```
Footnotes:
1:
                                                                                                                                                     1: char.h:42
2:
            This file is a part of Ftx2TeX, a convertor from "ftx" source files
                                                                                                                                                     2: char.h:44
3:
            to TeX-readable files. Ftx2TeX is a part of FarsiTeX, a Persian/English
4:
          * typesetting system.
5:
6:
            Copyright (C) 1996 Mohammad Mahdian <mahdian@mit.edu>
7:
            Copyright (C) 1998-2001 Roozbeh Pournader <roozbeh@sharif.edu>
8:
9:
          * FarsiTeX is free software; you can redistribute it and/or modify
10:
          * it under the terms of the GNU General Public License as published by
11:
            the Free Software Foundation; either version 2 of the License, or
12:
             (at your option) any later version.
13:
14:
          * FarsiTeX is distributed in the hope that it will be useful,
15:
            but WITHOUT ANY WARRANTY; without even the implied warranty of
16:
          * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17:
            GNU General Public License for more details.
18:
19:
          * You should have received a copy of the GNU General Public License
20:
          * along with FarsiTeX; if not, write to the Free Software
21:
          * Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307,
22:
            USA.
23:
24:
          * Any licensing or usage questions should be directed to Roozbeh
25:
            Pournader <roozbeh@sharif.edu>.
26:
          * /
27:
28:
         #include "char.h"
29:
30:
        char e[NUM_ERAB<sup>1</sup>] = "";
31:
32:
         struct chars<sup>2</sup> ch[256] =
33:
34:
             {0, 0, 1, 1, 0},
35:
             {1, 1, 1, 1, 0},
                                                            \fathe
                                           11
                                                   1
36:
                                                   2
             {2, 2, 1, 1, 0},
                                                            \kasre
37:
                                                   3
             {3, 3, 1, 1, 0},
                                                            \zamme
38:
             {4, 4, 1, 1, 0},
                                           11
                                                            \nasb
39:
                                                   5
             {5, 5, 1, 1, 0},
                                                            \tashdid
40:
             {6, 6, 1, 1, 0},
                                           11
                                                            \aleph
41:
             {7, 7, 1, 1, 0},
                                           11
                                                   7
                                                            \hamze
                                                                             11
42:
             {8, 8, 1, 1, 0},
43:
             {9, 9, 1, 1, 0},
44:
             {10, 10, 1, 1, 0},
                                           11
                                                   10
45:
             {11, 11, 1, 1, 0},
                                           11
                                                   11
46:
             {12, 12, 1, 1, 0},
                                          11
                                                   12
                                                       t gerd
                                                                             //
47:
             {13, 13, 1, 1, 0},
                                           11
                                                   13
48:
             {14, 14, 1, 1, 0},
                                           11
                                                   14
                                                                              11
49:
             {15, 15, 1, 1, 0},
                                          //
                                                   15
50:
             {16, 16, 1, 1, 0},
                                           11
                                                   16
51:
             {17, 17, 1, 1, 0},
                                           11
                                                   17
52:
             {18, 18, 1, 1, 0},
                                          11
                                                   18
53:
             {19, 19, 1, 1, 1},
                                                   19 taa aakhar
                                                                             11
```

	{20, 20, 1, 1, 1},	//	20 zaa aak	har	//
:	{21, 21, 1, 1, 0},	//	21 '		//
i:	{22, 22, 1, 1, 0},	//	22		
' :	{23, 23, 1, 1, 0},	//	23 *		//
3:	{24, 24, 1, 1, 0},	//	24		
):	{25, 25, 1, 1, 0},	//	25 ~		//
):	{26, 26, 1, 1, 0},	//	26		
:	{27, 27, 1, 1, 0},	//	27 +		//
2:	{28, 28, 1, 1, 0},	//	28		
3:	{29, 29, 1, 1, 0},	//	29 <	Space>	//
l:	{30, 30, 1, 1, 0},	//	30		
i:	{31, 31, 1, 1, 0},	//	31		
i:	{32, 32, 1, 0, 0},	//	32		
' :	{33, 33, 1, 0, 0},	// !	33		
3:	{34, 34, 1, 0, 0},	// "	34		
):	{35, 35, 1, 0, 0},	// #	35		
):	{36, 36, 1, 0, 0},	// \$	36		
:	{37, 37, 1, 0, 0},	// %	37		
2:	{38, 38, 1, 0, 0},	// &	38		
3:	{39, 39, 1, 0, 0},	// '	39		
l:	{40, 40, 1, 0, 0},	// (40		
i:	{41, 41, 1, 0, 0},	//)	41		
i:	{42, 42, 1, 0, 0},	// *	42		
' :	{43, 43, 1, 0, 0},	// +	43		
3:	{44, 44, 1, 0, 0},	// ,	44		
):	{45, 45, 1, 0, 0},	// -	45		
:	{46, 46, 1, 0, 0},	// .	46		

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

11 1

// 1

// 2

// 3

// 4

// 5

// 6

// 7

// 8

// 9

//:

// ;

// <

// =

// ?

// @

// A

// B

// C

// D

// E

// F

// G

// H

81:

82:

83:

84:

85:

86:

87:

88:

89:

90:

91:

92:

93:

94:

95:

96:

97:

98:

99:

100:

101:

102:

103:

104:

105:

106:

