

Unidade Curricular:

Integração de Sistemas de Informação

Tema da Ficha Prática:

Utilização de RPC – Remote Procedure Call

Objectivos:

Pretende-se com esta ficha prática que os alunos interajam com o conceito de RPC – Remote Procedure Call.

Bibliografia:

Para apoio a esta ficha os alunos devem consultar os apontamentos teóricos e práticos da disciplina bem como de outros recursos online. Mais informações sobre RPC no anexo e em:

<http://netbula.com/javarpc/msgsamp.html>

Exemplos Práticos: <http://ybwebservicesjrpc.blogspot.pt/>
<http://gbidsilva.blogspot.pt/2012/02/rpc-with-netbula-jrpc.html>

Índice

1. Utilização do conceito de RPC usando a linguagem de programação JAVA.....	3
1.1 Exercício 1 – Execução do exemplo de envio de mensagens.....	3
O PortMapper:	4
1.2 Exercício1 – Colocar um Ficheiro.....	6
1.2.1 FileClient.java.....	6
1.2.2 FileServer.java	7
1.2.2 putfile.x	9
2 Exemplo echo	9
2.1 MyServer.java	9
2.1 echoclient.java.....	9
2.2 echo.java.....	10
2.3 Execução	11
3 Exemplo colocaFicheiro	12
3.1 FileServer.java	12
3.2 FileClient.java.....	13
3.3 Execução	13
4 Exemplo Soma	14
4.1 MyServerSoma.java	14

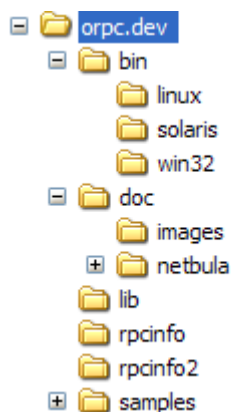
4.2 MyClientSoma.java	15
4.3 Execução	16
5 Exemplo TamanhoString	17
5.1 MyServerSize.java	17
5.1 MyClientSize.java	17
5.1 Execução	18
Anexo 1 – Documentação Online	20
Step 1. Compile the Msg.x with jrpcgen.....	20
Step 2. Code the main client application.....	21
Step 3. Compile the client	22
Step 4. Run the Msg client.....	22
Step 5 Code the Msg server	23
Step 6 Compile and run the java server.....	23

1. Utilização do conceito de RPC usando a linguagem de programação JAVA

Para apoio à elaboração do exemplo seguinte consulte a documentação disponível em:

[1] <http://netbula.com/javarpc/>

Para elaborar estes exercícios deve descompactar o ficheiro javarpc.zip que contém a seguinte estrutura:

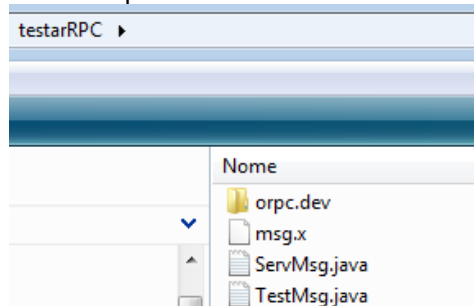


NOTA: Deverá redireccionar as variáveis de ambiente PATH e CLASSPATH.

1.1 Exercício 1 – Execução do exemplo de envio de mensagens

Exemplo Básico:

Crie uma pasta “testarRPC” em C: e descompacte o pacote orpc.dev



1º Configurar a variável PATH para se encontrar o programa “jrpcgen.exe” o qual está em orpc.dev\bin\win32

C:\testarRPC>set PATH=%PATH%;c:\testarRPC\orpc.dev\bin\win32

2º Compilar o ficheiro *.x usando o jrpcgen

C:\testarRPC>jrpcgen msg.x

Output:

Netbula JRPCGen V2.5.5.
Netbula JavaRPC demo, expires after a fixed date!
Generated XDR files:
Generated RPC Client files:
 msg_idl_cln.java
Generated RPC Server files:
 msg_idl_svcb.java

Observação: Foram criados os seguintes ficheiros:



3º Apontar o CLASSPATH para o ficheiro jar do módulo orpc.jar

Ex: Set the CLASSPATH to include the ../lib/orpc.jar file

C:\testarRPC>set CLASSPATH=%CLASSPATH%;C:\testarRPC\orpc.dev\lib\orpc.jar;

NOTA: coloque no fim ;.

