Hochschule Landshut
Fakultät Informatik

Prof. Dr. Jürgensen
SS 18

Praktikum Verteilte Systeme

Aufgabe 2: Remote Procedure Calls (RPC)

Erweitern Sie das in der Vorlesung angegebene RPC-Beispiel:

Count-Prozedur:

Ausgabe der *Anzahl* aller in der Datenbank enthaltenen Wörter.

Select-Prozedur 1:

Ausgabe der *Menge* aller in der Datenbank enthaltenen Wörter. Geben Sie die Wörter einfach in einem (langen) String zurück, in dem die einzelnen Wörter durch ein Sonderzeichen, z.B. Leerzeichen, getrennt sind.

Select-Prozedur 2:

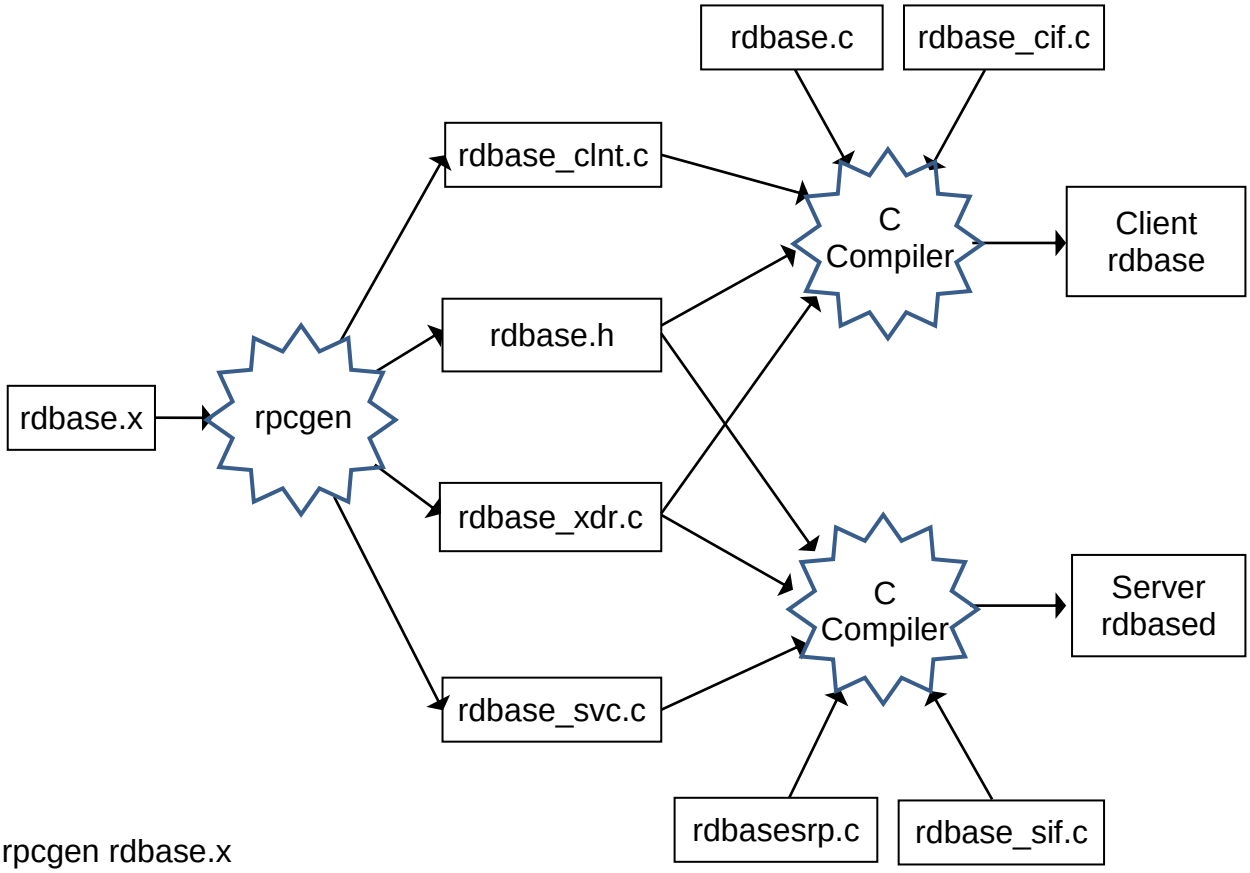
Wiederum Ausgabe der *Menge* aller in der Datenbank enthaltenen Wörter. Geben Sie die Wörter in einer geeigneten Datenstruktur zurück, die alle gefundenen Wörter enthält. Dieses Select ist dann die Vorstufe zu einem Select, in dem die Tupel einer Tabelle einer relationalen Datenbank zurückgegeben werden können.

Hinweise:

- Für die Datenstruktur *manywords* für die Select-Prozedur 2 empfiehlt es sich, in der von *rpcgen* generierten Datei *rdbase.h* nachzuschauen, welche C-Strukturen generiert worden sind (*words_len* bzw. **words_val*). Diese werden Sie zumindest in den Server-Prozeduren verwenden müssen, evtl. auch im Client. Für *words_val* ist in den Server-Prozeduren explizit Speicherplatz in ausreichender Größe zu allokieren: *(malloc(DICTSIZ*sizeof(oneword))).*
- Diese RPC-Aufgabe lässt sich sowohl mit *Terminal-Fenstern* für Server und Client (siehe den folgenden Anhang "RPC-Entwicklungsprozess") als auch mit *Eclipse* programmieren. Für Eclipse sind dafür zwei Projekte zu definieren. Das Server-Projekt enthält die Spezifikationsdatei und alle für den Server notwendigen .c- und .h-Dateien und das Client-Projekt entsprechend die Spezifikationsdatei und alle für den Client notwendigen .c- und .h-Dateien. Diese Dateien werden teilweise von Ihnen implementiert und teilweise mit dem *rpcgen*-Tool generiert. Das *rpcgen*-Tool läuft m.W. nicht unter Eclipse, muss also in einem Terminalfenster auf die Spezifikationsdatei angewandt werden, und danach werden die erzeugten Dateien in das entsprechende Eclipse-Projekt importiert (jeweils Refresh notwendig!). Anschließend kann man zuerst den Server übersetzen und laufen lassen (*rdbase_svc.c*, weil hier das *main()* enthalten ist) , dann den Client (*rdbase.c*).

Materialien auf dem Moodle-Server:
Beispiel aus der Vorlesung *(rdbase)*
RPC-Spezifikationsdatei *rdbase.x* für diese Aufgabe
Vorführung dieser Aufgabe im Praktikum.

Anhang: RPC Entwicklungsprozess



rpcgen rdbase.x

cc -c rdbase_clnt.c
cc -c rdbase_cif.c
cc -c rdbase.c

cc -c rdbase_xdr.c

cc -o rdbase rdbase_clnt.o rdbase_cif.o rdbase.o rdbase_xdr.o

cc -c rdbase_svc.c
cc -c rdbase_sif.c
cc -c rdbase_srp.c

cc -o rdbased rdbase_svc.o rdbase_sif.o rdbase_srp.o rdbase_xdr.o

./rdbased & // auf Server-Maschine
./rdbase // auf Client-Maschine