gsoap:gsoap使用方法及心得

疯狂代码 http://www.crazycoder.cn/ ĵ:http:/www.crazycoder.cn/VC/Article12662.html

gSOAP是个夸平台用于开发Web Service服务端和客户端工具在Windows、Linux、MAC OS和UNIX下使用C和C语言编码集合了SSL功能

下载地址:http://sourceforge.net/projects/gsoap2

官方网站WebSite:http://genivia.com/Products/gsoap/index.html

对于Windows平台下开发客户端首先下载最新gsoap_win32_2.7.6c.zip包具体在以下地址:http://optusnet.dl.sourceforge.net/sourceforge/gsoap2/gsoap_win32_2.7.6c.zip

首先查看gsoapUser's Guide基本就能对gsoap有个全面了解通过阅读Sample里例子深入然后搜索网上其它些文章比如:
gSOAP简单多线程服务器 http://blog.chinaunix.net/u1/55091/showart_430965.html
纯c gSoap实现WebService http://hi.baidu.com/2sky2sea/blog/item/40ec5555680279c1b745ae9b.html

接下来我结合自己实战和理解讲讲VC用gsoap下编写webService和客户端有不对地方还请大家指正谢谢
我以网上出现实现个简单加法为例讲讲我在操作过程中遇到问题

服务器端
1.首先编写 add.h文件:

```
//gsoap ns service name: add
//gsoap ns service : http://localhost/add.wsdl
//gsoap ns service location: http://localhost
//gsoap ns service executable: add.cgi
//gsoap ns service encoding: encoded
//gsoap ns schema : urn:add

ns__add(  num1,  num2, * sum );

```

2.用gsoap/bin目录下soapcpp2.exe生成些文件可以把soapcpp2.exe拷贝到add.h目录下用cmd执行soapcpp2.exe add.h就可以在这个目录下会自动生成许多将来有用文件如

add.namap,soapH.h,soapC.cpp,soapClient.cpp,soapServer.cpp等文件soapcpp2.exe可以带参数执行具体执行soapcpp2.exe -h查看

3.新建个win32控制台工程加入wsock32.lib库将刚才生成那些文件添加到工程中然后编写webserver.cpp主:

```
# "add.h"
# "add.nsmap"

 ( argc, char* argv)
{

 m, s; /**//* master and slave s */
struct soap add_soap;
soap_init(&add_soap);
//soap__s(&add_soap, add_s);

 (argc < 2)
{
prf("usage: %s <server_port> \n", argv[0]);
exit(1);
}

{
m = soap_bind(&add_soap, NULL, atoi(argv[1]), 100);
 (m < 0)
{
soap_pr_fault(&add_soap, stderr);
exit(-1);
}

fprf(stderr, "Socket connection successful: master  = %d\n", m);
for ( ; ; )
{
s = soap_accept(&add_soap);
 (s < 0)
{
soap_pr_fault(&add_soap, stderr);
exit(-1);
}
fprf(stderr, "Socket connection successful: slave  = %d\n", s);

soap_serve(&add_soap);//该句介绍说明该server服务
soap_end(&add_soap);
}
}
 0;
}
//server端实现和add.h中声明相同但是多了个当前soap连接参数
 ns__add(struct soap *add_soap,  num1,  num2,  *sum)
{
*sum = num1 + num2;
 0;
}
```

4. 编译这个会提示将gsoap_win32目录下stdsoap2.cppstdsoap2.h文件加入工程重新编译如果还有可能是你将add.h生成文件添加入工程出错原因实际上在编写server时无须带Client那些文件还有带Lib文件也无须添加到工程中再重新编译应该就没有问题了启动4567端口在ie中输入localhost:4567,如果显示xml页面介绍说明已经启动

 2 对应客户端

1客户端代码如下:

```
# <stdio.h>
# <stdlib.h>
# "soapH.h"
# "add.nsmap"


 add(const char* server,  num1,  num2,  *sum);

 ( argc, char **argv)
{
 result = -1;
char* server="http://localhost:4567";
 num1 = 0;
 num2 = 0;
 sum = 0;
( argc < 3 )
{
prf("usage: %s num1 num2 \n", argv[0]);
exit(0);
}

num1 = atoi(argv[1]);
num2 = atoi(argv[2]);

result = add(server, num1, num2, &sum);
 (result != 0)
{
prf("soap err,errcode = %d\n", result);
}

{
prf("%d+%d=%d\n", num1, num2, sum );
}
 0;
}

 add( const char* server,  num1,  num2,  *sum )
{
struct soap add_soap;
 result = 0;
soap_init(&add_soap);
// soap__s(&add_soap, add_s);

//该是客户端主要后面几个参数和add.h中声明样前面多了3个参数名是接口名ns__add前面加上soap_call_
soap_call_ns__add( &add_soap, server, "", num1, num2, sum );
(add_soap.error)
```

```
{
prf("soap error:%d,%s,%s\n", add_soap.error, *soap_faultcode(&add_soap), *soap_fault(&add_soap) );
result = add_soap.error;
}
soap_end(&add_soap);
soap_done(&add_soap);
 result;
}
```

2.客户端既可以新建个新win32控制台将刚才生成nsmap,soapH.h,soapClient.h等文件加入工程编译既可我是直接在原先工程中加入客户端代码将webserver.cpp文件移除并且将soapServer.cpp等server端需要文件移除将soapClient.cpp等client端需要cpp添加到工程编译既可

