



Aplikacije - programiranje kolegij: Baze podataka

I. Batistić

email: `ivo@phy.hr`

Fizički odsjek, PMF

Osnovni princip



Komunikacija:

- ⑥ Komunikacija između procesa
(ovisi o OSu, npr. Unix IPC SysV)
- ⑥ Komunikacija prekom LANa/Interneta
 - △ Isto računalo - *tz.* localhost
 - △ Različita računala - standardno internet adresiranje

Primjer: postgresQL

(primjer iz postgresQL dokumentacije. Npr:

http://grdelin.phy.hr/~ivo/Nastava/Baze_podataka/ekstra/postgresql/libpq-example.html)

Početak aplikacije:

```
/* Compile: gcc -I/usr/include/postgresql/ prog.c -o prog.x -lpq */

#include <stdio.h>
#include <stdlib.h>
#include "libpq-fe.h"

static void
exit_nicely(PGconn *conn)
{
    PQfinish(conn);
    exit(1);
}
```

Primjer: PostgreSQL - nastavak

Početak glavnog programa u aplikaciji (deklaracije):

```
int
main(int argc, char **argv)
{
    const char *conninfo;
    PGconn      *conn;
    PGresult    *res;
    int          nFields;
    int          i, j;
```

Primjer: postgresQL - nastavak



Definiranje varijabli komunikacije i provjera argumenata:

```
if (argc > 1)
    conninfo = argv[1];
else
    conninfo = "dbname = template1";
```

Primjer: postgresQL - nastavak

Iniciranje veze s DBMSom:

```
/* make a connection to the database */

    conn = PQconnectdb(conninfo);

/*
 * check to see that the backend connection was successfully made
 */

    if (PQstatus(conn) != CONNECTION_OK)
    {
        fprintf(stderr, "Connection to database '%s' failed.\n",
                PQdb(conn));
        fprintf(stderr, "%s", PQerrorMessage(conn));
        exit_nicely(conn);
    }
```

Primjer: postgresQL - nastavak

Početak transakcije:

```
/* start a transaction block */

res = PQexec(conn, "BEGIN");

if (PQresultStatus(res) != PGRES_COMMAND_OK)
{
    fprintf(stderr, "BEGIN command failed: %s",
            PQerrorMessage(conn));
    PQclear(res);
    exit_nicely(conn);
}

/*
 * should PQclear PGresult whenever it is no longer needed to avoid
 * memory leaks
 */
PQclear(res);
```

Primjer: PostgreSQL - nastavak

Zadavanje SQL naredbe DBMSu:

```
res = PQexec(conn,
    "DECLARE myportal CURSOR FOR select * from pg_database");
if (PQresultStatus(res) != PGRES_COMMAND_OK)
{
    fprintf(stderr, "DECLARE CURSOR failed: %s",
        PQerrorMessage(conn));
    PQclear(res); exit_nicely(conn);
}
PQclear(res);

res = PQexec(conn, "FETCH ALL in myportal");
if (PQresultStatus(res) != PGRES_TUPLES_OK)
{
    fprintf(stderr, "FETCH ALL failed: %s",
        PQerrorMessage(conn));
    PQclear(res); exit_nicely(conn);
}
```

Primjer: postgresQL - nastavak

Prihvati rezultata izvođenja naredbe:

```
/* first, print out the attribute names */

nFields = PQnfields(res);
for (i = 0; i < nFields; i++) printf("%-15s", PQfname(res, i));
printf("\n\n");

/* next, print out the rows */

for (i = 0; i < PQntuples(res); i++)
{
    for (j = 0; j < nFields; j++)
        printf("%-15s", PQgetvalue(res, i, j));
    printf("\n");
}
PQclear(res);
```

Primjer: postgresQL - nastavak

Završetak:

```
/* close the cursor */

    res = PQexec(conn, "CLOSE myportal");
    PQclear(res);

/* commit the transaction */

    res = PQexec(conn, "END");
    PQclear(res);

/* close the connection to the database and cleanup */

    PQfinish(conn);
    return 0;
} /* main end */
```

Primjer: postgresQL - nastavak

Kompajliranje:

```
racanalo> gcc -I/usr/include/postgresql/ prog.c -o prog.x -lpq
```

Izvođenje:

```
racanalo> ./prog.x
```

```
datname    datdba    encoding  datistemplate  dataallowconn  datlastsysoid  \\  
            datvacuumxid  datfrozenxid  datpath  
  
fakultet   100      8         f              t              16554          \\  
            7178          3221232651  
template1  1        8         t              t              16554          \\  
            7255          3221232728  
template0  1        8         t              f              16554          \\  
            49           49
```

Primjer: mono/C# (.NET) i mySql

Predradnje:

```
mysql> create table hello_world_test (  
    -> hello_id varchar(2),  
    -> hello_desc text);
```

Query OK, 0 rows affected (0.01 sec)

Upis podataka:

```
mysql> insert into hello_world_test (hello_id, hello_desc) values  
    ('HW', 'Hello World');
```

