer[1024] = "", httpheader[1024] = "";
int FileNumber = 0;
char e[2] = "@/";
WEBNODE * NodeHeader, * NodeTail, * NodeCurr;
char * mapped_mem;

int GetHost(char *, char **, char **, int *, char **); /**/
void AnalyzePage(WEBNODE *); /**/
void AddInitNode(char *, char *, int, char *); /**/
void HandleInitNode(WEBNODE *); /**/
void DisplayNode(WEBNODE *); /**/
void HandOneNode(WEBNODE *); /**/
void DoneWithList(int); /**/
void DoOnce(); /**/
void ConnectWeb(void); /**/
void SendRequest(void); /**/
void ReceiveResponse(void); /**/
void GetEmail(char *); /**/
void GetLink(char *); /**/
void GetBeforePos(char *, char **); /**/
void GetAfterPos(char *, char **); /**/
void AddChildNode(WEBNODE *, char *); /**/
void GetAfterPosWithSlash(char *, char **); /**/
void GetMemory(char **, int); /**/
int IsExistWeb(WEBNODE *, char *, char *, int, char *); /**/
void Rstrchr(char *, int, char **); /**/
int GetLocalAgent(char * UserAgent, char * Accept, char * AcceptLanguage, char * AcceptEncoding, char *
AcceptCharset, char * KeepAlive, char * Connection, char * ContentType); /**/

/*****
功能：设置 HTTP 协议头内容的一些固定值
*****/
int GetLocalAgent(char * UserAgent, char * Accept, char * AcceptLanguage, char * AcceptEncoding, char *
AcceptCharset, char * KeepAlive, char * Connection, char * ContentType)
{
    memcpy(UserAgent, USERAGENT, strlen(USERAGENT));
    memcpy(Accept, ACCEPT, strlen(ACCEPT));
    memcpy(AcceptLanguage, ACCEPTLANGUAGE, strlen(ACCEPTLANGUAGE));
    memcpy(AcceptEncoding, ACCEPTENCODING, strlen(ACCEPTENCODING));
    memcpy(AcceptCharset, ACCEPTCHARSET, strlen(ACCEPTCHARSET));
    memcpy(KeepAlive, KEEPALIVE, strlen(KEEPALIVE));
    memcpy(Connection, CONNECTION, strlen(CONNECTION));
    memcpy(ContentType, CONTENTTYPE, strlen(CONTENTTYPE));
    return 0;
}

/*****
功能：在字符串 s 里搜索 x 字符，并设置指针 d 指向该位置
*****/
void Rstrchr(char * s, int x, char ** d)
{

```

```

        int len = strlen(s) - 1;
        while(len >= 0)        {
            if(x == s[len]) {(*d) = s + len; return;}
            len--;
        }
        (*d) = 0;
    }

/*****
功能：连接一个网站服务器
*****/
void ConnectWeb(void) { /* connect to web server */
    /* create a socket descriptor */
    if((sockfd=socket(PF_INET,SOCK_STREAM,0))==-1)
    {
        fprintf(stderr,"Socket Error:%s\n",strerror(errno));
        exit(1);
    }

    /* bind address */
    bzero(&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(NodeCurr->port);
    server_addr.sin_addr = *((struct in_addr *)host->h_addr);

    /* connect to the server */
    if(connect(sockfd, (struct sockaddr *)&server_addr, sizeof(struct sockaddr)) == -1)
    {
        fprintf(stderr, "Connect Error:%s\n", strerror(errno));
        exit(1);
    }
}

/*****
功能：向网站发送 HTTP 请求
*****/
void SendRequest(void) { /* send my http-request to web server */
    dsend = 0; totalsend = 0;
    nbytes= strlen(request);
    while(totalsend < nbytes) {
        dsend = write(sockfd, request + totalsend, nbytes - totalsend);
        if(dsend==-1) {fprintf(stderr, "send error!\n", strerror(errno));exit(0);}
        totalsend+=dsend;
        fprintf(stdout, "\nRequest.%d %d bytes send OK!\n", reqn, totalsend);
    }
}