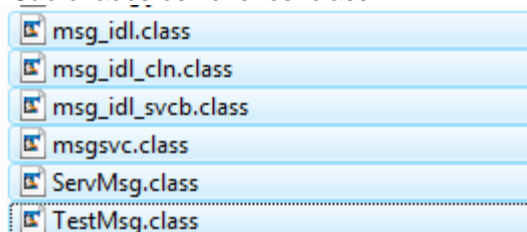
PONTO E VIRGULA PONTO Para indicar ao CLASSPATH que vai adicionar ao caminho “a directo-
ria atual”

4º Compilar os ficheiros .java

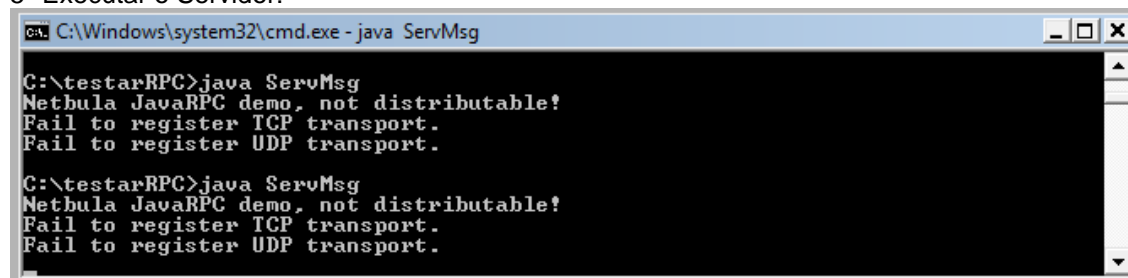
C:\testarRPC>set PATH=%PATH%;"c:\Program Files\Java\jdk1.6.0_20\bin"

C:\testarRPC>javac *.java

OUTPUT: São criados os ficheiros .class



5º Executar o Servidor:



NOTA: O registo dos protocolos de transporte não pode falhar.

O PortMapper:

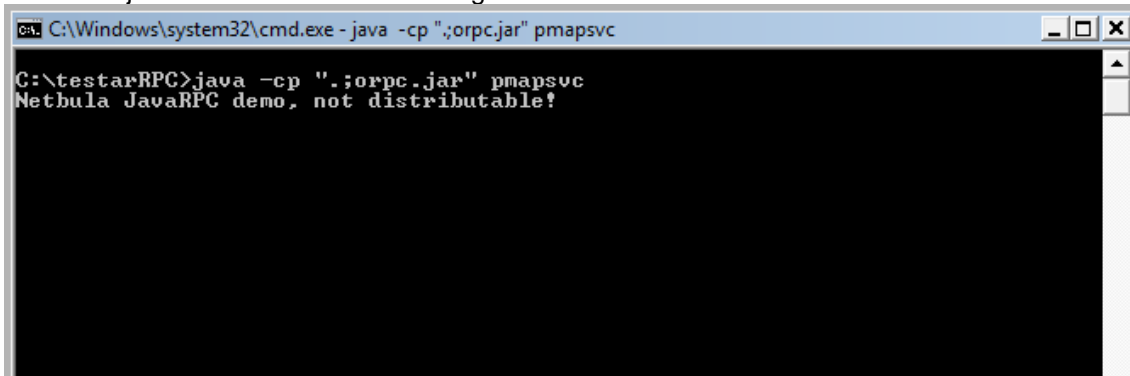
pmmapsvc: portmapper in Java(tm).

- Portmapper is like the RMI registry;
- RPC servers register their port numbers with the portmapper, and the RPC clients query portmapper to obtain the servers' ports.
- Portmapper runs on the RPC servers' machine.
- If your system does not yet have portmapper running, you can use this to start one. RPC server/client can also use fixed ports and by-pass portmapper registration and querying

Usando os pacotes java disponibilizados, o portmapper está disponível em:

- Copie para a pasta TestarRPC, o ficheiro orpc.jar que está em orpc.dev\lib
- Copie para a pasta TestarRPC, o ficheiro pmapsvc.class que está em orpc.dev\samples\pmapsvc

Abra uma janela de Dos e execute o seguinte comando:



```
C:\Windows\system32\cmd.exe - java -cp ".\orpc.jar" pmapsvc

C:\testarRPC>java -cp ".\orpc.jar" pmapsvc
Netbula JavaRPC demo, not distributable!
```

Agora abra duas janelas, uma para executar o servidor e outra para executar o cliente.
Para facilitar a abertura crie dois ficheiros .bat:

