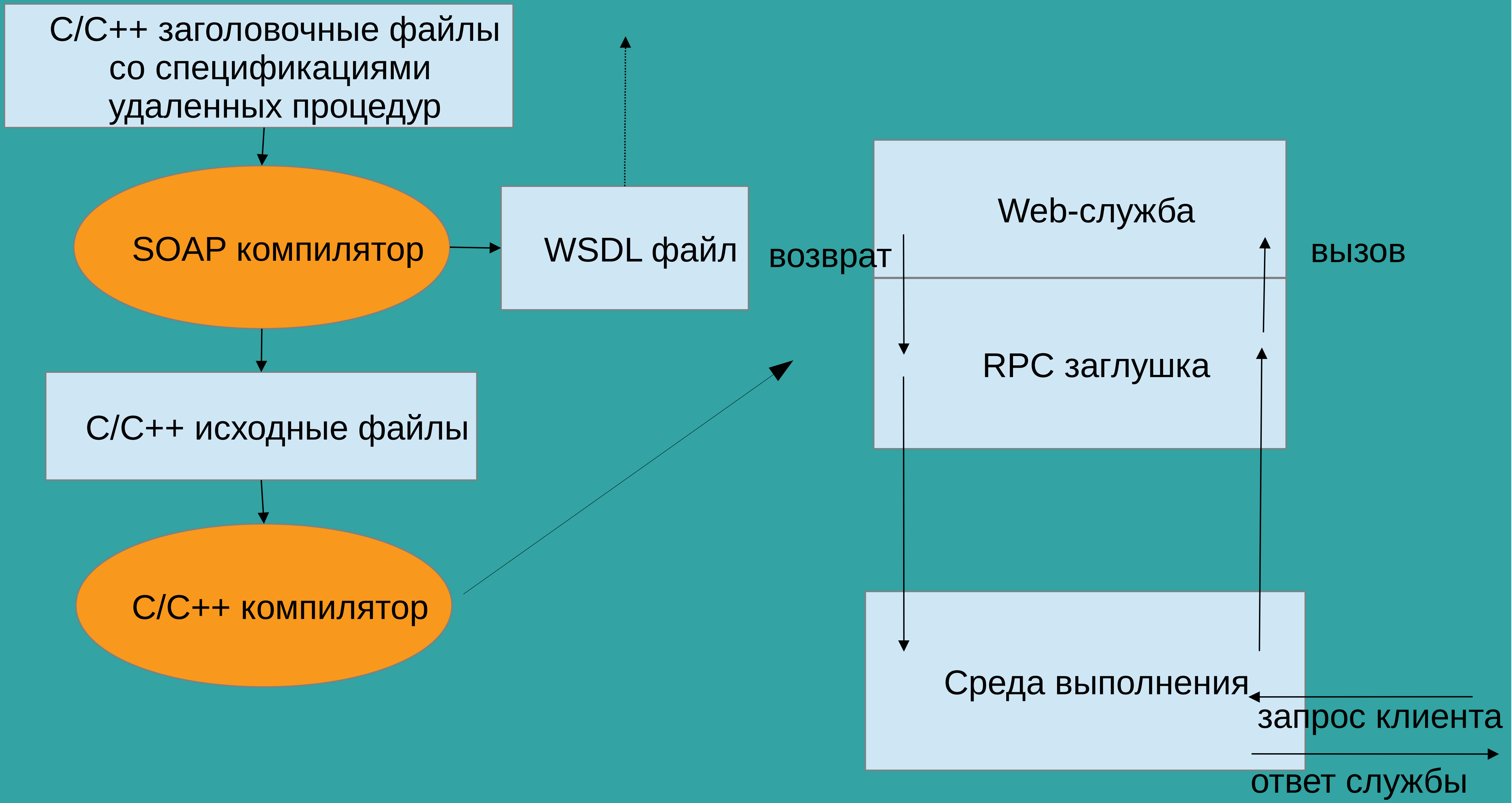
Лекция 7
XML Web services: реализация gSOAP

- Приложение идентифицируемое строкой URI (*Uniform Resource Identifier*).
- Интерфейсы приложения и его связи описываются документами XML (*eXtensible Markup Language*).
- Приложение взаимодействует с другими приложениями посредством XML-сообщений передаваемых по сети.