3.启动serverF5客户端经测试正常

 3 遇到问题

1.server端可以编译成CGI方式执行而并不是绑定到某个端口这种方式我没有实战

```
 (argc < 2) // no args: assume this is a CGI application
{
soap_serve(&soap); // serve request,
_disibledevent=>/**//////////////////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////////////////////
void * process_queue(void *); //线程入口
 enqueue(SOAP_SOCKET); //入队列
SOAP_SOCKET dequeue(void); //出队列

/**//////////////////////////////////////////////////////////////////////////
//线程入口
void * process_queue(void * soap)
{
struct soap * tsoap = (struct soap *)soap;
for(;;)
{
tsoap-> = dequeue;
```

```
 (!soap_valid_(tsoap->))
{
;
}
soap_serve(tsoap);
soap_destroy(tsoap);
soap_end(tsoap);
}
 NULL;
}

//入队列操作
 enqueue(SOAP_SOCKET sock)
{
 status = SOAP_OK;
 next;
pthread_mutex_lock(&queue_cs);
next = tail +1;
 (next >= MAX_QUEUE)
next = 0;
 (next  head)
status = SOAP_EOM;

{
queue[tail] =sock;
tail = next;
}
pthread_cond_signal(&queue_cv);
pthread_mutex_unlock(&queue_cs);
 status;
}

//出队列操作
SOAP_SOCKET dequeue
{
SOAP_SOCKET sock;
```

```
pthread_mutex_lock(&queue_cs);
while (head  tail )
{
pthread_cond_wait(&queue_cv,&queue_cs);
}
sock = queue[head];
 (head >= MAX_QUEUE)
{
head =0;
}
pthread_mutex_unlock(&queue_cs);
 sock;
}


/**///////////////////////////具体服务思路方法/////////////////////////////////////////
//加法实现
 ns__add(struct soap *soap, double a, double b, double *result)
{
*result = a + b;
 SOAP_OK;
}
//减法实现
 ns__sub(struct soap *soap, double a, double b, double *result)
{
*result = a - b;
 SOAP_OK;
}
//乘法实现
 ns__mul(struct soap *soap, double a, double b, double *result)
{
*result = a * b;
 SOAP_OK;
}
//除法实现
 ns__div(struct soap *soap, double a, double b, double *result)
```

```
{
 (b)
*result = a / b;

{
char *s = (char*)soap_malloc(soap, 1024);
sprf(s, "Can't">http://tempuri.org/">Can't divide %f by %f", a, b);
 soap_sender_fault(soap, "Division by zero", s);
}
 SOAP_OK;
}
//乘方实现
 ns__pow(struct soap *soap, double a, double b, double *result)
{
*result = pow(a, b);
 (soap_errno  EDOM) /**//* soap_errno 和errorno类似, 但是和widnows兼容 */
{
char *s = (char*)soap_malloc(soap, 1024);
sprf(s, "Can't take the power of %f to %f", a, b);
sprf(s, "Can't">http://tempuri.org/">Can't take power of %f to %f", a, b);
 soap_sender_fault(soap, "Power function do error", s);
}
 SOAP_OK;
}

/**////////////////////////////////////////////////////////////////////////////////////////////////
//主
 ( argc,char ** argv)
{
struct soap ServerSoap;
//话运行时环境
soap_init(&ServerSoap);
//如果没有参数当作CGI处理
 (argc <2)
{
//CGI 风格服务请求单线程
```

```
soap_serve(&ServerSoap);
//清除序列化类例子
soap_destroy(&ServerSoap);
//清除序列化数据
soap_end(&ServerSoap);
}
{
struct soap * soap_thr[MAX_THR];
pthread_t tid[MAX_THR];
 i,port = atoi(argv[1]);
SOAP_SOCKET m,s;
//锁和条件变量化
pthread_mutex_init(&queue_cs,NULL);
pthread_cond_init(&queue_cv,NULL);
//绑定服务端口
m = soap_bind(&ServerSoap,NULL,port,BACKLOG);
//循环直至服务套接字合法
while (!soap_valid_(m))
{
fprf(stderr,"Bind port error! ");
m = soap_bind(&ServerSoap,NULL,port,BACKLOG);
}
fprf(stderr," connection successful %d ",m);

//生成服务线程
for(i = 0; i <MAX_THR; i)

{
soap_thr[i] = soap_copy(&ServerSoap);
fprf(stderr,"Starting thread %d ",i);
pthread_create(&tid[i],NULL,(void*(*)(void*))process_queue,(void*)soap_thr[i]);
}

for(;;)
{
//接受客户端连接
```

```
s = soap_accept(&ServerSoap);
 (!soap_valid_(s))
{
 (ServerSoap.errnum)
{
soap_pr_fault(&ServerSoap,stderr);
continue;
}
{
fprf(stderr,"Server timed out ");
;
}
}
//客户端IP地址
fprf(stderr,"Accepted connection from IP= %d.%d.%d.%d  = %d ",
((ServerSoap.ip)>>24)&&0xFF,((ServerSoap.ip)>>16)&0xFF,((ServerSoap.ip)>>8)&0xFF,(ServerSoap.ip
)&0xFF,(ServerSoap.));
//请求套接字进入队列如果队列已满则循环等待
while(enqueue(s)  SOAP_EOM)
Sleep(1000);
}
//服务结束后清理工作
for(i = 0; i < MAX_THR; i)
{
while (enqueue(SOAP_INVALID_SOCKET)  SOAP_EOM)
{
Sleep(1000);
}
}
for(i=0; i< MAX_THR; i)
{
fprf(stderr,"Waiting for thread %d to terminate ..",i);
pthread_join(tid[i],NULL);
fprf(stderr,"terminated ");
soap_done(soap_thr[i]);
free(soap_thr[i]);
```

```
}
pthread_mutex_destroy(&queue_cs);
pthread_cond_destroy(&queue_cv);
}
//分离运行时环境
soap_done(&ServerSoap);
 0;
}
```

2009-2-12 3:46:20

疯狂代码 http://www.crazycoder.cn/