/*****
功能：接收网站的 HTTP 返回
*****/
void ReceiveResponse(void) { /* get response from web server */
    fd_set writefds;
    struct timeval tival;
    int retry = 0;
    FILE * localfp = NULL;

    i=0; j = 0;
__ReCeive:
    FD_ZERO(&writefds);
    tival.tv_sec = 10;
    tival.tv_usec = 0;
    if(sockfd > 0) FD_SET(sockfd, &writefds);
    else {fprintf(stderr, "\nError, socket is negative!\n"); exit(0);}

    ret = select(sockfd + 1, &writefds, NULL, NULL, &tival);
    if(ret ==0 ) {
        if(retry++ < 10) goto __ReCeive;
    }
    if(ret <= 0) {fprintf(stderr, "\nError while receiving!\n"); exit(0);}

    if(FD_ISSET(sockfd, &writefds)) {

```

```

memset(buffer, 0, 1024);
memset(httpheader, 0, 1024);
if((localfp = fopen(NodeCurr->file, "w")) == NULL) {if(DEBUG) fprintf(stderr, "create file '%s' error\n", NodeCurr->file); return;}
/* receive data from web server */
while((nbytes=read(sockfd,buffer,1))!=1)
{
    if(i < 4) { /* 获取 HTTP 消息头 */
        if(buffer[0] == '\r' || buffer[0] == '\n') i++;
        else i = 0;
        memcpy(httpheader + j, buffer, 1); j++;
    }
    else { /* 获取 HTTP 消息体 */
        fprintf(localfp, "%c", buffer[0]); /* print content on the screen */
        //fprintf(stdout, "%c", buffer[0]); /* print content on the screen */
        i++;
    }
}
fclose(localfp);
}
}

/*****
功能：执行一次 HTTP 请求
*****/
void DoOnce() { /* send and receive */
    ConnectWeb(); /* connect to the web server */

    /* send a request */
    SendRequest();

    /* receive a response message from web server */
    ReceiveResponse();

    close(sockfd); /* because HTTP protocol do something one connection, so I can close it after receiving */
}

/*****
功能：执行 HTTP 请求
*****/
void DoneWithList(int flag) {
    if(flag) fprintf(stdout, "\tRequest.%d is:\n%s", ++reqn, request);

    DoOnce();

    if(flag) fprintf(stdout, "\n\tThe following is the response header:\n%s", httpheader);
}

/*****
功能：从字符串 src 中分析出网站地址和端口，并得到文件和目录
*****/
int GetHost(char * src, char ** web, char ** file, int * port, char ** dir) {
    char * pA, * pB, * pC;
    int len;

    *port = 0;
    if(!(*src)) return -1;
    pA = src;
    if(!strcmp(pA, "http://", strlen("http://"))) pA = src+strlen("http://");
    /* else if(!strcmp(pA, "https://", strlen("https://"))) pA = src+strlen("https://"); */
    else return 1;
    pB = strchr(pA, '/');
    if(pB) {
        len = strlen(pA) - strlen(pB);
        GetMemory(web, len);
        memcpy((*web), pA, len);
        if(*(pB+1)) {
            Rstrchr(pB + 1, '/', &pC);
            if(pC) len = strlen(pB + 1) - strlen(pC);
            else len = 0;
            if(len > 0) {

```

```

    GetMemory(dir, len);
    memcpy((*dir), pB + 1, len);

    if(pC + 1) {
        len = strlen(pC + 1);
        GetMemory(file, len);
        memcpy((*file), pC + 1, len);
    }
    else {
        len = 1;
        GetMemory(file, len);
        memcpy((*file), e, len);
    }
}
else {
    len = 1;
    GetMemory(dir, len);
    memcpy((*dir), e + 1, len);

    len = strlen(pB + 1);
    GetMemory(file, len);
    memcpy((*file), pB + 1, len);
}
}
else {
    len = 1;
    GetMemory(dir, len);
    memcpy((*dir), e + 1, len);

    len = 1;
    GetMemory(file, len);
    memcpy((*file), e, len);
}
}
else {
    len = strlen(pA);
    GetMemory(web, len);
    memcpy((*web), pA, strlen(pA));
    len = 1;
    GetMemory(dir, len);
    memcpy((*dir), e + 1, len);
    len = 1;
    GetMemory(file, len);
    memcpy((*file), e, len);
}

pA = strchr((*web), ':');
if(pA) *port = atoi(pA + 1);
else *port = 80;

