

Vývoj rozhraní pro vzdálené ovládání systému mainframe

Radek Vláčil

Katedra matematiky
Fakulta jaderná a fyzikálně inženýrská
České vysoké učení technické v Praze
Školitel: Ing. Tomáš Oberhuber

Diplomová práce, 2009

- 1 Úvod
- 2 Cíl
- 3 Řešení
- 4 Popis funkce
- 5 Shrnutí výsledků

Obsah

- 1 Úvod
- 2 Cíl
- 3 Řešení
- 4 Popis funkce
- 5 Shrnutí výsledků

Mainframe

Historie

- Používají se déle než 40 let
- Vyvíjí se plynule vedle PC



Obrázek: System z10 Mainframe

Klady a zápory

Výhody

- Velmi výkonné
- Vysoce spolehlivé
- Zpětně kompatibilní

Nevýhody

- Zpětná kompatibilita
- Většina kódu psaná v assembleru
- Nepřívětivé uživatelské prostředí

Klady a zápory

Výhody

- Velmi výkonné
- Vysoce spolehlivé
- Zpětně kompatibilní

Nevýhody

- Zpětná kompatibilita
- Většina kódu psaná v assembleru
- Nepřívětivé uživatelské prostředí

Obsah

- 1 Úvod
- 2 Cíl**
- 3 Řešení
- 4 Popis funkce
- 5 Shrnutí výsledků

Prostředí ISPF

ISPF/PDF PRIMARY OPTION MENU

```

-----
OPTION  ===> █

      0  ISPF PARMs  - Specify terminal and user parameters      USERID  - VLARA80
      1  BROWSE     - Display source data or output listings    TIME     - 21:01
      2  EDIT       - Create or change source data              TERMINAL - 3278
      3  UTILITIES  - Perform utility functions                 PF KEYS  - 12
      4  FOREGROUND - Invoke language processors in foreground   SYSID   - XE44
      5  BATCH      - Submit job for language processing
      6  COMMAND    - Enter TSO command or CLIST
      7  DIALOG TEST - Perform dialog testing
      9  IBM PRODUCTS- Additional IBM program development products
      I  IPCS       - Interactive Problem Control System
      0  SDSF       - SDSF
      Z  z/XDC      - Interactive Debugging with z/XDC 1.8
      X  EXIT       - Terminate ISPF Using log and list defaults

Enter END command to terminate ISPF.
  
```

F1=HELP

F2=START

F3=END

F4=RETURN

F5=RFIND

F6=RCHANGE

F7=UP

F8=DOWN

F9=SWAP

F10=LEFT

F11=RIGHT

F12=RETRIEVE

4B █

A44XX002

002/015

Prostředí Eclipse

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with the test class `NbBundleTest.java` selected. The main editor shows the source code of `NbBundleTest.java`, which includes imports, annotations, and test methods. The Outline view on the right shows the class structure. The Console at the bottom indicates that the test run was successful, finishing after 0.129 seconds with 2/2 runs, 0 errors, and 0 failures.

```

27 import com.aramco.powers2.ui.NbBundle;
28
29 /**
30  * Tests the behavior of utility class NbBundle.
31  * Tests need to run against the background of a known set of objects.
32  * This set of objects is called a test fixture. (Refer to http://www.junit.org)
33  */
34  * @author Guanglin Du (dugl@petrochina.com.cn), Software Engineering Center, RIPED, PetroChina
35  */
36 public class NbBundleTest {
37
38     /**
39      * Uses the Bundle.properties to test NbBundle's behavior.
40      */
41     @Test
42     public void testExistingResource() {
43         String s1 = NbBundle.getMessage(ProjectView.class, "add_new_pvt_sat");
44         assertEquals("Add New PVT or SAT table", s1);
45     }
46
47     /**
48      * Uses the Bundle.properties to test NbBundle's behavior.
49      */
50     @Test
51     public void testNonExistingResource() {
52         String s1 = NbBundle.getMessage(ProjectView.class, "non-existing");
53         assertEquals("%non-existing", s1);
54     }
55
56     /**
57      * Method main to run this class directly.
58      * Can be run this way also on a command line:
59      * java org.junit.runner.JUnit4Core samples.SimpleTestFixture
60      */
61     public static void main(String args[]) {
62         JUnitCore.main("com.aramco.powers2.ui.test.NbBundleTest");
63     }
64 }
65

```

Console Output:

```

Finished after 0.129 seconds
Runs: 2/2      Errors: 0      Failures: 0