{47, 47, 1, 0, 0},

{48, 48, 1, 0, 0},

{49, 49, 1, 0, 0},

{50, 50, 1, 0, 0},

{51, 51, 1, 0, 0},

{52, 52, 1, 0, 0},

{53, 53, 1, 0, 0},

{54, 54, 1, 0, 0},

{55, 55, 1, 0, 0},

{56, 56, 1, 0, 0},

{57, 57, 1, 0, 0},

{58, 58, 1, 0, 0},

{59, 59, 1, 0, 0},

{60, 60, 1, 0, 0},

{61, 61, 1, 0, 0},

{62, 62, 1, 0, 0},

{63, 63, 1, 0, 0},

{64, 64, 1, 0, 0},

{65, 65, 1, 0, 1},

{66, 66, 1, 0, 1},

{67, 67, 1, 0, 1},

{68, 68, 1, 0, 1},

{69, 69, 1, 0, 1},

{70, 70, 1, 0, 1},

{71, 71, 1, 0, 1},

{72, 72, 1, 0, 1},

Modified on	Sun Jul	16 1	12:51:	36 2006

ivic	dilica dii dali dai 10 12.01.00 2000		
107:	{73, 73, 1, 0, 1},	// I	73
108:	{74, 74, 1, 0, 1},	// J	74
109:	{75, 75, 1, 0, 1},	// K	75
110:	{76, 76, 1, 0, 1},	// L	76
111:	{77, 77, 1, 0, 1},	// M	77
112:	{78, 78, 1, 0, 1},	// N	78
113:	{79, 79, 1, 0, 1},	// 0	79
114:	{80, 80, 1, 0, 1},	// P	80
115:	I : : : : : : : : : : : : : : : : : : :		
	{81, 81, 1, 0, 1},	// Q	81
116:	{82, 82, 1, 0, 1},	// R	82
117:	{83, 83, 1, 0, 1},	// S	83
118:	{84, 84, 1, 0, 1},	// T	84
119:	{85, 85, 1, 0, 1},	// U	85
120:	{86, 86, 1, 0, 1},	// V	86
121:	{87, 87, 1, 0, 1},	// W	87
122:	{88, 88, 1, 0, 1},	// X	88
123:	{89, 89, 1, 0, 1},	// Y	89
124:	{90, 90, 1, 0, 1},	// Z	90
125:	{91, 91, 1, 0, 0},	// [91
126:	{92, 92, 1, 0, 0},	// \	92
127:	{93, 93, 1, 0, 0},	//]	93
128:	{94, 94, 1, 0, 0},	// ^	94
129:	{95, 95, 1, 0, 0},	// _	95
130:	{96, 96, 1, 0, 0},	// =	96
131:	{97, 97, 1, 0, 1},	// a	97
132:	{98, 98, 1, 0, 1},	// b	98
133:	{99, 99, 1, 0, 1},	// c	99
134:	{100, 100, 1, 0, 1},	// d	100
135:	{101, 101, 1, 0, 1},	// e	101
136:	{102, 102, 1, 0, 1},	// f	102
137:	{103, 103, 1, 0, 1},	// g	103
138:	{104, 104, 1, 0, 1},	// h	104
139:	{105, 105, 1, 0, 1},	// i	105
140:	{106, 106, 1, 0, 1},	// j	106
141:	{107, 107, 1, 0, 1},	// k	107
142:	{108, 108, 1, 0, 1},	// 1	108
143:	{109, 109, 1, 0, 1},		
		// m	109
144:	{110, 110, 1, 0, 1},	// n	110
145:	{111, 111, 1, 0, 1},	// 0	111
146:	{112, 112, 1, 0, 1},	// p	112
147:	{113, 113, 1, 0, 1},	// q	113
148:	{114, 114, 1, 0, 1},	// r	114
149:	{115, 115, 1, 0, 1},	// s	115
150:	{116, 116, 1, 0, 1},	// t	116
151:	{117, 117, 1, 0, 1},	// u	117
152:			118
	{118, 118, 1, 0, 1}, {119, 119, 1, 0, 1},	// v	
153:		// w	119
154:	{120, 120, 1, 0, 1},	// x	120
155:	{121, 121, 1, 0, 1},	// y	121
156:	{122, 122, 1, 0, 1},	// z	122
157:	{123, 123, 1, 0, 0},	// {	123
158:	{124, 124, 1, 0, 0},	// Í	124
159:	{125, 125, 1, 0, 0},	// }	125
		. ,	

```
160:
                                             // ~
               {126, 126, 0, 0, 0},
                                                      126
161:
               {127, 127, 1, 0, 0},
                                              11
                                                      127
162:
               {48, 48, 1, 1, 0},
                                              11
                                                      128 zero
163:
               {49, 49, 1, 1, 0},
                                              11
                                                      129 one
164:
               {50, 50, 1, 1, 0},
                                              11
                                                      130
165:
               {51, 51, 1, 1, 0},
                                              11
                                                      131
166:
               {52, 52, 1, 1, 0},
                                              11
                                                      132
167:
               {53, 53, 1, 1, 0},
                                              11
                                                      133
168:
               {54, 54, 1, 1, 0},
                                              11
                                                      134
169:
               {55, 55, 1, 1, 0},
                                              11
                                                      135
170:
               {56, 56, 1, 1, 0},
                                              //
                                                      136
171:
               {57, 57, 1, 1, 0},
                                              11
                                                      137 nine
172:
               {44, 44, 1, 1, 0},
                                              11
                                                      138 fcomma
173:
               {139, 139, 0, 1, 0},
                                                      139 -
                                              11
174:
               {63, 63, 1, 1, 0},
                                              //
                                                      140
175:
               {141, 141, 1, 1, 1},
                                              11
                                                      141 aa
176:
               {142, 255, 0, 1, 1},
                                              11
                                                     142 hamze
177:
                                                      143 hamze tanhaa
               {143, 143, 1, 1, 1},
                                              11
178:
               {144, 144, 1, 1, 1},
                                              11
                                                      144 aleph avval
179:
                                                      145 aleph aakhar
               {145, 145, 1, 1, 1},
                                              11
180:
               {146, 176, 1, 1, 1},
                                              //
                                                      146 b bozorg
181:
               {147, 177, 0, 1, 1},
                                              11
                                                      147 b koochek
182:
               {148, 178, 1, 1, 1},
                                              11
                                                      148 p bozorg
183:
               {149, 179, 0, 1, 1},
                                              //
                                                      149 p k
184:
               {150, 180, 1, 1, 1},
                                              11
                                                      150 t b
185:
               {151, 181, 0, 1, 1},
                                              11
                                                      151 t k
186:
                                              //
               {152, 182, 1, 1, 1},
                                                      152 s b
187:
               {153, 183, 0, 1, 1},
                                              //
                                                      153 s k
188:
               {154, 184, 1, 1, 1},
                                              11
                                                      154 j b
189:
               {155, 185, 0, 1, 1},
                                              11
                                                      155 j k
190:
               {156, 186, 1, 1, 1},
                                              11
                                                      156 ch b
191:
               {157, 187, 0, 1, 1},
                                              //
                                                      157 ch k
192:
                                              11
               {158, 188, 1, 1, 1},
                                                      158 h b
193:
               {159, 189, 0, 1, 1},
                                              11
                                                      159 h k
194:
               {160, 190, 1, 1, 1},
                                              11
                                                      160 kh b
195:
               {161, 191, 0, 1, 1},
                                              11
                                                      161 kh k
196:
               {162, 192, 1, 1, 1},
                                                      162 dall
197:
               {163, 193, 1, 1, 1},
                                              11
                                                      163 zall
198:
               {164, 194, 1, 1, 1},
                                              11
                                                      164 r
199:
               {165, 195, 1, 1, 1},
                                              11
                                                      165 z
200:
               {166, 196, 1, 1, 1},
                                              11
                                                      166 zh
201:
               {167, 197, 1, 1, 1},
                                                      167 sin b
202:
               {168, 198, 0, 1, 1},
                                              11
                                                      168 sin k
203:
                                              //
               {169, 199, 1, 1, 1},
                                                      169 shin b
204:
               {170, 200, 0, 1, 1},
                                              11
                                                      170 shin k
205:
               {171, 201, 1, 1, 1},
                                              11
                                                      171 sad b
206:
               {172, 202, 0, 1, 1},
                                              11
                                                      172 sad k
207:
               {173, 203, 1, 1, 1},
                                              11
                                                      173 zad b
208:
               {174, 204, 0, 1, 1},
                                              11
                                                      174 zad k
209:
               {207, 206, 0, 1, 1},
                                              11
                                                      175 taa
210:
               {176, 176, 1, 1, 0},
                                              //
                                                      176 \fathe
211:
               {177, 177, 1, 1, 0},
                                             11
                                                      177 \kasre
212:
               {178, 178, 1, 1, 0},
                                              11
                                                      178 \zamme
```

```
213:
               {179, 179, 1, 1, 0},
                                             11
                                                      179 \nasb
214:
               {180, 180, 1, 1, 0},
                                             11
                                                      180 \tashdid
215:
               {35, 35, 1, 1, 0},
                                             11
                                                      181
216:
               {36, 36, 1, 1, 0},
                                             11
                                                      182
                                                                   $
217:
               {37, 37, 1, 1, 0},
                                             11
                                                      183
218:
               {38, 38, 1, 1, 0},
                                                      184
219:
               {39, 39, 1, 1, 0},
                                                      185
220:
               {186, 186, 1, 1, 0},
                                                      186 \alef
221:
               {187, 187, 1, 1, 0},
                                             11
                                                      187 \hamze
222:
               {188, 188, 1, 1, 0},
                                             11
                                                      188 momayyez
223:
               {40, 40, 1, 1, 0},
                                             11
                                                      189
224:
               {41, 41, 1, 1, 0},
                                             11
                                                      190
225:
               {134, 135, 1, 1, 1},
                                                      191 t gerd
226:
               {34, 34, 1, 1, 0},
                                                      192 " basteh farsi
                                             11
227:
               {175, 205, 1, 1, 1},
                                             11
                                                      193 taa aakhar
228:
               {224, 208, 1, 1, 1},
                                             11
                                                      194 zaa aakhar
229:
               {39, 39, 1, 1, 0},
                                             11
                                                      195 " baaz farsi
230:
               {196, 196, 1, 1, 0},
                                             11
                                                      196 \saken
231:
               {45, 45, 1, 1, 0},
                                                      197 farsi dash (-)
232:
               {46, 46, 1, 1, 0},
                                             11
                                                      198
233:
               {47, 47, 1, 1, 0},
                                             //
                                                      199
234:
               {42, 42, 1, 1, 0},
                                             11
                                                      200 farsi *
235:
                                                      201 farsi ~
               {126, 126, 1, 1, 0},
236:
               {58, 58, 1, 1, 0},
                                             //
                                                      202
237:
               {59, 59, 1, 1, 0},
                                                      203
                                             11
238:
               {62, 62, 1, 1, 0},
                                             11
                                                      204
                                                                         --> changed!
239:
               {43, 43, 1, 1, 0},
                                             //
                                                      205 farsi +
240:
               {61, 61, 1, 1, 0},
                                             //
                                                      206
241:
               {60, 60, 1, 1, 0},
                                             11
                                                      207
                                                                         --> changed!
242:
               {64, 64, 1, 1, 0},
                                                                   @
                                                      208
243:
               {93, 93, 1, 1, 0},
                                             11
                                                      209
244:
               {92, 92, 1, 1, 0},
                                             11
                                                      210
                                                                   1/1
245:
               {91, 91, 1, 1, 0},
                                             11
                                                      211
               {94, 94, 1, 1, 0},
246:
                                             11
                                                      212
247:
               {95, 95, 1, 1, 0},
                                             11
                                                      213
248:
               {96, 96, 1, 1, 0},
                                                      214
249:
               {125, 125, 1, 1, 0},
                                                      215
250:
               {124, 124, 1, 1, 0},
                                             11
                                                      216
251:
               {217, 217, 1, 1, 0},
                                             11
                                                      217
252:
               {32, 32, 1, 1, 0},
                                                      218 farsi space
253:
               {219, 219, 1, 1, 0},
                                                      219
254:
               {220, 220, 1, 1, 0},
                                                      220
                                                                <Space>
                                                                                       //
255:
               {33, 33, 1, 1, 0},
                                             11
                                                      221
256:
                                             //
               {123, 123, 1, 1, 0},
                                                      222
257:
               {223, 223, 0, 1, 0},
                                             11
                                                      223
                                                                              11
258:
               {210, 209, 0, 1, 1},
                                             11
                                                      224 zaa
259:
               {225, 225, 1, 1, 1},
                                                      225 ein tanhaa
260:
               {226, 226, 1, 1, 1},
                                             11
                                                      226 ein aakhar
261:
               {227, 227, 0, 1, 1},
                                             11
                                                      227 ein vasat
262:
               {228, 228, 0, 1, 1},
                                             11
                                                      228 ein avval
263:
               {229, 229, 1, 1, 1},
                                             11
                                                      229 ghein tanhaa
264:
               {230, 230, 1, 1, 1},
                                             11
                                                      230 ghein aakhar
265:
               {231, 231, 0, 1, 1},
                                             11
                                                      231 ghein vasat
```

```
266:
               {232, 232, 0, 1, 1},
                                            11
                                                     232 ghein avval
267:
               {233, 211, 1, 1, 1},
                                            11
                                                     233 f b
268:
               {234, 212, 0, 1, 1},
                                                     234 f k
                                            11
269:
                                                     235 ghaaf b
               {235, 213, 1, 1, 1},
                                            11
270:
                                                     236 ghaaf k
               {236, 214, 0, 1, 1},
                                            11
271:
               {237, 215, 1, 1, 1},
                                            11
                                                     237 k b
272:
               {238, 216, 0, 1, 1},
                                            11
                                                     238 k k
273:
               {239, 217, 1, 1, 1},
                                                     239 g b
274:
               {240, 218, 0, 1, 1},
                                            11
                                                     240 g k
275:
               {241, 219, 1, 1, 1},
                                            11
                                                     241 l b
276:
               {132, 133, 1, 1, 1},
                                            11
                                                     242 laa
277:
               {243, 220, 0, 1, 1},
                                            11
                                                     243 1 k
278:
               {244, 221, 1, 1, 1},
                                                     244 mim b