return 0;
}

/*****
*filename: mailaddrsearch.c
*purpose: 用 C 语言编写一个网络蜘蛛来搜索网上出现的电子邮件地址
*tidied by: zhoulifa(zhoulifa@163.com) 周立发(http://zhoulifa.bokee.com)
Linux爱好者 Linux知识传播者 SOHO族 开发者 最擅长C语言
*date time:2006-08-31 21:00:00
*Note: 任何人可以任意复制代码并运用这些文档，当然包括你的商业用途
* 但请遵循GPL
*Thanks to: www.gd-linux.org 广东省 Linux 公共服务技术支持中心
*****/

int main(int argc, char ** argv)
{
    int WebPort;
    char * WebHost = 0, * PageAddress = 0, * WebDir = 0;

    if(argc < 2) {if(DEBUG) fprintf(stdout, "Command error, you should input like this:\n\t%s WebPageAddress1
WebPageAddress2 WebPageAddress3 ...", argv[0]); exit(0);}

```

```

NodeHeader = NodeTail = NodeCurr = 0;
//setlocale(LC_ALL, "zh_CN.gb2312");
for(i = 1; i < argc; i++) {
    ret = GetHost(argv, &WebHost, &PageAddress, &WebPort, &WebDir); /* Get web page info */
    if(ret) {if(DEBUG) fprintf(stdout, "GetHost error from '%s'\n", argv); exit(0);}
    AddInitNode(WebHost, PageAddress, WebPort, WebDir); /* add this page to chain */
}
free(WebHost); free(PageAddress); free(WebDir);
if(DEBUG) {
    fprintf(stdout, "\nDisplay.%5d:", FileNumber);
    DisplayNode(NodeHeader); /* display every node */
}
HandleInitNode(NodeHeader); /* handle every page */
return 0;
}

/*****
功能：分析网页
*****/
void AnalyzePage(WEBNODE * node)
{
    int fd;
    int flength = 0;
    fd = open(node->file, O_RDONLY);
    if(fd == -1) goto __AnalyzeDone;
    flength = lseek(fd, 1, SEEK_END);
    write(fd, "\0", 1);
    lseek(fd, 0, SEEK_SET);
    mapped_mem = mmap(0, flength, PROT_READ, MAP_PRIVATE, fd, 0);
    GetEmail(mapped_mem);
    GetLink(mapped_mem);
    close(fd);
    munmap(mapped_mem, flength);
__AnalyzeDone:
    close(fd);
    node->IsHandled = 1;
    remove(node->file);
}

/*****
功能：为根节点设置兄弟节点
*****/
void AddInitNode(char * Host, char * Page, int Port, char * Dir)
{
    WEBNODE * NewNode;
    char filename[MAXFILENAME + 1] = "";

    if(NodeHeader == NULL) NewNode = NodeHeader = (WEBNODE *)malloc(sizeof(WEBNODE));
    else NodeTail->brother = NewNode = (WEBNODE *)malloc(sizeof(WEBNODE));
    memset(NewNode, 0, sizeof(WEBNODE));
    NewNode->host = (char *)malloc(strlen(Host) + 1);
    memset(NewNode->host, 0, strlen(Host) + 1);
    NewNode->page = (char *)malloc(strlen(Page) + 1);
    memset(NewNode->page, 0, strlen(Page) + 1);
    NewNode->dir = (char *)malloc(strlen(Dir) + 1);
    memset(NewNode->dir, 0, strlen(Dir) + 1);
    NewNode->file = (char *)malloc(MAXFILENAME + 1);
    memset(NewNode->file, 0, MAXFILENAME + 1);
    strcpy(NewNode->host, Host);
    strcpy(NewNode->page, Page);
    strcpy(NewNode->dir, Dir);
    sprintf(filename, "file%05d.html", FileNumber++);
    strcpy(NewNode->file, filename);
    NewNode->port = Port;
    NewNode->IsHandled = 0;
    NewNode->brother = 0;
    NewNode->child = 0;
    NodeTail = NewNode;
}

```

```

/*****
功能：处理根节点信息
*****/
void HandleInitNode(WEBNODE * node)
{
    WEBNODE * CurrentNode = 0;
    CurrentNode = node;
    if(CurrentNode) {
        while(CurrentNode) {
            if(CurrentNode->IsHandled == 0) {
                HandOneNode(CurrentNode);
                if(DEBUG) {
                    fprintf(stdout, "\nDisplay.%5d:", FileNumber);
                    DisplayNode(NodeHeader); /* display every node */
                }
                CurrentNode = CurrentNode->brother;
            }
            CurrentNode = node;
            while(CurrentNode) {
                if(CurrentNode->child && CurrentNode->child->IsHandled == 0) {
                    HandleInitNode(CurrentNode->child);
                }
                CurrentNode = CurrentNode->brother;
            }
        }
    }
}

