

图16-5汇总了一次远程过程调用中通常发生的步骤。编了号的步骤是按顺序执行的。

- (0) 服务器启动,它向所在主机上的端口映射器(port mapper)注册自身。然后客户启动,它调用clnt\_create,该函数则与服务器主机上的端口映射器联系,以找到服务器的临时端口。clnt\_create函数还建立一个与服务器的TCP连接(因为我们在图16-2中指定的协议为TCP)。我们在本图中没有展示这些步骤,留待16.3节中详细地讲述。
- (1) 客户调用一个称为客户程序存根(client stub)的本地过程。在图16-2中,该过程名为squareproc\_1,而含有这个客户程序存根的文件是由rpcgen产生的,名为square\_clnt.c。对于客户来说,客户程序存根看起来像是它想要调用的真正的服务器过程。存根的目的在于把有待传递给远程过程的参数打成包,可能的话把它们转换成某种标准格式,然后构造一个或多个网络消息。把客户提供的参数打包成一个网络消息的过程称为集结(marshaling)。客户程序的各个例程和存根通常调用RPC运行时函数库中的函数(例如我们早先的例子中的clnt\_create)。在Solaris下链接时,这些运行时库函数是从\_lnsl函数库中加载的,而BSD/OS下它们是在标准C函数库中。