Загрузка gSOAP toolkit и документация - http://www.cs.fsu.edu/~engelen/soap.html

Разработка и функционирование сервера web-службы



C/C++ заголовочные файлы со спецификациями удаленных процедур

SOAP компилятор

WSDL файл

C/C++ исходные файлы

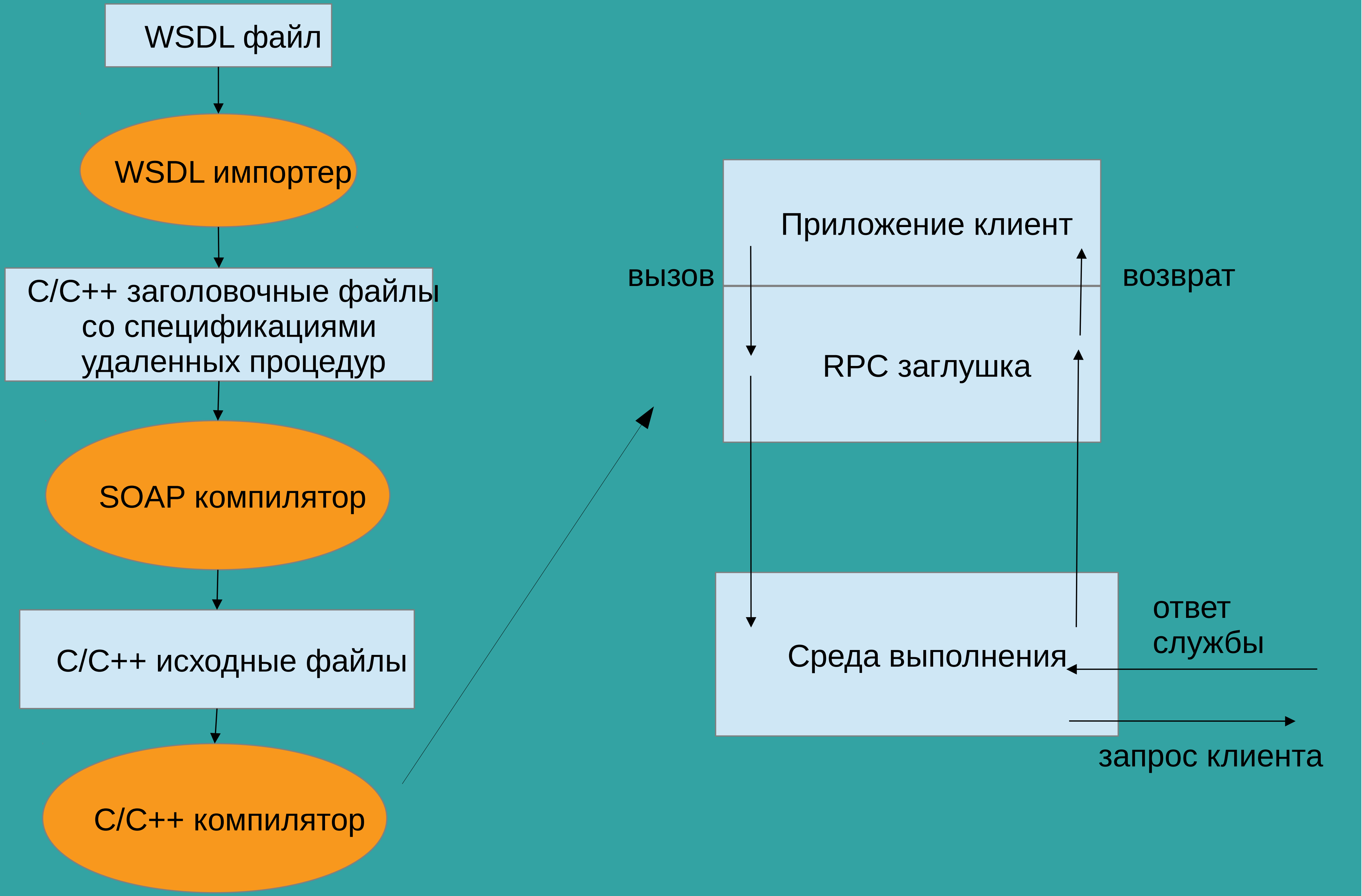
C/C++ компилятор

Web-служба

RPC заглушка

Среда выполнения

Разработка и функционирование клиента web-службы средствами gSOAP



WSDL файл

WSDL импортер

C/C++ заголовочные файлы со спецификациями удаленных процедур

SOAP компилятор

C/C++ исходные файлы

C/C++ компилятор

Приложение клиент

вызов

возврат

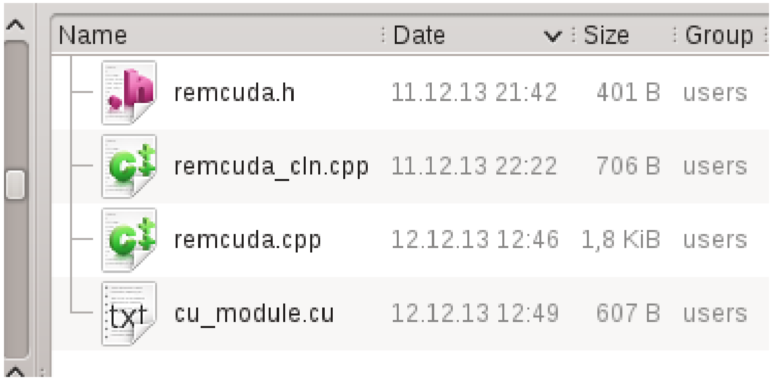
RPC заглушка

Среда выполнения

ответ службы

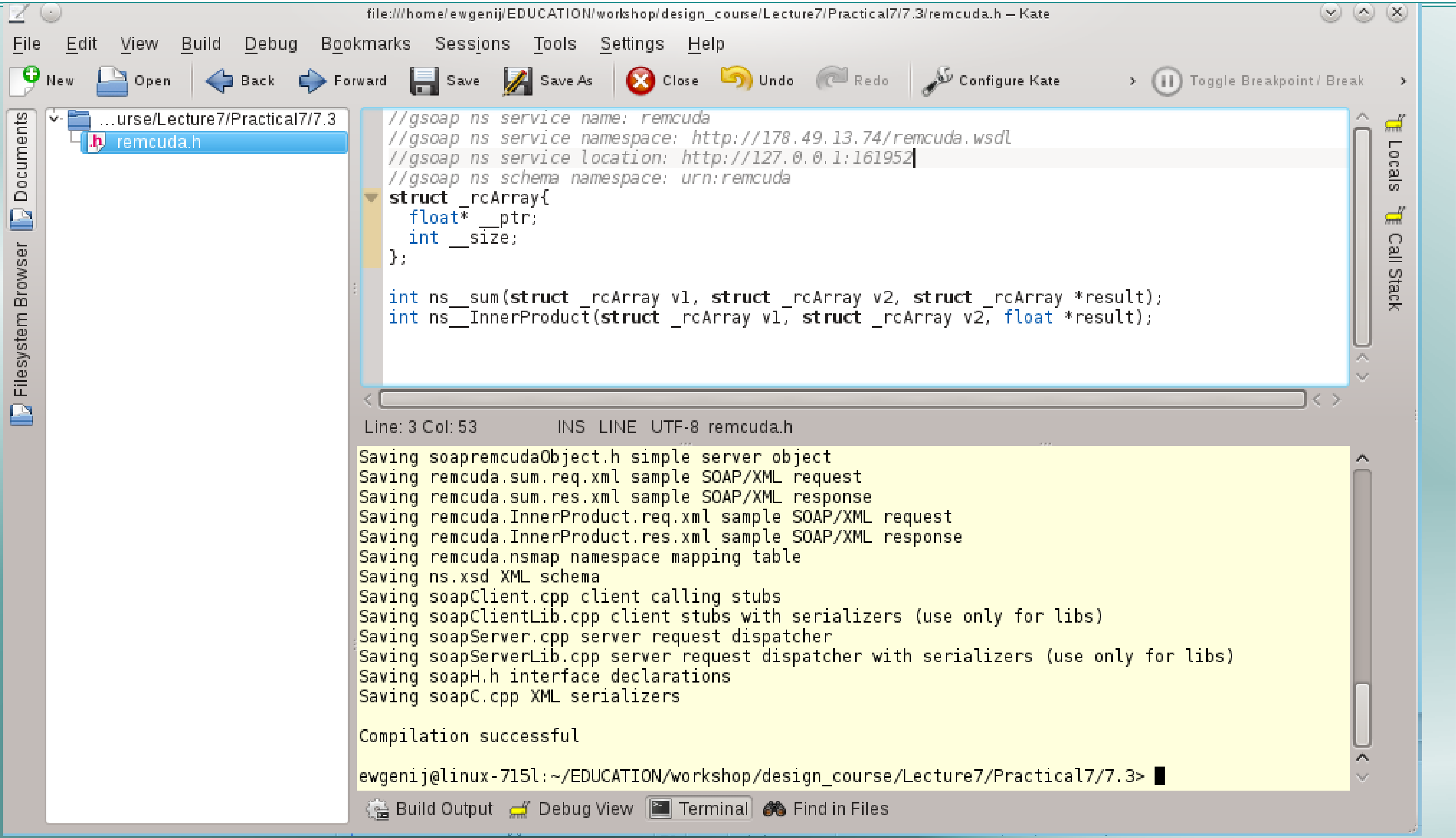
запрос клиента

“Source Code”



Name	Date	Size	Group
remcuda.h	11.12.13 21:42	401 B	users
remcuda_cln.cpp	11.12.13 22:22	706 B	users
