

Testování komunikačního rozhraní pro mainframe

Richard Salač

Katedra softwarového inženýrství v ekonomii
Fakulta jaderná a fyzikálně inženýrská
České vysoké učení technické v Praze
Školitel: Ing. Tomáš Oberhuber, Ph.D.

Bakalářská práce, 9.2.2012

Obsah

- 1 Systémy mainframe
- 2 Stav projektu a cíle práce
- 3 Testování
- 4 Ukázka testů
- 5 Shrnutí

Mainframe

Základní fakta

- Dlouhá historie - od roku 1964 z/360
- 90% společností v žebříčku Fortune 1000 spoléhá na jejich služby



IBM z196 Mainframe

Klady a zápory

Výhody

- Velmi výkoné
- Vysoká spolehlivost (až 99,999%)
- Důraz na zpětnou kompatibilitu

Nevýhody

- Nepřívětivé textové rozhraní
- Stále je používán MVS Assembler, Cobol...
- Náročné na administraci

Klady a zápory

Výhody

- Velmi výkoné
- Vysoká spolehlivost (až 99,999%)
- Důraz na zpětnou kompatibilitu

Nevýhody

- Nepřívětivé textové rozhraní
- Stále je používán MVS Assembler, Cobol...
- Náročné na administraci

Terminál ISPF

```

Menu Utilities Compilers Options Status Help
-----nerdc-
                ISPF Primary Option Menu
Option ==> 2

0 Settings      Terminal and user parameters      User ID . : USERID
1 View          Display source data or listings    Time. . . : 15:39
2 Edit          Create or change source data       Terminal. : 3278
3 Utilities     Perform utility functions          Screen. . : 1
4 Foreground    Interactive language processing    Language. : ENGLISH
5 Batch         Submit job for language processing  Appl ID . : ISR
6 Command       Enter TSO or Workstation commands  TSO logon : IKJACCT1
7 Dialog Test   Perform dialog testing             TSO prefix: NER
8 LM Facility   Library administrator functions    System ID : NER1
9 IBM Products  IBM program development products  MUS acct. : -
10 SCLM         SW Configuration Library Manager   Release . : ISPF 4.5
11 Workplace   ISPF Object/Action Workplace
N NERDC        NERDC Installed Products

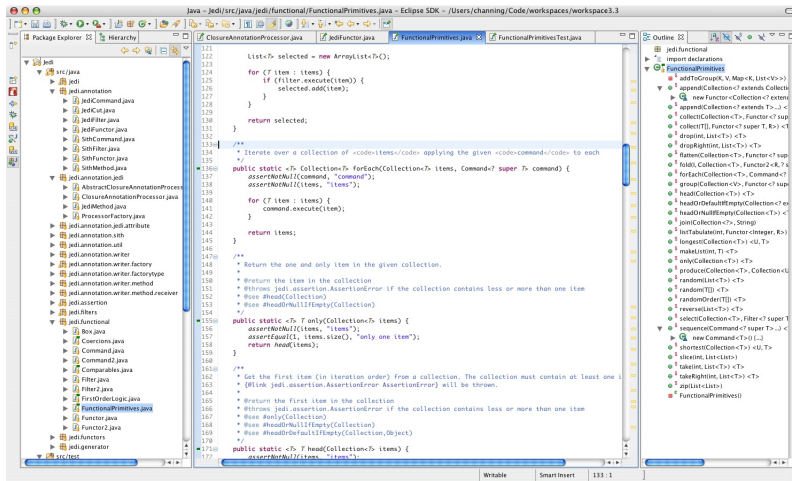
Enter X to Terminate using log/list defaults

F13=Help      F15=End      F16=Return   F17=RFind    F18=Rchange  F22=Left
F23=Right     F24=Cretrieve

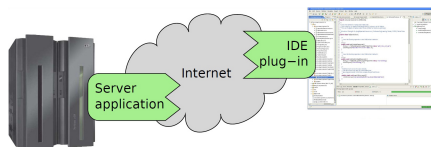
                                NUM                15:39:23 IBM-3278-2

```

Eclipse



Vzdálené ovládání mainframe



- Implementované jádro serverové části (R. Vláčil, 2009)
- Aplikace nebyla nikdy systematicky testována

Cíle práce

- Prostudujte základní postupy pro testování kódu
- Otestujte vybrané části existujícího kódu serveru

Testování

Testování jako prostředek ke zlepšení kvality softwaru

Testováním rozumíme cílené spouštění vyvíjeného programu nebo jeho části s předem definovanými a kontrolovanými provozními podmínkami a očekávaným chováním za účelem zlepšení jeho kvality.