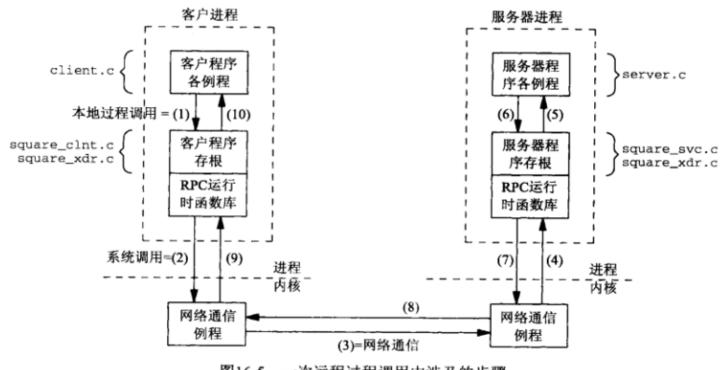
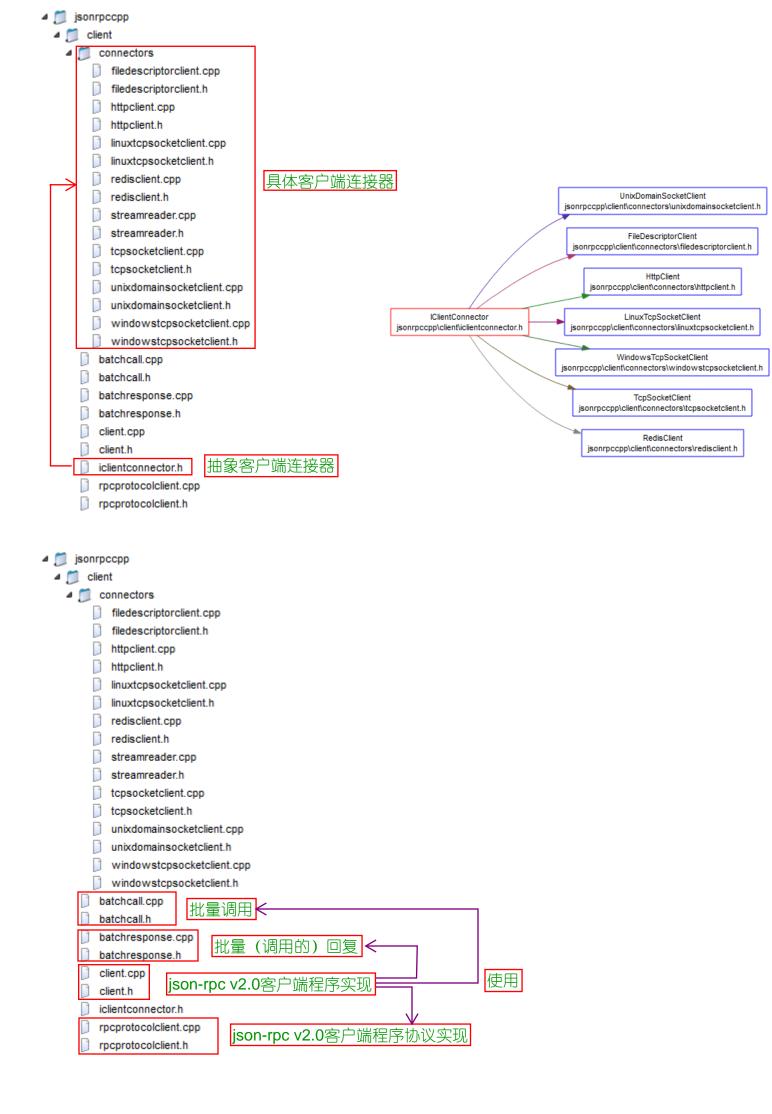
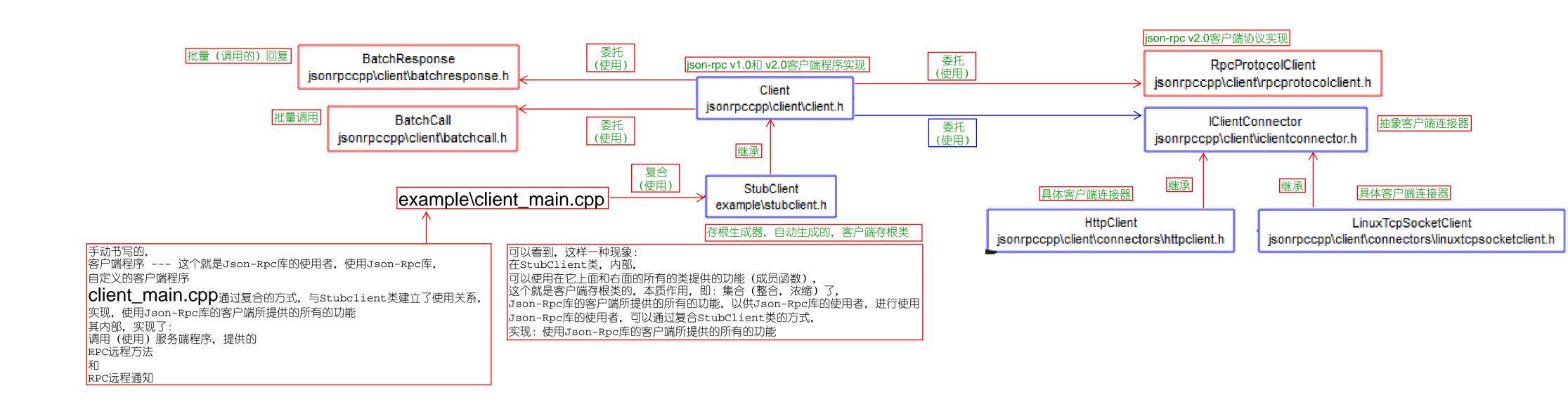


