

Vláknové programování

část I

Lukáš Hejmánek, Petr Holub

`{xhejtman, hopet}@ics.muni.cz`



Laboratoř pokročilých síťových technologií

PV192
2014-02-18

Přehled přednášky

Ada

Ada

“Regression testing?” What’s that? If it compiles, it is good, if it boots up it is perfect.

– *Linus Torvalds*

Compilable but broken code is even worse than working code.

– *Alan Cox*

during a bright moment on the
linux-kernel list

Ada:

Ideally, when compiles fine it also runs fine.

Ada

- Ada 83/95/2005
- jazyk orientovaný na spolehlivé aplikace: vojáci, letectví, ...
- WORM: write once read many
- silně typovaný jazyk
- podpora paralelismu
- enkapsulace, výjimky, objekty, správa paměti, atd.
- GNAT
 - volně dostupný překladač, frontend k GCC
- materiály na webu
 - (Annotated) Reference Manual
`http://www.adaic.org/standards/ada05.html`
 - `http://stwww.weizmann.ac.il/g-cs/benari/books/index.html#ase`
 - `http://en.wikibooks.org/wiki/Programming:Ada`
 - `http://www.adahome.com/Tutorials/Lovelace/lovelace.htm`
 - `http://www.pegasoft.ca/resources/boblap/book.html`

Na co se podívat

- Struktura balíčků, enkapsulace, pojmenování souborů
- Typový systém Ada, typy a podtypy, ukazatele (`access` VS. `access all`), generika, objekty, atributy proměnných
- Správa paměti
- Generické kontejnery
- Volby kompilace:

```
gnatmake hello.adb
```

VS.

```
gnatmake -gnatya -gnatyb -gnatyc -gnatye -gnatyf -gnatyi  
-gnatyk -gnatyl -gnatyn -gnatyp -gnatyr -gnatyt -g -gnato  
-gnatf -fstack-check hello.adb
```

Co v Adě neuděláte

... tedy co v ní neuděláte (*takhle*).

```
1 if(c = ntoab) {}
```

Co v Adě neuděláte

... tedy co v ní neuděláte (*takhle*).

```
1 send(to, from, count)
2 register short *to, *from;
3 register count;
4 {
5     register n = (count + 7) / 8;
6     switch(count % 8) {
7         case 0: do { *to = *from++;
8         case 7:     *to = *from++;
9         case 6:     *to = *from++;
10        case 5:     *to = *from++;
11        case 4:     *to = *from++;
12        case 3:     *to = *from++;
13        case 2:     *to = *from++;
14        case 1:     *to = *from++;
15                    } while(--n > 0);
16        }
17 }
```

Zdroj: https://en.wikipedia.org/wiki/Duff%27s_device

Kopírování pole do memory-mapped výstupu, nikoli kopie pole.

Co v Adě neuděláte

... tedy co v ní neuděláte (*takhle*).

```

1  if(c = ntoa(b)) {}
3  #define _ -F<00||--F-OO--;
   int F=00,OO=00;main(){F_OO();printf("%1.3f\n",4.*-F/OO/OO);}F_OO()
5  {
7      _ _ _ _ _
9      _ _ _ _ _
11     _ _ _ _ _
13    _ _ _ _ _
15   _ _ _ _ _
17  _ _ _ _ _
19 _ _ _ _ _
21  _ _ _ _ _
   }
```


Ada: kontejnery

```
2 with Ada.Containers.Ordered_Sets;
3 with Ada.Containers.Hashed_Sets;
4 with Ada.Containers.Hashed_Maps;
5
6 package Priklad_Kolekci is
7
8     function Unbounded_String_Hash (S : Unbounded_String)
9         return Hash_Type;
10
11     package UString_Int_Hash is new Ada.Containers.Hashed_Maps (
12         Key_Type => Unbounded_String,
13         Element_Type => Integer,
14         Hash => Unbounded_String_Hash,
15         Equivalent_Keys => "=");
16
17     package UStringSet is new Ada.Containers.Hashed_Sets (
18         Element_Type => Unbounded_String,
19         Hash => Unbounded_String_Hash,
20         Equivalent_Elements => "=");
```

Ada: kontejnery

```
1  procedure Output_Host_Set (HostSet : UStringSet.Set) is
2      use UStringSet;
3      Cur : UStringSet.Cursor;
4      Is_Tag_Open : Boolean := False;
5  begin
6      Cur := First (HostSet);
7      if Cur /= UStringSet.No_Element then
8          Open_Tag (Is_Tag_Open, "      <hosts>");
9          while Cur /= UStringSet.No_Element loop
10             Put_Line ("      <host>" & To
11 _String (Element (Cur)) & "</host>");
12             Cur := Next (Cur);
13         end loop;
14         Close_Tag (Is_Tag_Open, "      </hosts>"
15 );
16     end if;
17
18     exception
19         when An_Exception : others =>
20             Syslog (LOG_ERR, "State dump failed! " &
21 & Exception_Information (An_Exception));
22             Close_Tag (Is_Tag_Open, "      </hosts>"
23 );
24     end Output_Host_Set;
25
26 end Příklad_Kolekci
```

Instalace GNATu

- Instalační soubory jsou ve studijních materiálech
 - `/e1/1433/jaro2011/PV192/um/23046574/`
 - Linux x86 32 b
 - Linux x86 64 b
 - Windows
- Součásti:
 - GNAT GPL Ada Ada compiler, debugger, tools, libraries
 - AUnit gpl-2010 Ada unit testing framework
 - SPARK GPL SPARK Examiner and Simplifier (static prover)
 - GNATbench Ada plugins for the Eclipse/Workbench IDEs
 - PolyORB gpl-2010 CORBA and Ada distributed annex
 - AWS gpl-2.8.0 Ada web server library
 - ASIS gpl-2010 Library to create tools for Ada software
 - AJIS gpl-2010 Ada Java interfacing suite
 - GNATcoll gpl-2010 Reusable Ada components library
 - GtkAda gpl-2.14.1 Create modern native GUIs in Ada
 - XMLAda gpl-3.2.1 Library to process XML streams
 - Florist gpl-2010 Interface to POSIX libraries (pouze pro Linux)
 - Win32 Ada gpl-2010 Ada API to the Windows library (pouze pro Windows)

Instalace GNATu

- Instalace překladače
- Linux:
 - adacore-X86LNX-201102271805.zip:
`x86-linux/gnatgpl-gpl-2010/
gnat-gpl-2010-i686-gnu-linux-libc2.3-bin.tar.gz
gnat-2010-i686-gnu-linux-libc2.3-bin
./doinstall`
- Windows:
 - adacore-X86WIN-201102271212.zip:
`x86-windows/gnatgpl-gpl-2010/
gnat-gpl-2010-1-i686-pc-mingw32-bin.exe`

Úkol – Ada

- Naprogramujte ekvivalent `fgrep-u`
 - podpora voleb `-i`, `-r`, `-n`
 - smyslem je seznámit se s jazykem