remcuda.cpp	12.12.13 12:46	1,8 KiB	users
cu_module.cu	12.12.13 12:49	607 B	users

SOAP компиляция: *soapcpp2 remcuda.h*



file:///home/ewgenij/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3/remcuda.h – Kate
File Edit View Build Debug Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo Configure Kate Toggle Breakpoint / Break
Documents
Filesystem Browser
...urse/Lecture7/Practical7/7.3
remcuda.h
Locals
Call Stack
Line: 3 Col: 53 INS LINE UTF-8 remcuda.h
Build Output
Debug View
Terminal
Find in Files

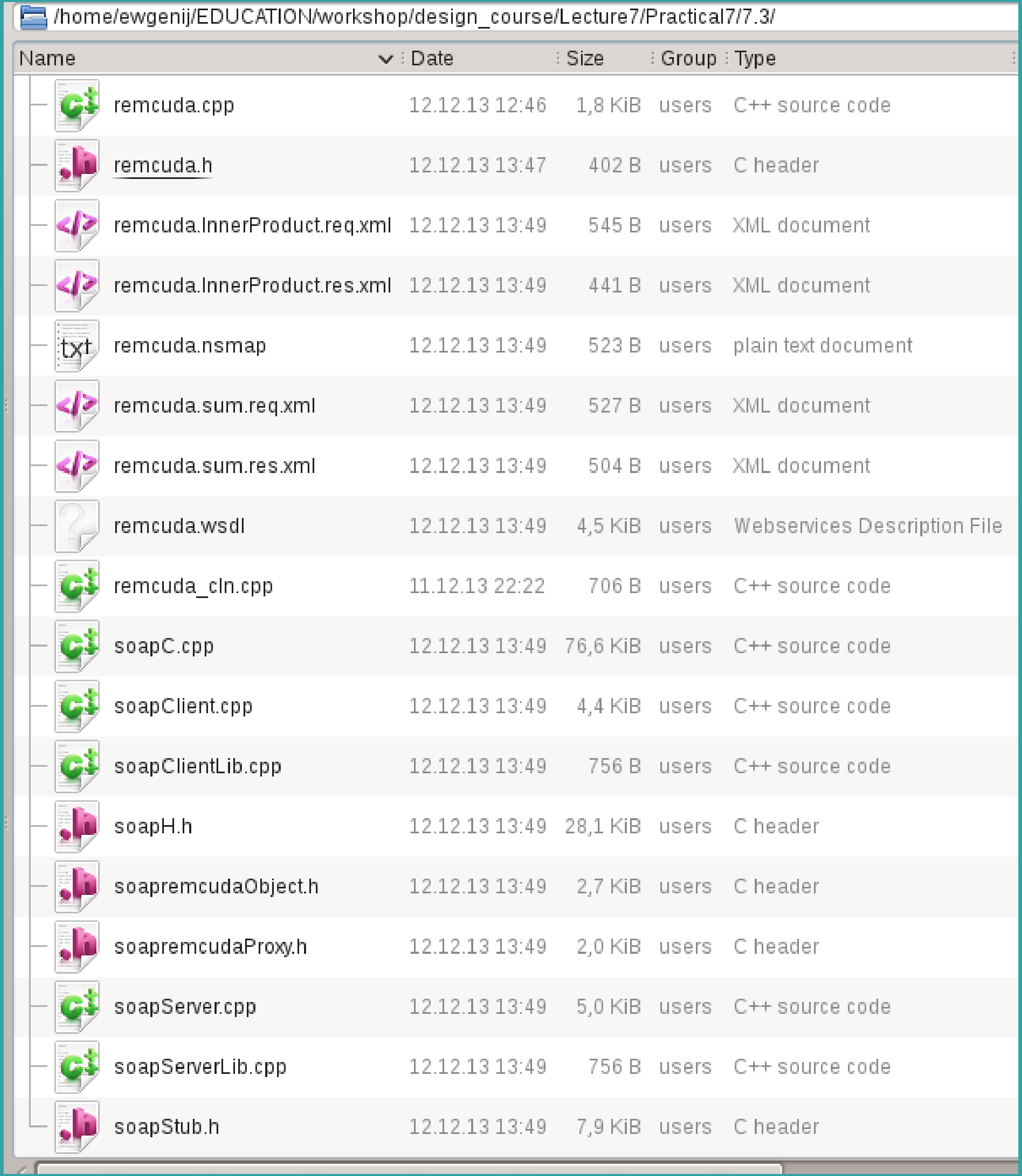
```
//gsoap ns service name: remcuda
//gsoap ns service namespace: http://178.49.13.74/remcuda.wsdl
//gsoap ns service location: http://127.0.0.1:161952
//gsoap ns schema namespace: urn:remcuda
struct _rcArray{
  float* __ptr;
  int __size;
};

int ns__sum(struct _rcArray v1, struct _rcArray v2, struct _rcArray *result);
int ns__InnerProduct(struct _rcArray v1, struct _rcArray v2, float *result);
```

```
Saving soapremcudaObject.h simple server object
Saving remcuda.sum.req.xml sample SOAP/XML request
Saving remcuda.sum.res.xml sample SOAP/XML response
Saving remcuda.InnerProduct.req.xml sample SOAP/XML request
Saving remcuda.InnerProduct.res.xml sample SOAP/XML response
Saving remcuda.nsmap namespace mapping table
Saving ns.xsd XML schema
Saving soapClient.cpp client calling stubs
Saving soapClientLib.cpp client stubs with serializers (use only for libs)
Saving soapServer.cpp server request dispatcher
Saving soapServerLib.cpp server request dispatcher with serializers (use only for libs)
Saving soapH.h interface declarations
Saving soapC.cpp XML serializers

Compilation successful

ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3>
```