Query OK, 1 row affected (0.00 sec)

Primjer: mono/C# (.NET) i mySql - nastavak

Provjera:

```
mysql> select * from hello_world_test;
```

```
+-----+-----+
| hello_id | hello_desc |
+-----+-----+
| HW       | Hello World |
+-----+-----+
```

```
1 row in set (0.02 sec)
```

ili

```
mysql> select hello_desc from hello_world_test where hello_id = 'HW';
```

```
+-----+
| hello_desc |
+-----+
| Hello World |
+-----+
```

```
1 row in set (0.02 sec)
```

Primjer: mono/C# (.NET) i mySql - nastavak

Početak aplikacije:

```
// MySqlHelloWorld.cs
//
// On Windows, compile like:
//     mono C:/cygwin/home/MyHome/mono/install/bin/mcs.exe \
//         MySqlHelloWorld.cs \
//         -lib:C:/cygwin/home/MyHome/mono/install/lib \
//         -r System.Data.dll -r Mono.Data.MySql.dll
// On Linux, compile like:
//     mcs MySqlHelloWorld.cs -r System.Data -r Mono.Data.MySql
//
// $ mono MySqlHelloWorld.exe
// Result: Hello World
```

```
using System;
using System.Data;
using Mono.Data.MySql;
```

Primjer: mono/C# (.NET) i mySql - nastavak

Početni dio programa:

```
class MySqlHelloWorld {  
  
    static void Main(string[] args) {  
        string connectionString = "Database=test";  
        string sql = "SELECT hello_desc FROM hello_world_test " +  
                    "WHERE hello_id = 'HW'";  
    }  
}
```

Primjer: mono/C# (.NET) i mySql - nastavak

Povezivanje s bazom i zadavanje SQL naredbe:

```
IDbConnection dbcon = new MySqlConnection(); // connect

dbcon.ConnectionString = connectionString; // the connection string

dbcon.Open(); // open a connection

IDbCommand dbcmd = dbcon.CreateCommand(); // create command

dbcmd.CommandText = sql; // set the command's SQL

IDataReader reader = dbcmd.ExecuteReader(); // execute the SQL
```

Primjer: mono/C# (.NET) i mySql - nastavak

Rezultati i kraj:

```
// read the first row in the result

if(reader.Read()) {
    Console.WriteLine("Result: " + reader["hello_desc"]);
}
else {
    Console.WriteLine("Row not found.");
}

reader.Close(); // clean up
reader = null;
dbcmd.Dispose();
dbcmd = null;
dbcon.Close();
dbcon = null;
}
}
```

Primjedbe:

Svaka baza podataka i platforma ima svoje aplikacijsko sučelje (API).

API je biblioteka programa / potprograma / procedura koji se koriste u aplikaciji kao funkcije ili naredbe za povezivanje s DBMSom.

- ⑥ Primjer za PostgreSQL: PQsetdb, PQstatus, PQexec, PQclear,

Napomena: Loša strana je da za svaku bazu podataka treba iznova učiti biblioteku programa.

Primjer: Perl i PostgreSQL

```
#!/usr/bin/perl

use Pg;

# niz izraza: opcija=vrijednost opcija1=vrijednost1 ....
# opcije: host, port, options, tty, dbname, user, password
$conn = Pg::connectdb("dbname=fakultet");

# Postavljanje upita
$sql = "SELECT ime, prezime FROM student WHERE indeks='F-2345'";

$result = $conn->exec($sql);

while (@row = $result->fetchrow)
{
    $ime = $row[0];    $prezime = $row[1];
    print "ime: $ime, prezime: $prezime\n";
}
```

Primjer: Perl i PostgreSQL - nastavak



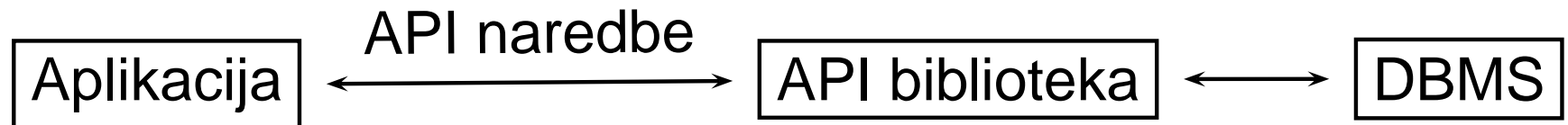
Izvođenje:

```
racunalo> ./primjer.pl  
ime: IVAN, prezime: ČOTI
```

