

Testování vzdáleného rozhraní pro mainframe

Martin Fous

Katedra matematiky
Fakulta jaderná a fyzikálně inženýrská
České vysoké učení technické
Školitel: Ing. Tomáš Oberhuber, Ph.D.

Bakalářská práce, 2011

Obsah

- 1 Úvod
- 2 Současný stav projektu
- 3 Testování
- 4 Shrnutí

Mainframe

- Platforma určená ke zpracování velkých objemů dat
- Více než 40ti letá historie



Obrázek: System z10 Mainframe

Pozitiva a negativa platformy mainframe

Výhody

- Spolehlivost
- Dostupnost
- Servisovatelnost
- Výkonnost
- Zpětná kompatibilita

Nevýhody

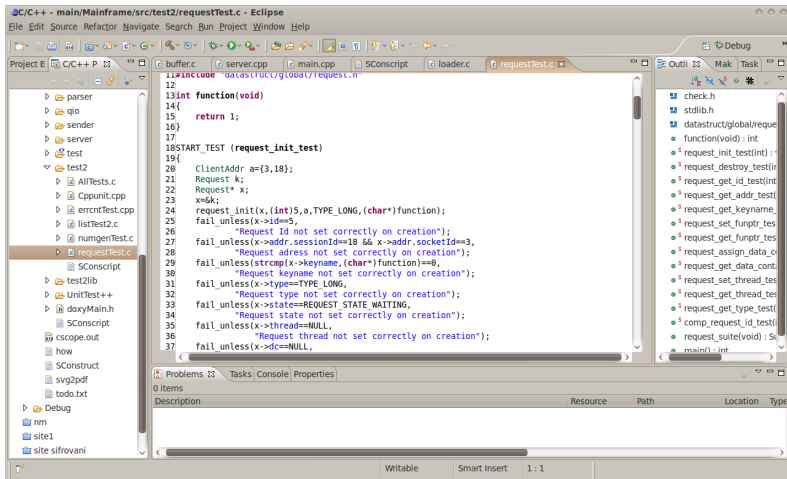
- Zpětná kompatibilita
- Nepřívětivé rozhraní a složitost konfigurace
- Mnoho kódu psaného v HLASM, COBOLu, atd.