runServer.bat

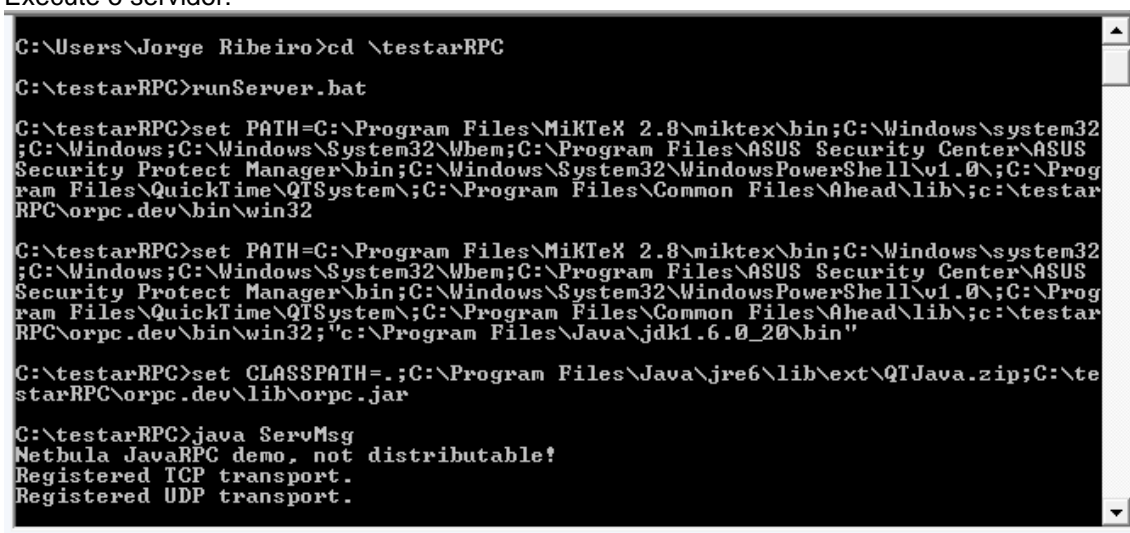
```
set PATH=%PATH%;c:\testarRPC\orpc.dev\bin\win32
set PATH=%PATH%;"c:\Program Files\Java\jdk1.6.0_20\bin"
set CLASSPATH=%CLASSPATH%;C:\testarRPC\orpc.dev\lib\orpc.jar
java ServMsg
```

E

runClient.bat

```
set PATH=%PATH%;c:\testarRPC\orpc.dev\bin\win32
set PATH=%PATH%;"c:\Program Files\Java\jdk1.6.0_20\bin"
set CLASSPATH=%CLASSPATH%;C:\testarRPC\orpc.dev\lib\orpc.jar
java TestMsg localhost
```

Execute o servidor:



```
C:\Users\Jorge Ribeiro>cd \testarRPC
C:\testarRPC>runServer.bat

C:\testarRPC>set PATH=C:\Program Files\MikTeX 2.8\miktex\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Program Files\ASUS Security Center\ASUS Security Protect Manager\bin;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\QuickTime\QTSystem\;C:\Program Files\Common Files\Ahead\lib;c:\testarRPC\orpc.dev\bin\win32
C:\testarRPC>set PATH=C:\Program Files\MikTeX 2.8\miktex\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Program Files\ASUS Security Center\ASUS Security Protect Manager\bin;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\QuickTime\QTSystem\;C:\Program Files\Common Files\Ahead\lib;c:\testarRPC\orpc.dev\bin\win32;"c:\Program Files\Java\jdk1.6.0_20\bin"
C:\testarRPC>set CLASSPATH=.;C:\Program Files\Java\jre6\lib\ext\QTJava.zip;C:\testarRPC\orpc.dev\lib\orpc.jar
C:\testarRPC>java ServMsg
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Execute o Cliente:

```
C:\Windows\system32\cmd.exe

do Servidor

Enviando: 0 glorioso...e mais nada!1
Recebido: 0 glorioso...e mais nada! 1
do Servidor

Enviando: 0 glorioso...e mais nada!2
Recebido: 0 glorioso...e mais nada! 2
do Servidor

Enviando: 0 glorioso...e mais nada!3
Recebido: 0 glorioso...e mais nada! 3
do Servidor

Enviando: 0 glorioso...e mais nada!4
Recebido: 0 glorioso...e mais nada! 4
do Servidor

C:\testarRPC>
```

Dispondo as três janelas:

```
C:\Windows\system32\cmd.exe - java -cp ".\orpc.jar" pmapsv
C:\testarRPC>java -cp ".\orpc.jar" pmapsv
Netbula JavaRPC demo, not distributable!

C:\Windows\system32\cmd.exe
do Servidor
Enviando: 0 glorioso...e mais nada!1
Recebido: 0 glorioso...e mais nada! 1
do Servidor
Enviando: 0 glorioso...e mais nada!2
Recebido: 0 glorioso...e mais nada! 2
do Servidor
Enviando: 0 glorioso...e mais nada!3
Recebido: 0 glorioso...e mais nada! 3
do Servidor
Enviando: 0 glorioso...e mais nada!4
Recebido: 0 glorioso...e mais nada! 4
do Servidor
C:\testarRPC>

