linux c语言http get post 请求 & 解析http报头 内容

2016/06/30 10:10 author wzb<wangzhibin_x@foxmail.com>

1.post

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <time.h>
#include <errno.h>
#include <signal.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define IPSTR "120.76.47.120"
#define PORT 80
#define BUFSIZE 1024

int main(int argc, char **argv)
{
        int sockfd, ret, i, h;
        struct sockaddr_in servaddr;
        char str1[4096], str2[4096], buf[BUFSIZE], *str;
        socklen_t len;
        fd_set   t_set1;
        struct timeval  tv;

        if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0 ) {
                printf("创建网络连接失败,本线程即将终止---socket error!\n");
                exit(0);
        };

        bzero(&servaddr, sizeof(servaddr));
        servaddr.sin_family = AF_INET;
        servaddr.sin_port = htons(PORT);
        if (inet_pton(AF_INET, IPSTR, &servaddr.sin_addr) <= 0 ){
                printf("创建网络连接失败,本线程即将终止--inet_pton error!\n");
                exit(0);
        };

        if (connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0){
                printf("连接到服务器失败,connect error!\n");
                exit(0);
        }
        printf("与远端建立了连接\n");

        //发送数据
        memset(str2, 0, 4096);
        strcat(str2, "data=5A25101010215H8880030000000691412011202 07C8_460040124507576_FFFFFFFFFFFFFFFFFFFFFFFFF");
        str=(char *)malloc(128);
        len = strlen(str2);
        sprintf(str, "%d", len);

        memset(str1, 0, 4096);
        strcat(str1, "POST /gsm/bpbg/upload_v1.php HTTP/1.1\n");
        strcat(str1, "Host: api.huayinghealth.com\n");
        strcat(str1, "Content-Type: application/x-www-form-urlencoded\n");
        strcat(str1, "Content-Length: ");
        strcat(str1, str);
        strcat(str1, "\n\n");
```

```
	strcat(str1, str2);
	strcat(str1, "\r\n\r\n");
	printf("%s\n",str1);

	ret = write(sockfd,str1,strlen(str1));
	if (ret < 0) {
		printf("发送失败！错误代码是%d，错误信息是'%s'\n",errno, strerror(errno));
		exit(0);
	}else{
		printf("消息发送成功，共发送了%d个字节！\n\n", ret);
	}

	FD_ZERO(&t_set1);
	FD_SET(sockfd, &t_set1);

	while(1){
		sleep(2);
		tv.tv_sec= 0;
		tv.tv_usec= 0;
		h= 0;
		printf("--------------->1");
		h= select(sockfd +1, &t_set1, NULL, NULL, &tv);
		printf("--------------->2");

		//if (h == 0) continue;
		if (h < 0) {
			close(sockfd);
			printf("在读取数据报文时SELECT检测到异常，该异常导致线程终止！\n");
			return -1;
		};

		if (h > 0){
			memset(buf, 0, 4096);
			i= read(sockfd, buf, 4095);
			if (i==0){
				close(sockfd);
				printf("读取数据报文时发现远端关闭，该线程终止！\n");
				return -1;
			}

			printf("*************** \n");
			printf("%s\n", buf);
			char *test;
			test=rindex(buf,'\r\n');
			printf("%s n",test);
		}
	}
	close(sockfd);


	return 0;
}
---------------------------------
```

2.get

```
#include <stdio.h>
 #include <unistd.h>
 #include <sys/types.h>
 #include <sys/socket.h>
 #include <sys/stat.h>
 #include <fcntl.h>
 #include <string.h>
 #include <arpa/inet.h>
 #include <netdb.h>
 #include <netinet/in.h>
 #include <stdlib.h>
```

```
#define BUFSIZE 0xF000
void geturl(char* url)
{
        int cfd;
        struct sockaddr_in cadd;
        struct hostent *pURL = NULL;
        char myurl[BUFSIZE];
        char *pHost = 0;
        char host[BUFSIZE],GET[BUFSIZE];
        char request[BUFSIZE];
        static char text[BUFSIZE];
        int i,j;


        //分离主机中的主机地址和相对路径
        memset(myurl,0,BUFSIZE);
        memset(host,0,BUFSIZE);
        memset(GET,0,BUFSIZE);
        strcpy(myurl,url);
        for(pHost = myurl;*pHost != '/' && *pHost != '\0';++pHost);


        //获取相对路径保存到GET中
        if((int)(pHost-myurl) == strlen(myurl))
        {
                strcpy(GET,"/");//即url中没有给出相对路径，需要自己手动的在url尾
//部加上/
        }
        else
        {
                strcpy(GET,pHost);//地址段pHost到strlen(myurl)保存的是相对路径
        }