279:
               {245, 222, 0, 1, 1},
                                            11
                                                     245 mim k
280:
               {246, 223, 1, 1, 1},
                                            11
                                                     246 n b
281:
               {247, 128, 0, 1, 1},
                                            11
                                                     247 n k
282:
               {248, 129, 1, 1, 1},
                                            11
                                                     248 v
283:
              {249, 130, 1, 1, 1},
                                            //
                                                     249 h aakhar
284:
               {250, 250, 0, 1, 1},
                                                     250 h vasat
285:
               {251, 251, 0, 1, 1},
                                            11
                                                     251 h avval
286:
              {252, 252, 1, 1, 1},
                                            11
                                                     252 y aakhar
287:
                                                     253 y tanhaa
               {253, 253, 1, 1, 1},
                                            11
288:
              {254, 131, 0, 1, 1},
                                            11
                                                     254 y vasat
289:
              {255, 255, 1, 1, 0}
                                            11
                                                     255
290:
```

```
Footnotes:
1:
                                                                                                                                                1: char.h:44
2:
         * This file is a part of Ftx2TeX, a convertor from "ftx" source files
                                                                                                                                                2: char.c:32
3:
         * to TeX-readable files. Ftx2TeX is a part of FarsiTeX, a Persian/English
                                                                                                                                                3: char.c:30
4:
         * typesetting system.
5:
6:
         * Copyright (C) 1996 Mohammad Mahdian <mahdian@mit.edu>
7:
            Copyright (C) 1998-2001 Roozbeh Pournader <roozbeh@sharif.edu>
8:
9:
         * FarsiTeX is free software; you can redistribute it and/or modify
10:
         * it under the terms of the GNU General Public License as published by
11:
         * the Free Software Foundation; either version 2 of the License, or
12:
         * (at your option) any later version.
13:
14:
         * FarsiTeX is distributed in the hope that it will be useful,
15:
         * but WITHOUT ANY WARRANTY; without even the implied warranty of
16:
         * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17:
            GNU General Public License for more details.
18:
19:
         * You should have received a copy of the GNU General Public License
20:
         * along with FarsiTeX; if not, write to the Free Software
21:
         * Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307,
22:
            USA.
23:
24:
         * Any licensing or usage questions should be directed to Roozbeh
25:
         * Pournader <roozbeh@sharif.edu>.
26:
         * /
27:
28:
        #define STRETCH
                          ((char) 137) // stretch character
29:
        #define F BSLASH ((char) 210) // farsi back slash
30:
        #define AT_SIGN
                          ((char) 208) // farsi at sign
31:
        #define FB_OPEN
                          ((char) 222) // farsi open brace
32:
        #define FB_CLOSE
                          ((char) 215) // farsi close brace
33:
        #define FK_OPEN
                           ((char) 211) // farsi open bracket
34:
        #define FK_CLOSE
                          ((char) 209) // farsi close bracket
35:
        #define FP_OPEN
                          ((char) 190) // farsi open parenthesis
36:
        #define FP_CLOSE ((char) 189) // farsi close parenthesis
37:
        #define F_MID
                           ((char) 216) // farsi vertical bar
38:
        #define F_SPC
                            ((char) 218) // farsi space
39:
        #define F_STAR
                            ((char) 200) // farsi star
40:
        #define F_MOM
                           ((char) 188) // farsi momayyez
41:
42:
        #define NUM_ERAB 8
43:
44:
        -struct chars {
45:
             char first;
                                         /* normal case of letter in changed alphabet */
46:
             char middle;
                                         /* middle case of letter in changed alphabet */
47:
            char last;
                                         /* indicates if the letter is in last case */
48:
             char farsi;
                                         /* indicates if the letter is a farsi letter */
49:
             char letter;
50:
51:
52:
        extern struct chars ch<sup>2</sup>[256];
53:
        extern char e<sup>3</sup>[];
```

```
Footnotes:
1:
                                                                                                                                                   1: char.h:42
2:
         * This file is a part of Ftx2TeX, a convertor from "ftx" source files
                                                                                                                                                   2: convert.c:36
3:
          * to TeX-readable files. Ftx2TeX is a part of FarsiTeX, a Persian/English
 4:
          * typesetting system.
 5:
6:
          * Copyright (C) 1996 Mohammad Mahdian <mahdian@mit.edu>
7:
          * Copyright (C) 1998-2001 Roozbeh Pournader <roozbeh@sharif.edu>
8:
9:
          * FarsiTeX is free software; you can redistribute it and/or modify
10:
          * it under the terms of the GNU General Public License as published by
11:
          * the Free Software Foundation; either version 2 of the License, or
12:
          * (at your option) any later version.
13:
14:
          * FarsiTeX is distributed in the hope that it will be useful,
15:
          * but WITHOUT ANY WARRANTY; without even the implied warranty of
16:
          * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17:
          * GNU General Public License for more details.
18:
19:
          * You should have received a copy of the GNU General Public License
20:
          * along with FarsiTeX; if not, write to the Free Software
21:
          * Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307,
22:
23:
24:
          * Any licensing or usage questions should be directed to Roozbeh
25:
          * Pournader <roozbeh@sharif.edu>.
26:
          * /
27:
28:
         #include <stdio.h>
29:
         #include <string.h>
30:
         #include <stdlib.h>
31:
         #include "char.h"
32:
        #include "convert.h"
33:
        #include "ftx2tex.h"
34:
35:
         #define MAX_LEN
                             500
                                          // Maximum length of a line
36:
         #define NUM_CMD1
                             13
37:
         #define NUM_CMD2
                             5
38:
39:
         int stretch = 1;
40:
         int noconvert = 0;
41:
        long int line no;
42:
        char *e_cmd[NUM_ERAB<sup>1</sup> + 1] =
43:
        -{"\\fathe ",
44:
          "\\kasre ",
45:
          "\\zamme ",
46:
         "\\nasb ",
47:
          "\\tashdid ",
48:
          "\\alef ",
49:
          "\\hamze ",
50:
          "\\saken ",
51:
         -"\\skasre "};
52:
53:
         char *command1[NUM_CMD1<sup>2</sup>] =
```

```
Footnotes:
 54:
             {"begin",
                                                                                                                                                                                               1: convert.c:37
 55:
             "è'\",
                                                                                                                                                                                               2: convert.c:75
 56:
              "end",
                                                                                                                                                                                               3: convert.c:35
 57:
              шп,
                                                                                                                                                                                               4: convert.c:91
 58:
                                                                                                                                                                                               5: convert.c:92
              "hspace",
                                                                                                                                                                                               6: convert.c:40
 59:
              "vspace",
                                                                                                                                                                                               7: convert.c:90
 60:
              "include",
                                                                                                                                                                                               8: convert.c:89
 61:
              "input",
                                                                                                                                                                                               9: convert.c:91
 62:
              "hspace*",
                                                                                                                                                                                               10: convert.c:94
 63:
              "vspace*",
                                                                                                                                                                                               11: convert.c:94
                                                                                                                                                                                               12: convert.c:83
 64:
              "label",
                                                                                                                                                                                               13: char.h:31
 65:
              "ref",
                                                                                                                                                                                               14: char.c:32
 66:
             -"cite"};
                                                                                                                                                                                               15: char.h:48
 67:
                                                                                                                                                                                               16: char.h:38
                                                                                                                                                                                               17: char.h:33
 68:
            char *command2[NUM CMD2<sup>1</sup>] =
 69:
             { "documentstyle",
 70:
              "documentclass",
 71:
 72:
              "usepackage",
 73:
             -"î\¤Â¢Æμ"};
 74:
 75:
            char *input_command[] =
 76:
 77:
                  "\\input",
                  "\\include"
 78:
 79:
 80:
 81:
            #define NUM_INPUT_CMD ((sizeof input_command²)/sizeof(char *))
 82:
 83:
            void fconvert(char *s)
 84:
 85:
                            This function converts a string in RL mode.
 86:
 87:
 88:
                  enum {
 89:
                       FIRST, MIDDLE
 90:
                  ·} state;
 91:
                  int i, j, k, l, temp, spc;
 92:
                  char t[MAX_LEN³], last[3];
 93:
                  char *cmd;
 94:
                 int is_cmd1, is_cmd2;
 95:
 96:
                  for (i^4 = 0; i^4 < MAX_LEN^3; i^4++)
 97:
                       t^{5}[i^{4}] = 0;
98:
                 if (noconvert<sup>6</sup>)
99:
                       return;
                  state<sup>7</sup> = FIRST<sup>8</sup>;
100:
                 -for (i<sup>4</sup> = j<sup>9</sup> = is_cmd1<sup>10</sup> = is_cmd2<sup>11</sup> = 0; s<sup>12</sup>[i<sup>4</sup>];) {
101:
                       if (s^{12}[i^4] == FB OPEN^{13})
102:
                            is cmd1^{10} = is cmd1^{10} == 1 ? 2 : 0;
103:
                       else if (ch<sup>14</sup>[(unsigned char) s<sup>12</sup>[i<sup>4</sup>]].farsi<sup>15</sup> && s<sup>12</sup>[i<sup>4</sup>] != F_SPC<sup>16</sup>)
104:
                            is\_cmd1^{10} = 0;
105:
                       if (s^{12}[i^4] == FK_OPEN^{17})
106:
```

```
Footnotes:
107:
                               is cmd2^1 = is cmd2^1 == 1 ? 2 : 0;
                                                                                                                                                                                                              1: convert.c:94
                         else if (s<sup>2</sup>[i<sup>3</sup>] == FK_CLOSE<sup>4</sup>)
108:
                                                                                                                                                                                                              2: convert.c:83
109:
                              is\_cmd2^1 = is\_cmd2^1 == 2 ? 3 : 0;
                                                                                                                                                                                                              3: convert.c:91
                         else if (s^2[i^3] == FB OPEN^5)
110:
                                                                                                                                                                                                              4: char.h:34
                               is cmd2^1 = is cmd2^1 & 1 ? 4 : 0;
                                                                                                                                                                                                              5: char.h:31
111:
                                                                                                                                                                                                              6: char.c:32
                         else if (ch<sup>6</sup>[(unsigned char) s<sup>2</sup>[i<sup>3</sup>]].farsi<sup>7</sup> && s<sup>2</sup>[i<sup>3</sup>] != F SPC<sup>8</sup>)
112:
                                                                                                                                                                                                              7: char.h:48
                              is\_cmd2^1 = 0;
113:
                                                                                                                                                                                                              8: char.h:38
114:
                                                                                                                                                                                                              9: char.h:40
115:
                        -if (ch<sup>6</sup>[(unsigned char) s<sup>2</sup>[i<sup>3</sup>]].farsi<sup>7</sup>) {
                                                                                                                                                                                                             10: convert.c:92
116:
                              -if (s^2[i^3] == F MOM^9) {
                                                                                                                                                                                                             11: convert.c:91
                                                                                                                                                                                                             12: char.c:30
                                    t^{10}[j^{11}] = | \setminus 0 | ;
117:
                                                                                                                                                                                                             13: convert.c:91
                                    strcat(t10, "\\mom ");
118:
                                                                                                                                                                                                             14: convert.c:92
                                     i^{11} = strlen(t^{10});
119:
                                                                                                                                                                                                             15: char.h:28
120:
                                     i^{3}++i
                                                                                                                                                                                                             16: convert.c:91
121:
                               \{-\} else if \{\text{strchr}(e^{12}, s^2[i^3])\} // e'raab
                                                                                                                                                                                                             17: convert.c:42
                                                                                                                                                                                                             18: char.h:35
122:
                                                                                                                                                                                                             19: char.h:36
                                     for (k^{13} = i^3; s^2[k^{13}] \&\& strchr(e^{12}, s^2[k^{13}]); k^{13}++);
123:
                                                                                                                                                                                                             20: char.h:37
124:
                                     t^{10}[i^{11}] = ' \setminus 0';
                                                                                                                                                                                                             21: char.h:45
                                     strcpy(last^{14}, t^{10} + (j^{11} - (t^{10}[j^{11} - 1] = STRETCH^{15}) + 1));
125:
                                                                                                                                                                                                             22: char.h:29
                                     if (last^{14}[0] == ' ' \&\& strchr(e^{12}, s^{2}[i^{3}]) == e^{12} + 1)
126:
                                                                                                                                                                                                             23: char.h:32
                                                                                                                                                                                                             24: convert.c:93
127:
                                          s^{2}[i^{3}] = ' \setminus 0';
