Lisp

- Rodina jazykov vynájdená v roku 1959 Johnom McCarthym
- Pôvodne pouzívaný vedeckou komunitou na výskum AI
- Jazyk kde boli vynájdené základné veci v IT:
  - Funcionálne programovanie, garbage collector, virtuálny stroj, DSL
  - Read Eval Print Loop (alebo “shell” v Pythone)
- Dnes sú dominantné dve vetvy:
  - Common Lisp (navrhnutý hlavne pre veľké priemyselné aplikácie)
  - Scheme (navrhnutý pre výuku s dôrazom na minimálnost jazyka)
- Obidva sú používané na rozličné účely:
    - Výskum, vývoj, výuka
    - Systémové, end-user aplikácie a backendy služieb na internete

Common Lisp

- Nie je dogmatický, integruje rozličné štýly programovania:
  - Procedurálny
  - Objektový
  - Funkcionálny
  - Metaprogramovanie
  - Doménové špecifické jazyky (DSL)
- Má veľa implementácií, každá s inými výhodami:
  - Komerčné: Allegro, LispWorks, Scieneer, CormanCL, MCL
  - OSS: SBCL, CMUCL, CLisp, ECL, GCL
- Prominentná OSS implementácia je SBCL:
  - Production ready na vsetkých platformách (Windows menej)
  - Generuje rýchly kód, často porovnateľný s C
  - Kľúčoví vývojári sú sponzorovaní komerčnými firmami
  - Nový release každý mesiac

Common Lisp

- Mýty a povery:
  - Lisp je pomalý:
    - Bola pravda desiatky rokov späť
    - Veľa techník kompilácie a optimalizácie kódu bolo vynájdených pre riešenie pomalosti Lispu na historických strojoch
  - Lisp je velký:
    - Bola pravda desiatky rokov späť keď 1 MB bol obrovské množstvo dát
    - Dnes má SBCL (32-bit verzia) menej ako 30 MB a obsahuje:
      - Kvalitný kompilátor do rýchleho natívneho kódu
      - Virtuálny stroj (runtime)
      - Celý Common Lisp štandard (vyše 1000 funkcií a makier)
      - Veľa rozšírení (natívny multithreading, interface s C, ...)

Common Lisp

- Mýty a povery:
  - Lisp nikto nepoužíva:
    - Nebola pravda nikdy
    - Životaschopnosť firiem poskytujúcich Common Lisp nástroje je toho dôkaz
    - Firmy častokrát:
      - Používanie Lispu nezmieňujú – je to ich “secret weapon” a výhoda pred konkurenciou
      - Používajú Lisp na rapídne vytvorenie prototypu ktorý je potom prepísaný do iných jazykov
      - Používajú Lisp a je to o nich známe: www.pchristensen.com/blog/lisp-companies
  - Neexistujú knižnice:
    - 10 rokov späť existovalo veľa vedeckých knižníc ale málo knižníc na aplikačný vývoj
    - Dnes existuje veľa kvalitných knižníc

Common Lisp

- Ideálny nástroj na tieto problémy:
  - Vývoj podľa vágne/nepresne definovaných požiadaviek
  - Vývoj podľa požiadaviek ktoré sa často menia
  - Načatie ťažkého problému implementovaním jeho časti iteratívne – zospodu (opak “vodopád” metódy)
  - Vývoj veľkých a komplexných aplikacií
- Prečo je to tak?
  - Jazyk ruší hranicu medzi kompiláciou a behom programu – je normálne za behu aplikácie kompilovať vygenerovaný kód a počas kompilácie volať ľubovolné funkcie a externé programy
  - Jazyk je veľmi “plastický” - je jednoduché aj veľkú aplikáciu “rearchitektovať” (úplne prestaviť) v krátkom čase
  - Jazyk nekladie žiadne byrokratické prekážky do práce programátora (zapúzdrenie, private/public, ...)