/*****
功能：显示年有节点信息
*****/
void DisplayNode(WEBNODE * NodeHeader)
{
    WEBNODE * TempNode;
    TempNode = NodeHeader;
    fprintf(stdout, "\n");
    while(TempNode) {
        if(!strcmp(TempNode->dir, "/")) fprintf(stdout, "\t%s:%d%s%s => %s %d\n", TempNode->host,
        TempNode->port, TempNode->dir, strcmp(TempNode->page, "@")?TempNode->page:"", TempNode->file,
        TempNode->IsHandled);
        else fprintf(stdout, "\t%s:%d/%s%s => %s %d\n", TempNode->host, TempNode->port, TempNode->
        >dir, strcmp(TempNode->page, "@")?TempNode->page:"", TempNode->file, TempNode->IsHandled);
        TempNode = TempNode->brother;
    }
    TempNode = NodeHeader;
    while(TempNode) {
        if(TempNode->child) DisplayNode(TempNode->child);
        TempNode = TempNode->brother;
    }
}

/*****
功能：处理单个节点信息
*****/
void HandOneNode(WEBNODE * node)
{
    char UserAgent[1024] = "", Accept[1024] = "", AcceptLanguage[1024] = "", AcceptEncoding[1024] = "",
    AcceptCharset[1024] = "", KeepAlive[1024] = "", Connection[1024] = "", ContentType[1024] = "";

    NodeCurr = node;
    if((host=gethostbyname(NodeCurr->host))==NULL) /* get ip address by domain */
    {
        if(DEBUG) fprintf(stderr, "\tGethostname '%s' error, %s\n", NodeCurr->host, strerror(errno));
        exit(1);
    }
    GetLocalAgent(UserAgent, Accept, AcceptLanguage, AcceptEncoding, AcceptCharset, KeepAlive, Connection,
    ContentType); /* Get client browser information */

    if(strcmp(NodeCurr->dir, "/")) sprintf(request, "GET /%s/%s HTTP/1.0\r\nHost: %s\r\nUser-Agent:
    %s\r\nAccept: %s\r\nConnection: %s\r\n\r\n", NodeCurr->dir, strcmp(NodeCurr->page, "@")?NodeCurr->page:"",
    NodeCurr->host, UserAgent, Accept, Connection);

```

```

        else    sprintf(request, "GET %s%s HTTP/1.0\r\nHost: %s\r\nUser-Agent: %s\r\nAccept: %s\r\nConnection:
%s\r\n\r\n", NodeCurr->dir, strcmp(NodeCurr->page, "@")?NodeCurr->page:"", NodeCurr->host, UserAgent, Accept,
Connection);
        DoneWithList(1);
        AnalyzePage(NodeCurr);
    }

/*****
功能：从字符串 src 中分析出邮件地址保存到文件
*****/
void GetEmail(char * src)
{
    char * pa, * pb, * pc, * pd;
    char myemail[1024] = "";
    FILE * mailfp = NULL;
    if((mailfp = fopen("email.txt", "a+")) == NULL)    return;
    pa = src;
    while((pb = strchr(pa, '@'))    {
        GetBeforePos(pb, &pc);
        GetAfterPos(pb, &pd);
        if(pc && pd && (strlen(pc) > (strlen(pd) + 3)))    {
            memset(myemail, 0, 1024);
            memcpy(myemail, pc, strlen(pc) - strlen(pd));
            if(strcmp(NodeCurr->dir, "/")) fprintf(mailfp, "%s\thttp://%s/%s/%s\n", myemail, NodeCurr->host,
NodeCurr->dir, strcmp(NodeCurr->page, "@")?NodeCurr->page:"");
            else fprintf(mailfp, "%s\thttp://%s%s%s\n", myemail, NodeCurr->host, NodeCurr->dir,
strcmp(NodeCurr->page, "@")?NodeCurr->page:"");
            if(*(pd + 1))    pa = pd + 1;
            else break;
        }
        else if(*(pb + 1))    pa = pb + 1;
        else    break;
    }
    fclose(mailfp);
}