                                                                                                                                                                                                             25: char.h:49
                                     for (temp^{16} = k^{13} - , t^{10}[j^{11}] = j \setminus 0; k^{13} >= j^3; k^{13} - )
128:
                                                                                                                                                                                                             26: convert.c:90
                                          strcat(t^{10}, e_cmd^{17}[strchr(e^{12}, s^2[k^{13}]) - e^{12}]);
129:
                                                                                                                                                                                                             27: convert.c:89
130:
                                    strcat(t<sup>10</sup>, last<sup>14</sup>);
                                                                                                                                                                                                             28: char.h:46
                                     i^{11} = strlen(t^{10});
131:
                                                                                                                                                                                                             29: char.h:47
                                                                                                                                                                                                             30: convert.c:89
132:
                                     i^3 = temp^{16};
                                                                                                                                                                                                             31: char.h:39
133:
                               \{ else if ((s^2[i^3] == FP_OPEN^{18} | | s^2[i^3] == FP_CLOSE^{19}) \& \& \} 
                                       ----i^3 != 0 \&\& s^2[i^3 - 1] == F_MID^{20}) 
134:
                                     if (s^2[i^3] == FP OPEN^{18})
135:
                                          s^{2}[i^{3}] = FP CLOSE^{19};
136:
137:
                                     else
                                          s^{2}[i^{3}] = FP OPEN^{18};
138:
139:
                                     t^{10}[j^{11}++] = ch^{6}[(unsigned char) s^{2}[i^{3}++]].first^{21};
140:
141:
                                    -if (i^3 \&\& s^2[i^3 - 1] == F_BSLASH^{22}) { // if it is a command
142:
143:
                                          if (s^2[i^3] == FB_OPEN^5)
                                                s^{2}[i^{3}] = FB CLOSE^{23};
144:
                                           else if (s^2[i^3] == FB CLOSE^{23})
145:
                                                s^{2}[i^{3}] = FB OPEN^{5};
146:
147:
                                          else {
                                               -for (k^{13} = i^3, cmd^{24} = t^{10} + j^{11}; ch^6[(unsigned char) s^2[i^3]].letter^{25};)
148:
                                                      t^{10}[j^{11}++] = (state^{26} = FIRST^{27}) ? ch^{6}[(unsigned char) s^{2}[i^{3}]].first^{21} : ch^{6}[(unsigned char) s^{2}[i^{3}]].middle^{28}
149:
                                                      state^{26} = ch^{6}[(unsigned char) s^{2}[i^{3}]].last^{29} ? FIRST^{27} : MIDDLE^{30};
150:
151:
                                                      i<sup>3</sup>++;
152:
                                                if (s^2[i^3] == F_STAR^{31})
153:
                                                      s^{2}[i^{3}] = **';
154:
                                                if (s^{2}[i^{3}] == '*')
155:
156:
                                                      t^{10}[j^{11}++] = s^2[i^3++];
                                                if (k^{13} == i^3)
157:
158:
                                                      t^{10}[j^{11}++] = ch^{6}[(unsigned char) s^{2}[j^{3}++]].first^{21};
```

```
Footnotes:
159:
                                                   t^{1}[\dot{j}^{2}] = \dot{j}^{0};
                                                                                                                                                                                                                         1: convert.c:92
                                                   for (1^3 = 0; 1^3 < NUM\_CMD1^4 \&\& strcmp(cmd^5, command1^6[1^3]);)
160:
                                                                                                                                                                                                                         2: convert.c:91
161:
                                                                                                                                                                                                                         3: convert.c:91
                                                   is cmd1^7 = 1^3 < NUM CMD1^4;
162:
                                                                                                                                                                                                                         4: convert.c:36
                                                   for (1^3 = 0; 1^3 < NUM CMD2^8 \&\& strcmp(cmd^5, command2^9[1^3]);)
163:
                                                                                                                                                                                                                         5: convert.c:93
                                                                                                                                                                                                                         6: convert.c:53
164:
                                                                                                                                                                                                                         7: convert.c:94
                                                   is cmd2^{10} = 1^3 < NUM_CMD2^8;
165:
                                                                                                                                                                                                                         8: convert.c:37
166:
                                                   continue;
                                                                                                                                                                                                                         9: convert.c:68
167:
                                                                                                                                                                                                                        10: convert.c:94
168:
                                                                                                                                                                                                                        11: convert.c:90
                                                                                                                                                                                                                        12: convert.c:89
                                       t<sup>1</sup>[j<sup>2</sup>++] = (state<sup>11</sup> == FIRST<sup>12</sup>) ? ch<sup>13</sup>[(unsigned char) s<sup>14</sup>[i<sup>15</sup>]].first<sup>16</sup> : ch<sup>13</sup>[(unsigned char) s<sup>14</sup>[i<sup>15</sup>]].middle<sup>17</sup>;
169:
                                                                                                                                                                                                                        13: char.c:32
                                       state^{11} = ch^{13}[(unsigned char) s^{14}[i^{15}]].last^{18}? FIRST^{12}: MIDDLE^{19};
170:
                                                                                                                                                                                                                        14: convert.c:83
                                      if (stretch<sup>20</sup> && !ch<sup>13</sup>[(unsigned char) s<sup>14</sup>[i<sup>15</sup>]].last<sup>18</sup>)
171:
                                                                                                                                                                                                                        15: convert.c:91
172:
                                            t^{1}[j^{2}++] = STRETCH^{21};
                                                                                                                                                                                                                        16: char.h:45
                                      i<sup>15</sup>++;
173:
                                                                                                                                                                                                                        17: char.h:46
                                                                                                                                                                                                                        18: char.h:47
174:
                                                                                                                                                                                                                        19: convert.c:89
175:
                         -} else {
                                                              // english
                                                                                                                                                                                                                        20: convert.c:39
176:
                                                                                                                                                                                                                        21: char.h:28
177:
                                for (k^{22} = i^{15}, spc^{23} = 1; s^{14}[k^{22}] \&\& !ch^{13}[(unsigned char) s^{14}[k^{22}]].farsi^{24}; spc^{23} \&= s^{14}[k^{22}++] == ! !);
                                                                                                                                                                                                                        22: convert.c:91
                                -if (i^{15} && s^{14}[i^{15} - 1] == AT_SIGN^{25} && s^{14}[k^{22}] == AT_SIGN^{25}) {
                                                                                                                                                                                                                        23: convert.c:91
178:
                                                                                                                                                                                                                        24: char.h:48
                                      for (s^{14}[k^{22}] = ' ', j^{2}--; i^{15} < k^{22}; t^{1}[j^{2}++] = s^{14}[i^{15}++]);
179:
                                                                                                                                                                                                                        25: char.h:30
                                      i 15++;
180:
                                                                                                                                                                                                                        26: char.h:29
                                -} else if (i^{15} \&\& s^{14}[i^{15} - 1] == F_BSLASH^{26})  { // if it is a command
181:
                                                                                                                                                                                                                        27: char.h:39
182:
                                      for (k^{22} = i^{15}, cmd^5 = t^1 + j^2; (s^{14}[i^{15}] >= 'A' && s^{14}[i^{15}] <= 'Z') | (s^{14}[i^{15}] >= 'a' && s^{14}[i^{15}] <= 'z'); t^1[j^2++] = s^{14}[i^{15}] <= 'z'
183:
              i<sup>15</sup>++]);
184:
                                      if (s^{14}[i^{15}] == F STAR^{27})
                                            s^{14}[i^{15}] = **';
185:
                                       if (s^{14}[i^{15}] == '*')
186:
                                            t^{1}[j^{2}++] = s^{14}[i^{15}++];
187:
                                      if (k^{22} == i^{15})
188:
                                            t^{1}[j^{2}++] = s^{14}[i^{15}++];
189:
190:
                                       t^{1}[j^{2}] = \lfloor \lfloor 0 \rfloor;
                                       for (1^3 = 0; 1^3 < NUM\_CMD1^4 \&\& strcmp(cmd^5, command1^6[1^3]); 1^3++);
191:
192:
                                       is cmd1^7 = 1^3 < NUM CMD1^4;
                                      for (1^3 = 0; 1^3 < \text{NUM CMD2}^8 \&\& \text{strcmp}(\text{cmd}^5, \text{command2}^9[1^3]); 1^3++);
193:
194:
                                       is\_cmd2^{10} = 1^3 < NUM\_CMD2^8;
195:
                                 } else {
                                       if (spc^{23} || is cmd1^7 & 6 || is cmd2^{10} & 6)
196:
197:
                                             for (; k^{22} - i^{15}; t^{1}[j^{2}++] = s^{14}[i^{15}++]);
198:
                                      else {
199:
                                            t^{1}[j^{2}] = ' \setminus 0';
                                             strcat(t<sup>1</sup>, "\\InE{}");
200:
201:
                                             j<sup>2</sup> = strlen(t<sup>1</sup>);
                                             for (; k^{22} - i^{15}; t^{1}[j^{2}++] = s^{14}[i^{15}++]);
202:
                                            t^{1}[\dot{j}^{2}] = ' \setminus 0';
203:
                                            strcat(t<sup>1</sup>, "\\EnE{}");
204:
205:
                                             j<sup>2</sup> = strlen(t<sup>1</sup>);
206:
207:
                                       if (is cmd1 & 1)
208:
                                            is cmd1^7 = 0;
                                      if (is_cmd2<sup>10</sup> & 1)
209:
                                            is cmd2^{10} = 0;
210:
```

```
Footnotes:
211:
                                                                                                                                                                                                                                1: convert.c:90
212:
                                  state<sup>1</sup> = FIRST<sup>2</sup>;
                                                                                                                                                                                                                                2: convert.c:89
213:
                                                                                                                                                                                                                                3: convert.c:91
214:
                                                                                                                                                                                                                                4: convert.c:91
215:
                                                                                                                                                                                                                                5: convert.c:91
                                                                                                                                                                                                                                6: convert.c:92
                     for (i^3 = k^4 = 0; i^3 < j^5; i^3++)
216:
                                                                                                                                                                                                                                7: convert.c:83
217:
                        --if (t<sup>6</sup>[i<sup>3</sup>] == 127) {
                                                                                                                                                                                                                                8: convert.c:35
                                 s^{7}[k^{4}] = ' \setminus 0';
218:
                                                                                                                                                                                                                                9: convert.c:40
                                 strcat(s<sup>7</sup>, "\\char127 ");
219:
                                                                                                                                                                                                                               10: convert.c:235
220:
                                 k<sup>4</sup> += strlen("\\char127 ");
                                                                                                                                                                                                                               11: convert.c:235
                                                                                                                                                                                                                               12: convert.c:227
221:
                          -} else
                                                                                                                                                                                                                               13: char.c:32
222:
                                 s^{7}[k^{4}++] = t^{6}[i^{3}];
                                                                                                                                                                                                                               14: char.h:48
223:
                     s^{7}[k^{4}] = 0;
                                                                                                                                                                                                                               15: convert.c:236
224:
                                                                                                                                                                                                                               16: char.h:40
225:
                                                                                                                                                                                                                               17: convert.c:235
                                                                                                                                                                                                                               18: convert.c:235
226:
                                                                                                                                                                                                                               19: char.h:38
227:
               void econvert(char *s)
