

UNIVERZITA PALACKÉHO V OLMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

BAKALÁŘSKÁ PRÁCE

Fonty v $\text{T}_{\text{E}}\text{X}_{\text{u}}$



Vedoucí bakalářské práce:
RNDr. Miloslav Závodný
Rok odevzdání: 2014

Vypracoval:
Markéta Beranová
ME, III. ročník

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně pod vedením pana RNDr. Miloslava Závodného s použitím uvedené literatury.

V Olomouci dne 7. dubna 2014

Poděkování

Ráda bych poděkovala vedoucímu bakalářské práce panu RNDr. Miloslavu Závodnému za spolupráci i za čas, který mi věnoval při konzultacích, a také mé rodině za podporu při vytváření mé bakalářské práce.

Obsah

Úvod	4
1 Historie typografie	5
2 T _E X	7
3 METAFONT	10
4 Fonty v T _E Xu	11
4.1 Rodina Computer Modern	11
4.2 Metrika a tvar znaků	14
4.3 Standardní postscriptové fonty	18
4.4 Zařazení fontů do plainT _E Xu	20
4.5 Zařazení fontů do L ^A T _E Xu	23
4.6 Nové fontové selekční schéma (NFSS)	23
4.7 Zařazení fontu do textové sazby	25
4.8 Zařazení fontu do matematické sazby	30
Závěr	34
Literatura	36

Úvod

Tématem bakalářské práce jsou *Fonty v T_EXu*. Ve druhém ročníku jsem měla možnost zapsat si předmět, kde se sazba T_EXem vyučovala, a mohla jsem si tak vyzkoušet způsob práce s tímto programem. Filosofie zpracování textu programem T_EX a hlavně výsledná sazba se mi líbila a já chtěla, aby mé dokumenty byly také tak krásné. Proto jsem se rozhodla věnovat se tomuto systému více. Téma Fonty v T_EXu mě zaujalo vzhledem k tomu, že použité písmo podstatně ovlivňuje vzhled výsledku.

Hlavním cílem mé bakalářské práce je ukázat, jak T_EX s fonty pracuje, a představit některé standardní fonty, které jsou v T_EXu k dispozici.

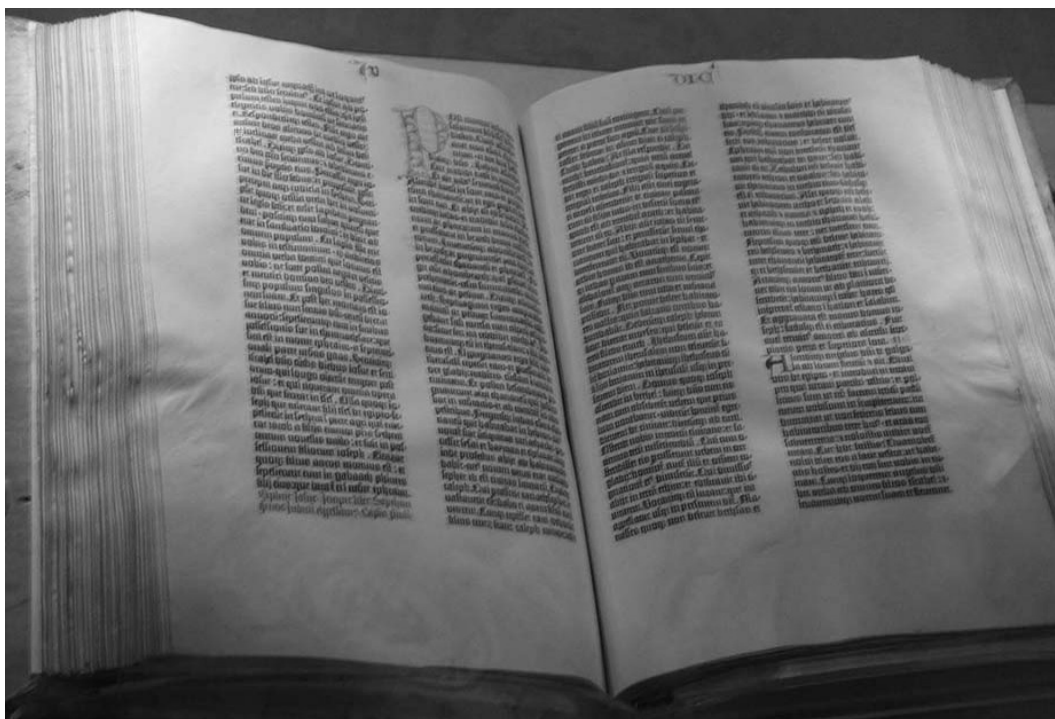
Dalším cílem bylo vytvořit stručný přehled T_EXovských fontů tak, aby bylo jednoduché vyhledat vhodný font podle speciální potřeby. Informací o T_EXovských fontech ani publikací, které se tímto tématem zabývají, není mnoho.

Práce je rozdělena do čtyř kapitol. V první kapitole je stručně uvedena historie typografie. Ve druhé kapitole je přidána informace o programu T_EX a jeho zakladateli. Ve třetí kapitole jsou uvedeny základní informace o programu METAFONT. Čtvrtá kapitola obsahuje základní informace o tom, jak T_EX pracuje s fonty. Práce dále představuje fonty programu T_EX, ukazuje, jak se s fonty pracuje v T_EXu a nahlíží do souborů tfm a pk. Práce vysvětluje pojmy virtuální font a postscriptový font, uvádí přehled standardních fontů, které jsou v T_EXu k dispozici, a popisuje i některé speciální fonty.

1. Historie typografie

Nejprve byly knihy přepisovány písaři, knih bylo málo a jejich cena značná. Tisk písem vznikl v Číně, kdy v 2. století n. l. vznikly kamenné desky s vyrývanými znaky a skládaly se do kamenných knihoven, na které se psaly díla čínských klasiků [11]. V 6. století vznikly dřevěné desky s vyrývanými znaky – deskotisky. Na reliéf znaků se štětcem nanasla barva, přiložil se navlhčený papír, kartáčkem se přejela zadní strana a vznikl otisk. V 9. století nahradil dřevo kov.

Johannes Gensfleisch zur Laden zum Gutenberg (1397/1400–1468), zvaný Gutenberg, je objevitelem knihtisku (tisk z výšky). Tiskařský lis, který koncepčně vycházel z vinařského lisu, doplnil o rozebíratelnou předlohu. Na připravenou mřížku vedle sebe sázel jednotlivé litery, odlité z kovu, [12]. První pokusy s knihtiskem vykonal asi r. 1450. V roce 1454 byla vytištěna první kniha – Gutenbergova Bible.



Obr. 1: Gutenbergova Bible (zdroj: http://upload.wikimedia.org/wikipedia/commons/b/b0/Gutenberg_Bible.jpg)

Litery měl knihtiskový sazeč před sebou v přihrádkách, odkud je vybíral a sázel do sázítka, které držel v ruce. Když dokončil řádek, musel jej ještě doplnit na přesnou délku (mezislovní mezery, v některých případech i mezery mezi písmeny) tak, aby se vyrovnal pravý okraj sazby. Hotové řádky skládal na sazebnici (plech opatřený okraji ve tvaru U) a hotový sloupec (příp. stránku) převázal motouzem. Stránka se pro kontrolu ručně otiskla a vznikl tzv. kontrolní otisk (obtah) a případné chyby se ručně opravily. Teprve po vysázení celého archu se mohly stránky sestavit ve správném pořadí, případně doplnit obrázky (dřevoryty, štočky), linkami a ornamenty a opět kontrolně otisknout. Tímto způsobem se sázely všechny knihy a časopisy téměř až do konce 19. století a mnohdy ještě déle [13].

Teprve roku 1896 zkonstruoval německý hodinář Ottmar Mergenthaler v Baltimore prakticky použitelný sázečí stroj *Linotype*. Na něm se pomocí klávesnice sázely negativní matrice a teprve hotový řádek se vcelku odlil. Matrice měly různé zářezy, takže se po použití automaticky roztřídily. Stroje Linotype se neustále zdokonalovaly a vyráběly téměř 80 let.

Roku 1897 patentoval americký vynálezce Talbert Lanston sázečí stroj na jiném principu. Jeho *Monotype* se skládal z pracoviště sazeče, kde se vyrobila děrovaná papírová páska, která pak řídila odlévání a sázení jednotlivých liter. Také tento stroj se zdokonaloval a vyráběl až do poloviny 20. století.

Složitě sazby např. matematických vzorců se ovšem nadále dělaly ručně [13].