Common Lisp

- Prečo je to tak?:
  - Jazyk nevnucuje programátorovi konkrétnu metodiku ako riešiť problém
  - Všetko čo je zadefinované je možné vždy (aj počas behu aplikácie) zobraziť, zmeniť, zrušiť
  - REPL (shell) je veľmi mocný interaktívny nástroj a vždy prístupný
  - Možnosť uložiť celkový stav “sveta” ako obraz na disku a jeho nahratie a pokračovanie od miesta uloženia
  - Jazyk kombinuje 2 veci každú z iného sveta:
    - 1. rýchlosť výsledného kódu (bez upísanosti a byrokracie) z C
    - 2. dynamickosť vývoja (bez pomalosti výsledneho programu z Pythonu/Ruby)

Common Lisp

- Prečo vypadá ako vypadá – alebo prečo je tam toľko ((((())))):
  - Program a dáta sú reprezentované rovnakou notáciou
  - Absentujúca syntax a program a dáta písané jednotnou formou umožňujú jednoduchú manipuláciu programov Lispom samotným:
    - Umožňuje pracovať na META úrovni (metaprogramovanie):
      - Programovanie ktorého výsledok je ďalší (iný) program:
        - V praxi to znamená naprogramovanie funkcie ktorá ako výsledok vráti program ako dáta, tento program je potom možno ďalej skompilovat ako bežný kód napísaný programátorom
        - V Lispe bežná prax, a veľmi mocný nástroj
        - Umožňuje vytváranie DSL uz vyše 40 rokov
        - Umožňuje rozšíriť jazyk o nové vlastnosti ktoré by normálne museli byť naprogramované tvorcom jazyka alebo kompilátora (lazy evaluation, perzistencia objektov, unifikácia ako v prologu, ...)

Common Lisp

- Zaujímavé OSS projekty:
  - Movitz – Common Lisp priamo na hardware, OS v Lispe
  - Climacs – Emacs klon
  - Maxima – matematický software
  - Stumpwm – X Window manager
  - Hunchentoot – web server
  - Weblocks – web framework
  - ACL2 – theorem prover
  - PWGL – vizuálny jazyk na kompozíciu hudby
  - Acclaim – prezentačný software
  - Cells – reaktívne programovanie
  - Cl-ppcre – regulárne výrazy (rýchlejsie ako v C)
  - CLPython – Python implementovaný v Lispe

Common Lisp

- Zaujímavé OSS projekty:
  - CXML – XML procesor s XSLT a XPATH
  - Drakma – web klient
  - Elephant – perzistentná objektová databáza
  - Jnil – preklad Javy do Lispu
  - Parenscript – Lisp syntax nad Javascriptom
  - Spray – raytracer
  - Gsharp – interaktívny zápis nôt
  - Fucc – univerzálny parser generátor
  - Cl-xmpp – jabber protokol
  - Linj – Lisp jazyk prekladaný do Javy
- Mnoho ďalších tu: www.common-lisp.net, www.cliki.net, www.cl-user.net

Common Lisp

- Komunita:
  - Vekovo veľmi rozmanitá – od 15 do 80 rokov
  - Združuje sa na news://comp.lang.lisp, www.lispforum.com, a na irc.freenode.net #lisp
- Novinky o svete Lispu:
  - http://planet.lisp.org
  - http://planet.sbcl.org
  - http://www.reddit.com/r/lisp

Happy hacking!! ;-)

- Moje veci v Lispe:
  - www.neuroarena.com - realtime stratégia vo web browseri (backend v Lisp/SBCL, frontend Flash, management server Erlang) (nie OSS zatiaľ)
  - http://common-lisp.net/project/patg - (OSS) graph rewriting engine
  - http://github.com/ks/X.LET-STAR - (OSS) rozšíriteľný “binder” - utilitka pre vývojarov
  - Ďaľšie OSS veci čakajúce na vyleštenie...

karol.skocik@gmail.com