                                                                                                                                                                                                                               20: convert.c:234
228:
                                                                                                                                                                                                                               21: convert.c:233
229:
                           This function converts a string in LR mode.
                                                                                                                                                                                                                               22: char.c:30
230:
231:
232:
                     -enum {
233:
                           FIRST, MIDDLE
234:
                     -} state;
235:
                     int i, j, k, l, tmp, spc;
236:
                     char t[MAX LEN<sup>8</sup>], last[3];
237:
238:
                     if (noconvert9)
239:
                           return;
240:
                     -for (i^{10} = j^{11} = 0; s^{12}[i^{10}];) {
                           if (!ch<sup>13</sup>[(unsigned char) s<sup>12</sup>[i<sup>10</sup>]].farsi<sup>14</sup>)
241:
                                 t^{15}[j^{11}++] = s^{12}[i^{10}++];
242:
243:
                          -else {
                                                                 // farsi
244:
245:
                                -if (s^{12}[i^{10}] == F_MOM^{16}) 
                                        t^{15}[j^{11}] = ' \setminus 0';
246:
                                        strcat(t<sup>15</sup>, "\\mom ");
247:
                                        j^{11} = strlen(t^{15});
248:
                                        i<sup>10</sup>++;
249:
250:
                                 -} else {
                                        for (k^{17} = i^{10}, spc^{18} = 1; s^{12}[k^{17}] && ch<sup>13</sup>[(unsigned char) s<sup>12</sup>[k<sup>17</sup>]], farsi<sup>14</sup>; spc<sup>18</sup> &= s<sup>12</sup>[k<sup>17</sup>++] == F SPC<sup>19</sup>);
251:
                                        -if (!spc<sup>18</sup>) {
252:
                                              t^{15}[j^{11}] = ' \setminus 0';
253:
254:
                                              strcat(t<sup>15</sup>, "\\InF{}");
                                              j^{11} = strlen(t^{15});
255:
256:
                                        for (state<sup>20</sup> = FIRST<sup>21</sup>; s<sup>12</sup>[i<sup>10</sup>] && ch<sup>13</sup>[(unsigned char) s<sup>12</sup>[i<sup>10</sup>]].farsi<sup>14</sup>; i<sup>10</sup>++) {
257:
258:
                                             -if (s^{12}[i^{10}] == F_MOM^{16}) 
                                                     t^{15}[\dot{7}^{11}] = \dot{0};
259:
                                                     strcat(t<sup>15</sup>, "\\mom ");
260:
261:
                                                     j^{11} = strlen(t^{15});
                                                     i<sup>10</sup>++;
262:
263:
                                               \{e\} else if \{\operatorname{strchr}(e^{22}, s^{12}[i^{10}])\} // e'raab
```

```
Footnotes:
264:
                                                                                                                                                                                                                      1: convert.c:235
265:
                                                   for (1^1 = i^2; s^3[1^1] \&\& strchr(e^4, s^3[1^1]); 1^1++);
                                                                                                                                                                                                                      2: convert.c:235
266:
                                                   t^{5}[j^{6}] = ' \setminus 0';
                                                                                                                                                                                                                      3: convert.c:227
                                                   strcpy(last^{7}, t^{5} + (j^{6} -= (t^{5}[j^{6} - 1] == STRETCH^{8}) + 1));
267:
                                                                                                                                                                                                                      4: char.c:30
                                                   for (tmp^9 = --1^1, t^5[j^6] = '\setminus 0'; 1^1 >= i^2; 1^1--)
268:
                                                                                                                                                                                                                      5: convert.c:236
                                                                                                                                                                                                                      6: convert.c:235
                                                        strcat(t^{5}, e cmd^{10}[strchr(e^{4}, s^{3}[1^{1}]) - e^{4}]);
269:
                                                                                                                                                                                                                      7: convert.c:236
270:
                                                   strcat(t<sup>5</sup>, last<sup>7</sup>);
                                                                                                                                                                                                                      8: char.h:28
                                                  j<sup>6</sup> = strlen(t<sup>5</sup>);
271:
                                                                                                                                                                                                                      9: convert.c:235
272:
                                                   i^2 = tmp^9;
                                                                                                                                                                                                                      10: convert.c:42
273:
                                             else {
                                                                                                                                                                                                                     11: char.h:31
                                                                                                                                                                                                                     12: char.h:32
                                                  if (t^{5}[j^{6}-1] == ' \  \& s^{3}[i^{2}] == FB_OPEN^{11})
274:
                                                                                                                                                                                                                     13: convert.c:234
275:
                                                        s^3[i^2] = FB CLOSE^{12};
                                                                                                                                                                                                                     14: convert.c:233
                                                   if (t^{5}[j^{6} - 1] == ' \  \&  s^{3}[i^{2}] == FB\_CLOSE^{12})
276:
                                                                                                                                                                                                                     15: char.c:32
                                                        s^3[i^2] = FB OPEN^{11};
277:
                                                                                                                                                                                                                     16: char.h:45
                                                   t^{5}[j^{6}++] = (state^{13} == FIRST^{14})? ch^{15}[(unsigned\ char)\ s^{3}[i^{2}]].first<sup>16</sup>: ch^{15}[(unsigned\ char)\ s^{3}[i^{2}]].middle<sup>17</sup>;
278:
                                                                                                                                                                                                                     17: char.h:46
                                                   state<sup>13</sup> = ch<sup>15</sup>[(unsigned char) s<sup>3</sup>[i<sup>2</sup>]].last<sup>18</sup> ? FIRST<sup>14</sup> : MIDDLE<sup>19</sup>;
                                                                                                                                                                                                                     18: char.h:47
279:
                                                                                                                                                                                                                     19: convert.c:233
280:
                                                   if (stretch<sup>20</sup> && !ch<sup>15</sup>[(unsigned char) s<sup>3</sup>[i<sup>2</sup>]].last<sup>18</sup>)
                                                                                                                                                                                                                     20: convert.c:39
281:
                                                        t^{5}[i^{6}++] = STRETCH^{8};
                                                                                                                                                                                                                     21: convert.c:235
282:
                                                                                                                                                                                                                     22: convert.c:235
                                                                                                                                                                                                                     23: char.h:40
283:
                                                                                                                                                                                                                     24: convert.c:313
                                      -if (!spc<sup>21</sup>) {
284:
                                                                                                                                                                                                                      25: convert.c:307
285:
                                            t^{5}[j^{6}] = ' \setminus 0';
                                            strcat(t<sup>5</sup>, "\\EnF{}");
286:
287:
                                            j<sup>6</sup> = strlen(t<sup>5</sup>);
288:
289:
290:
291:
                    for (i^2 = k^{22} = 0; i^2 < j^6; i^2++)
292:
293:
                         -if (t<sup>5</sup>[i<sup>2</sup>] == 127) {
294:
                                s^{3}[k^{22}] = ' \setminus 0';
                                strcat(s³, "\\char127 ");
295:
                                k<sup>22</sup> += strlen("\\char127 ");
296:
297:
                          -} else if (t^{5}[i^{2}] == F_{MOM}^{23}) {
                                s^{3}[k^{22}] = ' \setminus 0';
298:
                                strcat(s<sup>3</sup>, "\\mom ");
299:
300:
                                k<sup>22</sup> += strlen("\\mom ");
301:
                          } else
                                s^{3}[k^{22}++] = t^{5}[i^{2}];
302:
303:
                    s^{3}[k^{22}] = 0;
304:
305:
306:
307:
              void detab(char *s)
308:
309:
                          This function changes all of the tab characters
310:
                          in the string s to spaces.
311:
312:
313:
                    int i, j;
314:
                    char temp[500];
                   -for (i^{24} = 0; i^{24} < strlen(s^{25});) {
315:
                         -if (s^{25}[i^{24}] == '\t') {
316:
```

```
Footnotes:
317:
                           strcpy(temp^1, s^2 + i^3 + 1); // remainder of the string
                                                                                                                                                                                        1: convert.c:314
318:
                                                                                                                                                                                        2: convert.c:307
319:
                           for (j^4 = i^3; i^3 < 8 * (1 + j^4 / 8); s^2[i^3++] = ' ');
                                                                                                                                                                                        3: convert.c:313
320:
                           s^2[i^3] = \langle 0 \rangle ;
                                                                                                                                                                                        4: convert.c:313
                           strcat(s2, temp1);
321:
                                                                                                                                                                                        5: convert.c:35
                                                                                                                                                                                        6: convert.c:329
322:
                      -} else
                                                                                                                                                                                        7: convert.c:327
323:
                           i<sup>3</sup>++;
                                                                                                                                                                                        8: convert.c:329
324:
                                                                                                                                                                                        9: convert.c:327
325:
                                                                                                                                                                                        10: convert.c:41
326:
                                                                                                                                                                                       11: convert.c:330
                                                                                                                                                                                       12: convert.c:307
327:
            int convert(char *infile_name, char *outfile_name)
                                                                                                                                                                                       13: convert.c:227
328:
                                                                                                                                                                                        14: convert.c:83
329:
                 FILE *infile, *outfile;
                                                                                                                                                                                       15: convert.c:39
330:
                 char s[MAX LEN<sup>5</sup>];
                                                                                                                                                                                       16: convert.c:40
331:
                 int i;
332:
                 char *p, *q, *r;
333:
                 char temp_name[MAX_LEN<sup>5</sup>];
334:
335:
                -if (!(infile = fopen(infile name, "r"))) {
336:
                      printf("ftx2tex: cannot open input file \"%s\"\n", infile_name<sup>7</sup>);
337:
                      return -1;
338:
339:
                 -if (!(outfile<sup>8</sup> = fopen(outfile_name<sup>9</sup>, "w"))) {
340:
                      printf("ftx2tex: cannot open output file \"%s\"\n", outfile_name*);
341:
                      return -1;
342:
343:
                 fprintf(stderr, "Converting \"%s\" to \"%s\"", infile_name, outfile_name);
344:
345:
                 line no^{10} = 0;
346:
                 -while (fgets(s<sup>11</sup>, MAX_LEN<sup>5</sup>, infile<sup>6</sup>)) {
347:
                      line_no<sup>10</sup>++;
348:
                      detab<sup>12</sup>(s<sup>11</sup>);
349:
                      if (!(line_no<sup>10</sup> % 50))
350:
                           fprintf(stderr, " [%ld]", line_no<sup>10</sup>);
351:
                      if (s^{11}[strlen(s^{11}) - 1] == '\n')
                           s^{11}[strlen(s^{11}) - 1] = ' \setminus 0'; // delete \setminus n from end of line
352:
353:
354:
                      if (s^{11}[0] == '>')
                           econvert^{13}(s^{11}+1);
355:
                      else if (s^{11}[0] == '<')
356:
                           fconvert^{14}(s^{11} + 1);
357:
358:
359:
                           fprintf(stderr, "\nftx2tex: file \"%s\" line %ld: every line must begin with < or >.\n", infile_name<sup>7</sup>, line_no<sup>10</sup>);
360:
                           return -1;
361:
                      if (!strcmp(s<sup>11</sup> + 1, "\\nostretch"))
362:
363:
                           stretch^{15} = 0;
364:
                      else if (!strcmp(s<sup>11</sup> + 1, "\\stretch"))
                           stretch^{15} = 1;
365:
                      else if (!strcmp(s<sup>11</sup> + 1, "\\convert"))
366:
367:
                           noconvert^{16} = 0;
                      else if (!strcmp(s<sup>11</sup> + 1, "\\noconvert"))
368:
                           noconvert^{16} = 1;
369:
```

```
370:
                       —else {
371:
                             -for (i<sup>1</sup> = 0; i<sup>1</sup> < NUM_INPUT_CMD<sup>2</sup>; ++i<sup>1</sup>) {
372:
                                    p^3 = s^4 + 1;
373:
                                    -while ((p³ = strstr(p³, input_command⁵[i¹])) != NULL) {
374:
                                          r^6 = q^7 = p^3 + strlen(input_command^5[i^1]);
375:
                                          while (*r^6 == ' ')
                                                ++r<sup>6</sup>;
376:
                                          if (*r<sup>6</sup> == '{')
377:
378:
                                                ++r<sup>6</sup>;
379:
                                          while (*r<sup>6</sup> == ' ')
                                                ++r<sup>6</sup>;
380:
381:
                                          -if (r^6 != q^7) {
                                                q^7 = r^6;
382:
                                                while (*r^6 != ' ' \&\& *r^6 != ' )')
383:
384:
                                                strncpy(temp_name^8, q^7, r^6 - q^7);
385:
                                                temp_name{}^{8}[r^{6} - q^{7}] = ' \setminus 0';
386:
                                                add_file<sup>9</sup>(temp_name<sup>8</sup>, infile_name<sup>10</sup>);
387:
388:
389:
                                          p^3 = r^6;
390:
391:
392:
                              fprintf(outfile<sup>11</sup>, "%s\n", s<sup>4</sup> + 1);
393:
394:
395:
                   fprintf(stderr, " [%ld]\n", line_no<sup>12</sup>);
396:
                   return 0;
397:
```