Od roku 1985 postupně ovládá pole počítačová sazba (DTP – Desktop publishing). Zasloužil se o to zejména programovací jazyk PostScript, program PageMaker firmy Aldus a počítače Macintosh firmy Apple. Výstup počítače se buď používá pro osvit filmu případně desky, anebo přímo ovládá digitální tisk. Zejména v akademických kruzích se velmi rozšířily sázečí systémy $\text{T}_{\text{E}}\text{X}$ a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, které umožňují i složitou sazbu např. matematických vzorců [13].

2. T_EX



Program T_EX vytvořil Donald Ervin Knuth, který se řadí mezi zakladatele počítačových věd a je profesorem na Stanfordské univerzitě. Napsal velmi známou učebnici programování s názvem „*The Art of Computer Programming*“, na které stále pracuje. Když viděl předlohu 4. dílu své knihy vysazené pomocí DTP (první díly byly sázeny klasicky), byl s ní tak „spokojen“, že se pustil do práce na vytvoření vlastního DTP – T_EXu. Program T_EX¹ je typografický systém pro sazbu pěkných knih ve vysoké kvalitě, zvláště obsahují-li spoustu matematiky.

Knuth dal T_EX veřejnosti k dispozici zdarma. T_EX je koncipován jako značkovací systém, tj. informace o vzhledu sazby se zapisují přímo do zpracovávaného textu. Systém značek je rozšiřitelný, T_EX je i programovacím jazykem. Od jeho uživatele je tak vyžadována schopnost matematického myšlení a jistá inteligence, proto se T_EX používá hlavně v akademickém prostředí, využívá ho většina matematiků, fyziků a informatiků. Silnou stránkou T_EXu je totiž sazba matematických výrazů, což mnoho jiných komerčních systémů postrádá.

T_EX je konstruován jako systém, který je nezávislý na operačním systému i na výstupním zařízení. Nezávislost T_EXu na operačním systému značí, že k sazbě připravený dokument je zpracovatelný na jakémkoliv počítači s libovolným operačním systémem – Windows, Unix, IBM. Nezávislost T_EXu na výstupním zařízení je dána filosofií jeho práce. Při sazbě stránky (vytváření abstraktního popisu jejího vzhledu) nepracuje s konkrétními znaky, ale pouze s tzv. boxy, pravouhelníky s přesně definovanou výškou, šířkou a hloubkou. Tyto boxy jsou přiřazeny každému objektu sazby (tedy i každému písmenu) a na konkrétní objekt pouze odkazují. Nahrazení boxů konkrétním objektem řeší až ovladač výstupního zařízení – laserové tiskárny, osvitové jednotky apod.

¹T_EX vyslovujeme jako „tech“

\TeX můžeme uzpůsobit pro sazbu v jakémkoliv jazyku, i když byl původně určen především k sazbě anglických textů – „ \TeX je Američan“. Bez problémů lze používat akcenty či i jiné než latinkové abecedy – azbuku, arabské písmo, japonské znaky atd. Lze psát zprava doleva a klidně i zdola nahoru. Nic není nemožné, vše lze naprogramovat, i když to nemusí být zrovna jednoduché. \TeX může samozřejmě pracovat s postskriptovými objekty – obrázky, písmem, kódy.

Na rozdíl od formátování souboru pomocí programu typu MS Word², se do programu \TeX přímo vypíše požadavky na formátování pomocí *značek*. Značky (řídící sekvence) jsou procedury, které vyvolají příslušnou činnost. Formálně mají tvar posloupnosti písmen (rozlišují se velká a malá písmena), končící mezerou a začínající tzv. únikovým znakem, obvykle zpětným lomítkem. Mezery za značkou jsou při sazbě ignorovány. Značka může mít též podobu jediného znaku, který není písmeno, např. $\backslash\@$, pak nejsou mezery za značkou ignorovány. Textový soubor se značkami se nazývá *dokument*.

\TeX je nutno vidět jako programovací jazyk s příkazy umožňujícími efektivní sazbu textu. Při spuštění \TeX ovského souboru se nejprve načte *formát* – binární knihovna předdefinovaných značek sestavených z tzv. primitivů (asi 300 značek v jádru programu \TeX tvořících „bázi prostoru typografických příkazů“ – pomocí nich lze postihnout jakoukoliv činnost sazeče). S primitivy nepracuje začátečník, práce s nimi je náročná i pro zkušené programátory. První formát vytvořil Knuth, je to tzv. *plain*. Tento formát zakázal dále upravovat, dokument používající značek tohoto formátu dá i ve vzdálené budoucnosti stejný výsledek. \TeX , který je spuštěný s formátem *plain* se nazývá Plain \TeX .

Jiný formát – \LaTeX – byl vytvořen skupinou kolem Leslieho Lamporta. Práce s tímto formátem je vhodná pro uživatele, kteří nechtějí značky programovat, nabízí jim totiž kromě značek definovaných ve vlastním formátu i spoustu dalších knihoven, postihujících téměř vše, co může sazeč potřebovat. Práce v \LaTeX u se tak více podobá inženýrské činnosti, kdy pro své dílo (sazbu) vybíráme podle nás nejlepší z již naprogramovaných značek.

²z angl. WYSIWYG — what you see is (all) what you get

Výstupem \TeX u je soubor obsahující abstraktní popis vzhledu stránek dle Knutha, tzv. *dvi* (device independent) soubor. V tomto souboru jsou pouze informace o rozmístění boxů na jednotlivých stránkách a informace o jejich obsahu. Soubor je pak zpracován ovladačem (device driver) daného výstupního zařízení a informace o znacích jsou nahrazeny těmito znaky v příslušném rozlišení.

Pokud se rozhodneme používat \TeX , budeme potřebovat jak samotný program \TeX , tak i další soubory potřebné k práci s \TeX em – formáty, doplňující balíčky, fonty apod. Tyto soubory získáme v tzv. distribuci \TeX u. Distribuce \TeX u jsou na internetu k dispozici zdarma a s nimi i plno studijního materiálu a hlavně veškerá dokumentace. V současné době se využívá především distribucí *TeXLive* nebo *MikTeX*. Mezi oběma distribucemi jsou rozdíly jak ve způsobu instalace \TeX u, tak v uspořádání adresářů, přístupu k práci s fonty aj.

MikTeX potřebuje připojení k internetu, aby se jeho komponenty mohly doinstalovat podle aktuální potřeby uživatele, například knihovny použité v dokumentu. *TeXLive* umožňuje zvolit si při instalaci komponenty, které chceme nainstalovat, pokud si uživatel není jistý tím, co bude potřebovat, zvolí úplnou instalaci. *TeXLive* můžeme kdykoliv aktualizovat pomocí programu „*TeXLive Manager*“, jenž je součástí instalace, stejně jako internetová podpora.

Další programy, které je dobré instalovat, jsou *ghostscript* (interpret postskriptu, program *GS*) a *ghostview* (jeho preprocesor, program *GSView*), ty ale nejsou součástí distribuce, ale lze je zdarma stáhnout z internetu. Většinou chceme pracovat i se soubory ve formátu PDF, ty můžeme prohlížet, například programem *Acrobat Reader* nebo také pomocí *GSView*.

Ohledně práce s fonty, *MikTeX* zachovává původní přístup k uložení bitmapových fontů do adresářů podle jejich rozlišení. Soubory s fonty různého rozlišení (generované systémem ze zdrojových textů) mají stejný název, například *cmr10.pk*, ale jsou uloženy v odlišných adresářích pojmenovaných podle rozlišení, např. 600DPI, 300DPI. *TeXLive* rozlišuje soubory s fonty pomocí přípon s číslem rozlišení, například *cmr10.600pk* a ukládá je do jednoho adresáře.

3. METAFONT

Program METAFONT je úzce spojen s programem \TeX . Byl vytvořen stejným autorem jako \TeX a je uveden v práci *Computers and Typesetting*, dílu *The METAFONTbook*. I tento program byl vytvořen tak, aby nebyl závislý na operačním systému. Program METAFONT digitalizuje ze zdrojového souboru (napsaného v jazyce METAFONTu) písmo pro výstupní zařízení, které rastruje výstupní prostor. Jsou to vlastně skoro všechna zařízení, která používáme jako výstup z počítačů. Zařízení se rozdělují podle hustoty rastru. Rastr se udává v počtu bodů na palec – dpi (dot per inch). Máme různé hustoty rastru, například pro hrubší obrazovky počítačů (okolo 90 dpi), jehličkové tiskárny (180–360 dpi), inkoustové tiskárny (asi 300 dpi), laserové tiskárny (600–1200 dpi) a osvitové jednotky (1270–4000 dpi).