图16-5 一次远程过程调用中涉及的步骤

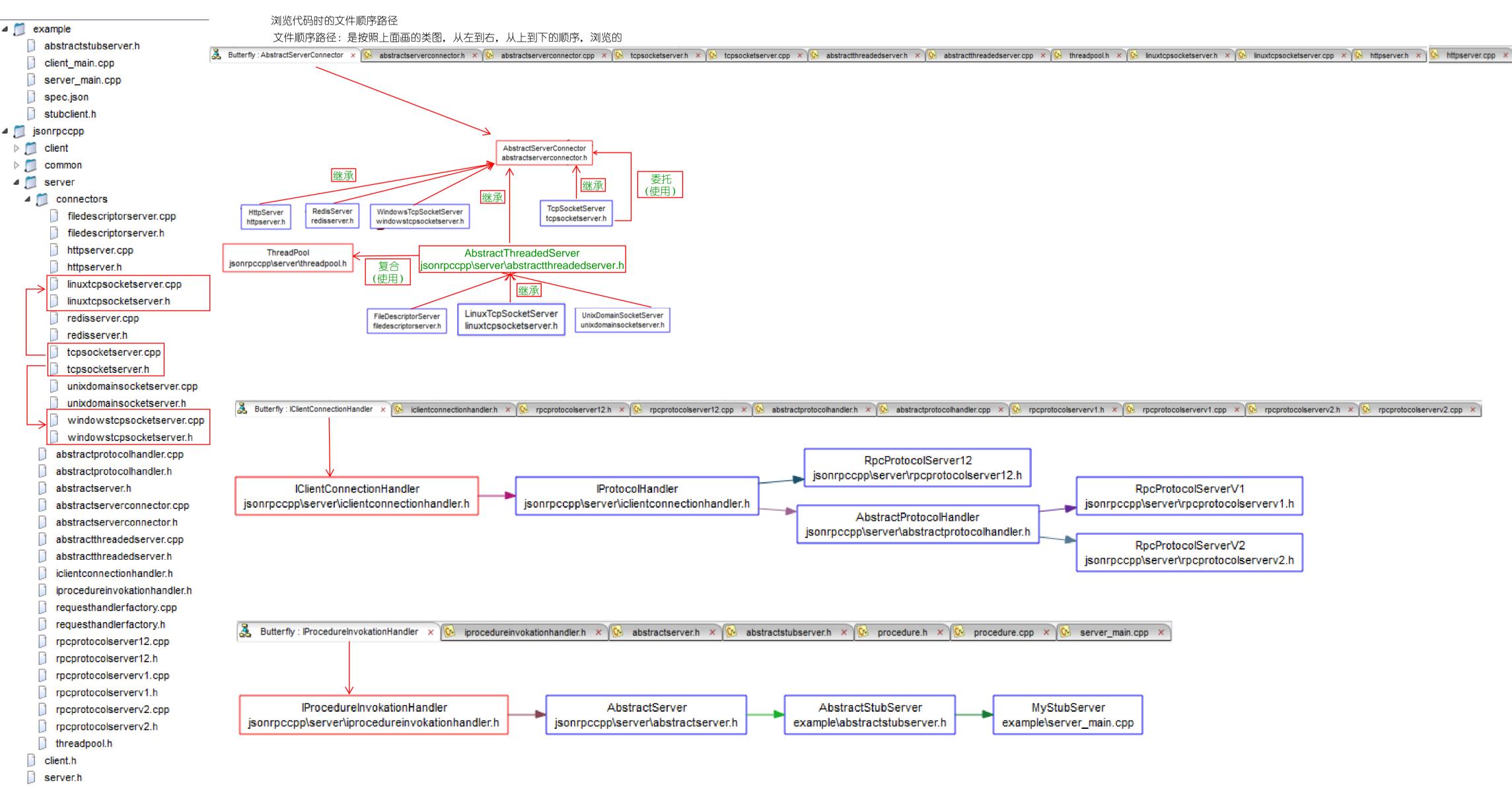
- (2) 这些网络消息由客户程序存根发送给远程系统。这通常需要一次陷入本地内核的系统调用(例如write或sendto)。
  - (3) 这些网络消息传送到远程系统。这一步所用的典型网络协议为TCP或UDP。
- (4) 一个服务器程序存根(sever stub)过程一直在远程系统上等待客户的请求。它从这些网络消息中解散(unmarshaling)出参数。
- (5) 服务器程序存根执行一个本地过程调用以激活真正的服务器函数(图16-3中我们的squareproc\_1\_svc过程),传递给该函数的参数是它从来自客户的网络消息中解散出来的。
  - (6) 当服务器过程完成时,它向服务器程序存根返回其返回值。
- (7) 服务器程序存根在必要时对返回值进行转换,然后把它们集结到一个或多个网络消息中,以便发送回客户。
  - (8) 这些消息通过网络传送回客户。
  - (9) 客户程序存根从本地内核中读出这些网络消息(例如read或recvfrom)。
- (10) 对返回值进行可能的转换后,客户程序存根最终返回客户函数。这一步看起来像是一个普通的过程返回客户。





