Lühikene juhatus $\LaTeX\,2_\varepsilon$kasutamiseks neile kes juba natuke asja tunnevad

Ott Toomet

14. november 2003. a.

Kokkuvõte

Siin on lühidalt kirjeldatud installerimise ning eesti ja vene keele kasutamise võimalusi ühes dokumendis, vanade (gooti-) kirjade kasutamist $\LaTeX\,2_\varepsilon$-s ja *pdf*-failide tegemist. Aluseks on *tetex 1.0.7* TEX-i pakett.

1 TEX-i globaalne ja lokaalne paigaldamine

Failide leidmiseks kasutab `tetex` süsteemi mida kutsutakse *kpathsearch*. Selle põmõte on lihtne ja kaval – selle asemel et otsida faile mööda ketast (mis on suhteliselt aeglane), koostatakse kõigist TEX-i failidest nimekiri ja otsitakse faili säält. See on palju kiirem viis.

Teiseks teeb *kpathsearch* installeerimise lihtsaks. Põhimõtteliselt võib uued failid installeerida ükskõik kuhu (siiski *kpathsearch* otsimispiirides), kui failid on nimekirjas olemas, siis nad säält ükskõik kust ka leitakse.

Globaalne installatsioon asub enamasti kataloogis `/usr/share/texmf` (sellel kataloogile viitan ma edaspidi kui `$TEXMF`). Lokaalne installatsioon võib asuda mitmes kohas:

- Globaalses TEX-i konfiguratsioonifailis failis `$TEXMF/web2c/texmf.cnf`. Sääl on umbes niisugused read:

  ```
  TEXMFMAIN = /usr/share/texmf
  TEXMFLOCAL = /usr/local/share/texmf
  HOMETEXMF = $HOME/texmf
  TEXMF = {!!$HOMETEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN}
  ```

  Ehk siis süsteemne installatsioon asub kataloogis `/usr/share/texmf`, lokaalne `/usr/local/share/texmf` ja koduinstallatsioon kataloogis `$HOME/texmf`.

- Kõiki eelpool toodud katalooge ja konfiguratsioonifaile saab ka keskkonnamuutujatega kohaldada.

Seega vaikimisi vaatab TEXkolme kohta vajalike failide otsimiseks. Oluline on jxrjekord – vaikimisi on too esiteks kodu-, siis lokaalne- ja lõpuks süsteemne kataloog. Niisuguse järjekorra mõte on et omainstalleeritud asjadel on alati eelis süsteemsete ees.

Uute pakettide, fontide või muude asjade paigaldamine käib nii:

1. Mõtle välja mis tasemele tahad asju paigaldada. Kui nood peaks olema kättesaadavad ka teistele kasutajatele, siis soovitan paigaldada lokaalselt. Kui ainult Sulle, siis koju. Süsteemset paigaldust ma üldiselt ei soovita, kui TEX-i uuendada, siis võivad muudetud süsteemsed failid kergesti kaduma minna. Vastava taseme juurkataloogile on edaspidi viidatud kui `$TEXMF`.

2. Tee vajalik juurkataloog (vaikimisi väärtuste korral siis kas `/usr/local/share/texmf` või `$HOME/texmf` (kui teda juba ei ole).

3. Tee vastav alamkataloog, et kuhu failid kopeerida. Soovitav järgida süsteemset kataloogipuud (kuigi *kpathsearch* puhul pole too tegelikult oluline, v.t. eespool).

4. Kopeeri failid sinna alamkataloogi.

5. Värskenda vastav *kpathsearch*-i nimekiri. See käib käsuga

   ```
   mktexlsr <\$TEXMF>
   ```

   kus kataloog on siis vastava installatsiooni asukoht (näiteks kodu puhul `$HOME/texmf`). `mktexlsr` käib ise kõik vajalikud alamkataloogid läbi.

6. Vajadusel tuleb kohandada ka konfiguratsioonifaile.

2 Poolitusreeglite häälestamine

LATEX-i poolitusreeglid on üldiselt kirjas igale keelele (kirjaviisile) vastavas failis, failid asuvad kataloogis `/usr/share/texmf/tex/generic/hyphen`. Vastav konfiguratsioonifail on `/usr/share/texmf/tex/generic/config/language.dat`. Lokaalselt eesti keele poolituse lisamiseks on vaja redigeerida `language.dat`-faili (ehk teha sellest lokaalne koopia) ning lisada TEX-i kataloogi eesti keele poolituse fail (leiab võrgus nime all `hyphen.estonian`). Sobivad kohad on `$TEXMF/tex/generic/config` konfiguratsiooni- ning `$TEXMF/tex/generic/hyphen` poolitusreeglite tarvis. Kui failid on paigaldatud, tuleb *kpathsearch*-i nimekiri uuendada (v.t. 1. osa).