Fonty jsou popsány ve „zdrojových“ souborech s příponou *mf*. Tak jako \TeX má své formáty, které načítá a následně ukládá do binární podoby, tak METAFONT má takzvané báze. Donald Knuth vytvořil a popsal nejznámější a nejpoužívanější bázi, kterou popsal v knize s názvem *The METAFONTbook*, tuto bázi pojmenoval *plain*. Kromě této báze Knuth vytvořil bázi *cm* ke svým fontům Computer Modern.

Zdrojový soubor je vstupním souborem do programu METAFONT, v tomto souboru jsou METAFONTovým jazykem (stejně jako \TeX je i METAFONT programovací jazyk) popsány veškeré tahy písma (jde o textový soubor). Výstupem programu METAFONT je jednak metrický soubor *.tfm* (\TeX font metric) s boxy k jednotlivým znakům, údaji o slitcích a podřezávání, jednak bitová mapa³, přímo použitelná pro každé výstupní zařízení, s hustotou rastru odpovídající tomuto zařízení, např. *.600pk*, *.1200pk*, apod. v \TeX Live.

³matice bílých a černých bodů

4. Fonty v T_EXu

Fonty jsou definovány jako kompletní sada znaků abecedy, které mají jednu velikost a jednotný styl, společně s dalšími informacemi o těchto znacích (slitky, podřezávání). Krátce, font⁴ je souhrn všech znaků jednoho typu a řezu. Fonty můžeme rozlišit nejen podle toho jak jednotlivé znaky vypadají, ale také podle toho, zda obsahují akcenty, slitky a vyrovnané páry.

Akcent neboli diakritické znaménko pomáhá ve výslovnosti různých slov a to v podobě čárek, teček, háčků a dalších. *Vyrovnaní* (*podřezávání*) neboli kerning je zase informace o tom, jak by měly vypadat mezery mezi dvěma po sobě jdoucími znaky. Dvě po sobě jdoucí písmena na sebe musí dobře navazovat a dobře do sebe zapadat, aby nenarušovala estetický dojem ze sazby. A nakonec *slitek* neboli ligatura, což znamená, že po sobě jdoucí znaky budou nahrazeny jiným znakem (znaky se slévají). Např. ve slově filosofie by měla být po sobě jdoucí písmena f i nahrazena slítkem fi – filosofie, nebo --- je nahrazeno —.

Knuthův koncept tedy vyžaduje naprogramování fontu ve speciálním programovacím jazyku, určeném pro zpracování speciálním programem nazvaným METAFONT. S odborníky na písmo a typografii připravil profesor Donald Knuth pomocí METAFONTu rodinu písem nazvanou Computer Modern, základní písmo pro sazbu T_EXem. Rodina Computer Modern je základní součástí každé instalace T_EXu. Toto písmo vyniká především při velkém rozlišení, kde vynikne jeho krása.

4.1. Rodina Computer Modern

Donald Knuth rodinu Computer Modern zdokumentoval v knize The Computer Modern Typefaces. Celá rodina obsahuje přibližně 75 řezů. Computer Modern je moderní klasicistické písmo (tzv. statická antikva), které vychází z rytecké předlohy písma Modern Monotype 8a. Hlavními znaky tohoto písma jsou výrazné rozdíly mezi tenkými a silnými tahy a výborně propracované znaky pro sazbu matematiky. Písmo Computer Modern obsahuje osm základních řezů [10].

⁴ „font“ v češtině znamená „znaková sada“

Různé řezy písma Computer Modern:

stojatý (roman) – jako tato ukázka

polotučný (bold face) – **jako tato ukázka**

kurzívu (italic) – *jako tato ukázka*

skloněný (slanted) – *jako tato ukázka*

bezpatkový (sans serif) – jako tato ukázka

polotučnou kurzívu – ***jako tato ukázka***

strojopisný (typewriter style) – jako tato ukázka

malé kapitálky (small caps) – JAKO TATO UKÁZKA

matematickou kurzívu (math italic) – *jakotatoukzka*

matematické symboly – $\Sigma \oplus \Pi = f$

Pro první tři řezy existují varianty pro použití ve velikostech (5, 7, 8, 9, 10 a 12 bodů), které se neliší jen velikostí znaků, ale i jejich kresbou a šířkou (pětibodové písmo není desetibodové zmenšené na polovinu). T_EX umí ve spolupráci s METAFONTEM písmo zvětšovat či dokonce připravit v libovolné velikosti, zásady typografie však pro písma větší než 10 bodů připouští pouze 11 bodů a pak 1,2 násobky velikosti 10 bodů, tj. 12, 14,4, 17,28 atd. Do rodiny Computer Modern patří též různé matematické a experimentální řezy. Rodina Computer Modern byla vytvořena jako bitmapová, v současnosti existuje také ve vektorových formátech – TrueType, Type1 a OpenType.

Originální písmo Computer Modern není vhodné na sazbu českých dokumentů, protože neobsahuje potřebné akcentované znaky (á, é, í, š apod.) – je pouze 7bitové. Např. font cmr10 obsahuje tyto znaky:

0: Γ	1: Δ	2: Θ	3: Λ	4: Ξ	5: Π	6: Σ	7: Υ	8: Φ	9: Ψ
10: Ω	11: ff	12: fi	13: fl	14: ffi	15: ffl	16: l	17: J	18: `	19: ´
20: ˘	21: ˘	22: ¯	23: °	24: ‚	25: ß	26: æ	27: œ	28: ø	29: Æ
30: Œ	31: Ø	32: ˆ	33: !	34: ”	35: #	36: \$	37: %	38: &	39: ’
40: (41:)	42: *	43: +	44: ,	45: -	46: .	47: /	48: 0	49: 1
50: 2	51: 3	52: 4	53: 5	54: 6	55: 7	56: 8	57: 9	58: :	59: ;
60: j	61: =	62: ¿	63: ?	64: @	65: A	66: B	67: C	68: D	69: E

70: F	71: G	72: H	73: I	74: J	75: K	76: L	77: M	78: N	79: O
80: P	81: Q	82: R	83: S	84: T	85: U	86: V	87: W	88: X	89: Y
90: Z	91: [92: “	93:]	94: ^	95: ·	96: ‘	97: a	98: b	99: c
100: d	101: e	102: f	103: g	104: h	105: i	106: j	107: k	108: l	109: m
110: n	111: o	112: p	113: q	114: r	115: s	116: t	117: u	118: v	119: w
120: x	121: y	122: z	123: -	124: —	125: ”	126: ~	127: ”		

Pro sazbu českých dokumentů existuje upravená 8bitová česko-slovenská varianta, tzv. CS-fonty. Např. font cstil0 obsahuje tyto znaky:

0: <i>Γ</i>	1: <i>Δ</i>	2: <i>Θ</i>	3: <i>Λ</i>	4: <i>Ξ</i>	5: <i>Π</i>	6: <i>Σ</i>	7: <i>Υ</i>	8: <i>Φ</i>	9: <i>Ψ</i>
10: <i>Ω</i>	11: <i>ff</i>	12: <i>fi</i>	13: <i>fl</i>	14: <i>ffi</i>	15: <i>ffl</i>	16: <i>ι</i>	17: <i>Ͽ</i>	18: `	19: ´
20: ˘	21: ˘	22: ¯	23: °	24: ˘	25: <i>β</i>	26: <i>æ</i>	27: <i>œ</i>	28: <i>ø</i>	29: <i>Æ</i>
30: <i>Œ</i>	31: <i>Ø</i>	32: ¨	33: !	34: ”	35: #	36: <i>£</i>	37: %	38: <i>ℳ</i>	39: ’
40: (41:)	42: *	43: +	44: ,	45: -	46: .	47: /	48: 0	49: 1
50: 2	51: 3	52: 4	53: 5	54: 6	55: 7	56: 8	57: 9	58: :	59: ;
60: <i>i</i>	61: =	62: <i>ı</i>	63: ?	64: @	65: <i>A</i>	66: <i>B</i>	67: <i>C</i>	68: <i>D</i>	69: <i>E</i>
70: <i>F</i>	71: <i>G</i>	72: <i>H</i>	73: <i>I</i>	74: <i>J</i>	75: <i>K</i>	76: <i>L</i>	77: <i>M</i>	78: <i>N</i>	79: <i>O</i>
80: <i>P</i>	81: <i>Q</i>	82: <i>R</i>	83: <i>S</i>	84: <i>T</i>	85: <i>U</i>	86: <i>V</i>	87: <i>W</i>	88: <i>X</i>	89: <i>Y</i>
90: <i>Z</i>	91: [92: “	93:]	94: ^	95: ·	96: ‘	97: <i>a</i>	98: <i>b</i>	99: <i>c</i>
100: <i>d</i>	101: <i>e</i>	102: <i>f</i>	103: <i>g</i>	104: <i>h</i>	105: <i>i</i>	106: <i>j</i>	107: <i>k</i>	108: <i>l</i>	109: <i>m</i>
110: <i>n</i>	111: <i>o</i>	112: <i>p</i>	113: <i>q</i>	114: <i>r</i>	115: <i>s</i>	116: <i>t</i>	117: <i>u</i>	118: <i>v</i>	119: <i>w</i>
120: <i>x</i>	121: <i>y</i>	122: <i>z</i>	123: -	124: —	125: ”	126: ~	127: ”	128:	129:
130:	131:	132:	133:	134:	135:	136:	137:	138:	139:
140:	141: ‰	142:	143:	144:	145:	146:	147:	148:	149:
150:	151:	152: <i>Á</i>	153:	154:	155:	156: ¯	157: ˘	158: «	159: »
160:	161:	162:	163:	164:	165: <i>Ĺ</i>	166:	167:	168:	169: <i>Š</i>
170:	171: <i>Ť</i>	172:	173:	174: <i>Ž</i>	175:	176:	177:	178:	179:
180:	181: <i>ř</i>	182:	183:	184: <i>à</i>	185: <i>š</i>	186:	187: <i>ť</i>	188:	189:
190: <i>ž</i>	191:	192: <i>Ř</i>	193: <i>Á</i>	194:	195:	196: <i>Ā</i>	197: <i>Ĺ</i>	198:	199:
200: <i>Č</i>	201: <i>Ě</i>	202:	203:	204: <i>Ě</i>	205: <i>Í</i>	206:	207: <i>Ď</i>	208:	209:
210: <i>Ň</i>	211: <i>Ó</i>	212: <i>Ô</i>	213:	214: <i>Ö</i>	215:	216: <i>Ř</i>	217: <i>Ů</i>	218: <i>Ú</i>	219:

220: Ů	221: Ŷ	222:	223:	224: ř	225: á	226:	227:	228: ä	229: í
230:	231:	232: ě	233: é	234:	235:	236: ě	237: í	238:	239: đ
240:	241:	242: ň	243: ó	244: ô	245:	246: õ	247:	248: ř	249: ů
250: ú	251:	252: ü	253: ý	254: „	255: “				

\TeX může v současnosti pracovat s jakýmkoliv fonty. Je ale nutné poznamenat, že s objevením se verze pdf \TeX vznikl malý problém. Pdf \TeX může pracovat přímo s TrueType fonty, zatímco klasický \TeX nikoliv. Zde je nutné transformovat TTF font do postskriptové podoby Type1, mnohdy lze ale vyrobit jen font Type42, s nímž ale pro změnu nepracuje pdf \TeX .

4.2. Metrika a tvar znaků

Uvedla jsem, že „font“ je jistý souhrn údajů, vztahujících se k jednotlivým písmenům abecedy nebo znakům nějakého konkrétního vzhledu. Do fontu se zahrnují nejen písmena abecedy, ale také i jiné symboly, které s abecedou souvisí, například číslice, interpunkce a další údaje. Existují speciální fonty, které nejsou složeny z písmen, tvoří je např. pouze matematické symboly. Veškeré údaje, které jsou uloženy ve fontu rozdělujeme tímto způsobem:

1. rozměry symbolů
2. mezery mezi slovy
3. metrické údaje o vzájemných polohách objektů z fontu, například v matematické sazbě informace o polohách indexů
4. údaje vztahující se k dané skupině symbolů, například údaje o ligaturách
5. tvary jednotlivých symbolů

Veškeré údaje o fontech jsou rozděleny do dvou částí:

1. V metrice fontu (soubor s příponou `tfm`), s níž pracuje \TeX jsou uloženy informace o rozměrech symbolů a údaje vztahující se k dané skupině symbolů, body 1–4 uvedeného výčtu.

2. V souborech s příponou pk jsou uloženy bitové mapy tvarů jednotlivých symbolů. V souborech s příponou pfb nebo pfa jsou uloženy postskriptové procedury popisující tvary jednotlivých symbolů. S těmito soubory samotný T_EX vůbec nepracuje. S nimi pracují ovladače výstupních zařízení.

Postskriptový font

Máme-li k dispozici postskriptový font (soubory afm a pfb), pak jediné co potřebujeme, je odpovídající tfm soubor. Ovladač dvips i pdfT_EX dokáží s postskriptovým fontem pracovat přímo.

Metrické informace postskriptových fontů jsou do tfm generovány z textového souboru s příponou afm (Adobe Font Metric). K tomuto účelu slouží program afm2tfm.

AFM soubor se skládá z několika částí:

- řídicí informace
- celkové údaje o fontu
- informace o směru psaní
- metriky jednotlivých znaků
- kerningové informace
- údaje o kompozitních znacích

Afm soubor musí začínat `StartFontMetrics version` a končit `EndFontMetrics`. Tato slova jsou povinná a musí být vždy na začátku, resp. na konci, celého afm souboru, version je specifikovaný afm formát, např 3.0.

Program Afm2tfm

Program Afm2tfm napsal Tomas Rokicki a tento program můžeme získat z instalace C_ST_EX v dvipsovém balíku. Pomocí Afm2tfm souboru převádíme postskriptovou metriku afm na metriku tfm a lze vytvářet i virtuální fonty. Můžeme měnit kódování výstupních souborů.

Program `afm2tfm`, společně s programy pro konverzi `pfb` souboru `t1disasm` a `t1asm`, patří k základním nástrojům pro manipulaci s postscriptovými fonty. Máme-li k dispozici např. font `Casting`, tj. soubory `Casting.afm` a `Casting.pfb`, pak příkazem `afm2tfm Casting.afm Casting.tfm` k fontu `Casting.afm` připravíme metriku `Casting.tfm`. Přitom program vypíše informaci potřebnou pro zařazení fontu do \TeX u: `Casting Casting`.

Kdybychom chtěli překódovat původní font a připravit virtuální font, pak by syntax příkazu byla komplikovanější:

```
afm2tfm Casting.afm -t xl2.enc -v Casting8z rCasting8z
vptovf Casting8z.vpl Casting8z.vf Casting8z.tfm
```

Význam parametrů nalezne čtenář v dokumentaci k programu `afm2tfm`, nebo po zavolání programu `afm` s parametrem `help` – `afm2tfm --help`, program `vptovf` převede `vpl` soubor, který připraví v prvním řádku program `afm2tfm`, na virtuální font. Program `afm2tfm` vypíše informaci potřebnou pro zařazení fontu do \TeX u: `rCasting8z Casting`.

O metrice `tfm`

Jak jsem se dříve již zmínila, samotný program \TeX pracuje pouze s metrickými informacemi o fontu a tyto informace jsou uloženy v souboru `tfm` (\TeX Font Metric). Kdybychom chtěli sázet nestandardními fonty, které nejsou součástí instalace \TeX u, pak stačí získat metrické informace tohoto fontu ve formátu `tfm`.

Formát `tfm` je binární, tedy přímo nečitelný. Aby bylo možné přečíst informace v tomto formátu uložené, připravil Donald Knuth k tomuto účelu konverzní programy, které konvertují `tfm` soubor do čitelné podoby v textovém souboru a zase zpět. Metrika má svoji čitelnou podobu v textovém souboru s názvem „property list“ a mívá příponu `pl`. Z `tfm` do `pl` provede konverzi program `tftopl`, obráceně pracuje program `pltotf`.

Font a ovladač

Dvi-ovladače pracují se souborem ve formátu `pk` je to komprimovaná verze formátu `gf` (generic font). Pro konverzi používáme programy `gftopk` a `pktogf`.

Formát `pk` (pack bitmap) obsahuje bitmapové obrazy písem, se kterými dokáže pracovat každý dvi-ovladač. Bitmapy musíme generovat pro každé rozlišení zvlášť, například jehličkové 180 dpi, inkoustové a laserové tiskárny 300 a 600 dpi apod. Dříve se používaly knihovny těchto bitmap, tzv. soubory `fli`. `TeXLive` je nepoužívá.