/*****
功能：从 src 中找出前面的字母、数字等内含，即 email 地址中 @ 的前面部分
*****/
void GetBeforePos(char * src, char ** d)
{
    char * x;
    if(src - 1)    x = src - 1;
    else {*d = 0; return ;}
    while(x)    {
        if(*x >= 'a' && *x <= 'z') {x--; continue;}
        else if(*x >= 'A' && *x <= 'Z') {x--; continue;}
        else if(*x >= '0' && *x <= '9') {x--; continue;}
        else if(*x == '.' || *x == '-' || *x == '_') {x--; continue;}
        else {break;}
    }
    x++;
    if(x) *d = x;
    else *d = 0;
}

/*****
功能：从 src 中找出后面的字母、数字等内含，即 email 地址中 @ 的后面部分
*****/
void GetAfterPos(char * src, char ** d)
{
    char * x;
    if(src + 1)    x = src + 1;
    else {*d = 0; return ;}
    while(x)    {
        if(*x >= 'a' && *x <= 'z') {x++; continue;}
        else if(*x >= 'A' && *x <= 'Z') {x++; continue;}
        else if(*x >= '0' && *x <= '9') {x++; continue;}
        else if(*x == '.' || *x == '-' || *x == '_') {x++; continue;}
        else {break;}
    }
}

```



```
if(x) *d = x;
else *d = 0;
}
```

本版精华

- 多线程程序中利用管道控制 [select](#) 行为
- 转贴 - 如何在 [Linux](#) 下调试动态链接库
- [Linux](#)下各类TCP网络服务器的实现源代码
- 做了个下载器给大家试试[已升级到0.8.0版]
- [redhat9](#)中支持的最大线程
- [原创]基于[proc](#)文件系统的简易主机端口扫描器
- >>共享一个包发生器
- [Linux\(Unix\)](#)下MySQL数据库访问接口程序MCI (MySQL Call
- 利用[libtool](#)自动生成动态库的Makefile的生成方法
- 用[tcpdump](#)分析协议后用C语言自己编写一个BBS发贴机器人

光顾一下周立发的Web log 参加周立发的Linux讨论

对于大表多层hash join，手工增加hash_area ... | 很高兴今天又做了一个ORACLE NMAES 服务器 | 【求助】Oracle 10g 如何收集索引的使用情况？ ... | 读取数据块到内存的问题

zhoulifa



丰衣足食



帖子165
主题29
精华8
可用积分740
专家积分0
在线时间1 小时
注册时间2005-09-06
最后登录2009-10-25

串门好友
博客消息

论坛徽章：0

发表于 2006-09-01 09:54:37 | 只看该作者

[报告] 2楼

```
/******
功能：从 src 中找出前面的字母、数字等内含，即一个网页地址中主机名后面的部分
*****/
void GetAfterPosWithSlash(char * src, char ** d)
{
    char * x;
    if(src) x = src;
    else { *d = 0; return ;}
    while(x) {
        if(*x >= 'a' && *x <= 'z') {x++; continue;}
        else if(*x >= 'A' && *x <= 'Z') {x++; continue;}
        else if(*x >= '0' && *x <= '9') {x++; continue;}
        else if(*x == '.' || *x == '-' || *x == '_' || *x == '=') {x++; continue;}
        else if(*x == ':' || *x == '/' || *x == '?' || *x == '&') {x++; continue;}
        else {break;}
    }
    if(x) *d = x;
    else *d = 0;
}

/******
功能：为 myanchor 分配 len 大小的内存
*****/
void GetMemory(char ** myanchor, int len)
{
    if(!(*myanchor)) (*myanchor) = (char *)malloc(len + 1);
    else (*myanchor) = (char *)realloc((void *)(*myanchor), len + 1);
    memset((*myanchor), 0, len + 1);
}

/******
功能：从 src 中分析出网页链接，并加入到当前节点的子节点上
*****/
void GetLink(char * src)
{
    char * pa, * pb, * pc;
    char * myanchor = 0;
    int len = 0;

    pa = src;
    do {
        if((pb = strstr(pa, "href="))) {
            pc = strchr(pb + 6, '"');
            len = strlen(pb + 6) - strlen(pc);
            GetMemory(&myanchor, len);
            memcpy(myanchor, pb + 6, len);
        }
        else if((pb = strstr(pa, "href=\"")) {
            pc = strchr(pb + 6, '"');
            len = strlen(pb + 6) - strlen(pc);
            GetMemory(&myanchor, len);
            memcpy(myanchor, pb + 6, len);
        }
    } while(pb);
}
```

