

# Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

3. samostatná úloha z předmětu

## **Počítačové sítě**

RMI

Jiří Kučera, A08N0092P

kalwi@students.zcu.cz

23. 6. 2009

## Zadání třetí samostatné úlohy – RMI

Sestavte program pro vzdálené volání metody (RMI) v Javě, který bude jako parametry přenášet netriviální datové struktury. Vlastní úlohu pro výpočet si zvolte dle vlastního uvážení (např. pro zadané vrcholy trojúhelníka vypočtete jeho obvod, obsah a polohu těžiště).

Úlohu řeší student samostatně. Výsledkem bude funkční dvojice programů (server a klient) a referát v elektronické podobě. Struktura adresářů by měla být následující: v kořenovém adresáři soubor \*.pdf s referátem, případně READ.ME a dále podadresáře se zdrojovými kódy klienta a serveru. Mezi zdrojovými kódy budou i dávkové soubory pro překlad a sestavení.

Mnou implementované zadání vypočítá determinant matic 3x3, dále jejich součet a součin.

## Programátorská dokumentace

Jako referenční aplikaci při psaní programu jsem použil vzorový příklad ze stránky:

<http://www.kiv.zcu.cz/~ledvina/vyuka/PDS/index.htm#Java>

Aplikace je členěna do čtyř tříd:

*IRmiServer*

Rozhraní definující RMI metody volané klientem. Toto rozhraní implementuje třída se serverovou aplikací.

*RmiServer*

Serverová aplikace. Implementuje RMI metody, které volá klient.

*RmiClient*

Klientská aplikace. Vytváří náhodné matice, volá RMI metody serveru a vypisuje získané výsledky.

*Matrix3*

Reprezentace matice o rozměrech 3x3.

Součástí aplikace jsou také skripty pro kompilaci a spouštění aplikace (podrobněji jsou popsány v uživatelské dokumentaci):

*compile.sh*

*server.sh*

*client.sh*

## Uživatelská dokumentace

Aplikace je platformně nezávislá. Pro kompilaci a spuštění jsou k dispozici skripty psané pro Linuxový shell bash.

## Kompilace

Aplikace již obsahuje přeložené zdrojové kódy a spouštěcí skript, nicméně opětovnou kompilaci lze provést skriptem `compile.sh`, který smaže staré binární soubory (existují-li) a znovu zkompiluje aplikaci.

## Spuštění a používání aplikace

Program sestává ze serverové a klientské aplikace. Pro přehlednější orientaci ve vstupech a výstupech je vhodnější spustit obě aplikace v novém terminálovém okně. Z demonstračních důvodů klientská aplikace generuje náhodná data (ruční zadávání by bylo zdlouhavé) a připojuje se pouze na `localhost`, pro připojení k jinému serveru je třeba změnit příslušnou proměnnou v klientské aplikaci a program překompilovat.

Nejprve je třeba spustit server:

```
./server.sh
```

*Pozn.: Zároveň se serverem se spouští i registr RMI (rmiregistry). Je-li již registr spuštěn, dojde při jeho opakovaném spuštění k výjimce, činnost serveru to ale neohrozí.*

Poté je možno spustit klient:

```
./client.sh
```

Vstupem jsou dvě matice o rozměrech 3x3 a, jak již bylo řečeno, jsou generovány náhodně. Výstupem je determinant obou matic a dále jejich součet a součin.

## Závěr

Aplikaci jsem odladil na školním serveru `ul403ds11-kiv.fav.zcu.cz`, na kterém je nainstalován operační systém Debian 4.1.1-19 (jádro Linux 2.6.18.2, JDK 1.5.0\_09-b03).

Přesné určení doby trvání vývoje aplikace je zpětně obtížný. Odhaduji, že práce trvala včetně lazení a psaní dokumentace okolo dvou dnů (po 12 – 16 hod.).