Virtuální fonty

Donald Knuth navíc přišel s konceptem virtuálních fontů. `TeX` pracuje pouze s fontovými metrikami, které obsahují veškeré informace potřebné pro správné zformátování textu. Teprve ovladače jednotlivých zařízení (obrazovky, tiskárny) pracují s vlastními fonty a umísťují skutečnou podobu znaků na obrazovku nebo papír. Např. dokud nebyly k dispozici CS-fonty s kresbami akcentovaných znaků, byly tyto znaky sestavovány také pomocí virtuálního fontu: font `cmr10.600pk` obsahuje 128 znaků písma roman ve velikosti 10 bodů. Metrika tohoto fontu je uložena v souboru `cmr10.tfm`. Virtuální font `csr10.vf` udával, že znak `A` fontu `csr10` je roven znaku `A` fontu `cmr10` a znak `Á` je složen ze znaků `A` a `´` fontu `cmr10`. Metrika tohoto fontu byla uložena v souboru `csr10.tfm`. `TeX` pracuje pouze s metrikami a nestará se o to, zda daný font je virtuální či skutečný. Teprve driver příslušného zařízení buď použije přímo daný font nebo v případě virtuálního skládá daný znak ze znaků jiného fontu.

Virtuální fonty umožnily nazývat písmenem či znakem více symbolů, například písmeno `ř` bylo v Knuthových Computer modern fontech specifikováno jako `\v{r}`, tj. složeno z písmene `r` a háčku `ˇ`. Virtuální font vlastně obsahuje informace o tom, jak poskládat písmena daného fontu z písem jiných fontů na základě informací o znacích z metriky. Zjistí-li dvi-ovladač sazbu virtuálním fontem, poradí si. Virtuální fonty mají příponu `vf` a dvi-ovladač z informací, které tyto virtuální fonty obsahují, sestaví příslušné znaky. Pomocí virtuálního fontu můžeme:

- nahradit znak z jiného pk fontu, můžeme sestavit jinak kódovaný virtuální font, než jak ho známe z pk fontu
- vytvořit znak pomocí jazyka výstupního zařízení
- znak virtuálního fontu sestavit z většího množství elementárních znaků, z nichž každý může být realizován jiným způsobem

V praxi se nejčastěji používají virtuální fonty při sestavování znaků s akcentem.

4.3. Standardní postscriptové fonty

Instalace \TeX Live obsahuje počestěné všechny standardní postscriptové fonty. Jedná se o tyto fonty:

- AvantGarde
- Bookman
- Courier
- Helvetica
- New Century Schoolbook
- Palatino
- Times Roman
- ZapfChancery

Sada metrik balíčku `cspfonts.tar.gz` zahrnuje následující fonty, viz [4]:

<code>pagk8z</code>		AvantGarde-Book
<code>pagko8z</code>		AvantGarde-BookOblique
<code>pagd8z</code>		AvantGarde-Demi
<code>pagdo8z</code>		AvantGarde-DemiOblique
<code>pagkc8z</code>	*	AvantGarde-Book Caps
<code>pagdc8z</code>	*	AvantGarde-Demi Caps
<code>pbkl8z</code>		Bookman-Light
<code>pbkli8z</code>		Bookman-LightItalic
<code>pbkd8z</code>		Bookman-Demi
<code>pbkdi8z</code>		Bookman-DemiItalic
<code>pbklc8z</code>	*	Bookman-Light Caps
<code>pbkdc8z</code>	*	Bookman-Demi Caps

pcrr8u	Courier
pcro8u	Courier-Oblique
pcrb8u	Courier-Bold
pcrbo8u	Courier-BoldOblique
pcrrc8u	* Courier Caps
pcrbc8u	* Courier-Bold Caps
phvr8z	Helvetica
phvro8z	Helvetica-Oblique
phvb8z	Helvetica-Bold
phvbo8z	Helvetica-BoldOblique
phvrc8z	* Helvetica Caps
phvbc8z	* Helvetica-Bold Caps
phvrn8z	* Helvetica-Narrow
phvron8z	* Helvetica-Narrow-Oblique
phvbn8z	* Helvetica-Narrow-Bold
phvbon8z	* Helvetica-Narrow-BoldOblique
phvrc8z	* Helvetica-Narrow Caps
phvbnc8z	* Helvetica-Narrow-Bold Narrow Caps
pncr8z	NewCenturySchlbk-Roman
pncri8z	NewCenturySchlbk-Italic
pncb8z	NewCenturySchlbk-Bold
pncbi8z	NewCenturySchlbk-BoldItalic
pncrc8z	* NewCenturySchlbk-Roman Caps
pncbc8z	* NewCenturySchlbk-Bold Caps
pplr8z	Palatino-Roman
pplr8z	Palatino-Italic
pplb8z	Palatino-Bold
pplbi8z	Palatino-BoldItalic
pplrc8z	* Palatino-Roman Caps
pplbc8z	* Palatino-Bold Caps
ptmr8z	Times-Roman
ptmri8z	Times-Italic
ptmb8z	Times-Bold
ptmbi8z	Times-BoldItalic
ptmrc8z	* Times-Roman Caps
ptmbc8z	* Times-Bold Caps
pzcmi8z	ZapfChancery-MediumItalic

V tabulce jsou hvězdičkou označeny ty metriky, které neodkazují přímo na speciální postskriptový font, ale jsou implementovány pomocí virtuálního fontu.

4.4. Zařazení fontů do plainTeXu

Nejprve je potřeba uvést, že se zásadně liší práce s fonty v textovém režimu (horizontální mód, kdy TeX sází běžný text do odstavců) a matematickém režimu (kdy TeX sází matematické výrazy v matematickém módu).

Textový režim

Pro textový režim jsou předdeklarovány základní přepínače `\rm`, `\it`, `\bf`, `\sf`, `\tt` a `\sl`. K dispozici jsou obvykle i desetibodové verze `\tenrm`, `\tenit`, `\tenbf`, `\tentt` a `\tensl`, resp. analogické `\twl...` aj.

Všechny takové přepínače jsou deklarovány použitím primitivního příkazu `\font`. Např. `\font\tenrm=cmr10` deklaruje přepínač `\tenrm`, který načte font uložený v souboru `cmr10`. Tento přepínač se chová stejně, jako standardní přepínač `\rm`.

Příkaz `\font\MAN=logo10 scaled \magstep1` deklaruje přepínač fontu s názvem `\MAN` ve velikosti písma 12pt (`\magstep1`).

Příkaz `\font\tucny=csssxbx10` deklaruje přepínač `\tucny`, který vede k sazbě fontem uloženým v souboru `csssxbx10`.

Co vlastně znamená `csssxbx10` [3]:

- `cs` znamená, že se jedná o CS-font, který rozšiřuje písmo rodiny Computer Modern o akcentované znaky. Knuthův font má zkratku `cm`.
- `ss` je zkratka „sans serif“ – písmo bez patek
- `bx` je zkratka „bold extended“ – tučné písmo s rozšířeným horizontálním rozměrem
- desítka znamená, že jde o desetibodové písmo

Můžeme říci, že k tomu, abychom mohli nějaké písmo ihned používat, stačí mít k dispozici soubor `tfm` a dále metafontový zdrojový soubor `.mp` nebo připravený soubor `.pk` nebo poskriptové písmo `.pfb`.

Pro zavedení standardního postskriptové fontu do plainTeXu jsou předdefinovány styly, které přepnou sazbu textu do příslušného fontu:

- `cavantga` – Avantgarde Book
- `cbookman` – Bookman
- `chelvet` – Helvetica
- `cncent` – New Century Schoolbook
- `cpalatin` – Palatino
- `ctimes` – Times Roman

Chceme-li v plain \TeX u používat toto písmo, stačí načíst příslušný styl, např. `\input cavantga` nebo `\input ctimes`. Tak nastavíme normální řez daného písma pro sazbu textu.

Matematický režim

Zde je nutné deklarovat tzv. matematickou rodinu a načíst font ve velikostech nutných pro matematickou sazbu. Je-li písmo velikosti n , pak první index má velikost $n - 3$, druhý a další $n - 5$. Např. pro sazbu matematického skriptu `rsfs` definujeme

```

\newfam\scrfam
\font\tenscr=rsfs10
\font\sevenscr=rsfs7
\font\fivescr=rsfs5
\textfont\scrfam=\tenscr
\scriptfont\scrfam=\sevenscr
\scriptscriptfont\scrfam=\fivescr
\def\scr #1{\fam\scrfam #1}

```

Připojíme font `rsfs` v požadovaných velikostech, přiřadíme ho „členům rodiny“ a definujeme přepínač `\scr`. Ten v plain \TeX u umožní matematickou sazbu tímto fontem.