```

    }
    else if((pb = strstr(pa, "href=")))    {
        GetAfterPosWithSlash(pb + 5, &pc);
        len = strlen(pb + 5) - strlen(pc);
        GetMemory(&myanchor, len);
        memcpy(myanchor, pb + 5, len);
    }
    else {goto __returnLink ;}

/*
    if(DEBUG)    {
        if(strcmp(NodeCurr->dir, "/"))    fprintf(stdout, "%s\\thttp://%s/%s/%s\\n", myanchor, NodeCurr->host, NodeCurr->dir, strcmp(NodeCurr->page, "")?NodeCurr->page:"");
        else    fprintf(stdout, "%s\\thttp://%s%s%s\\n", myanchor, NodeCurr->host, NodeCurr->dir, strcmp(NodeCurr->page, "")?NodeCurr->page:"");
    }
*/

    if(strlen(myanchor) > 0)    AddChildNode(NodeCurr, myanchor);
    if(pc + 1)    pa = pc + 1;
}while(pa);
__returnLink:
    return;
}

/*****
功能：为当前节点增加子节点
*****/

void AddChildNode(WEBNODE * node, char * src)
{
    int WebPort, len;
    char * WebHost = 0, * PageAddress = 0, * WebDir = 0, * pC = 0;
    WEBNODE * NewNode;
    char filename[MAXFILENAME + 1] = "";
    char IsFromRoot = 0;

    if(!src)    return;
    if(!strncasecmp(src, "mailto:", strlen("mailto:")))    return ;
    if(strstr(src, ".css"))    return;
    if(strstr(src, ".xml"))    return;
    if(strstr(src, ".ico"))    return;
    if(strstr(src, ".jpg"))    return;
    if(strstr(src, ".gif"))    return;
    if(strstr(src, "javascript:"))    return;
    if(strstr(src, "+"))    return;

    ret = GetHost(src, &WebHost, &PageAddress, &WebPort, &WebDir);
    if(ret)    {
        len = strlen(node->host);
        GetMemory(&WebHost, len);
        strcpy(WebHost, node->host);

        WebPort = node->port;

        IsFromRoot = !strncmp(src, "/", 1);
        if(IsFromRoot && (src + 1))    Rstrchr(src + 1, '/', &pC);
        else if(!IsFromRoot)    Rstrchr(src, '/', &pC);
        else    pC = 0;

        if(pC)    {
            if(IsFromRoot)    len = strlen(src + 1) - strlen(pC);
            else    len = strlen(src) - strlen(pC) + strlen(node->dir) + 1;
            GetMemory(&WebDir, len);
            if(IsFromRoot)    memcpy(WebDir, src + 1, len);
            else    {memcpy(WebDir, node->dir, strlen(node->dir)); strcat(WebDir, "/"); memcpy(WebDir +
strlen(node->dir) + 1, src, strlen(src) - strlen(pC));}

            if(pC + 1)    {
                len = strlen(pC + 1);
                GetMemory(&PageAddress, len);
                strcpy(PageAddress, pC + 1);
            }
            else    {

```

```

        len = 1;
        GetMemory(&PageAddress, len);
        memcpy(PageAddress, e, len);
    }
}
else {
    if(IsFromRoot) {
        len = 1;
        GetMemory(&WebDir, len);
        memcpy(WebDir, e + 1, len);

        len = strlen(src + 1);
        GetMemory(&PageAddress, len);
        memcpy(PageAddress, src + 1, len);
    }
    else {
        len = strlen(node->dir);
        GetMemory(&WebDir, len);
        memcpy(WebDir, node->dir, len);

        len = strlen(src);
        GetMemory(&PageAddress, len);
        memcpy(PageAddress, src, len);
    }
}
}
ret = IsExistWeb(NodeHeader, WebHost, PageAddress, WebPort, WebDir);
if(ret) goto __ReturnAdd;