/home/ewgenij/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3/

Name	Date	Size	Group	Type
remcuda.cpp	12.12.13 12:46	1,8 KiB	users	C++ source code
remcuda.h	12.12.13 13:47	402 B	users	C header
remcuda.InnerProduct.req.xml	12.12.13 13:49	545 B	users	XML document
remcuda.InnerProduct.res.xml	12.12.13 13:49	441 B	users	XML document
remcuda.nsmap	12.12.13 13:49	523 B	users	plain text document
remcuda.sum.req.xml	12.12.13 13:49	527 B	users	XML document
remcuda.sum.res.xml	12.12.13 13:49	504 B	users	XML document
remcuda.wsdl	12.12.13 13:49	4,5 KiB	users	Webservices Description File
remcuda_cln.cpp	11.12.13 22:22	706 B	users	C++ source code
soapC.cpp	12.12.13 13:49	76,6 KiB	users	C++ source code
soapClient.cpp	12.12.13 13:49	4,4 KiB	users	C++ source code
soapClientLib.cpp	12.12.13 13:49	756 B	users	C++ source code
soapH.h	12.12.13 13:49	28,1 KiB	users	C header
soapremcudaObject.h	12.12.13 13:49	2,7 KiB	users	C header
soapremcudaProxy.h	12.12.13 13:49	2,0 KiB	users	C header
soapServer.cpp	12.12.13 13:49	5,0 KiB	users	C++ source code
soapServerLib.cpp	12.12.13 13:49	756 B	users	C++ source code
soapStub.h	12.12.13 13:49	7,9 KiB	users	C header

Web sevice как cgi-приложение и stand-alone сервер (*remcuda.cpp*)

```
#include "soapH.h"
#include "remcuda.nsmap"
#include <stdio.h>

void Sum_vec(float* v1, float* v2, float* w, int N);

int main(int argc, char* argv[])  {
  int m, s;
  struct soap *soap = soap_new();

  if (argc < 2)
    soap_serve(soap); /*  CGI  */
  else{
    m = soap_bind(soap, NULL, atoi(argv[1]), 100);
    if (m < 0){
       soap_print_fault(soap, stderr);
       exit(-1);
    }
    fprintf(stderr, "Socket connection successful: master socket = %d\n", m);
    for (;;){
      s = soap_accept(soap);
      fprintf(stderr, "Socket connection successful: slave socket = %d\n", s);
      if (s < 0){
        soap_print_fault(soap, stderr);
        exit(1);
      }
      soap_serve(soap);
      soap_end(soap);
    }
  }
  soap_done(soap);
  free(soap);
  return 0;
}
```

Контекст времени выполнения

адрес

Диспетчер запросов

порт

Реализация функций web-службы

```
int ns__sum(struct soap *soap, struct _rcArray v1, struct _rcArray v2, struct _rcArray *result){
  int n=v1.__size;
  result->__size=n;
  result->__ptr=new float[n];
  Sum_vec(v1.__ptr, v2.__ptr, result->__ptr, v1.__size);
  /*
  int i;
  int n=v1.__size;

  result->__size=n;
  result->__ptr=new float[n];
  for(i=0;i<n;i++){
    (result->__ptr)[i] = v1.__ptr[i] + v2.__ptr[i];
   }
  */
  return SOAP_OK;
}
int ns__InnerProduct(struct soap *soap, struct _rcArray v1, struct _rcArray v2, float *result){
  int i;
  int n=v1.__size;
  (*result)=0.0;
  for(i=0;i<n;i++)
    (*result)+= v1.__ptr[i]*v2.__ptr[i];

  return SOAP_OK;
}
```