C:\Windows\system32\cmd.exe - runServer.bat
C:\testarRPC>set PATH=C:\Program Files\Microsoft 2.0\jdk6\bin;C:\Windows\system32
;C:\Windows\System32\bin;C:\Program Files\MSUS Security Center\MSUS
Security Protect Manager\bin;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Progr
am Files\QuickTime\QTSystem\;C:\Program Files\Common Files\Ahead\Lib\;c:\testar
RPC\orpc.dev\bin\win32;"c:\Program Files\Java\jdk1.6.0_20\bin"
C:\testarRPC>set CLASSPATH=.;C:\Program Files\Java\jre6\lib\ext\QTJava.zip;C:\tes
tarRPC\orpc.dev\lib\orpc.jar
C:\testarRPC>java ServMsg
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Mensagem recebida: 0 glorioso...e mais nada! 0
Mensagem recebida: 0 glorioso...e mais nada! 1
Mensagem recebida: 0 glorioso...e mais nada! 2
Mensagem recebida: 0 glorioso...e mais nada! 3
Mensagem recebida: 0 glorioso...e mais nada! 4
```

NOTA: Analise o código dos ficheiros ServMsg.java e o TestMsg.java, assim como o output da execução dos mesmos.

1.2 Exercício1 – Colocar um Ficheiro

1.2.1 FileClient.java

```
import netbula.ORPC.*;

public class FileClient{

    static public void main(String args[]) {
```

```
try {
    if(args == null || args.length < 2) {
        System.out.println("syntax: java fileClient server_hostname file1 [file2 file3 ..]");
        System.exit(1);
    }
    String servhost = args[0];

    /* utilizar TCP, UDP não consegue enviar grande número de bytes*/
    putfile_cln cl = new putfile_cln(servhost, "tcp");

    System.out.println("Ligado ao servidor " +servhost);

    /* envia todos os ficheiros identificados na linha de comando */
    NFiles nf = new NFiles();
    nf.files = new XDFile[args.length -1];
    for(int i=0; i< args.length-1; i++)
        nf.files[i] = new XDFile(args[i+1]);

    cl.sendFile(nf);

    for(int i=0; i< args.length-1; i++)
        System.out.println(args[i+1]+ " " + nf.files[i].byteCount()+ " bytes enviados");

} catch (Exception e) {
    System.out.println("rpc: " + e.toString());
    e.printStackTrace();
}

}

}
```

1.2.2 FileServer.java

```
import netbula.ORPC.*;
import java.io.*;

public class FileServer extends putfile_svcb{

    public void sendFile(NFiles in_arg){
        for(int i=0; i<in_arg.files.length; i++) {
            System.out.println("Ficheiro recebido: "+ in_arg.files[i].receivedFilepath()+ " " +
in_arg.files[i].byteCount()+ " bytes transferidos");
            System.out.println("Ficheiro gravado: "+ in_arg.files[i].savedFilename());
        }
    }

    static public void main(String args[]) {

        rpc_err.debug=true;
        FileServer server = new FileServer();
        try {
            server.run();
            System.out.println("servidor terminou");
        } catch (rpc_err e) {
            System.out.println("Problemas para executar o servidor:"+e.toString());
        }
    }
}
```

FICHA PRÁTICA n.º 3

Integração de Sistemas de Informação

}

1.2.2 putfile.x

/*The following got copied to the generated java files */

```
%import netbula.ORPC.*;

struct NFiles {
    XDFile files<>;
};

program putfile{
    version v1{
        void sendFile(NFiles)=1;
    } = 1;
} = 12345678;
```

2 Exemplo echo

2.1 MyServer.java

```
import netbula.ORPC.*;

public class MyServer extends echo_svcb
{
    public static void main(String[] args)
    {
        try
        {
            new MyServer().run();
            System.out.println("server exited");
        }
        catch (rpc_err e)
        {
            System.out.println("Fail to run server:"+e.toString());
        }
    }

    public MyServer() throws rpc_err
    {
        super();
    }

    public int echo(int a)
    {
        return(a*2);
    }
}
```

2.1 echoclient.java

```
import netbula.ORPC.*;
import java.net.InetAddress;
```

```
public class echoclient
{

    public static void main(String[] argv)
    {
        /*
        //Trabalhar com DNS names
        try {
            InetAddress address = InetAddress.getByName(argv[0]);
            byte ip[] = address.getAddress();

            for (int octet=0; octet < ip.length; octet++)
            {

                System.out.print(((int)ip[octet]) & 0xff);
                if (octet+1<ip.length) System.out.print (" ");

            }
            System.out.println();
        }
        catch (Exception e)
        {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    /*

    echo_cln client;
    if(argv.length <2)
    {
        System.out.println("Sintax: java echoclient host number");
        System.exit(0);
    }
    try
    {
        client = new echo_cln(argv[0],"tcp");
        int result = client.echo(Integer.parseInt(argv[1]));

        System.out.println("\nResultado: " + result);
    }
    catch(Exception e)
    {
        System.out.println("\nErro: " + e.toString());
    }
    System.exit(0);
}
}
```

2.2 echo.java

//Generated by Netbula JRPCGEN V2.5.5.
//Netbula JavaRPC demo, expires after a fixed date!