Základní rodiny matematických fontů

Petr Olšák ve své knize [2] podrobně rozebírá problém matematické sazby. Mimo jiné uvádí: „Aby byl T_EX ‚matematicky gramotný‘, musí mít bezpodmínečně deklarované rodiny číslo 2 a 3. Z fontů těchto rodin čerpá T_EX prostřednictvím parametrů `\fontdimen` veškeré znalosti o tom, jak sestavovat matematickou sazbu. Například, jak vysoko posunout exponent nebo kolik místa udělat kolem zlomkové čáry. Pouze rodiny 2 a 3 mají tuto významnou roli.

V rodině 2 jsou soustředěny parametry, které jsou rozdílné pro každou ze tří velikostí fontu (například vzdálenost matematické osy od účaří). V rodině 3 jsou pak parametry, které nezávisí na velikosti fontu (například tloušťka zlomkové čáry bude ve všech matematických stylech stejná).“

- Rodina 0: Antikva; běžně v textu a méně v matematice (`cos`, `lim` apod.)
- Rodina 1: Kurzíva; sazba matematických proměnných (a , β , x , f apod)
- Rodina 2: Značky v matematice, binární operace a relace (\odot , \leq , atd.)
- Rodina 3: Operátory a „zvětšující se“ objekty pomocí následníků (např.

$$\sqrt{a}, \sqrt{\frac{a}{b}}, \sqrt{\frac{a}{\frac{c}{b}}}$$

Rodiny 0 až 3 jsou v plainu deklarovány takto [2]:

	<code>\textfont</code>	<code>\scriptfont</code>	<code>\scriptscriptfont</code>
Rodina 0:	<code>cmr10 (\tenrm)</code>	<code>cmr7 (\sevenrm)</code>	<code>cmr5 (\fiverm)</code>
Rodina 1:	<code>cmmi10 (\teni)</code>	<code>cmmi7 (\seveni)</code>	<code>cmmi5 (\fivei)</code>
Rodina 2:	<code>cmsy10 (\tensy)</code>	<code>cmsy7 (\sevensy)</code>	<code>cmsy5 (\fivey)</code>
Rodina 3:	<code>cmex10 (\tenex)</code>	<code>cmex10 (\tenex)</code>	<code>cmex10 (\tenex)</code>

V závorce je uveden přepínač, který byl u každého fontu deklarován při zavedení primitivem `\font`. Přepínač je tedy možné použít při sazbě v textovém režimu. Pro rodinu 1 není použita běžná textová kurzíva `cmti`, ale speciální matematická. Kresby znaků jsou nepatrně širší, kerningové a „slitkové“ (ligační) tabulky fontů jsou rozdílné. V rodině 3 máme ve všech třech variantách zaveden jediný font. To většinou stačí, protože velké operátory se v matematických vzorcích používají obvykle jen v základní velikosti [2].

4.5. Zařazení fontů do L^AT_EXu

Můžeme font připojit stejným způsobem jako v plainT_EXu, můžeme také použít přepínačů definovaných ve verzi L^AT_EX2.09, tj. `\rm`, `\bf`, `\it`, `\tt`, `\sl`, `\sf`, `\sc`, `\em`. Tyto přepínače pracují analogicky jak přepínače fontů v plainT_EXu, tj. zavedou daný font bez ohledu na použitý font předcházející – takže např. `\bf\it` je totéž co `\it`.

Velikost písma můžeme samozřejmě měnit, např. je-li náš dokument připraven ve 12bodovém písmu, je:

- Příkaz `\tiny` – Markéta 7 pt
- Příkaz `\scriptsize` – Markéta 9 pt
- Příkaz `\footnotesize` – Markéta 10 pt
- Příkaz `\small` – Markéta 11 pt
- Příkaz `\normalsize` – Markéta 12 pt
- Příkaz `\large` – Markéta 14,4 pt
- Příkaz `\Large` – Markéta 17,28 pt
- Příkaz `\LARGE` – Markéta 20,74 pt
- Příkaz `\huge` – Markéta 24,88 pt

V případě 10bodového písma, je `\normalsize` 10pt, menší velikosti jsou 9pt, 8pt, 7pt a 5pt, pro 24,88 pt je k dispozici přepínač `\Huge`.

Fonty můžeme snadno přepínat z **velkého** na **malý**, z **polotučného** na **polotučnou kurzívu**. Aby text vypadal krásně, měli bychom volit jeden typ písma a měli bychom se také vyhýbat přílišnému měnění velikostí písma v textu.

4.6. Nové fontové selekční schéma (NFSS)

L^AT_EX 2_ε přinesl do přepínání fontů radikální změnu – nové fontové selekční schéma NFSS (New Font Selection Scheme).

Při psaní této části jsem vycházela z [9].

Každý font v $\text{\LaTeX} 2_{\epsilon}$ má pět atributů:

1. kódování fontu (encoding) – uspořádání znaků v znakové sadě, máme např.:

- OT1 — fonty jak je uspořádal Knuth (ordinary \TeX text)
- T1 — fonty v kódování Cork (\TeX extended text, dohodnuté na konferenci v Corku r. 1990, nazývané též ‘Cork encoding’, obsahuje téměř všechny střeoevropské znaky)
- OML — fonty pro matematickou italiku
- OMS — fonty pro matematické symboly
- OMX — fonty pro velké matematické symboly
- L — lokální kódování (pro místní potřebu)
- U — univerzální (pro speciální potřebu, např. AMS fonty)

2. rodina písma (family) – rodiny, které máme k dispozici, jsou např.:

- cmr — Computer Modern Roman
- cmss — Computer Modern Sans Serif
- cmtt — Computer Modern Typewriter
- cmm — Computer Modern Math Italic
- cmsy — Computer Modern Math Symbols
- cmex — Computer Modern Math Extensions
- ptm — Adobe Times
- phv — Adobe Helvetica
- ppl — Adobe Palatino
- pbk — Adobe Bookman
- pcr — Adobe Courier

3. váha, duktus (series) – dalo by se říci tučnost tahů písma. Nejužívanější váhy jsou:

- m — Medium
- b — Bold
- bx — Bold Extended
- sb — Semi-bold
- c — Condensed

4. tvar, varianta (shape):

- n — Normal (vzpřímené písmo)
- it — Italic (kurzíva)
- sl — Slanted (skloněné písmo)
- sc — Small caps (malé kapitálky)

5. stupeň – velikost písma, např. 10pt pro desetibodové písmo

4.7. Zařazení fontu do textové sazby

Standardní příkazy pro volbu fontu

Přepínače písma jsou tzv. *ortogonální*, přepínají právě jeden atribut, ostatní se zachovávají. K dispozici máme následující přepínače, resp. příkazy:

- `\textrm{...}` nebo `\rmfamily` — nastavení antikvy
- `\textsf{...}` nebo `\sffamily` — nastavení grotesku
- `\texttt{...}` nebo `\ttfamily` — nastavení strojopisného
- `\textmd{...}` nebo `\mdseries` — nastavení netučného
- `\textbf{...}` nebo `\bfseries` — **nastavení polotučného**
- `\textup{...}` nebo `\upshape` — nastavení vzpřímeného
- `\textit{...}` nebo `\itshape` — *nastavení kurzívy*

- `\textsl{...}` nebo `\slshape` — nastavení skloněného
- `\textsc{...}` nebo `\scshape` — NASTAVENÍ MALÝCH KAPITÁLEK
- `\emph{...}` — nastavení vyznačovacího řezu

Informace o použitém fontu je pak T_EXem zobrazena např. ve tvaru `\OT1/cmr/m/n/10`. Jde o font v kódování OT1, rodina cmr, duktus médium, tvar normální (antikva), velikost 10pt.

Na nižší úrovni můžeme k výběru fontu použít následující příkazy:

```
\fontencoding{encoding}
\fontfamily{family}
\fontseries{series}
\fontshape{shape}
\fontsize{size}{baselineskip}
\linespread{factor}
```

Význam uvedených příkazů odpovídá jejich názvům, `\linespread{factor}` násobí factorem proklad. Následovat musí příkaz `\selectfont`, který daný font zavede. Parametry, které nebyly nastaveny jsou převzaty z dosavadního fontu.

Např. `\fontfamily{ptm}\fontseries{b}\selectfont` nějaký text vysází **nějaký text** polotučným řezem písma Times Roman normálního tvaru, velikosti 12pt a kódování IL2 (sázíme česky a styl `czech` v preambuli dokumentu přepne do tohoto kódování). Existuje stručnější verze uvedeného postupu – příkaz `\usefont{encoding}{family}{series}{shape}`. V tomto příkazu nesmíme žádný parametr vynechat. Např. `\usefont{IL2}{ppl}{m}{it}` vybere písmo palatino ve tvaru italic, tedy *vybere písmo palatino ve tvaru italic*.