Последний параметр всегда указатель,
и служит для возврата значений

Модуль на CUDA C (*cu_module.cu*)

```
__global__  void gSum_vec(float* v1, float* v2, int N){
        int i=threadIdx.x+blockIdx.x*blockDim.x;
        v1[i]+=v2[i]+1.0;
}

void Sum_vec(float* v1, float* v2, float *w, int N){
  float *u1,*u2;

  cudaMalloc((void **) &u1, N*sizeof(float));
  cudaMalloc((void **) &u2, N*sizeof(float));

  cudaMemcpy(u1, v1, N*sizeof(float), cudaMemcpyHostToDevice);
  cudaMemcpy(u2, v2, N*sizeof(float), cudaMemcpyHostToDevice);

  gSum_vec<<<dim3(N/512+((N%512)?1:0)),dim3(512)>>>(u1,u2,N);
  cudaDeviceSynchronize();

  cudaMemcpy(w, u1, N*sizeof(float), cudaMemcpyDeviceToHost);

  cudaFree(u1);
  cudaFree(u2);
  }
```

Приложение-клиент web sevice (*remcuda_cln.cpp*)

```
#include "soapremcudaProxy.h"
#include "remcuda.nsmap"
#include <iostream>

int main(int argc, char* argv[]){
  int n=atoi(argv[1]);
  int i;
  struct _rcArray V1, V2,result;
  remcuda c;

  V1.__ptr=new float[n];
  V2.__ptr=new float[n];

  for(i=0;i<n;i++){
    (V1.__ptr)[i]=(float)(i+1);
    (V2.__ptr)[i]=(float)(i+1);
  }

  V1.__size=n;
  V2.__size=n;

  if (c.ns__sum(V1, V2, &result) == SOAP_OK)
    for(i=0;i<n;i++)
        std::cout <<result.__ptr[i]<<std::endl;
  else
    soap_print_fault(c.soap, stderr);

  delete[] V1.__ptr;
  delete[] V2.__ptr;
  delete[] result.__ptr;

  return 0;
}
```

Генерируемое объявление класса remcuda

```
#ifndef soapremcudaProxy_H
#define soapremcudaProxy_H
#include "soapH.h"
class remcuda
{   public:
        /// Runtime engine context allocated in constructor
        struct soap *soap;
        /// Endpoint URL of service 'remcuda' (change as needed)
        const char *endpoint;
        /// Constructor allocates soap engine context, sets default endpoint URL, and sets namespace mapping table
        remcuda()
        { soap = soap_new(); endpoint = "http://127.0.0.1:161952"; if (soap && !soap->namespaces) { static const struct
Namespace namespaces[] =
{
        {"SOAP-ENV", "http://schemas.xmlsoap.org/soap/envelope/", "http://www.w3.org/*/soap-envelope", NULL},
        {"SOAP-ENC", "http://schemas.xmlsoap.org/soap/encoding/", "http://www.w3.org/*/soap-encoding", NULL},
        {"xsi", "http://www.w3.org/2001/XMLSchema-instance", "http://www.w3.org/*/XMLSchema-instance", NULL},
        {"xsd", "http://www.w3.org/2001/XMLSchema", "http://www.w3.org/*/XMLSchema", NULL},
        {"ns", "urn:remcuda", NULL, NULL},
        {NULL, NULL, NULL, NULL}
};
        soap->namespaces = namespaces; } };
        /// Destructor frees deserialized data and soap engine context
        virtual ~remcuda() { if (soap) { soap_destroy(soap); soap_end(soap); soap_free(soap); } };
        /// Invoke 'sum' of service 'remcuda' and return error code (or SOAP_OK)
        virtual int ns__sum(struct _rcArray v1, struct _rcArray v2, struct _rcArray *result) { return soap ?
soap_call_ns__sum(soap, endpoint, NULL, v1, v2, result) : SOAP_EOM; };
        /// Invoke 'InnerProduct' of service 'remcuda' and return error code (or SOAP_OK)
        virtual int ns__InnerProduct(struct _rcArray v1, struct _rcArray v2, float *result) { return soap ?
soap_call_ns__InnerProduct(soap, endpoint, NULL, v1, v2, result) : SOAP_EOM; };
};
#endif
```

Компиляция web-службы и клиента

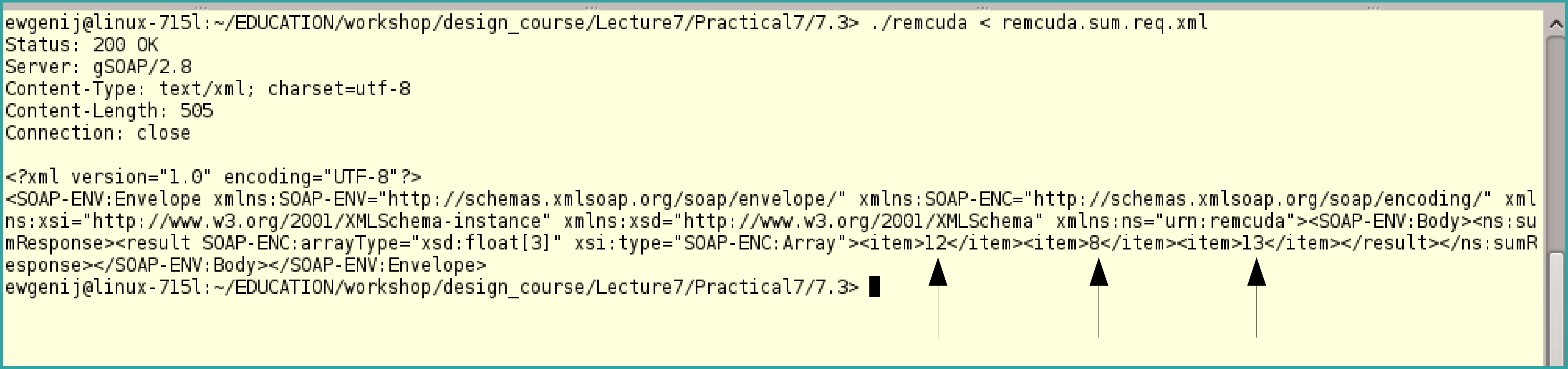
```
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> nvcc
-arch=sm_20 remcuda.cpp soapC.cpp soapServer.cpp cu_module.cu -lgsoap++ -o remcuda

ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> g++
remcuda_cln.cpp   soapC.cpp   soapClient.cpp    -lgsoap++   -o remcuda_cln
```

soapC.cpp, *soapServer.cpp* и *soapClient.cpp* содержат объявления soap-функций, структур и код заглушек, необходимых для маршалинга/демаршалинга и сериализации/десериализации. То есть для представления запросов в виде xml-строк с последующей упаковкой и передачей web-службе в виде потока байт и наоборот, возврата клиенту.