if(node->child == NULL)    NewNode = node->child = (WEBNODE *)malloc(sizeof(WEBNODE));
else NodeTail->brother = NewNode = (WEBNODE *)malloc(sizeof(WEBNODE));
memset(NewNode, 0, sizeof(WEBNODE));
NewNode->host = (char *)malloc(strlen(WebHost) + 1);
memset(NewNode->host, 0, strlen(WebHost) + 1);
NewNode->page = (char *)malloc(strlen(PageAddress) + 1);
memset(NewNode->page, 0, strlen(PageAddress) + 1);
NewNode->dir = (char *)malloc(strlen(WebDir) + 1);
memset(NewNode->dir, 0, strlen(WebDir) + 1);
NewNode->file = (char *)malloc(MAXFILENAME + 1);
memset(NewNode->file, 0, MAXFILENAME + 1);
strcpy(NewNode->host, WebHost);
strcpy(NewNode->page, PageAddress);
strcpy(NewNode->dir, WebDir);
sprintf(filename, "file%05d.html", FileNumber++);
strcpy(NewNode->file, filename);
NewNode->port = WebPort;
NewNode->IsHandled = 0;
NewNode->brother = 0;
NewNode->child = 0;
NodeTail = NewNode;
__ReturnAdd:
    free(WebHost); free(PageAddress); free(WebDir);
}