Seejärel tuleb poolitusfailis märkida ära et milliste keelte poolitust soovitakse kasutada, valida mõnede keelte puhul meeldivad variandid (nt. briti või ameerika inglise keel) ning lisada märge eesti keele kohta. Näiteks soome keele rea ette võib lisada rea

```
estonian  hyphen.estonian
```

Seejärel tuleb tekitada uus LATEX-i baasformaat. Too käib käsuga

```
initex /usr/share/texmf/tex/latex/config/latex.ini
```

Oluline on et `initex` kirjutab uue formaadi (`latex.fmt`) ja logifaili (`latex.log`) käesolevasse kataloogi, seega saab `initex`-i kasutada ainult kataloogis kuhu Sul on kirjutamisõigus. `initex`-i loodud `latex.fmt` (soovitavalt ka `latex.log`, et hiljem teada mis täpselt tehti) tuleb seejärel kopeerida kuhugi TEX-i kataloogipuusse (sobiv koht on `$TEXMF/web2c`) ja seejärel värskendada *kpathsearch* nimekirji (kirjeldatud 1. osas).

3 Eesti ja vene keele kasutamine LaTeX-is

TeXon originaalkujul 7- või 8-bitine tekstitöötlussüsteem ja ei luba otseselt kasutada unicode 16 või 32 bitiseid sümboleid, mis toetavad korraga erinevaid keeli (TeX-i laiendatud variant *omega*, mis tuleb *tetex*-iga kaasa, on unikoodi põhine). Seega on vajalik 8-bitistesse märkidesse mahutada erinevate keelte sümbolid. Korraga on võimalik ühtses märgisüsteemis kasutada näiteks ladina- ja vene tähestikku aga siis ilma täpitähtedeta (täpitahti saab ikkagi teha „originaalkujul", nii nagu 7-bitise sisendiga TeX-is).

3.1 Üksikud venekeele katked eestikeelses tekstis

Et LaTeXteaks, milline tegelik kirjakuju tuleb vastavusse panna 8-bitiste koodidega, tuleb kasutada pakette `\inputenc`, `\fontenc` ja `\babel`. Kui soovid kasutada korraga vene- ja eesti keelt koos täpitähtedega, tuleb preambulasse (enne `\begin{document}`-i) kirjutada nii:

```
\usepackage[OT2,T1]{fontenc}
\usepackage[russian,estonian]{babel}
```

Siin `T1` tähendab „uut" LaTeX-i koodingut, mis vastab enam-vähem eesti keele (ja teiste ladina tähestikuga keelte) standardile, ning `OT2` on „vana" vene kodeering, mis paneb ladina tähtedele vastavusse vastavad vene tähed. Niimoodi et *L*-le vastab Л ja umbes nii (täpsemalt vt. tabel 1[1]). Siinkohal on oluline teada, et vaikimisi kasutatakse toda keelt, mis on käsus `\usepackage[...]{babel}` nurksulgudes viimane. Praeguse meetodi korral on mõistlik kasutada viimasena eesti keelt, kuna siin kirjeldatud meetod sobib ainult lühikeste venekeelsete katkete tarvis. Pika venekeelse jutu tarvis on parem kasutada koodingut *T2A* (vt. osa 3.2).