Тестирование web-службы. Файл *remcuda.sum.req.xml*

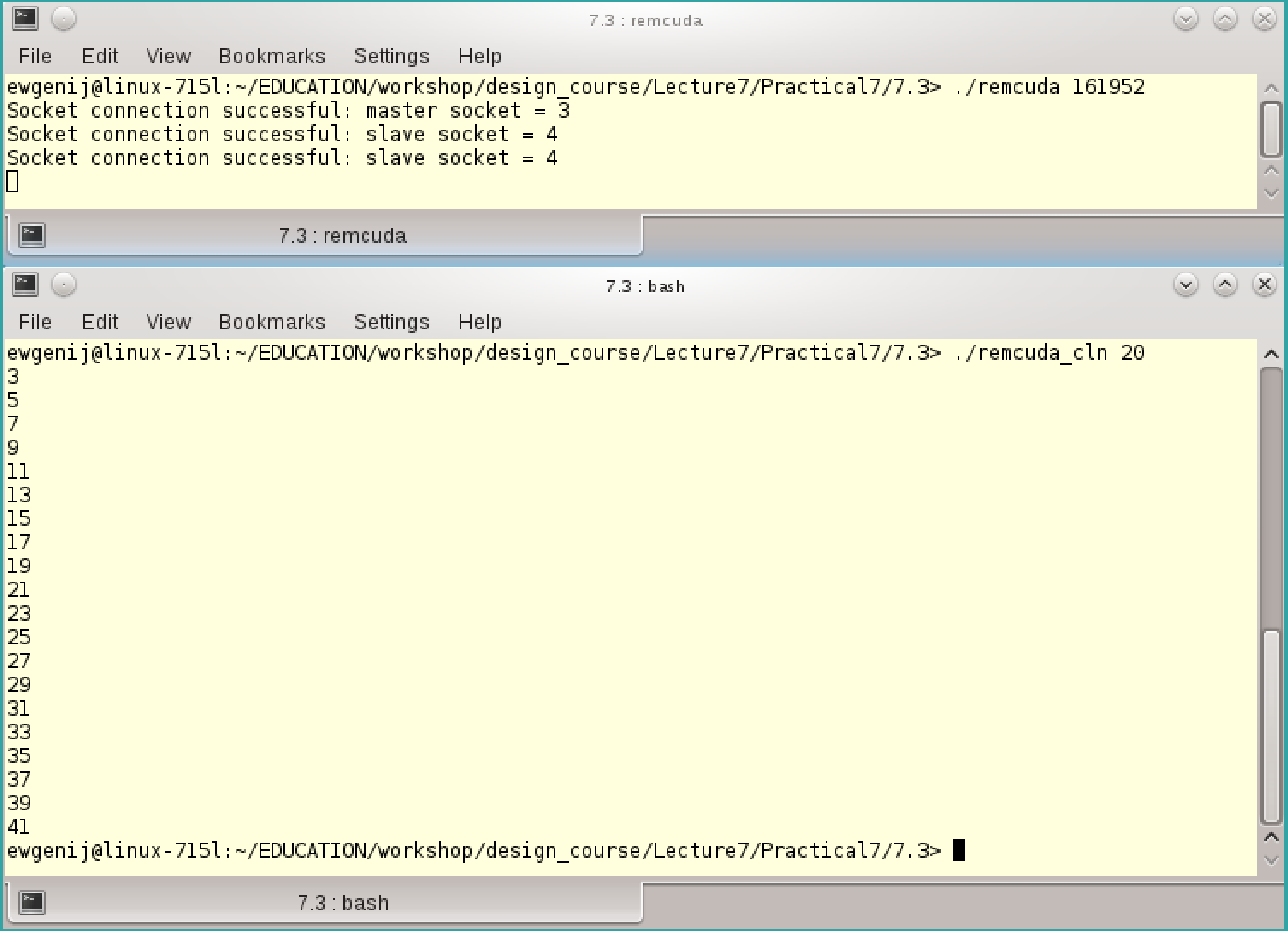
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:ns="urn:remcuda">
 <SOAP-ENV:Body>
  <ns:sum>
   <v1 SOAP-ENC:arrayType="xsd:float[3]">
    <item>2.0</item>
    <item>3.0</item>
    <item>7.0</item>
   </v1>
   <v2 SOAP-ENC:arrayType="xsd:float[3]">
    <item>9.0</item>
    <item>4.0</item>
    <item>5.0</item>
   </v2>
  </ns:sum>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> ./remcuda < remcuda.sum.req.xml
Status: 200 OK
Server: gSOAP/2.8
Content-Type: text/xml; charset=utf-8
Content-Length: 505
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xml
ns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns="urn:remcuda"><SOAP-ENV:Body><ns:su
mResponse><result SOAP-ENC:arrayType="xsd:float[3]" xsi:type="SOAP-ENC:Array"><item>12</item><item>8</item><item>13</item></result></ns:sumR
esponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> █
```

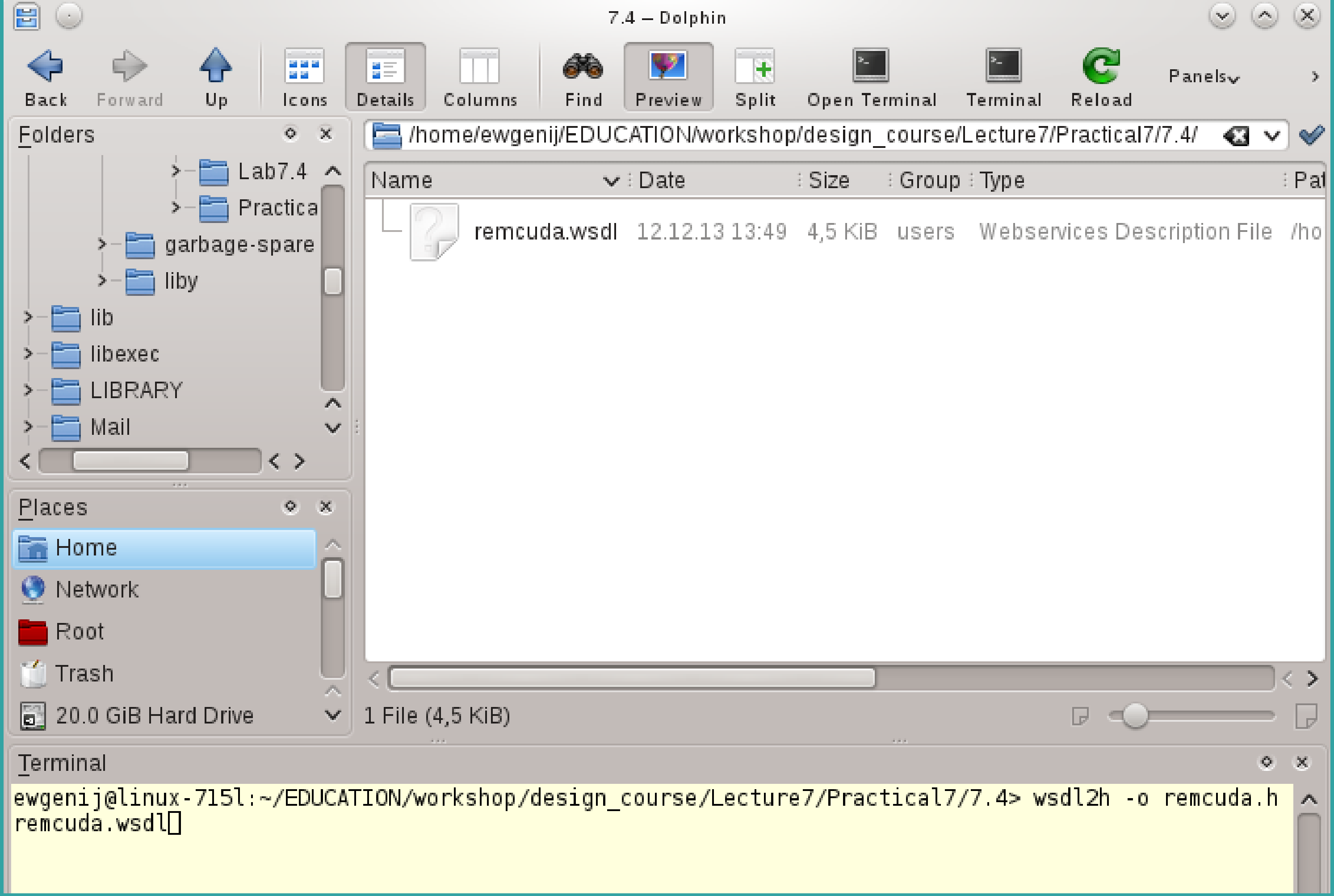
Тестирование web-службы. Stand-alone сервер.


7.3 : remcuda
File Edit View Bookmarks Settings Help
7.3 : remcuda
7.3 : bash
File Edit View Bookmarks Settings Help
7.3 : bash