        //将主机信息保存到host中
        //此处将它置零，即它所指向的内容里面已经分离出了相对路径，剩下的为host信
//息(从myurl到pHost地址段存放的是HOST)
        *pHost = '\0';
        strcpy(host,myurl);
        //设置socket参数
        if(-1 == (cfd = socket(AF_INET,SOCK_STREAM,0)))
        {
                printf("create socket failed of client!\n");
                exit(-1);
        }

        pURL = gethostbyname(host);//将上面获得的主机信息通过域名解析函数获得域>名信息

        //设置IP地址结构
        bzero(&cadd,sizeof(struct sockaddr_in));
        cadd.sin_family = AF_INET;
        cadd.sin_addr.s_addr = *((unsigned long*)pURL->h_addr_list[0]);
        cadd.sin_port = htons(80);
        //向WEB服务器发送URL信息
        memset(request,0,BUFSIZE);
        strcat(request,"GET ");
        strcat(request,GET);
        strcat(request," HTTP/1.1\r\n");//至此为http请求行的信息
        strcat(request,"HOST: ");
        strcat(request,host);
        strcat(request,"\r\n");
        strcat(request,"Cache-Control: no-cache\r\n\r\n");
        //连接服务器


int cc;
        if(-1 == (cc = connect(cfd,(struct sockaddr*)&cadd,(socklen_t)sizeof(cadd))))
        {
                printf("connect failed of client!\n");
```

```
            exit(1);
        }
        printf("connect success!\n");

        //向服务器发送url请求的request
        int cs;
        if(-1 == (cs = send(cfd,request,strlen(request),0)))
        {
            printf("向服务器发送请求的request失败!\n");
            exit(1);
        }
        printf("发送成功,发送的字节数:%d\n",cs);

        //客户端接收服务器的返回信息
        memset(text,0,BUFSIZE);
        int cr;
        if(-1 == (cr = recv(cfd,text,BUFSIZE,0)))
        {
            printf("recieve failed!\n");
            exit(1);
        }
        else
        {
            printf("receive succecc!\n");
            printf("%s \n ",text);

            printf("*************************\n");
            char *test;
            test=rindex(text,'\r\n');
            char test1[100];
            strcpy(test1,test+1);
            printf("data=%s",test1);



        }
        close(cfd);
 }

int main(int argc,char* argv[])
 {
        if(argc<2)
        {
            //printf("用法:%c url网页网址\n",argv[0]);
           // exit(1);
        }
        geturl("huayinghealth.com/test.php");
        return 0;
 }
----------------------------------
3.#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <errno.h>

#define BUFSIZE 1024
#define DestIp "120.76.47.120"
#define DestPort 80
#define Req "GET /gsm/bpbg/upload_v1.php HTTP/1.1\r\nHost: api.huayinghealth.com\r\nConnection: Close\r\n\r\n"
#define ReqLen sizeof(Req)


int main(int argc, char *argv[]) {
            ssize_t i;
            int nRequestLen;

            char strResponse[BUFSIZE]={0};
            char strRequest[BUFSIZE]={0};


            int sockfd, numbytes;
            struct sockaddr_in dest_addr; /* connector's address information */

            if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
                perror("socket");
                exit(1);
            }

            dest_addr.sin_family = AF_INET; /* host byte order */
            dest_addr.sin_port = htons(DestPort); /* short, network byte order */
            dest_addr.sin_addr.s_addr = inet_addr(DestIp);

            /* Create and setup the connection */
            if (connect(sockfd, (struct sockaddr *)&dest_addr,sizeof(struct sockaddr)) == -1) {
                perror("connect");
                exit(1);
            }

            /* Send the request */
            strncpy(strRequest, Req,ReqLen);
            nRequestLen = ReqLen;
            if (write(sockfd,strRequest,nRequestLen) == -1) {
              perror("write");
              exit(1);
            }

            /* Read in the response */
            while(1) {
                i = read(sockfd,strResponse,BUFSIZE-1);
                if(0 >= i){
                    break;
                }
                strResponse[i]='\0';
                printf(strResponse);

            }

            printf("\n-------------------------------- \n");
            char *test;
            test=rindex(strResponse,'\r\n');
            printf(test);



            /* Close the connection */
            close(sockfd);
}
```

----

4. rindex函数,rindex(char *str,int c)
获取字符串str中c字符最后出现才位置

```
char *str1="abcdeffeaaa";
char *str2;
str2=rindex(str1,'e');
printf(str2)="eaaa";
```