```
public interface echo{

    public static final int _def_pno = 77;
    public static final int _def_vno = 1;
```

```
public static final int _echo_proc = 1;

public int echo(int in_arg) throws netbula.ORPC.rpc_err;

}
```

2.3 Execução

```
Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Vale>cd..
C:\Users>cd..
C:\>cd testar
C:\testar>cd testarRPC
C:\testar\testarRPC>set PATH=%PATH%;c:\testar^Z
C:\testar\testarRPC>set PATH=%PATH%;c:\testar\testarRPC\orpc.dev\bin\win32
C:\testar\testarRPC>jrpcgen MyServer
Netbula JRPCGen U2.5.5.
Netbula JavaRPC demo, expires after a fixed date!
jrpcgen: MyServer: No such file or directory
C:\testar\testarRPC>
```

```
C:\TESTAR\testarRPC>javac *.java
C:\TESTAR\testarRPC>java MyServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Executar o Portmapper

```
C:\TESTAR\testarRPC>portmapper.bat
C:\TESTAR\testarRPC>java -cp ".;orpc.jar" pmapsvc
Netbula JavaRPC demo, not distributable!
```

Executar numa nova linha de comandos o servidor

```
C:\TESTAR\testarRPC>runServer.bat

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Jav
a\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java MyServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Executar numa nova linha de comandos o cliente

```
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Jav
a\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de progra
mas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Jav
a\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de progra
mas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de
programas\Java\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar;C:\
TESTAR\testarRPC\orpc.dev\lib\orpc.jar;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java echoclient PC 11976
Netbula JavaRPC demo, not distributable!

Resultado: 23952

C:\TESTAR\testarRPC>
```

3 Exemplo colocaFicheiro

3.1 FileServer.java

```
import netbula.ORPC.*;
import java.io.*;

public class FileServer extends putfile_svcb{

    public void sendFile(NFiles in_arg){
        for(int i=0; i<in_arg.files.length; i++) {
            System.out.println("Ficheiro recebido: "+ in_arg.files[i].receivedFilepath()+" " +
in_arg.files[i].byteCount()+" bytes transferidos");
            System.out.println("Ficheiro gravado: "+ in_arg.files[i].savedFilename());
        }
    }

    static public void main(String args[]) {

        rpc_err.debug=true;
        FileServer server = new FileServer();
        try {
            server.run();
            System.out.println("servidor terminou");
        }catch(rpc_err e) {
```

```
        System.out.println("Problemas para executar o servidor:"+e.toString());
    }
}
```

3.2 FileClient.java

```
import netbula.ORPC.*;

public class FileClient{

    static public void main(String args[]) {
        try {
            if(args == null || args.length < 2) {
                System.out.println("syntax: java fileClient server_hostname file1 [file2 file3 ..]");
                System.exit(1);
            }
            String servhost = args[0];

            /* utilizar TCP, UDP não consegue enviar grande número de bytes*/
            putfile_cln cl = new putfile_cln(servhost, "tcp");

            System.out.println("Ligado ao servidor " +servhost);

            /* envia todos os ficheiros identificados na linha de comando */
            NFiles nf = new NFiles();
            nf.files = new XDFile[args.length -1];
            for(int i=0; i< args.length-1; i++)
                nf.files[i] = new XDFile(args[i+1]);

            cl.sendFile(nf);

            for(int i=0; i< args.length-1; i++)
                System.out.println(args[i+1]+ " "+ nf.files[i].byteCount()+ " bytes enviados");

        }catch (Exception e) {
            System.out.println("rpc: " + e.toString());
            e.printStackTrace();
        }

    }
}
```

3.3 Execução

```

Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Vale>cd..

C:\Users>cd..

C:\>cd testar

C:\testar>cd testarRPC

C:\testar\testarRPC>set PATH=%PATH%;c:\testar\testarRPC\orpc.dev\bin\win32

C:\testar\testarRPC>jrpcgen FileServer
Netbula JRPCGen U2.5.5.
Netbula JavaRPC demo, expires after a fixed date!
jrpcgen: FileServer: No such file or directory

C:\testar\testarRPC>
    
```

```

C:\TESTAR\testarRPC>javac *.java

C:\TESTAR\testarRPC>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
    
```

Executar o portmapper numa consola:

```

C:\TESTAR\testarRPC>portmapper.bat

C:\TESTAR\testarRPC>java -cp ".;orpc.jar" pmapsvc
Netbula JavaRPC demo, not distributable!
    
```

Executar numa nova janela o servidor

```

Mbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Comon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Comon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=.;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: NOU01.txt 0 bytes transferidos
Ficheiro gravado: NOU01.txt
Closing socket.
    