Footnotes:
1: convert.c:331
2: convert.c:332
4: convert.c:332
4: convert.c:75
6: convert.c:332
7: convert.c:333
9: ftx2tex.c:70
10: convert.c:329
11: convert.c:329

1: int convert¹(char *, char *);
Footnotes:
1: convert.c:327

```
1:
2:
             This file is a part of Ftx2TeX, a convertor from "ftx" source files
3:
             to TeX-readable files. Ftx2TeX is a part of FarsiTeX, a Persian/English
 4:
             typesetting system.
 5:
6:
             Copyright (C) 2001 Roozbeh Pournader <roozbeh@sharif.edu>
7:
8:
          * FarsiTeX is free software; you can redistribute it and/or modify
9:
          * it under the terms of the GNU General Public License as published by
10:
          * the Free Software Foundation; either version 2 of the License, or
11:
          * (at your option) any later version.
12:
13:
          * FarsiTeX is distributed in the hope that it will be useful,
14:
             but WITHOUT ANY WARRANTY; without even the implied warranty of
15:
          * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16:
          * GNU General Public License for more details.
17:
18:
          * You should have received a copy of the GNU General Public License
19:
             along with FarsiTeX; if not, write to the Free Software
20:
          * Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307,
21:
          * USA.
22:
23:
          * Any licensing or usage questions should be directed to Roozbeh
24:
          * Pournader <roozbeh@sharif.edu>.
25:
26:
27:
         #include "filefunc.h"
28:
29:
         #include <stdio.h>
30:
         #include <sys/stat.h>
31:
32:
         #ifdef __BORLANDC__
33:
         #include <io.h>
34:
         #else
35:
         #include <unistd.h>
36:
         #endif
37:
38:
         int file_exists(char *filename)
39:
40:
             return (access(filename<sup>1</sup>, 0) == 0);
41:
42:
43:
         int file_is_newer(char *first, char *second)
44:
45:
             struct stat first_stat, second_stat;
46:
47:
             stat(first<sup>2</sup>, &first_stat<sup>3</sup>);
48:
             stat(second<sup>4</sup>, &second_stat<sup>5</sup>);
49:
             return (first stat<sup>3</sup>.st mtime > second stat<sup>5</sup>.st mtime);
50:
```