Vene keele sisselülitamiseks pakub *babel* kolm võimalust. Kõik võimalused lülitavad ümber nii kirjakuju kui ka poolitusreeglid (juhul kui vene poolitus on sisse lülitatud). Esiteks võib paar venekeelset sõna sisestada niimoodi:

```
\foreignlanguage{russian}{Kopeek\cyrhrdsn}
```

Копеекъ

Teiseks on võimalik kasutada käsku `\selectlanguage`. Näiteks nii:

```
\selectlanguage{russian}
po-russki
\selectlanguage{estonian}
ja jälle maakeeles
```

по-русски ja jälle maakeeles

Kolmandaks on võimalik kasutada keskkonda *otherlanguage*:

[1] Kooditabel põhineb paljuski Matthias Ellerti vastaval tabelil. Aitäh.

```
\begin{otherlanguage}{russian}
  Se\U{i}chas dlinny\U{i} primer po-russki
  \begin{verse}
    Beleet parus odinoki\u{i}\\
    v tumane morya golubom,

    chto ishchet on v strane daleko\u{i},\\
    chto kinul on krayu rodnom...
  \end{verse}
\end{otherlanguage}
```

Сейчас длинный пример по-русски

Белеет парус одинокий
в тумане моря голубом,

что ищет он в стране далекой,
что кинул он краю родном...

Kõiki meetodeid on võimalik makrodega lühendada. Näiteks võib preambulasse panna niisugused definitsioonid:

```
\newcommand{\textrus}[1]{\foreignlanguage{russian}{#1}}
\newcommand{\rus}{\selectlanguage{russian}}
\newcommand{\est}{\selectlanguage{estonian}}
\newenvironment{russian}
  {\begin{otherlanguage}{russian}}
    {\end{otherlanguage}}
```

Nüüd võib mõne venekeelse sõna tekitada käsuga

```
\textrus{Nu zayac, pogodi!}
```

Ну заяц, погоди!

Ja pikema lõigu niiviisi:

```
\begin{russian}
  Zhil-byl os\"{e}l.  Kogda on sostarilsya, khozyain stal
  durno s nim obrashchat{\cyrsftsn}sya...
\end{russian}
```

Жил-был осёл. Когда он состарился, хозяин
стал дурно с ним обращаться...

Kõigil kolmel meetodil on omad eelised. Esimene meetod sobib eeskätt paari sõna või üksikute tähtede teksti kirjutamiseks. Teine meetod sobib pikkadeks lõikudeks (terve peatükk vms.) ja viimane meetod millekski vahepäälseks, nagu näiteks paar lõiku, üks luuletus vms.

Eraldi tuleb märkida niisuguste vene märkide kasutamist mis antakse latehhi käsuga. Sõnavahe on LaTeX-is kahetähenduslik – ta tähendab ühest küljest sõnavahe, aga teisest küljest ka käsu lõppu. Käsu lõpu järel tähendab ta kokkuleppeliselt *ainult käsu lõppu aga mitte sõnavahet*! Seega kirjutades `Zdes\cyrsftsn d\cyrsftsn yavol!` on tulemuseks Здесьдьявол!. Kui soovitakse saada Здесь дьявол! tuleb kirjutada näiteks
`Zdes{\cyrsftsn} d\cyrsftsn yavol!`. Kellele see viis tüütu tundub võiks heita pilgu paketile `xspace`.

3.2 Pikk venekeelne tekst

Pikk venekeelne tekst on mõtekas kirjutada kohe mingis vene koodingus (näiteks *KOI-8*-s). Siis tuleb faili preambulasse kirjutada niisugused read:

```
\usepackage[koi-8]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[russian]{babel}
```

Ja ülejäänud tekst *KOI-8* koodingus, kus siis pole enam küsimust üksikute vene märkide tekitamiseks.