```

Executar numa nova janela o cliente

4 Exemplo Soma

4.1 MyServerSoma.java

```
import netbula.ORPC.*;

public class MyServerSoma extends add_svcb
{
    public static void main(String[] args)
    {
        try
        {
            new MyServerSoma().run();
        }
        catch (rpc_err e)
        {
            System.out.println(e.getMessage());
        }
    }

    public MyServerSoma() throws rpc_err
    {
        super();
    }

    public result add(request arg)
    {
        result r=new result();
        r.value=arg.x+arg.y;
        return (r);
    }
}
```

4.2 MyClientSoma.java

```
import netbula.ORPC.*;
import java.net.*;

public class MyClientSoma
{
    public static void main(String[] args)
    {
        add_cln client;
        if(args.length <3)
        {
            System.out.println("Sintaxa: java MyClientSoma hostName num1 num2");
            System.exit(0);
        }
        try
        {
            client = new add_cln(args[0],"tcp");
            request r = new request();
            r.x=Integer.parseInt(args[1]);
            r.y=Integer.parseInt(args[2]);
            result result = client.add(r);
            System.out.println("\nResultado: " + result.value);
        }
        catch(Exception e)
        {
            System.out.println("\nErro: " + e.getMessage());
        }
        System.exit(0);
    }
}
```

}

4.3 Execução

```
Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
C:\Users>cd..
C:\>cd..
C:\>cd testar
C:\testar>cd testarRPC
C:\testar\testarRPC>set PATH=%PATH%;c:\testar\testarRPC\orpc.dev\bin\win32
C:\testar\testarRPC>jrpcgen MyServerSoma
Netbula JRPCGen U2.5.5.
Netbula JavaRPC demo, expires after a fixed date!
jrpcgen: MyServerSoma: No such file or directory
C:\testar\testarRPC>_
```

```
C:\TESTAR\testarRPC>javac *.java
C:\TESTAR\testarRPC>java MyServerSoma
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Executar o portmapper numa linha de comandos

```
C:\TESTAR\testarRPC>portmapper.bat
C:\TESTAR\testarRPC>java -cp ".;orpc.jar" pmapsvc
Netbula JavaRPC demo, not distributable!
```

Numa nova linha de comandos executar o servidor

```
C:\TESTAR\testarRPC>runServer.bat
C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\C
ommon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32
C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\C
ommon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Jav
a\jdk1.7.0_13\bin"
C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar
C:\TESTAR\testarRPC>java MyServerSoma
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Numa nova linha de comandos executar o cliente


```
C:\TESTAR\testarRPC>runClient.bat

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\C
ommon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\C
ommon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Jav
a\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java MyClientSoma PC 11 33
Nethula JavaRPC demo, not distributable!

Resultado: 44

C:\TESTAR\testarRPC>
```

5 Exemplo TamanhoString

5.1 MyServerSize.java

```
import netbula.ORPC.*;

public class MyServerSize extends strcnt_svcb
{
    public static void main(String[] args)
    {
        try
        {
            new MyServerSize().run();
            System.out.println("server terminou");
        }
        catch (rpc_err e)
        {
            System.out.println(e.getMessage());
        }
    }

    public MyServerSize() throws rpc_err
    {
        super();
    }

    public int strcount(String arg1)
    {
        return(arg1.length());
    }
}
```

5.1 MyClientSize.java

```
import netbula.ORPC.*;
import java.net.*;

public class MyClientSize
{
    public static void main(String[] args)
    {
```

```

        strcnt_cln client;
    if(args.length <2)
    {
        System.out.println("Sintax: java MyClientSize host string");
        System.exit(0);
    }
    try
    {
        client = new strcnt_cln(args[0],"tcp");
        int result = client.strcount(args[1]);
        System.out.println("\nTamanho da string: " + result);
    }
    catch(Exception e)
    {
        System.out.println("\nErro: " + e.getMessage());
    }
    System.exit(0);
}
}

```

5.1 Execução

```

Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
C:\Users>cd..
C:\>cd testar
C:\testar>cd testarRPC
C:\testar\testarRPC>set PATH=%PATH%;c:\testar\testarRPC\orpc.dev\bin\win32
C:\testar\testarRPC>jrpgen MyServerSize
'jrpgen' não é reconhecido como um comando interno ou externo,
programa operacional ou ficheiro batch.
C:\testar\testarRPC>jrpgen MyServerSize
Nethula JRPCGen U2.5.5.
Nethula JavaRPC demo, expires after a fixed date!
jrpgen: MyServerSize: No such file or directory
C:\testar\testarRPC>

```

```

C:\TESTAR\testarRPC>javac *.java
C:\TESTAR\testarRPC>java MyServerSize
Nethula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.

```

Executar numa linha de comandos o portmapper