Footnotes:

1: *filefunc.c:38*

2: filefunc.c:43

3: *filefunc.c:45*

4: *filefunc.c:43*

5: *filefunc.c:45*

ftx2tex.src\filefunc.c

```
Footnotes:
1:
          #ifdef __MSDOS__
                                                                                                                                                                              1: filefunc.h:4
2: filefunc.c:38
3: filefunc.c:43
          #define PATH_SEP1 '\\'
2:
3:
          #else
4:
          #define PATH_SEP '/'
5:
          #endif
6:
7:
          #define FILENAME_LEN 1024
8:
          extern int file_exists<sup>2</sup>(char *);
9:
10:
          extern int file_is_newer³(char *, char *);
```

ftx2tex.src\filefunc.h

```
Footnotes:
1:
                                                                                                                                                       1: filefunc.h:4
2:
          * This file is a part of Ftx2TeX, a convertor from "ftx" source files
                                                                                                                                                       2: ftx2tex.c:41
 3:
          * to TeX-readable files. Ftx2TeX is a part of FarsiTeX, a Persian/English
 4:
          * typesetting system.
 5:
 6:
          * Copyright (C) 1996 Mohammad Mahdian <mahdian@mit.edu>
7:
             Copyright (C) 1998-2001 Roozbeh Pournader <roozbeh@sharif.edu>
8:
9:
          * FarsiTeX is free software; you can redistribute it and/or modify
10:
          * it under the terms of the GNU General Public License as published by
11:
          * the Free Software Foundation; either version 2 of the License, or
12:
          * (at your option) any later version.
13:
14:
          * FarsiTeX is distributed in the hope that it will be useful,
15:
          * but WITHOUT ANY WARRANTY; without even the implied warranty of
16:
          * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17:
             GNU General Public License for more details.
18:
19:
          * You should have received a copy of the GNU General Public License
20:
          * along with FarsiTeX; if not, write to the Free Software
21:
          * Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307,
22:
             USA.
23:
24:
          * Any licensing or usage questions should be directed to Roozbeh
25:
          * Pournader <roozbeh@sharif.edu>.
26:
          * /
27:
28:
         #include <stdio.h>
29:
         #include <stdlib.h>
30:
         #include <string.h>
31:
         #include "convert.h"
         #include "filefunc.h"
32:
33:
34:
         #ifdef __BORLANDC__
35:
         #define strcasecmp stricmp
36:
         #endif
37:
38:
         #define MAX_FILES
                                   50
39:
40:
         /* changes all slash characters to PATH_SEP */
41:
         void change slashes(char *filename)
42:
43:
         #if (PATH_SEP<sup>1</sup> != '/')
44:
             int i;
45:
46:
             for (i = 0; filename<sup>2</sup>[i] != ' \setminus 0'; ++i)
47:
                 if (filename<sup>2</sup>[i] == '/')
48:
                     filename<sup>2</sup>[i] = PATH_SEP<sup>1</sup>;
49:
         #endif
50:
51:
52:
         /* add an extension to the file if it didn't contain one */
53:
         void add_extension(char *filename, char *ext)
```