Nastavení atributů se děje prostřednictvím parametrů určujících aktuální font `\encodingdefault`, `\familydefault`, `\seriesdefault`, `\shapedefault`.

Hodnoty `\familydefault` jsou `\rmdefault`, `\sfdefault`, `\ttdefault`.

Hodnoty `\seriesdefault` jsou `\bfdefault`, `\mddefault` a teprve jejich hodnoty jsou `bx` a `m`.

Podobně hodnoty `\shapedefault` jsou `\itdefault`, `\sldefault`, `\scdefault`, `\updefault` a teprve jejich hodnoty jsou `it`, `sl`, `sc`, `n`.

Speciální příkazy pro volbu fontu

Příkaz `\DeclareFixedFont{cmd}{encod.}{family}{series}{shape}{size}` umožňuje definovat příkaz pro zavedení vybraného fontu. Např. font `msam10` Americké matematické společnosti můžeme přiřadit přepínači `\znacky` příkazem `\DeclareFixedFont{\znacky}{U}{msa}{m}{n}{10}`. Konstrukce `\znacky abcd` pak vysází $\wedge\in\exists\psi$.

Příkaz s parametrem pro volání fontu můžeme na této úrovni v preambuli dokumentu deklarovat příkazem `\DeclareTextFontCommand{cmd}{font-switches}`, např. `\DeclareTextFontCommand{\mujtext}{\sffamily\ffseries}`. Napíšeme-li `\mujtext{abcd}` dostaneme **abcd**.

Propojení mezi touto úrovní a primitivem `font`, na nějž všechny tyto značky expandují, zajišťuje fontový vzor (Font Design) – soubor s příponou `fd`. Chceme-li používat `cs`-fonty v \LaTeX u bez českého stylu `czech`, připravíme si souboru `OT1CSR.FD`

```
%%
%% This is file 'OT1csr.fd', generated
%% on <1994/12/30> with the docstrip utility (2.2i).
%%
%% The original source files were:
%%
%% cmfonts.fdd (with options: 'fd,OT1cmr')
%%
%% Copyright (C) 1994 LaTeX3 project, Frank Mittelbach and Rainer Schoepf.
%% All rights reserved.
%%
%% This file is part of LaTeX2e.
%% -----
```

```

%%
%% This is a generated file. Permission is granted to to customize the
%% declarations in this file to serve the needs of your installation.
%% However, no permission is granted to distribute a modified version of
%% this file under its original name.
%%
%% If you want to customize this file, it is best to make a copy of the
%% source file cmfonts.fdd under a different name and modify this copy.
%% Otherwise your changes might get lost when you install a new release
%% of LaTeX2e.
%%
\ProvidesFile{OT1csr.fd}
      [1994/12/02 v2.3e Standard LaTeX font definitions]
\DeclareFontFamily{OT1}{csr}{}
\DeclareFontShape{OT1}{csr}{m}{n}
  { <5> <6> <7> <8> <9> <10> <12> gen * csr
    <10.95> csr10
    <14.4> csr12
    <17.28><20.74><24.88>csr17}{}
\DeclareFontShape{OT1}{csr}{m}{sl}
  {
    <5><6><7>cssl8%
    <8> <9> gen * cssl
    <10><10.95>cssl10%
    <12><14.4><17.28><20.74><24.88>cssl12%
  }{}
\DeclareFontShape{OT1}{csr}{m}{it}
  {
    <5><6><7>csti7%
    <8>csti8%

```

```

    <9>csti9%
    <10><10.95>csti10%
    <12><14.4><17.28><20.74><24.88>csti12%
  }{}

\DeclareFontShape{OT1}{csr}{m}{sc}
  {
    <5><6><7><8><9><10><10.95><12>
    <14.4><17.28><20.74><24.88> cscsc10
  }{}

% Warning: please note that the upright shape below is
%         used for the \pounds symbol of LaTeX. So this
%         font definition shouldn't be removed.
%
\DeclareFontShape{OT1}{cmr}{m}{ui}
  {
    <5><6><7><8><9><10><10.95><12>%
    <14.4><17.28><20.74><24.88>cmu10%
  }{}

%%%%%%%%% bold series

\DeclareFontShape{OT1}{csr}{b}{n}
  {
    <5><6><7><8><9><10><10.95><12>
    <14.4><17.28><20.74><24.88> csb10
  }{}

%%%%%%%%% bold extended series

\DeclareFontShape{OT1}{csr}{bx}{n}
  {
    <5> <6> <7> <8> <9> gen * csbx
    <10><10.95> csbx10
    <12><14.4><17.28><20.74><24.88>csbx12
  }{}

```

```

    }{}
\DeclareFontShape{OT1}{csr}{bx}{sl}
  {
    <5> <6> <7> <8> <9>
    <10> <10.95> <12> <14.4> <17.28> <20.74> <24.88> csbxs110
  }{}
\DeclareFontShape{OT1}{csr}{bx}{it}
  {
    <5> <6> <7> <8> <9>
    <10> <10.95> <12> <14.4> <17.28> <20.74> <24.88> csbxti10
  }{}
% Again this is necessary for a correct \pounds symbol in
% the cmr fonts Hopefully the dc/ec font layout will take
% over soon.
%
\DeclareFontShape{OT1}{cmr}{bx}{ui}
  {<->ssub * cmr/m/ui}{}
\endinput
%%
%% End of file 'OT1csr.fd'.

```

Příkaze `\renewcommand{\rmdefault}{csr}` pak nahradí Knuthův americký `cmr`, českým `csr`.

4.8. Zařazení fontu do matematické sazby

Deklarace fontů pro sazbu matematiky je velmi odlišná od deklarace fontů pro sazbu v textovém režimu. I když 5 atributů fontům stále zůstává, v matematické sazbě fonty vystupují v roli matematické abecedy *math alphabets* nebo matematických symbolů *math symbol fonts*.

Pro matematickou abecedu jsou předdefinovány příkazy s jedním parametrem:

Abeceda	Popis	Ukázka
<code>\mathnormal</code>	default	<i>abcXYZ</i>
<code>\mathrm</code>	roman	abcXYZ
<code>\mathbf</code>	bold roman	abcXYZ
<code>\mathsf</code>	sans serif	abcXYZ
<code>\mathit</code>	text italic	<i>abcXYZ</i>
<code>\mathtt</code>	typewriter	abcXYZ
<code>\mathcal</code>	calligraphic	\mathcal{XYZ}

Na vysvětlení posledního řádku přikládáme výpis fontu cmsy10:

0: $-$	1: \cdot	2: \times	3: $*$	4: \div	5: \diamond	6: \pm	7: \mp	8: \oplus	9: \ominus
10: \otimes	11: \odot	12: \odot	13: \circ	14: \circ	15: \bullet	16: \asymp	17: \equiv	18: \subseteq	19: \supseteq
20: \leq	21: \geq	22: \preceq	23: \succeq	24: \sim	25: \approx	26: \subset	27: \supset	28: \ll	29: \gg
30: \prec	31: \succ	32: \leftarrow	33: \rightarrow	34: \uparrow	35: \downarrow	36: \leftrightarrow	37: \nearrow	38: \searrow	39: \simeq
40: \Leftarrow	41: \Rightarrow	42: \Uparrow	43: \Downarrow	44: \Leftrightarrow	45: \curvearrowright	46: \sphericalangle	47: \propto	48: $!$	49: ∞
50: \in	51: \ni	52: \triangle	53: ∇	54: $/$	55: \dagger	56: \forall	57: \exists	58: \neg	59: \emptyset
60: \Re	61: \Im	62: \top	63: \perp	64: \aleph	65: \mathcal{A}	66: \mathcal{B}	67: \mathcal{C}	68: \mathcal{D}	69: \mathcal{E}
70: \mathcal{F}	71: \mathcal{G}	72: \mathcal{H}	73: \mathcal{I}	74: \mathcal{J}	75: \mathcal{K}	76: \mathcal{L}	77: \mathcal{M}	78: \mathcal{N}	79: \mathcal{O}
80: \mathcal{P}	81: \mathcal{Q}	82: \mathcal{R}	83: \mathcal{S}	84: \mathcal{T}	85: \mathcal{U}	86: \mathcal{V}	87: \mathcal{W}	88: \mathcal{X}	89: \mathcal{Y}
90: \mathcal{Z}	91: \cup	92: \cap	93: \uplus	94: \wedge	95: \vee	96: \vdash	97: \dashv	98: \lfloor	99: \rfloor
100: \lceil	101: \rceil	102: $\{$	103: $\}$	104: \langle	105: \rangle	106: $ $	107: \parallel	108: \updownarrow	109: \Updownarrow
110: \backslash	111: \wr	112: \surd	113: \amalg	114: ∇	115: \int	116: \sqcup	117: \sqcap	118: \sqsubseteq	119: \sqsupseteq
120: \S	121: \dagger	122: \ddagger	123: \clubsuit	124: \clubsuit	125: \diamond	126: \heartsuit	127: \spadesuit		

Předdefinované matematické symboly jsou zařazeny do jedné z těchto tříd:

Symbolický font	Popis	Ukázka
operators	symboly z <code>\mathrm</code>	$[+]$
letters	symboly z <code>\mathnormal</code>	$\langle f \rangle$
symbols	většina symbolů	$\leq * \geq$
largesymbols	velké symboly	$\Sigma \Pi f \cup \cap$

Některé znakové sady vystupují v roli matematické abecedy i matematických symbolů. Matematickým symbolům je přiřazena verze (math version). Jsou předdefinovány 2 verze `normal` – standardní verze (default) – a `bold` – polotučné znaky. Přepínat verze lze příkazem `\mathversion{version}`. Pro `\mathversion{bold}` existuje starší přepínač `\boldmath`.