```

C:\TESTAR\testarRPC>portmapper.bat
C:\TESTAR\testarRPC>java -cp ".;orpc.jar" pmapsvc
Nethula JavaRPC demo, not distributable!

```

Executar numa nova linha de comandos o servidor

```
C:\TESTAR\testarRPC>runServer.bat

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
When;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
When;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java
a\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java MyServerSize
Nethula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
```

Executar numa nova linha de comandos o cliente

```
ommon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java
a\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de progra
mas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de
programas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32

C:\TESTAR\testarRPC>set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
When;C:\WINDOWS\system32\WindowsPowerShell\v1.0;C:\Arquivos de programas\Microso
ft SQL Server\100\Tools\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\
DTS\Binn\;C:\Arquivos de programas\Microsoft SQL Server\100\Tools\Binn\USShell\Co
mmon7\IDE\;C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de programas\Java
a\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arquivos de progra
mas\Java\jdk1.7.0_13\bin";C:\TESTAR\testarRPC\orpc.dev\bin\win32;"C:\Arqu
ivos de programas\Java\jdk1.7.0_13\bin"

C:\TESTAR\testarRPC>set CLASSPATH=;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar;C:\
TESTAR\testarRPC\orpc.dev\lib\orpc.jar;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar
;C:\TESTAR\testarRPC\orpc.dev\lib\orpc.jar

C:\TESTAR\testarRPC>java MyClientSize 127.0.0.1
Nethula JavaRPC demo, not distributable!

Tamanho da string: 7

C:\TESTAR\testarRPC>
```

Anexo 1 – Documentação Online

The screenshot shows a web browser displaying the Netbula LLC website. The address bar shows 'netbula.com/javarp/mgsamp.html'. The website has a blue header with the Netbula LLC logo and navigation links: ABOUT US, PRODUCTS, SUPPORT, DOWNLOAD, PURCHASE. Below the header, there's a 'Products' section with tabs for PowerRPC, ONC RPC, JRPC, Anyboard, and Anyemail. The 'JRPC' tab is selected, showing a 'JRPC Programming Tutorial' section. The tutorial is titled 'A step by step illustration on using the JRPC SDK'. The text describes it as a Hello World example and provides instructions on downloading the SDK and looking at the code under samples/ directory. It also mentions that the tutorial illustrates how to build a JRPC client for a simple Msg RPC server/client and references the book 'Power Programming RPC' from O'Reilly.

JRPC Programming Tutorial

A step by step illustration on using the JRPC SDK

This is a Hello World example, for more elaborate examples, please download the SDK and look at the code under samples/ directory.

In this tutorial, we illustrate how to build a JRPC client for a simple Msg RPC server/client. This tutorial is not about programming ONC RPC, but about how to use the JRPC tool. If you are not familiar with ONC RPC, the book "Power Programming RPC" from O'Reilly is a good guide.

The Msg server is defined by the following RPC IDL

```
%cat msg.x
```

```
program msgserv {
    version MSGSERV_V1 {
        string sendmsg(string)=2;
    }= 1;
} = 1234567;
```

The interface defines an RPC program with a single procedure **sendmsg**, the client sends a string to the server, and the server returns a string back.

The **C/C++** version of the Msg client/server is available from the [Netbula ONC RPC For Win32 SDK](#).

Now let's build the Msg RPC client in Java(tm).

Step 1. Compile the Msg.x with jrpgen

At the command prompt, run

```
% jrpgen msg.x
```

The demo package includes jrpcgen binaries for win32, solaris and linux, they generate identical Java(tm) code.

This would produce the following files:

- [msgserv.java](#) The RPC program interface definition, including constant definition such as program number.
- [msgserv_cln.java](#) The client stub class. This class implements the RPC interface defined above. An RPC client program instantiate an instance of this class and call its methods (remote call).
- [msgserv_svcb.java](#) The RPC service. This class inherits the RPC interface and is abstract, the programmer needs to extend this class and supply the implementation for the interface.

Normally, jrpcgen would produce four kinds of Java(tm) source code files

- XDT classes for user defined types. These classes can be serialized through XDR streams.
- RPC program interface
- Client stub class
- Server stub class

In our case, there is no XDT classes, because **string** is a built-in type.

Step 2. Code the main client application

This is very easy, we just need to create an instance of the generated msgserv_cln class and call its methods.

%cat [ClientTest.java](#)

```
import netbula.ORPC.*;

import java.net.*;
public class ClientTest {
    public ClientTest () {}

    static public void main(String args[]) {

        try {

            msgserv_cln cl = new msgserv_cln(args[0], "udp");

/*
            cl.setAuth(new AuthUnix
                ("localhost", 501, 100, new int[2]));
*/

            String msg = "hello world\n";

            System.out.println("sending.. ");

            for(int i=0; i<5; i++){

                String reply = cl.sendmsg(msg);

                System.out.println(
                    "got " + reply + "\n");

