

Práca so súbormi v PLT-Scheme 3xx

RNDr. Radovan Palík

9. februára 2007

Abstrakt

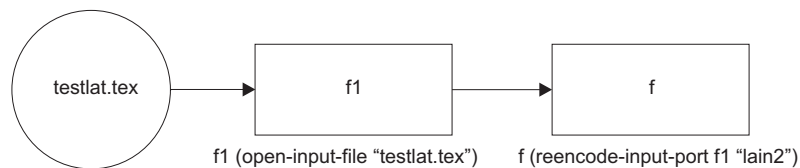
Ukážka práce so súbormi s národným kódovaním v PLT-Scheme v 3xx.

1 Úvod

Systém PLT-Scheme od verzie 299 pracuje vnútorne v kódovaní Unicode. Je to spôsobené prevodom celého systému pod toto kódovanie. Prináša to síce mnohé výhody, ale aj komplikácie pri spracovaní súborov v iných kódovaniach, keďže dnešné operačné systémy na Slovensku stále používajú predovšetkým *CP1250* alebo *ISO 8859-2*.

2 Reencoder

Riešením je použitie prekódovača — *reencodera*. Ten sa napája na port pracujúci s pôvodným súborom a prevádza príslušnú konverziu medzi národným kódovaním a Unicode.



Reencoder nepatrí medzi základné procedúry jazyka Scheme, ale je súčasťou knižnice *port.ss*. Jej zavedenie sa vykoná príkazom (*require (lib "port.ss")*).

Nech súbor `testlat.tex`, uložený na disku, je vo formáte *latin2*. Pri jeho otvorení pomocou

```
(define f1 (open-input-file "testlat.txt" 'text))
```

systém predpokladá, že je v kódovaní *UTF-8*. Znaký anglickej abecedy by síce boli v poriadku, no národné znaky by sa zobrazovali ako „?“. Na port *f1* tak musíme použiť *reencoder*

```
(define f (reencode-input-port f "latin2"))
```

V ňom uvedieme port, na ktorý sa má pripojiť a v akom kódovaní tento port poskytuje znaky. Z takto otvoreného portu *f* môžeme čítať znaky v kódovaní *UTF-8*, s ktorým pracuje Scheme.

V prípade zápisu do súboru (povedzme v kódovaní *CP1250*), je postup rovnaký

```
(define g1 (open-output-file "testwin.tex" 'replace 'text))
(define g (reencode-output-port g1 "CP1250"))
```

Ak budeme teraz v Scheme zapisovať do portu *g*, súbor `testwin.tex` bude na disku uložený v kódovej stránke *CP1250*.

Pri ukončení práce s portom (napríklad *g*) musíme uzavrieť aj pôvodný port *g1* a to v opačnom poradí, ako pri ich otvorení

```
(close-output-port g)
(close-output-port g1)
```

3 Príklad

Uveďme si teraz príklad konvertoru, ktorý vykoná prevod súboru z kódovania *latin2* do *CP1250*.

```
(require (lib "port.ss"))
```

```
(define (konvertuj vstup vystup)
  (let* ((f1 (open-input-file vstup 'text)) ;;vstup
         (f (reencode-input-port f1 "latin2"))
         (g1 (open-output-file vystup 'replace 'text)) ;;vystup
         (g (reencode-output-port g1 "CP1250")))
    (let loop ((a (read-line f)))          ;prave nacistany riadok
      ;ak je koniec vstupneho suboru?
      (if (eof-object? a)
          ;-> uzatvor vstupny aj vystupny subor
          [begin (close-input-port f)
                 (close-input-port f1)
                 (close-output-port g)
                 (close-output-port g1)]
          ;inak zapis nacistany riadok v 'a' do vystupneho suboru,
          ;precitaj dalsi zo vstupneho a v cykle loop vsetko opakuj
          [begin
            (display a g)
            (newline g)
            (loop (read-line f))])))
```