Kvalita softwaru

FURPS+

- F - Funkčnost
- U - Použitelnost
- R - Spolehlivost
- P - Výkon
- S - Rozšiřitelnost
- Implementace
- Rozhraní
- Operační systémy
- Obchodní a právní aspekty

Vývojářské testování

Vývojářské testování

- Funkční testování (angl. black-box testing)
- Strukturní testování (angl. white-box testing)

Testování jednotek

Testování komponent

Testy integrace

Testování jednotek

- Testování elementárních stavebních prvků aplikace
- Jednotky jsou testovány oddělené od ostatních součástí
- Cílem je odstranit chyby na nejnižší úrovni programu

Požadavky na testy

- Přenositelnost na mainframe
- Rychlá dostupnost testů během dalšího vývoje
- Možnost automatizace testů
- Snadná správa testů a jednoduchý přehled výsledků

Testovací framework

- C/C++
- Dobrá a dostupná dokumentace
- Dostupnost pod licencí svobodného softwaru
- CppUnit

Příklad - Testovaná funkce

Inicializace kruhové fronty se zarážkou

- `int buffer_init(Buffer *thee, int n)`
- vrací 1 při úspěchu | 0 jinak

Testování - analýza okrajových hodnot

- inicializace fronty o délce 1
- inicializace fronty o délce 0
- inicializace fronty o délce -1

Příklad - Výsledky testů

Negativní test

- alokace fronty o délce 1 proběhne v pořádku
 - návratová hodnota 1

Pozitivní test

- inicializace fronty o délce 0 proběhne „v pořádku“
 - návratová hodnota 1
- inicializace fronty o délce -1 „v pořádku“
 - návratová hodnota 1

Příklad - zdrojový kód

Chyba

```
int buffer_ init(Buffer *thee, int n) {  
thee->data = (char *) malloc(n+1);  
...  
return 1;}
```

Oprava

```
int buffer_ init(Buffer *thee, int n) {  
if(n<=0) return 0;  
thee->data = (char *) malloc(n+1);  
...  
return 1;}
```

Příklad - zdrojový kód

Chyba

```
int buffer_ init(Buffer *thee, int n) {  
thee->data = (char *) malloc(n+1);  
...  
return 1;}
```

Oprava

```
int buffer_ init(Buffer *thee, int n) {  
if(n<=0) return 0;  
thee->data = (char *) malloc(n+1);  
...  
return 1;}
```


Shrnutí

Výsledky

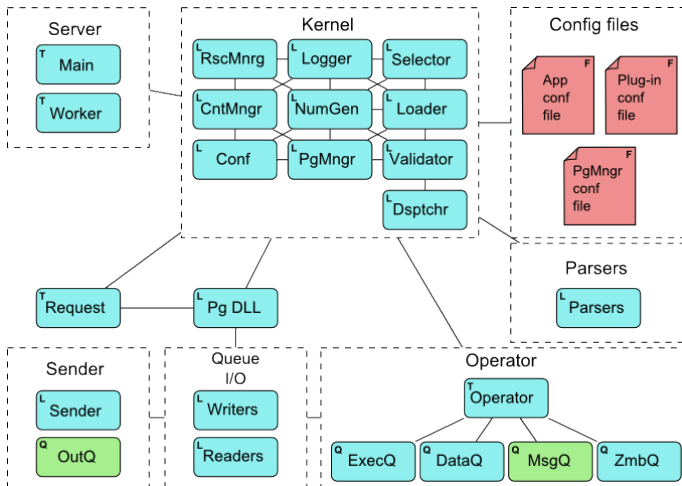
- Rešerše na téma způsoby a druhy testování
- Zvoleny a použity nástroje pro podporu testování přenositelné na mainframe
- Začlenění testů do procesu překladu
- Implementovány testy pro datové struktury

malloc(0)

C99 (článek 7.20.3)

The order and contiguity of storage allocated by successive calls to the `calloc`, `malloc`, and `realloc` functions is unspecified ... If the space cannot be allocated, a null pointer is returned. **If the size of the space requested is zero, the behavior is implementation- defined:** either a null pointer is returned, or the behavior is as if the size were some nonzero value, except that the returned pointer shall not be used to access an object.

Návrh aplikace



Implementace aplikace



Implementace aplikace