/*****
功能：检查是否已经处理过的网页
*****/
int IsExistWeb(WEBNODE * node, char * host, char * page, int port, char * dir)
{
    WEBNODE * t;
    t = node;
    while(t) {
        if(!strcmp(t->host, host) && !strcmp(t->page, page) && t->port == port && !strcmp(t->dir, dir)) return 1;
        t = t->brother;
    }
    t = node;
    while(t) {
        if(t->child) {
            ret = IsExistWeb(t->child, host, page, port, dir);
            if(ret) return 2;
        }
    }
}

```

<pre> t = t->brother; } return 0; } [/code]</pre> <p>编译这个程序：</p> <pre>gcc mailaddrsearch.c -o mailsearcher</pre> <p>输入一个网址作为参数运行一下试试吧：</p> <pre>./mailsearcher http://zhoulifa.bokee.com/5531748.html</pre> <p>程序首先找出 http://zhoulifa.bokee.com/5531748.html 页面上的邮件地址保存到当前目录下 email.txt 文件里，每行一条记录，格式为邮件地址和出现该邮件地址的网页。然后分析这个页面上出现的网页链接，把各链接作为子节点加入链表，再去处理子节点，重复上述操作。</p> <p>这只是一个示例程序，并不完善，如果要使其达到实用的目的，还需要让这个程序效率更高点，比如加入 epoll（在 2.4 内核中只有 select 了）实现 I/O 多路复用。又比如对每个子节点实现多线程，每个线程处理一个节点。</p> <p>如果对 I/O 多路复用不熟悉，您可以看一下我这篇文章 http://zhoulifa.bokee.com/5345930.html 里关于“Linux 下各类TCP网络服务器的实现源代码”</p> <p>光顾一下周立发的Web log 参加周立发的Linux讨论</p> <p>Mysql存储引擎之生产应用 2012数据库技术大会PPT下载 IDC行业云计算应用有奖调查 桌面虚拟化，您现在还在等什么？</p>	
<div>醉卧水云间</div> <div></div> <div>腰缠万贯 🐶🐶</div> <div><div>帖子33513</div><div>主题5955</div><div>精华4</div><div>可用积分8922</div><div>专家积分246</div><div>在线时间5003 小时</div><div>注册时间2006-07-19</div><div>最后登录2012-05-19</div><div><div>串门</div><div>好友</div><div>博客</div><div>消息</div></div><div>论坛徽章：0</div></div>	<div> 发表于 2006-09-01 19:29:54 只看该作者<div>[报告] 3楼</div></div> <div>支持</div>
<div>fanyunfei</div> <div></div> <div>白手起家</div> <div><div>帖子96</div><div>主题14</div><div>精华0</div><div>可用积分79</div><div>专家积分0</div><div>在线时间0 小时</div><div>注册时间2006-06-16</div><div>最后登录2007-01-30</div><div><div>串门</div><div>好友</div><div>博客</div><div>消息</div></div><div>论坛徽章：0</div></div>	<div> 发表于 2006-09-01 23:13:26 只看该作者<div>[报告] 4楼</div></div> <div>up</div>
<div>flw</div> <div>外法猎手</div>	<div> 发表于 2006-09-02 09:55:00 只看该作者<div>[报告] 5楼</div></div> <div>真佩服这些用 C 语言的高手们——要换了我，我就用 Perl。</div>

<div></div> <div>版主 😄😄</div> <div>帖子29097 主题1498 精华22 可用积分80964 专家积分1309 在线时间13953 小时 注册时间2002-08-12 最后登录2012-05-20</div> <div>串门好友博客消息</div> <div>论坛徽章: 0</div>	<div></div> <div>2012 高薪诚聘大量研发工程师，站内联系</div> <div>Mysql存储引擎之生产应用 2012数据库技术大会PPT下载 IDC行业云计算应用有奖调查 桌面虚拟化，您现在还在等什么？</div>
<div>周若水</div> <div>唏嘘的猪肉佬</div> <div></div> <div>家境小康 👤</div> <div>帖子130 主题31 精华0 可用积分1398 专家积分0 在线时间45 小时 注册时间2005-02-28 最后登录2011-12-22</div> <div>串门好友博客消息</div> <div>论坛徽章: 0</div>	<div> 发表于 2006-09-02 12:05:21 只看该作者<div>【报告】 6楼</div></div> <div>强，对我这个初学者来说很难，但感谢楼主这种精神</div> <div>Mysql存储引擎之生产应用 2012数据库技术大会PPT下载 IDC行业云计算应用有奖调查 桌面虚拟化，您现在还在等什么？</div>
<div>liuyishao</div> <div></div> <div>稍有积蓄 ⭐</div> <div>帖子842 主题97 精华0 可用积分352 专家积分0 在线时间1 小时 注册时间2005-04-24 最后登录2007-04-21</div> <div>串门好友博客消息</div> <div>论坛徽章: 0</div>	<div> 发表于 2006-09-02 13:26:08 只看该作者<div>【报告】 7楼</div></div> <div>好东西</div> <div>树欲静而风不止， 子欲养而亲不待。</div> <div>Mysql存储引擎之生产应用 2012数据库技术大会PPT下载 IDC行业云计算应用有奖调查 桌面虚拟化，您现在还在等什么？</div>
<div>醉卧水云间</div>	<div> 发表于 2006-09-02 14:29:12 只看该作者<div>【报告】 8楼</div></div> <div>原帖由 flw 于 2006-9-2 09:55 发表 真佩服这些用 C 语言的高手们——要换了我，我就用 Perl。</div>



腰缠万贯



帖子33513

主题5955

精华4

可用积分8922

专家积分246

在线时间5003 小时

注册时间2006-07-19

最后登录2012-05-19

串门

好友

博客

消息

论坛徽章: 0

我相信google的spider不是Perl写的。

flw

外法猎手



版主

帖子29097

主题1498

精华22

可用积分80964

专家积分1309

在线时间13953 小时

注册时间2002-08-12

最后登录2012-05-20

串门

好友

博客

消息

论坛徽章: 0

发表于 2006-09-02 15:06:40 | 只看该作者

[报告] 9楼

原帖由 醉卧水云间 于 2006-9-2 14:29 发表

我相信google的spider不是Perl写的。

这句话的正确性如同我（flw）相信 google 的 spider 不是 C 写的一样。



2012 高薪诚聘大量研发工程师，站内联系

Mysql存储引擎之生产应用 | 2012数据库技术大会PPT下载 | IDC行业云计算应用有奖调查 | 桌面虚拟化，您现在还在等什么？

flw

外法猎手



版主

帖子29097

主题1498

精华22

可用积分80964

专家积分1309

在线时间13953 小时

注册时间2002-08-12

最后登录2012-05-20

串门

好友

博客

消息

论坛徽章: 0

发表于 2006-09-02 15:09:26 | 只看该作者

[报告] 10楼

像这一类程序，耗时的操作主要在于网络通讯上，处理本身是不占用什么时间的，反过来灵活性和扩展性就显得至关重要，用 C 来做爬虫，的确罕见！

BTW：可以请蜘蛛给大家介绍一下。




2012 高薪诚聘大量研发工程师，站内联系

Mysql存储引擎之生产应用 | 2012数据库技术大会PPT下载 | IDC行业云计算应用有奖调查 | 桌面虚拟化，您现在还在等什么？

论坛 操作系统 **Linux论坛** 程序开发 [原创] 用 C 语言编写一个网络蜘蛛来搜索网上出现的电子 ...

高级模式

您需要登录后才可以回帖 登录 | 注册  [用QQ帐号登录](#)

[发表回复](#) ☐ 回帖后跳转到最后一页