ftx2tex.src\ftx2tex.c

```
Footnotes:
 54:
                                                                                                                                                                                                             1: ftx2tex.c:55
 55:
                   int i;
                                                                                                                                                                                                             2: ftx2tex.c:53
 56:
                                                                                                                                                                                                             3: filefunc.h:4
 57:
                   i<sup>1</sup> = strlen(filename<sup>2</sup>);
                                                                                                                                                                                                             4: ftx2tex.c:53
 58:
                   while (i^1 >= 0 \&\& filename^2[i^1] != '.' \&\& filename^2[i^1] != PATH_SEP^3)
                                                                                                                                                                                                             5: ftx2tex.c:38
                                                                                                                                                                                                             6: filefunc.h:7
 59:
                                                                                                                                                                                                             7: ftx2tex.c:74
                   -if (i^1 < 0 || filename^2[i^1] != '.') {
 60:
                                                                                                                                                                                                             8: ftx2tex.c:70
 61:
                         strcat(filename<sup>2</sup>, ".");
                                                                                                                                                                                                             9: ftx2tex.c:73
 62:
                         strcat(filename<sup>2</sup>, ext<sup>4</sup>);
                                                                                                                                                                                                            10: ftx2tex.c:70
 63:
                                                                                                                                                                                                            11: ftx2tex.c:72
                                                                                                                                                                                                            12: ftx2tex.c:41
 64:
                                                                                                                                                                                                            13: ftx2tex.c:53
 65:
                                                                                                                                                                                                            14: ftx2tex.c:67
 66:
             char *infile_name[MAX_FILES<sup>5</sup>]; *outfile_name[MAX_FILES<sup>5</sup>];
                                                                                                                                                                                                            15: ftx2tex.c:66
 67:
             int file num;
                                                                                                                                                                                                            16: ftx2tex.c:72
                                                                                                                                                                                                            17: filefunc.c:38
 68:
                                                                                                                                                                                                            18: filefunc.c:43
 69:
             /* add a file to the list of the files to be converted afterwards */
                                                                                                                                                                                                            19: ftx2tex.c:75
 70:
             void add_file(char *filename, char *basefilename)
                                                                                                                                                                                                            20: ftx2tex.c:66
 71:
 72:
                   char sourcename[FILENAME LEN<sup>6</sup>], realname[FILENAME LEN<sup>6</sup>];
 73:
                   char *p;
 74:
                   int i;
 75:
                   static int flag = 0;
 76:
 77:
                   i<sup>7</sup> = strlen(filename<sup>8</sup>);
                   while (i^7 >= 0 \&\& filename^8[i^7] != '.'
 78:
 79:
                             && filename<sup>8</sup>[i^7] != '/')
 80:
 81:
 82:
                   /* if extension-less or extension == '.tex' */
 83:
                   -if (i^7 < 0 | filename<sup>8</sup>[i^7] != '.' | strcmp(filename<sup>8</sup> + i^7 + 1, "tex") == 0) {
 84:
                        -if ((p<sup>9</sup> = strrchr(basefilename<sup>10</sup>, PATH_SEP<sup>3</sup>)) != NULL) {
                              strncpy(realname<sup>11</sup>, basefilename<sup>10</sup>, p<sup>9</sup> - basefilename<sup>10</sup> + 1);
 85:
                               strcpy(realname<sup>11</sup> + (p<sup>9</sup> - basefilename<sup>10</sup> + 1), filename<sup>8</sup>);
 86:
 87:
 88:
                               strcpy(realname<sup>11</sup>, filename<sup>8</sup>);
 89:
 90:
                         change_slashes<sup>12</sup>(realname<sup>11</sup>);
                         add_extension<sup>13</sup>(realname<sup>11</sup>, "tex");
 91:
 92:
                         for (i^7 = 0; i^7 < file num^{14}; ++i^7)
 93:
                               if (strcasecmp(realname<sup>11</sup>, outfile_name<sup>15</sup>[i<sup>7</sup>]) == 0)
 94:
                                    return;
                         strcpy(sourcename<sup>16</sup>, realname<sup>11</sup>);
 95:
                         strcpy(strrchr(sourcename<sup>16</sup>, '.') + 1, "ftx");
 96:
                         if (file_exists 17 (sourcename 16) &&
 97:
                         -(!file exists<sup>17</sup>(realname<sup>11</sup>) || file is newer<sup>18</sup>(sourcename<sup>16</sup>, realname<sup>11</sup>))) {
98:
                             -if (file num<sup>14</sup> >= MAX FILES<sup>5</sup>) {
 99:
                                    -if (!flag<sup>19</sup>) {
100:
                                          fprintf(stderr, "ftx2tex: maximum number of files exceeded, won't convert any more");
101:
102:
                                          flaq^{19} = 1;
103:
104:
                              -} else {
                                    infile_name<sup>20</sup>[file_num<sup>14</sup>] = strdup(sourcename<sup>16</sup>);
105:
                                    outfile name<sup>15</sup>[file num<sup>14</sup>] = strdup(realname<sup>11</sup>);
106:
```

ftx2tex.src\ftx2tex.c

```
107:
                                   ++file num<sup>1</sup>;
108:
109:
110:
111:
112:
113:
             int main(int argc, char *argv[])
114:
115:
                  int i;
116:
117:
                  -if (argc<sup>2</sup> <= 1) {
118:
                        printf("\nftx2tex usage: ftx2tex input_file [output_file]\n");
119:
                        return 1;
120:
121:
                  -if (!(infile_name<sup>3</sup>[0] = (char *) malloc(strlen(argv<sup>4</sup>[1]) + 5))) {
122:
                        printf("ftx2tex: cannot allocate memory\n");
123:
                        return 1;
124:
125:
                  strcpy(infile name<sup>3</sup>[0], argy<sup>4</sup>[1]);
126:
                  change_slashes<sup>5</sup>(infile_name<sup>3</sup>[0]);
127:
                  add_extension<sup>6</sup>(infile_name<sup>3</sup>[0], "ftx");
128:
129:
                  -if (argc<sup>2</sup> > 2) {
130:
                       -if (!(outfile_name<sup>7</sup>[0] = (char *) malloc(strlen(argv<sup>4</sup>[2]) + 5))) {
131:
                             printf("ftx2tex: cannot allocate memory\n");
132:
                             return 1;
133:
134:
                        strcpy(outfile_name<sup>7</sup>[0], argv<sup>4</sup>[2]);
135:
                        change slashes<sup>5</sup>(outfile name<sup>7</sup>[0]);
136:
                        add_extension<sup>6</sup>(outfile_name<sup>7</sup>[0], "tex");
137:
                  -} else {
                       -if (!(outfile_name<sup>7</sup>[0] = (char *) malloc(strlen(infile_name<sup>3</sup>[0]) + 4))) {
138:
139:
                             printf("ftx2tex: cannot allocate memory\n");
                             return 1;
140:
141:
                        strcpy(outfile_name<sup>7</sup>[0], infile_name<sup>3</sup>[0]);
142:
143:
                        i<sup>8</sup> = strrchr(outfile_name<sup>7</sup>[0], '.') - outfile_name<sup>7</sup>[0];
                        outfile_name^{7}[0][i^{8} + 4] = 0;
144:
                       -if (strcasecmp(outfile_name<sup>7</sup>[0] + i<sup>8</sup> + 1, "tex")) {
145:
                             outfile_name^{7}[0][i^{8}] = 0;
146:
147:
                             strcat(outfile name<sup>7</sup>[0], ".tex");
148:
149:
                             outfile_name^{7}[0][i^{8}] = 0;
150:
151:
152:
                  file num^1 = 1;
                  for (i^8 = 0; i^8 < file_num^1; ++i^8)
153:
154:
                        convert<sup>9</sup>(infile_name<sup>3</sup>[i<sup>8</sup>], outfile_name<sup>7</sup>[i<sup>8</sup>]);
155:
156:
                  return 0;
157:
```

Footnotes:

1: ftx2tex.c:67

2: ftx2tex.c:113

3: ftx2tex.c:66

4: ftx2tex.c:113

5: ftx2tex.c:41 **6**: ftx2tex.c:53

7: ftx2tex.c:66

8: ftx2tex.c:115

9: convert.c:327

ftx2tex.src\ftx2tex.c

1: int add_file¹(char *, char *);

Footnotes:
1: ftx2tex.c:70

ftx2tex.src\ftx2tex.h

Symbols "char.h"; 1,9

"char.h"; 1,9
"convert.h"; 9,20
"filefunc.h"; 18,20
"ftx2tex.h"; 9
<io.h>; 18
<<stdio.h>; 9,18,20
<<stdlib.h>; 9,20
<<string.h>; 9,20
<<string.h>; 18
<unistd.h>; 18
_BORLANDC__; 18,20
_MSDOS__; 19

Α

access; 18 add_extension; 20,21,22 add_file; 16,21,23 argc; 22 argv; 22 AT_SIGN; 7,12

В

basefilename; 21

C

ch; 1,7,10,11,12,13,14 change_slashes; 20,21,22 chars; 1,7 command1; 9,12 command2; 10,12 convert; 15,17,22

D

detab; 14,15

Ε

e; 1,7,11,13,14 e_cmd; 9,11,14 econvert; 13,15 ext; 20,21

F

F_BSLASH; 7,11,12 F_MID; 7,11 F_MOM; 7,11,13,14 F SPC; 7,10,11,13

F STAR; 7,11,12 farsi; 7,10,11,12,13 FB_CLOSE; 7,11,14 FB OPEN; 7,10,11,14 fconvert; 10,15 fgets; 15 FILE; 15 file_exists; 18,19,21 file is newer; 18,19,21 file num; 21,22 filename; 18,20,21 FILENAME LEN; 19,21 first; 7,11,12,14,18 FIRST; 10,11,12,13,14 FK_CLOSE; 7,11 FK_OPEN; 7,10 fopen; 15 FP CLOSE; 7,11 FP OPEN; 7,11 fprintf; 15,16,21

ı

infile_name; 15,16,21,22 input_command; 10,16

L

last; 7,11,12,14 letter; 7,11 line_no; 9,15,16

M

main; 22 malloc; 22 MAX_FILES; 20,21 MAX_LEN; 9,10,13,15 MIDDLE; 10,11,12,13,14 middle; 7,11,12,14

N

noconvert; 9,10,13,15 NULL; 16,21 NUM_CMD1; 9,12 NUM_CMD2; 9,10,12 NUM_ERAB; 1,7,9 NUM_INPUT_CMD; 10,16

0

outfile_name; 15,21,22

Ρ

PATH_SEP; 19,20,21 printf; 15,22

S

s; 10,11,12,13,14,15 second; 18 st mtime; 18 stat; 18 stderr; 15,16,21 strcasecmp; 20,21,22 streat; 11,12,13,14,15,21,22 strchr; 11,13,14 strcmp; 12,15,21 strcpy; 11,14,15,21,22 strdup; 21 stretch; 9,12,14,15 STRETCH; 7,11,12,14 strlen; 11,12,13,14,15,16,21,22 strncpy; 16,21 strrchr; 21,22 strstr; 16

Index