А	а	A	a	Rus., Ukr., Blr., Bul., Mak., Serb
Б	б	B	b	Rus., Ukr., Blr., Bul., Mak., Serb
В	в	V	v	Rus., Ukr., Blr., Bul., Mak., Serb
Г	г	G	g	Rus., Ukr., Blr., Bul., Mak., Serb
Ѓ	ѓ	\'{G}	\'{g}	Mak.
Д	д	D	d	Rus., Ukr., Blr., Bul., Mak., Serb
Ђ	ђ	Dj	dj	Serb
Е	е	E	e	Rus., Ukr., Blr., Bul., Mak., Serb
Ё	ё	\"{E}	\"{e}	Rus., Blr.
Є	є	\CYRIE	\cyrie	Ukr.
Ж	ж	Zh	zh	Rus., Ukr., Blr., Bul., Mak., Serb
З	з	Z	z	Rus., Ukr., Blr., Bul., Mak., Serb
Ѕ	ѕ	\CYRDZE	\cyrdze	Mak.
И	и	I	i	Rus., Ukr., Bul., Mak., Serb
І	і	\CYRII	\cyrii	Ukr., Blr., Russian until 1917
Ї	ї	\"{\CYRII}	\"{\i}	Ukr.
Й	й	\U{I}	\U{i}	Rus., Ukr., Blr., Bul.
Ј	ј	J	j	Mak., Serb
К	к	K	k	Rus., Ukr., Blr., Bul., Mak., Serb
Ќ	ќ	\'{K}	\'{k}	Mak.
Л	л	L	l	Rus., Ukr., Blr., Bul., Mak., Serb
Љ	љ	Lj	lj	Mak., Serb
М	м	M	m	Rus., Ukr., Blr., Bul., Mak., Serb
Н	н	N	n	Rus., Ukr., Blr., Bul., Mak., Serb
Њ	њ	Nj	nj	Mak., Serb
О	о	O	o	Rus., Ukr., Blr., Bul., Mak., Serb
П	п	P	p	Rus., Ukr., Blr., Bul., Mak., Serb
Р	р	R	r	Rus., Ukr., Blr., Bul., Mak., Serb
С	с	S	s	Rus., Ukr., Blr., Bul., Mak., Serb
Т	т	T	t	Rus., Ukr., Blr., Bul., Mak., Serb
Ћ	ћ	\CYRTSHE	\cyrtshe	Serb
У	у	U	u	Rus., Ukr., Blr., Bul., Mak., Serb
Ў	ў	\U{U}	\U{u}	Blr.
Ф	ф	F	f	Rus., Ukr., Blr., Bul., Mak., Serb
Х	х	Kh	kh	Rus., Ukr., Blr., Bul., Mak., Serb
Ц	ц	Ts	ts	Rus., Ukr., Blr., Bul., Mak., Serb
Ч	ч	Ch	ch	Rus., Ukr., Blr., Bul., Mak., Serb
Џ	џ	\CYRDZHE	\cyrdzhe	Mak., Serb
Ш	ш	Sh	sh	Rus., Ukr., Blr., Bul., Mak., Serb
Щ	щ	Shch	shch	Rus., Ukr., Bul.
Ъ	ъ	\CYRHRDSN	\cyrhrdsn	Rus., Bul.
Ы	ы	Y	y	Rus., Blr.
Ь	ь	\CYRSFTSN	\cyrsftsn	Rus., Ukr., Blr., Bul.
Э	э	\CYREREV	\cyrerev	Rus., Blr.
Ю	ю	Yu	yu	Rus., Ukr., Blr., Bul.
Я	я	Ya	ya	Rus., Ukr., Blr., Bul.
Ѵ	ѵ	\CYRIZH	\cyrizh	Russian until 1917
Ѣ	ѣ	\CYRYAT	\cyryat	Russian until 1917
Ѳ	ѳ	\CYRFITA	\cyrfita	Russian until 1917
№		NO		
«	»	<	>	

Tabel 1: OT2 olulisemad sümbolid

4 Vana stiili kirjad

Pakett `oldgerm` pakub kolme liiki vana tüüpi kirju: gooti-, swaabi- ja fraktuurkirja. Vastavate kirjade kasutamiseks tuleb päises öelda

```
\usepackage{oldgerm}
```

Vastavad käsud on antud tabelis 2.

kiri	lühike käsk	pikk käsk
Gooti	`\textgoth{}`	`\gothfamily`
Swaabi	`\textswab{}`	`\swabfamily`
Fraktuur	`\textfrak{}`	`\frakfamily`

Tabel 2: Vanad kirjad ja vastavad käsud

Täpitähti saab kasutada kas TEX-i standardkujul või valides õige sisendkooding. Nii võib kirjutada näiteks et

```
\frakfamily
wiis: s\"{o}av\"{a}e w\"{a}lja
```

wiis ſöaväe wälja

Õige sisendkoodingu tarvis tuleb preambulas öelda (näiteks kui kasutad *latin-1* koodingut):

```
\usepackage[latin1]{inputenc}
```

Nii saab kasutada tavalisi *latin-1* täpitähti. Ainuke häda on et õ-kriips kipub kole pikk olema. Veel tuleb arvestada et gooti kirjas pole õ-tähte! Õ-d ei saa tekitada ei vastava koodinguga ega TEX-i standardkujul (`\~o`). Eesti vanale kirjaviisile mõeldes pole too tõtt-öelda suur probleem…

Küll on vanades kirjades kasutusel kahte tüüpi s-d. “Pikk” s on kasutusel silbi sees, “lühike” s silbi lõpus (aga pane tähele kirjaviisi!). Viimase saab kui sisestada `s:`. Nagu näiteks:

```
\frakfamily
Ja pärrast sedda sündis:, et Nahas:, Ammonitte kunningas:
ärrasurri, ja ta poeg sai kunningaks ta as:semele.
```

Ja pärraſt ſedda ſündis, et Nahas, Ammonitte kunningas
ärraſurri, ja ta poeg ſai kunningakſ ta asſemele.