浏览代码时的文件顺序路径

Client.h × 🕟 client.cpp × 🕟 rpcprotocolclient.h × 🕟 rpcprotocolclient.cpp × 🔯 batchresponse.h × 🔯 batchresponse.cpp × 🔯 batchresponse.cpp × 🔯 httpclient.cpp × 🔯 linuxtcpsocketclient.h × 🔯 linuxtcpsocketclient.cpp × 🔯 stubclient.h ×





我们在本图中没有展示这些步骤,留待16.3节中详细地讲述。

(1) 客户调用一个称为客户程序存根(client stub)的本地过程。在图16-2中,该过程名为squareproc\_1,而含有这个客户程序存根的文件是由rpcgen产生的,名为square\_clnt.c。对于客户来说,客户程序存根看起来像是它想要调用的真正的服务器过程。存根的目的在于把有待传递给远程过程的参数打成包,可能的话把它们转换成某种标准格式,然后构造一个或多个网络消息。把客户提供的参数打包成一个网络消息的过程称为集结(marshaling)。客户程序的各个例程和存根通常调用RPC运行时函数库中的函数(例如我们早先的例子中的clnt\_create)。在Solaris下链接时,这些运行时库函数是从\_lnsl函数库中加载的,而BSD/OS下它们是在标准C函数库中。

