

Sissejuhatus

Tegemist on C-keele projekti LihtsamLatex dokumentatsiooniga. Et tegu on meie enda välja mõeldud keele kompilaatoriga, siis peaks dokumentatsioon olema põhjalik.

Programmi töötamiseks peab kasutajal installitud olema mõni \TeX keele kompilaator, näiteks \MiKTeX või muu \TeX i kompilaator, mis kasutaks käsku `pdflatex`. Seda käsku kasutab programm, et peale lähtekoodi tõlgendamist ja \LaTeX i koodi genereerimist see ka kompileerida pdf-failiks. Samuti on tarvilik `config.txt` faili olemasolu programmi enda kaustas ning mingi template faili olemasolu `templates` kaustas.

Programmi kasutamiseks peab kasutaja looma ühe tekstifaili, mida edaspidi dokumnetatsioonis nimetatakse *lähtekoodifailiks*. Sinna hakkab kasutaja kirjutama koodi, mida meie programm tõlgib. Seda kasutaja kirjutatud koodi nimetatakse edaspidi *lähtekoodiks*.

1 Template failid

Esmalt tegeleme template failidega. Template folder on mõeldud kõikide erinevate template failide hoidmiseks. Hetkel on seal ainult `defaultTemplate.txt`, mille sisu võiks olla järgmine

```
\documentclass{article}
\usepackage{amsfonts}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage[estonian]{babel}

\usepackage{parskip}

\begin{document}
{{content}}
\end{document}
```

Siia saab oma valikul juurde lisada nii palju pakete, kui neid vaja võiks minna. Samuti võib kausta juurde lisada veel template faile, kusjuures nende nimed võivad olla suvalised. Igas template failis peab aga olema märksõna

```
{{content}}
```

mis enamasti käib koostatava dokumendi alustava ja lõpetava definitsiooni vahele. See koht, kus template failis on kirjas `{{content}}`, asendatakse kasutaja kirjutatud lähtekoodifaili tõlkega. Tähendab, lõpuks kompileeritava \TeX -faili päis ja jalus on pärit template failist, aga sisuks on kood, mille genereerib programm tõlkides kasutaja kirjutatud lähtekoodi. Seda, kuidas kasutaja kirjutatud lähtekoodifaili tõlgitakse, kontrollib `config.txt` fail.

2 Config.txt

Nüüd saame liikuda järgmise faili juurde. Peamised käskude ja keskkondade definitsioonid, sealjuures ka muud lipud, lähevad kõik `config.txt` faili. Algselt võiks selle sisuks olla

```
template = KAAREL

TEXTMODE KÄSUD
//(arg1)// -> \emph{arg1}
pealk (arg1) -> \secton{arg1}

MATHMODE KÄSUD
sum(al)(ül) -> \sum_{al}^{\ül}
to -> \to
inf -> \infty
lim(arg1) -> \lim_{arg1}

KESKKONNAD
enum [multiline, end:{--}] -> \begin{enumerate} #content \end{enumerate} | (item(arg1) -> \item arg1)
```

Välimuseltki on selge, et selle faili sisu on neljas eri osas. Iga osa kirjeldatakse allpool eraldi.

2.1 Template faili nimi

Reaga `template = KAAREL` määratakse template faili nimi, mida kompileerimisel kasutatakse. Selle faili nimi peab olema kirjutatud ilma `.txt` laiendusega. Samanimeline fail peab leiduma kaustas nimega `templates`. Näite puhul peab leiduma selles kaustas fail `KAAREL.txt`.

2.2 Mathmode käsud

Peamine asi, mis meie programmiga teeb \LaTeX i kirjutamist kiiremaks on käskude defineerimine. Nendega on võimalik suhteliselt lihtsal moel kirjutatud tekst asendada keerulise \LaTeX i koodiga. Kasutaja saab defineerida kaks komplekti käske. Esimene neist on mõeldud kasutamiseks olles mathmode sees. Teine on kasutamiseks textmode sees. Esmalt kirjeldame mathmode käske. Käsu definitsiooni saame jaotada kaheks pooleks

Tekst, mida lähtekoodis otsitakse. -> Tekst, mis sellega asendatakse.

Niimoodi saame defineerida lihtsad tekstasenduse käsud, näiteks

```
alfa -> \alpha
kord -> \cdot
RR -> \mathbb{R}
```

Vahel sellest aga ei piisa. Peame suutma ka defineerida integraale, summasid ning muid abikäske, mis võtavad endale teatud arv argumente. Selleks saab defineeritavas käsus anda sisse n argumenti, kujul

`käsuNimi(arg1)[arg2]... -> \mathcal{F} (arg1, arg2,...)`

kus

$\mathcal{F}(\text{arg1}, \text{arg2}, \dots)$

on vastava käsu \LaTeX i definitsioon, mis sisaldab endas vajalikke argumente (kusjuures need peavad olema sama nimega, mis käsu definitsiooni vasakul pool). Samuti tuleb eristada käsu defineerimisel argumentide tüüpe. Nendeks on

- `(argumendiNimi)` – Ümarad sulud tähendavad pikemat argumenti. See tähendab, et lähtekoodis argumendi lõppu tähistab ainult tühi või rea lõpp. Sellele vastandub lühem argumentitüüp.
- `[argumendiNimi]` – Kandilised sulud tähendavad lühemat argumenti. See tähendab, et lähtekoodis tähistavad argumendi lõppu peale tühiku ka tähemärgid `+`, `-`, `*`, `=` ja `,`. Tihti peale ei ole vahet, kas definitsioonis on argumentitüüp märgitud pikaks või lühikeseks. Vahe tuleb aga sisse siis kui tahetakse näiteks kirjutada polünoome. Definitsiooni `^(arg1) -> ^{arg1}` puhul tõlgitakse lähtekoodi tekst `a^n+b^m` koodiks

`a^{n+b^m}`

Polünoomi kirjutamise soovi jaoks see on vae. Õige tõlge oleks `a^{n}+b^{m}`, mis saavutatakse, kui definitsioonis oleks kasutatud kandilisi sulgusid järgmiselt `^[arg1] -> ^{arg1}`.