Univerzalni API

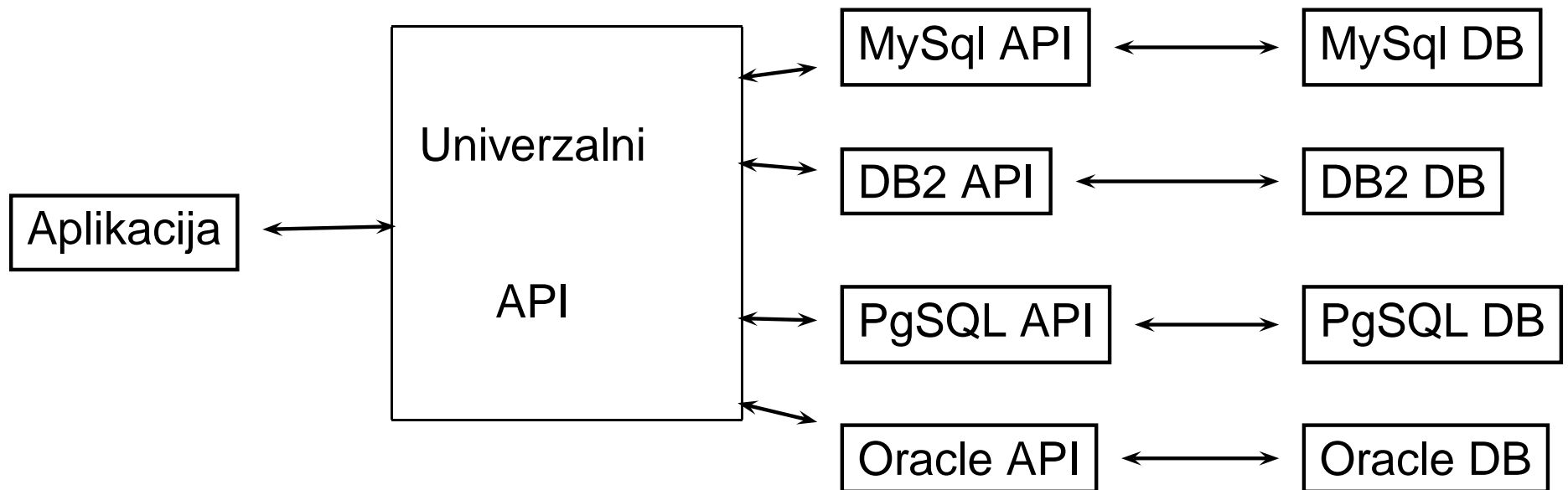
Uvodi se posebna biblioteka funkcija koja *stoji* između aplikacije i biblioteke DB API-a i kojom se uvodi univerzalni skup naredbi, isti za sve vrste baza podataka.

Standardni pristup:



Univerzalni API - princip

Univerzalni API:



Primjer za univerzalni API

```
#!/usr/bin/perl

use DBI;
$conn = DBI->connect("dbi:Pg:dbname=fakultet");

# Postavljanje upita
$sql = "SELECT ime, prezime FROM student WHERE indeks='F-2345'";

$res = $conn -> prepare($sql); $res -> execute();

while (@row = $res->fetchrow_array)
{
    $ime = $row[0];
    $prezime = $row[1];
    print "ime: $ime, prezime: $prezime\n";
}
$conn -> disconnect;
```

Univerzalni API

Razlika između različitih baza podataka dolazi u drugačijem izrazu za povezivanje s bazom podataka.

- ⑥ mySql: "dbi:mysql:dbname=fakultet",
- ⑥ miniSQL: "dbi:mSQL:dbname=fakultet",
- ⑥ Oraclea: "dbi:Oracle:dbname=fakultet"
- ⑥ ...

Aplikacijski programi trebaju minimalne izmjene pri prelasku s jedne baze podataka na drugu.

Primjer: Java programiranje

```
import java.sql.*;
public class ProfFizInf {
    public static void main(String args[]) {
        String url = "jdbc:odbc:PostgreSQL Fakultet";
String myLogin = "ivo";
String myPassword = "";
        Connection con;
        String traziString;
        traziString = "SELECT ime,prezime FROM student " +
                    "WHERE godina=3 AND smjer='pfi'";
        Statement stmt;

        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        } catch (java.lang.ClassNotFoundException e) {
            System.err.print("ClassNotFoundException: ");
            System.err.println(e.getMessage());
        }
    }
}
```

Primjer: Java programiranje - nastavak

```
try {
    con = DriverManager.getConnection(url, myLogin, myPassword);
    stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(traziString);
    while (rs.next()) {
        String i = rs.getString("ime");
        String p = rs.getString("prezime");
        System.out.println(i + " " + p);
    }

    stmt.close();
    con.close();

} catch (SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
}
```

Primjer: Java programiranje - nastavak



⑥ Unix/linux

```
racunalo> javac ProfFizInf.java
```

```
racunalo> java ProfFizInf
```

⑥ MS Windows

```
C:> javac ProfFizInf.java
```

```
C:> java ProfFizInf
```

Tutorial:

<http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html>

Java programiranje - PostgreSQL driver



⑥ `url = "jdbc:odbc:PostgreSQL Fakultet"`
zamjeniti s `"jdbc:postgresql:fakultet"` ili
`"jdbc:postgresql://localhost/fakultet"`

⑥

`Class.forName("sun.jdbc.odbc.JdbcOdbcDriver")`
zamjeniti s

`Class.forName("org.postgresql.Driver")`