```
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> ./remcuda 161952
Socket connection successful: master socket = 3
Socket connection successful: slave socket = 4
Socket connection successful: slave socket = 4
```

```
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3> ./remcuda_cln 20
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.3>
```

ДОПОЛНЕНИЕ: WSDL-импортер (*wsdl2h*)

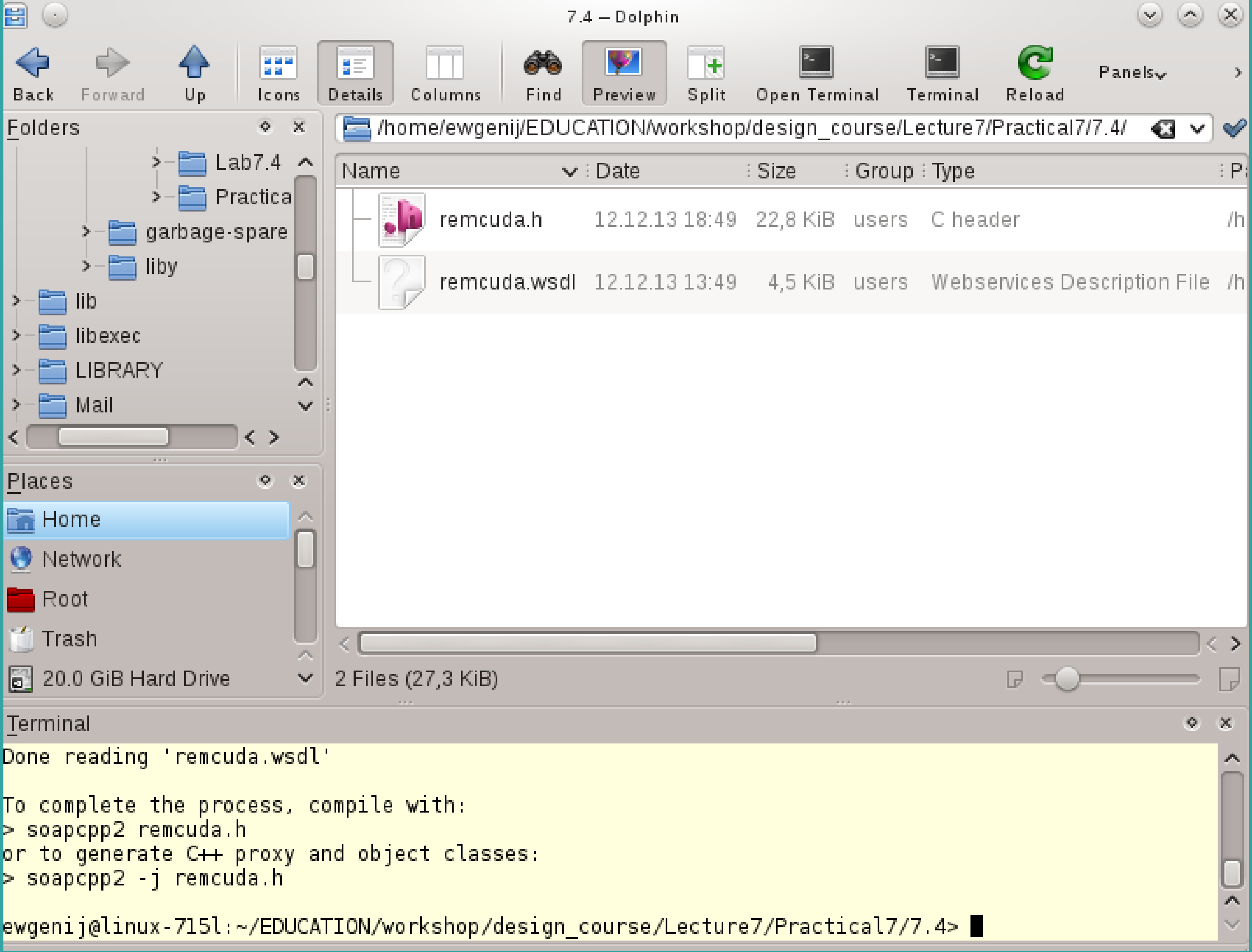


/home/ewgenij/EDUCATION/workshop/design_course/Lecture7/Practical7/7.4/

Name	Date	Size	Group	Type	Pat
remcuda.wsdl	12.12.13 13:49	4,5 KiB	users	Webservices Description File	/ho

1 File (4,5 KiB)

```
ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.4> wsdl2h -o remcuda.h
remcuda.wsdl
```



/home/ewgenij/EDUCATION/workshop/design_course/Lecture7/Practical7/7.4/

Name	Date	Size	Group	Type	P:
remcuda.h	12.12.13 18:49	22,8 KiB	users	C header	/h
remcuda.wsdl	12.12.13 13:49	4,5 KiB	users	Webservices Description File	/h

2 Files (27,3 KiB)

```
Done reading 'remcuda.wsdl'

To complete the process, compile with:
> soapcpp2 remcuda.h
or to generate C++ proxy and object classes:
> soapcpp2 -j remcuda.h

ewgenij@linux-715l:~/EDUCATION/workshop/design_course/Lecture7/Practical7/7.4> 
```

Фрагмент файла *remcuda.wsdl*

```
……………………………………………………………………………………………………………
 <complexType name="ArrayOffloat">
   <complexContent>
    <restriction base="SOAP-ENC:Array">
     <sequence>
      <element name="item" type="xsd:float" minOccurs="0" maxOccurs="unbounded" nillable="false"/>
     </sequence>
     <attribute ref="SOAP-ENC:arrayType" WSDL:arrayType="xsd:float[]"/>
    </restriction>
   </complexContent>
  </complexType>
  <!-- operation request element -->
  <element name="sum">
   <complexType>
    <sequence>
     <element name="v1" type="ns:ArrayOffloat" minOccurs="1" maxOccurs="1" nillable="true"/><!-- ns__sum::v1 -->
     <element name="v2" type="ns:ArrayOffloat" minOccurs="1" maxOccurs="1" nillable="true"/><!-- ns__sum::v2 -->
    </sequence>
   </complexType>
  </element>
  <!-- operation response element -->
  <element name="sumResponse">
   <complexType>
    <sequence>
     <element name="result" type="ns:ArrayOffloat" minOccurs="0" maxOccurs="1" nillable="true"/><!--
ns__sum::result -->
    </sequence>
   </complexType>
  </element>
  <!-- operation request element -->
  <element name="InnerProduct">
   <complexType>
……………………………………………………………………………………………………………
```

Фрагмент файла *remcuda.h*

```
…………………………………………………………………………………………………………………………………………
/// Top-level root element "urn:remcuda":sum
/// "urn:remcuda":sum is a complexType.
class _ns2__sum
{ public:
/// Element v1 of type "urn:remcuda":ArrayOffloat.
    ArrayOffloat*                        v1                          1;       ///< Nullable pointer.
/// Element v2 of type "urn:remcuda":ArrayOffloat.
    ArrayOffloat*                        v2                          1;       ///< Nullable pointer.
/// A handle to the soap struct that manages this instance (automatically set)
    struct soap                         *soap                        ;
};
/// Top-level root element "urn:remcuda":sumResponse

/// "urn:remcuda":sumResponse is a complexType.
class _ns2__sumResponse
{ public:
/// Element result of type "urn:remcuda":ArrayOffloat.
    ArrayOffloat*                        result                       0;      ///< Nullable pointer.
/// A handle to the soap struct that manages this instance (automatically set)
    struct soap                         *soap                        ;
};
/// Top-level root element "urn:remcuda":InnerProduct

/// "urn:remcuda":InnerProduct is a complexType.
class _ns2__InnerProduct
{ public:
/// Element v1 of type "urn:remcuda":ArrayOffloat.
    ArrayOffloat*                        v1                          1;       ///< Nullable pointer.
/// Element v2 of type "urn:remcuda":ArrayOffloat.
    ArrayOffloat*                        v2                          1;       ///< Nullable pointer.
/// A handle to the soap struct that manages this instance (automatically set)
    struct soap                         *soap                        ;
};
…………………………………………………………………………………………………………………………………………
```