            }

        }catch (rpc_err e) {

            System.out.println("rpc: " + e.toString());

        }

    }

}
```

Here, we construct a Msg client which connects to the Msg server on host argv[0] with UDP protocol, send a message, and print out the reply.

Step 3. Compile the client

Make sure that the **netbula.ORPC** package is in the classpath (simply add the orpc.jar file to the **CLASSPATH** environment variable).

Run the source through the Java(tm) compiler:

```
% javac ClientTest.java MSGSERV_1.java
```

This would produce two class files: ClientTest.class and MSGSERV_1.class.

Step 4. Run the Msg client

1. Make sure the Msg server (C version or Java(tm) version) is running on localhost

2. Run the client

```
% java ClientTest
```

If the server is running, you should see the client print out the reply from the server, otherwise, it will print out an RPC error: Program not registered.

That is it!

Now, let's build the Msg server in Java(tm)

Step 5 Code the Msg server

The jrpcgen generates [msgserv_svcb.java](#), which defines an abstract class [msgserv_svcb](#) with an abstract function **sendmsg**. To fully implement the server, one needs to derive a class which supplies a body for the sendmsg function.

```
import netbula.ORPC.*;

class msgsvc extends msgserv_svcb {
    //implement the server function,
    //let's just echo the msg back

    String sendmsg(String msg) {
        System.out.println("got msg from client "+ msg);
        return msg;
    }

    //main function runs the server

    public static void main(String sargv[]) {

        //let's run the server using the run() method in Svc
        //For more flexibility, one could use the TCPServer and UDPServer directly

        new msgsvc().run();

    }
}
```

Step 6 Compile and run the java server

```
%javac msgsvc.java
%java msgsvc
```

*) Implement the Msg server/client in C

1) Use rpcgen to compile msg.x file into client server stubs. rpcgen is available on unix, [rpcgen for win32](#) is available from Netbula.

2) Code the server implementation

See the **cservs** directory in the JRPC package for sample code.

File Transfer Server/Client with JRPC

Next, we look at a more interesting example, a JRPC server/client that transfer multiple files via RPC mechanism. This example is under **samples/filexfer** directory of the JRPC package.

The Netbula JRPC API has a class named XDFile, this is a class to serialize a disk file to and from an XDR stream.

The .x file for the file transfer RPC interface is listed below:

```
%import netbula.ORPC.*;

struct NFiles { XDFile files<> };
program FileXFER{
    version v1{
        void xferFile(NFiles)=1;
    } = 1;
} = 12345678;
```

This is a very simple interface, we defined a struct NFiles, which contains a variable length of XDFile. The xferFile function takes NFiles as an argument, so it can transfer any number of files.

Now, the server code (FileServer.java). The server saves the files (which is done by XDFile) received and print out a message.

```
import netbula.ORPC.*;
import java.io.*;

public class FileServer extends filexfer_svc{

    public void xferfile(NFiles in_arg){
        for(int i=0; i<in_arg.files.length; i++) {
            System.out.println("Received file: "+
                in_arg.files[i].receivedFilepath()+
                " " + in_arg.files[i].byteCount()+
                " bytes transfered");
            System.out.println("saved file: "+
                in_arg.files[i].savedFilename());
        }
    }

    static public void main(String args[]) {

        rpc_err.debug=true;

        FileServer server = new FileServer();
```



```
try {
    server.run();
    System.out.println("server exited");
} catch (rpc_err e) {
    System.out.println("Fail to run server:" + e.toString());
}
}
```

The client code (FileClient.java). The client sends the files listed on the command line to the server.

```
import netbula.ORPC.*;

public class FileClient{

    static public void main(String args[]) {
        try {
            if(args == null || args.length < 2) {
                System.out.println("Usage: java fileClient serv-
er_hostname file1 [file2 file3 ..]");
                System.exit(1);
            }
            String servhost = args[0];

            /* use TCP, UDP is not reliable */
            filexfer_cln cl = new filexfer_cln(servhost, "tcp");

            System.out.println("Connected to " + servhost);

            /* send the files listed on command line args to the server */

            NFiles nf = new NFiles();
            nf.files = new XDFile[args.length - 1];
            for(int i=0; i< args.length-1; i++)
                nf.files[i] = new XDFile(args[i+1]);

            cl.xferfile(nf); //send all the files over

            for(int i=0; i< args.length-1; i++)
                System.out.println(args[i+1] + " " + nf.files[i].byteCount() + " bytes
sent");
        }
    }
}
```

FICHA PRÁTICA n.º 3

Integração de Sistemas de Informação

```
}catch (Exception e) {  
    System.out.println("rpc: " + e.toString());  
    e.printStackTrace();  
}  
  
}  
  
}
```

You will probably agree that the above is a small amount of code which does some useful work, transfer any number of files to another machine.