图16-5汇总了一次远程过程调用中通常发生的步骤。编了号的步骤是按顺序执行的。

它调用clnt\_create,该函数则与服务器主机上的端口映射器联系,以找到服务器的临时端口。

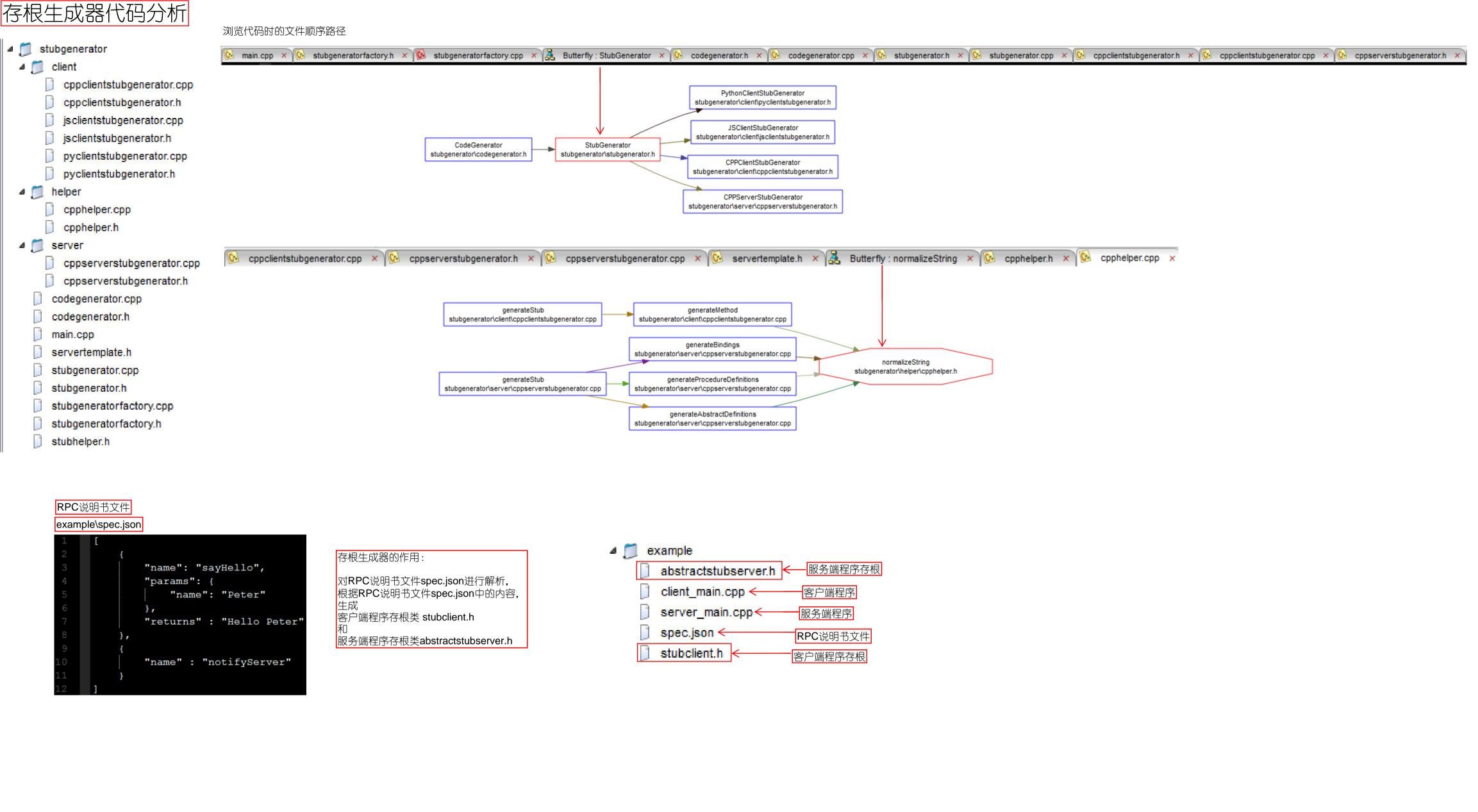
clnt\_create函数还建立一个与服务器的TCP连接(因为我们在图16-2中指定的协议为TCP)。

(0) 服务器启动,它向所在主机上的端口映射器(port mapper)注册自身。然后客户启动,

服务器进程 ┌---r----client. 序各例程 \_\_\_\_ 服务器程 square\_svc.c 序存根 square\_xdr.c square\_xdr.c \_\_\_\_ 时函数库 系统调用=(2) ----进程 -图16-5 一次远程过程调用中涉及的步骤

- (2) 这些网络消息由客户程序存根发送给远程系统。这通常需要一次陷入本地内核的系统调用(例如write或sendto)。
  (3) 这些网络消息传送到远程系统。这一步所用的典型网络协议为TCP或UDP。
- (4)一个服务器程序存根(sever stub)过程一直在远程系统上等待客户的请求。它从这些网络消息中解散(unmarshaling)出参数。 (5)服务器程序存根执行一个本地过程调用以激活真正的服务器函数(图16-3中我们的squareproc\_1\_svc过程),传递给该函数的参数是它从来自客户的网络消息中解散出来的。
- (6) 当服务器过程完成时,它向服务器程序存根返回其返回值。
  (7) 服务器程序存根在必要时对返回值进行转换,然后把它们集结到一个或多个网络消息中,以便发送回客户。
- r,以使及医固各户。 (8)这些消息通过网络传送回客户。 (9)客户程序存根从本地内核中读出这些网络消息(例如read或recvfrom)。
- (9) 客户程序存根从本地内核中读出这些网络消息(例如read或recvfrom)。 (10) 对返回值进行可能的转换后,客户程序存根最终返回客户函数。这一步看起来像是一个普通的过程返回客户。





common代码分析