Pro jiné verze musíme připravit fontový vzor a písmo připojit.

Například pro font rsfs je obsah souboru OMSRSFS.FD

```
%% This is file 'OMSrsfs.fd'
\def\fileversion{2.1}
\def\filedate{94/04/20}
\def\docdate{94/04/15}
%%
\wlog{Fontdef\space file\space OMSrsfs\space (\filedate)}
\typeout{File \space OMSrsfs.fd\space loading \space rsfs\script}%
\DeclareFontFamily{OMS}{rsfs}{\skewchar\font'60}%
\DeclareFontShape{OMS}{rsfs}{m}{n}{%
  <5> rsfs5
  <6> <7> rsfs7
  <8> <9> <10> rsfs10
  <10.95> <12> <14.4> <17.28> <20.74> <24.88> rsfs10
}{\}
\endinput
%% End of file.
```

a jeho připojení je zařízeno knihovnou script.sty:

```
%%% rsfs script
\DeclareSymbolFont{rsfs\script}{OMS}{rsfs}{m}{n}
\DeclareSymbolFontAlphabet{\script}{rsfs\script}
\endinput
```

Připojíme-li knihovnu script.sty a napíšeme-li \script{ABC} dostaneme *ABC*.

Postskriptové fonty v L^AT_EXu – NFSS

V textovém režimu jsou kdispozici styly

- AvantGarde — avant.sty
- Bookman — bookman.sty

- Courier — courier.sty
- Helvetica — helvet.sty
- New Century Schoolbook — newcent.sty
- Palatino — palatino.sty
- Times Roman — times.sty
- ZapfChancery — chancery.sty
- ZapfDingbat — pifont.sty

po jejichž načtení je sazba provedena daným fontem.

Podobně pro změnu matematické sazby existuje např. knihovna mathptm, resp. mathptmx, která sází matematiku ve fontu Times. Podrobně je problém připojení postskriptových fontů do matematicky řešen v souboru psfonts.pdf v dokumentaci `texmf-dist\doc\latex\psnfss\psfonts.pdf`. Např. použít font Palatino při sazbě matematiky lze po deklaraci v preambuli:

```
\DeclareSymbolFont{operators} {OT1}{ppl}{m}{n}
\SetSymbolFont{operators}{bold} {OT1}{ppl}{b}{n}
\DeclareMathAlphabet{\mathip} {OT1}{ppl}{m}{it}
\SetMathAlphabet{\mathip}{bold} {OT1}{ppl}{b}{it}
```

Příkazem `\mathip{...}` tímto fontem sázíme matematiku: `$_\mathip{A+b=C}$` dá $A + b = C$, ve verzi bold `\mathversion{bold}_\mathip{A+b=C}` dá $A + \mathbf{b} = \mathbf{C}$.

Analogicky bychom mohli pracovat se všemi uvedenými postskriptovými fonty.

Na závěr je třeba poznamenat, že $\text{\LaTeX} 2_{\epsilon}$ se neustále vyvíjí a předdefinované styly se mohou změnit, knihovny mohou měnit jméno, mohou vznikat nové styly.

Závěr

Na začátku práce jsem se stručně zmínila o historii typografie. Domnívám se, že je dobré vědět, kdy a kde se typografie objevila.

Ve druhé kapitole jsem krátce pojednala o systému $\text{T}_{\text{E}}\text{X}$, jenž je dnes jedním z nejdokonalejších systémů pro počítačovou sazbu. Ve třetí kapitole jsem se zmínila o METAFONTu , který je pro generování „Knuthových fontů“ v systému $\text{T}_{\text{E}}\text{X}$ nezbytný.

Psaní čtvrté kapitoly bylo pro mne již náročnější. Bylo těžké seřadit všechny informace do přehledného a srozumitelného textu. Způsob práce programu $\text{T}_{\text{E}}\text{X}$ s fonty je poměrně složitý, pronikáte-li hlouběji do této problematiky, narážíte na další a další problémy. Bylo těžké sehnat literaturu, ve které by bylo vše přehledně sepsáno, musela jsem čerpat z mnoha knížek, které ne vždy obsahují dostatek informací. Také proto musím znovu poděkovat vedoucímu mé práce RNDr. Miloslavu Závodnému za poskytnutou literaturu a spolupráci, které si velice cením.

Ve čtvrté kapitole jsem tedy pojednala o nejpoužívanějších Knuthových Computer Modern fontech a uvedla malou ukázkou jeho 7bitového fontu `cmr10` a jeho 8bitové verze z CS-fontů. Ty jsou upraveny pro češtinu – obohaceny o akcentované znaky. Dále jsem představila metriku fontu, která je pro činnost $\text{T}_{\text{E}}\text{Xu}$ nejdůležitější součástí fontů, protože $\text{T}_{\text{E}}\text{X}$ pracuje výhradně s metrikou. V ní jsou uloženy veškeré informace o boxech, z nichž je sestaven `dvi` soubor – abstraktní popis vzhledu sazby podle Knutha. Dále jsem ukázala, jak připravit tuto metriku z metricky `afm` firmy Adobe dodávanou k jejím postskriptovým fontům – konverzního programu `afm2tfm`. Pojednala jsem i o virtuálních fontech, které mohou změnit původní kódování fontu, poskládat znaky z jiných znaků apod. V další části jsem popsala standardní postscriptové fonty. Předvedla jsem, jak zařadit fonty do `plainTEXu`, a to jak v textovém, tak i v matematickém režimu. V tomto duchu jsem zpracovala i zařazení fontů do $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{Xu}$.

Důležité bylo ukázat schéma NFSS, které je dnes standardem pro práci s fonty v $\text{\LaTeX} 2_{\varepsilon}$. I zde mně práci komplikoval zásadní rozdíl v přístupu NFSS k matematickým a textovým fontům.

Práce na zadané problematice byla pro mě velmi zajímavá a přínosná.

Literatura

- [1] Rybička, J.: LATEX pro začátečníky (2. vydání). Konvoj, Brno, 1993.
- [2] Olšák, P.: T_EXbook naruby. Konvoj, Brno, 2001.
- [3] Olšák, P.: Typografický systém T_EX. CSTug, Praha, 1995.
- [4] Olšák, P.: Manuál k C_ST_EXu. Dokumentace T_EXLive: `..\texmf-dist\doc\cstex\cstexman.pdf`.
- [5] Doob, M.: JEMNÝ ÚVOD DO T_EXu. Karolinum, Praha, 1990.
- [6] Oetiker, T.: Ne příliš stručný úvod do systému L^AT_EX 2_ε.
- [7] Bakalářská práce, Šolcová T., Databáze v T_EXu (Zpracování výsledků soutěže)
- [8] Diplomová práce, Černý M., Znakové sady v typografických systémech
- [9] L^AT_EX 2_εfont selection. Dokumentace T_EXLive: `..\texmf-dist\doc\latex\base\fntguide.pdf`, 2005.
- [10] http://sk.wikipedia.org/wiki/Computer_Modern
- [11] <http://technologie.webgarden.cz/rubriky/graficke-techniky/vznik-tisku-a-tiskovych-pisem>
- [12] http://technet.idnes.cz/knihtisk-zacatek-informacni-revoluce-d8e-/tec_technika.aspx?c=A070216_183142_tec_technika_pka
- [13] <http://cs.wikipedia.org/wiki/Sazba>