5 *pdf*-failide tegemine

pdf-faile saab TEX-s teha õige mitut moodi. Viimasel ajal on üha paremaks muutunud `pdflatex`. Kirjeldan siin viisi kuidas teha *pdf*-`i` `.dvi`-failidest.

pdf-fail on laias laastus pakitud postiskript ja nii teda ka tehakse – kõigepäält `dvips`-iga *dvi*-st postskriptiks ja siis `ghostscriptiga` postskriptist *pdf*-iks. `dvips`-i käivitamisel tuleb talle öelda et teeks *pdf*-le vastava postskripti (`-Ppdf`) ja et ligatuure ära ei rikuks (`-G0`). Seega siis:

```
dvips -Ppdf -G0 <failinimi.dvi>
```

`ghostscripti` abil *pdf*-i destilleerimisel tuleks öelda et tegu on A4 paberiga (`-sPAPERSIZE=a4`):

```
ps2pdf -sPAPERSIZE=a4 <failinimi.ps>
```

Nii teib eelnevast postskript-failist *pdf*. Probleemideta teisenevad ka kõik postskript-joonised.

Mõistlik on need käsud kombineerida ühes skriptiks. Näiteks nii:

```
#!/bin/bash
fname=${1%.dvi}
echo "Teen ps-iks..."
dvips -Ppdf -G0 $fname.dvi
echo "Nüüd teen pdf-iks..."
ps2pdf -sPAPERSIZE=a4 $fname.ps
echo "Kommbes: fail on $fname.pdf"
```

Korralik *pdf* fail sisaldab kõik vajalikud kirjad vektorkujul (*scalable fonts*). Selleks sobivad postskripti (*Type1*) kirjad. Kui vastavaid kirju vektorkujul pole, siis kasutatakse tavaliselt vaikimisi bitmap-kirju. Nondega on aga mitmeid probleeme. Esiteks on bitmap-kirjad tehtud kindla lahutuse tarvis ja võivad teise lahutusega esitades koledad välja näha. Teiseks probleemiks on et `xpdf 1.x` ei saa nendega üldse hakkama, ja `acroread` näitab neid koledasti (kuigi trükib ilusti kui lahutus õige on). `ghostscript` ja `gv` näitavad neid aga kiiresti ja tipptopp. Moodsad *ec*-kirjad ei ole *tetex 1.0.7*-ga vektorkujul kaasas. Küll on nad vabalt saadaval *CTAN* serverites.

Fontide (lokaalne või kodune, vt. osa 1) installeerimine käib nii:

1. Tõmba *CTAN*-ist *Type1* *ec*-kirjad ja nende juurde kuuluv *.map* fail (kokku 60MB).

2. Paki kirjad lahti. Hää kataloog kuhu nad panna oleks `$TEXMF/fonts/type1/ec`. *.map*-fail sobib näiteks kataloogi `$TEXMF/dvips/config` kus on juba kari sarnaseid faile ees.

3. Tee omale koopia `dvips-i` pääkonfiguratsioonifailist `/usr/share/texmf/dvips/config/config.ps`, sobiv koht kuhu koopia panna on `$TEXMF/dvips/config`, *.map*-faili juurde.

4. Lisa `dvips-i` konfiguratsioonile *ec*-fontide *.map* fail. Selleks asenda failis niisuguses kohas

```
% This shows how to add your own map file.
% Remove the comment and adjust the name:
% p +myfonts.map
```

viimane rida reaga

```
p +ecpfb.map
```

või kuidas iganes Sa *.map*-faili nimetasid.

5. Värskenda failinimekirja:

```
mktexlsr <$TEXMF>
```

Nüüd peaks ka *ec*-fontidega tulema ilusad *pdf*-failid.