```

Obsah

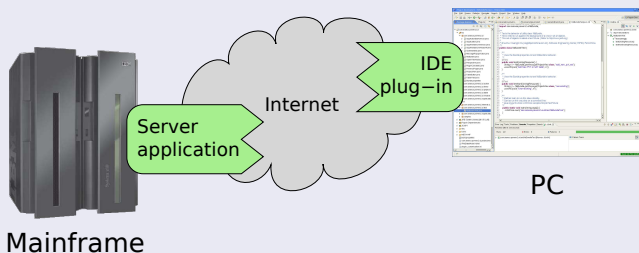
- 1 Úvod
- 2 Cíl
- 3 Řešení**
- 4 Popis funkce
- 5 Shrnutí výsledků

Obecně

Obecné požadavky

- Vyhnout se textovému rozhraní
- Poskytnout kompletní IDE
- Minimalizovat datový tok sítí
- Přenositelnost klienta

Obecný návrh

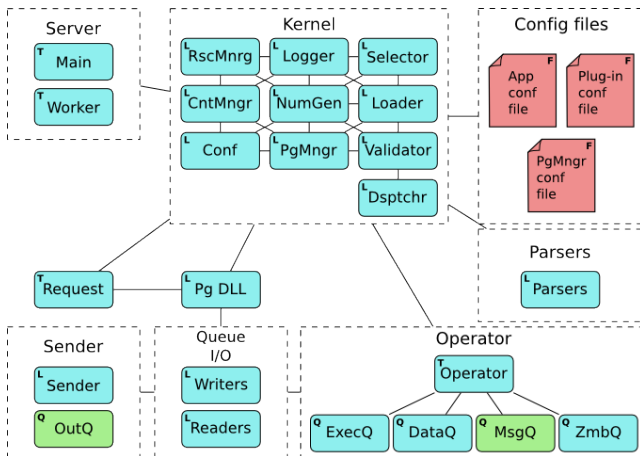


Server

Požadavky na server

- Programovací jazyk
 - C, kde to nepůjde C++ nebo HLASM
- Komunikace
 - TCP/IP sockety, poskytuje z/OS USS
- Souběžnost
 - POSIX thready, poskytuje z/OS USS
- Rozšiřitelnost
 - Použití dynamických knihoven pro funkce požadavků

Návrh serveru



Obsah

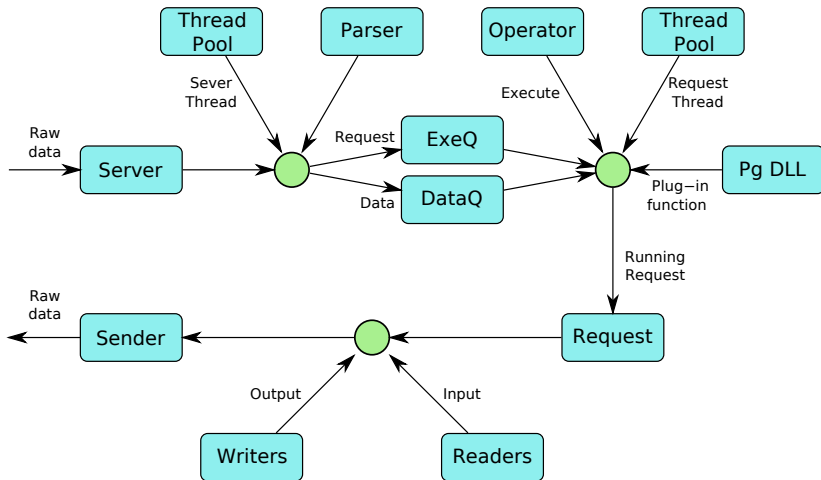
- 1 Úvod
- 2 Cíl
- 3 Řešení
- 4 Popis funkce**
- 5 Shrnutí výsledků

Příklad

Porovnání datasetů

- 1 Klient chce srovnat dataset, který má uložený na svém počítači, s datasetem na mainframe
- 2 Klient zašle požadavek o srovnání datasetů
- 3 Zašle jméno datasetu na mainframe, případně další atributy
- 4 Poté zasílá postupně jednotlivé bloky datasetu
- 5 Server odpovídá zda jsou stejné či ne
- 6 Pokud nejsou stejné, server zašle obsah rozdílného bloku

Funkce serveru



Obsah

- 1 Úvod
- 2 Cíl
- 3 Řešení
- 4 Popis funkce
- 5 Shrnutí výsledků**

Shrnutí

Výsledky

- Vytvoření návrhu serveru
- Implementace serveru
- Implementace testovacího plug-inu
- 10000 LOC