Niisiis toome mõned näited

```
sum(al)(ül) -> \sum_{al}^{ül}
sin(uuga) -> \sin{uuga}
^[mingiAsi] -> ^{mingiAsi}
```

Ainus käsk, mida defineerima ei pea, on jagamine, mida automaatselt tõlgitakse kui

`a/b -> \frac{a}{b}`

2.3 Texmode käsud

Mathmode sees kasutamiseks mõeldud mathmode käskudele lisaks saab kasutaja defineerida texmode käskude komplekti, mille käske otsitakse tavalise teksti seest. Näiteks on textmode käske ideaalne kasutada pealkirjade kiireks vormistamiseks, kui on vaja korduvalt panna mingit teksti \LaTeX i `center` keskkonna sisse.

Kuna textmode käsud on mõneti nüansikamad, kui mathmodekäsud, siis on parim neid selgitada näitega. Allpool on esitatud üks keskmise käsu definitsioon. Numbritega on tähistatud selle definitsiooni osad ja seejärel selgitatakse, millega on tegu.

1	2	3	2	4	5	6	5	8	_____9_____	10	_9_	11	_____9_____
edevpealkiri (pealkiri) (alune) -> \begin{center}\n\section*{pealkiri}\nalune\n\end{center}\n													
								7		12		12	12

1. Tegu on käsku alustava tekstiga. Kui see tekst lähtekoodist leitakse, ss sellele järgnevat teksti tõlgitakse erilisel moel nagu käsu definitsioonis kirjeldatud on. Panna tähele, et näidatud juhul on käsku alustavaks tekstiks `edevpealkiri` – sellel on viimane täht tühik.
2. Tegu on esimest argumenti avava ja sulgeva suluga. Argumentide eraldatakse muust käsudefinitsiooni tekstist neid ümbritsevate ümarsulgude järgi. Seega ei või käsu nimes ümarsulgi kasutada, sest asja võidaks tõlgendada valesti.
3. See on esimese argumenti nimi. See tähistab teksti, mida kasutaja kirjutab lähtekoodis käsku alustava teksti järele kuni selle tekstini, mis on argumenti sulgeva sulu järel (4). Argumente saab käsul olla praktiliselt lõputu kogus.
4. Tegu on esimest argumenti lõpetava tekstiga. Lähtekoodis käsku alustava teksti ja esimet argumenti lõpetava teksti vahele jääv tekst loetakse esimeseks argumentiks ja see asendatakse käsu definitsiooni paremas pooles kohale, kus on esimese argumenti nimi (10). Selles näites on mõeldud, et esimene argument läheb latexi sectioni loogeliste sulgude vahele.
5. Need sulud eraldavad muust definitsioonist teise argumenti nime (6). Järelikult kasutaja tahab peale esimese argumenti lõppu ka teise argumenti anda.
6. See on teise argumenti nimi. See tähistab teksti, mis jääb lähtekoodis esimest argumenti lõpetava teksti (4) ja teist argumenti lõpetava teksti (7) vahele.
7. See on teist argumenti lõpetav tekst. Näites on teist argumenti lõpetav tekst tühi sõne. Kui kumbagi argumenti lõpetav tekst on tühi sõne, siis loetakse vastava argumenti lõpuks realõpp e uuereamärk.
8. Tegu on tähistusega, mis eraldab definitsiooni vasakut poolt paremast poolest. Selleks on tekst `->` – selles on tühikud kummalgi pool. Oluline on meeles pidada, et tühikud on selle tähistuse osa. Vasakul on kirjeldatud, kuidas käsk algab, kuidas argumentid eraldatud on ja kui palju neid on. Parempool aga kirjeldab seda, milliseks tekstiks vasaku poole järgi vormistatud tekst lähtekoodis tõlgendada.
9. Kõik see tekst on see, mis lähtekoodis kirjutatud käsualguse teksti asemel pannakse \LaTeX i faili. Kohtades, kus käsu paremal poolel on kirjas argumentide nimed, saadetakse \LaTeX i faili tekstid, mis loeti argumentideks.
10. Esimene argument pannakse kohale (10) ja teine kohale (11).
12. Textmodekäsu paremale poolele saab kirjutada teksti `\n`, mis on eriline tekstilõik, mille asemel pannakse \LaTeX i faili reavahetusmärk, mitte tekst `\n`. Kui tahta tõlkesse panna teksti `\n`, siis tuleb kirjutada `\\n`

Selle käsu kasutamine lähtekoodifailis näeks muu teksti seas välja järgmine.

```
Tere mina olen Kaarel ja see on minu fancy ass pealkiri.
edevpealkiri MIMMA 6. kodune töö Kaarel Parve, 3. variant, 15.12.2024
Uuga buuga veel matemaatilist juttu mitme muutuja funktsionaalreak koondumine blaa.
```

Ja see tõlgitakse järgmiseks \LaTeX i koodiks.

```
Tere mina olen Kaarel ja see on minu fancy ass pealkiri.
\begin{center}
\section*{MIMMA 6. kodune töö}
Kaarel Parve, 3. variant, 15.12.2024
\end{center}
Uuga buuga veel matemaatilist juttu mitme muutuja funktsionaalreak koondumine blaa.
```