Prostředí ISPF

```

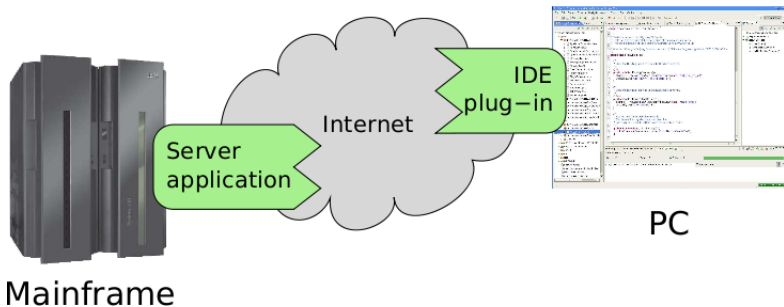
x3270-4 141.202.28.20
File Options
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW F0UMAB0.ASM.SOURCE(EXER10) - 01.03 Columns 00001 00000
Command ==> SCROLL ==> HALF
***** Top of Data *****
==MSD> Warning- The UNDO command is not available until you change
==MSD> your edit profile using the command RECOVERY ON.
000001 EXER10 *FILE
000002 *****
000003 * ===== Write a program which will copy entire input data set to *
000004 * to the output data set, use macros OPEN, CLOSE, GET, PUT, *
000005 * =====
000006 OPEN (INFILE,(INPUT))
000007 OPEN (OUTFILE,(OUTPUT))
000008 PUT OUTFILE,READING
000009 PUT OUTFILE,BELIM
000010 READLOOP CS OH
000011 GET INFILE,INREC
000012 MVI PRINTREC,C' '
000013 MVC PRINTREC,I(1,32),PRINTREC
000014 MVC PRNAME,INAME
000015 MVC PRENTSY,INCNTRY
000016 PACK PRSALARY,INSALARY
000017 * UNPK PRSALARY,PRSALARY
000018 MVC PRSALARY,C'00000020202068202020'
000019 ED PRSALARY,PRSALARY
000020 PUT OUTFILE,PRINTREC
000021 CP PRSALARY,PIMAXSLR
000022 BH HIGH
000023 B SKIP
000024 HIGH DS OH
000025 MVC PIMAXSLR,PRSALARY
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up F9=Down
F9=Swap F10=Left F11=Right F12=Cancel
Display Editer View Print Options Help
-----
SDSF OUTPUT DISPLAY F0UMAB0J 00006631 DSIO 181 LINE 160 COLUMNS 02- 181
COMMAND INPUT ==> SCROLL ==> HALF
00000A 5810 0000 00000 101** FREE DYNAMICALLY ACQUIRED SAVE AREA
102* E 1,8,(13) 01-ENDIT
103* FREEMAIN 8,LV=72,A=(1) 01-ENDIT
Page 5
Active Usings: EXER06,R12
Loc Dbrect Code Addr1 Addr2 Stat Source Statement HLRAS R5.0 2810/04/12 12.36
00000E 4100 0048 00000 106+ LA 8,72(0,0) LOAD LENGTH 02-FREEM
0000A2 4110 1000 00000 107+ LA 1,0(0,1) LOAD AREA ADDRESS 02-FREEM
0000A6 040A 00000 108+ SVC 10 ISSUE FREEMAIN SVC 02-FREEM
110** RESTORE CALLING PROGRAMS REGISTERS, SET RETURN CODE, AND EXIT
111* RETURN (14,12),T,RC=0 01-ENDIT
112* LM 14,12,I(1,12) 02-RETURN
0000AC 9601 00BF 0000F 115+ DI 15(13),X(0) SET RETURN INDICATION 02-RETURN
0000B8 41F0 0000 00000 116+ LA 15,0(0,0) LOAD RETURN CODE 02-RETURN
0000B4 07FE 00000 117* BR 14 RETURN 02-RETURN
118 *****
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
03  A44XK033 039/021
    
```

Prostředí Eclipse



Stav projektu

- Hotová implementace jádra serveru a testovacího pluginu
- Potřeba otestovat funkčnost napsaného kódu



Testování software

Definice

Testování software je souhrnný název pro postupy vedoucí k nalezení chyb v kódu, kontrole funkčnosti vzhledem ke specifikacím, porozumění omezením programu, obecně měření kvality softwaru.

Rozdělení podle stadia vývoje

- Jednotkové testování (Unit testing)
- Integrovaní testování (Integration testing)
- Systémové testování (System testing)
- Akceptační testování (Acceptance testing)
- Regresní testování (Regression testing)

Jednotkové testování

Definice

Jednotkový test je kód (obvykle procedura) volající jiný kus kódu a poté ověřuje správnost svých předpokladů. Když se předpoklad ukáže chybným, jednotkový test ohlásí chybu. Jednotkou myslíme proceduru nebo funkci.

Vlastnosti unit testu

- automatizovaný a opakovatelný
- snadno implementovatelný
- rychlý a spustitelný bez další přípravy

Nástroje usnadňující testování

- Testovací frameworky
- Nástroje sledující pokrytí (gcov)
- Nástroje sledující výkon (gprof)
- Nástroje kontrolující paměť (mpatrol)

Testovací frameworky

- Knihovny poskytující nástroje a usnadňující psaní testů
- Automatizace testů
- Frameworky pro C/C++ : Check, CppUnit, CuTest, UnitTest++

Mock a Stub objekty

- Umožňují izolaci testovaného objektu
- Nahrazují volání mimo testovanou jednotku, nedostupnou funkcionalitu (jiné objekty, dotazy do databáze,...)
- Stub = pomocný objekt, vrací předem definovanou hodnotu
- Mock = pomocný objekt, testuje správnost interakce testovaného objektu s jiným objektem

Výsledky

- Nastudování implementace serveru
- Porovnání frameworků pro C/C++
- Vyzkoušení dalších testovacích nástrojů
- Pokrytí vybraných datových struktur unit testy
- Odstranění několika chyb z těchto struktur

Děkuji za pozornost