

論文

点字・墨字間の変換プログラムについて

早坂匡*・塚越雅斗*

On the Code Switching Computer Program
between Braille and Ink-print

Tadashi Hayasaka* and Masato Tsukagoshi*

Abstract

We have developed a computer program that switches between Braille and ink-print codes on Linux. The program is consist of three components: (1) the conversion from ink-print to Braille, (2) the conversion from Braille to nonary number (3) the conversion from nonary number to ink-print.

The basic performance of the program is confirmed. We mainly report on it's data structure and usage in this paper.

1. はじめに

点字と墨字（一般に用いられている普通の文字）間の変換用ソフトウェアは各種存在するが、われわれが調べた限りでは、それらは全て Microsoft Windows 上で動作するものである。良く知られている例としては、フリーウェアとして IBM が公開している Win-BES99 がある (<http://www-6.ibm.com/jp/accessibility/soft/winbes99.html> 参照)。

しかし、視覚障害者やボランティアの人たちの利便性を考えると、UNIX 系 OS 上で動作するプログラムを開発することにも価値があると思われる。

このような観点から、われわれは、UNIX 系 OS で、かつ比較的広く使用されている Linux 上において開発をおこない、基本的な動作を確認した。ここでは、そのデータ構造やプログラムの構成および

使用方法を中心に報告する。

ただし、今回使用した開発環境とそのバージョンは、Table 1 のようなものである。

2. 点字について

2.1 点字の表記と構成

6 点点字は、1825 年、ルイ・ブライユ⁽¹⁾により考案された。日本の点字は、1890 年、いくつかの案の中から選定され、石川倉次の案が認められたものである。

日本の点字は、現在、全国的に統一されてきているが、点字記号については問題点も残されている。本稿の表記は、1990 年版「日本点字表記法⁽²⁾」に従う。

点字は、Fig. 1 のように、縦 3 点が 2 列に並んだ 6 点を 1 つの単位（一マスと呼ぶ）として構成され、その各点を凸面にするることにより、左から右へ、指先で読むことができる。

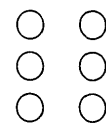


Fig. 1 点字の構成（一マス）

なお、本稿では、上記の「日本点字表記法」にならない、凸面の点を「●」、平らな点を「○」で表す。

Table 1 開発環境とバージョン

Linux kernel	2.4.17
X (XFree86)	4.1.0
gcc	2.95.2
Tcl/Tk	8.2

2002年7月22日受理

* 秋田大学工学資源学部電気電子工学科. Department of Electrical and Electronic Engineering, Faculty of Engineering and Resource Science, Akita University.

Table 2 清音・撥音・促音・長音

あ	い	う	え	お
か	き	く	け	こ
さ	し	す	せ	そ
た	ち	つ	て	と
な	に	ぬ	ね	の
は	ひ	ふ	へ	ほ
ま	み	む	め	も
や		ゆ		よ
ら	り	る	れ	ろ
わ	ゐ		ゑ	を
ん	っ	ー		

Table 3 濁音・半濁音・拗音・拗濁音
拗半濁音の一部

が	ぎ	ぐ	げ	ご
ぱ	ぴ	ぷ	ぺ	ぽ
きゃ	きゅ	きょ		
ぎゃ	ぎゅ	ぎょ		
ぴゃ	ぴゅ	ぴょ		

Table 4 数字・アルファベット

数符				
1	2	3	4	5
6	7	8	9	0
外字符				
a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y
z				

2.2 点字仮名の構成

Table 2のように、日本の点字の清音・撥音・促音・長音は一マスで表される。これに対して、濁音・半濁音・拗音・拗濁音・拗半濁音は一マス前置して、二マスで表される。Table 3には、繁雑さを避けるためその一部を示してある。また、Table 4からわかるように、数字は数符、アルファベットは外字符を前置して二マスとなる。

これら以外に、外来音などで用いられる特殊音として、開拗音（イェ等）や合拗音（ウイ等）があるが、本稿では、未対応部分である。

Table 5 句読符

句点 (.)	読点 (,)
疑問符 (?)	中点 (⋯)
感嘆符 (!)	

2.3 句読符

Table 5 に示された句読符は、それ自身は一マスで表されるが、その後続く空白部が異なっている。すなわち、句点・疑問符・感嘆符は二マスあけ、読点・中点は一マスあける。日本の点字の符合に関しては、上に挙げたもの以外に、外国語引用符・カッコ・波線・矢印等多数の符合があるが、本稿では未対応部分である。

2.4 点字による仮名遣い

点字による仮名遣いは、現代仮名遣いに対応しているが、以下の2点で異なっている。その第1点は、助詞の“は”と“へ”を、発音するとおり“わ”と“え”と書くことである。そして、その第2点は、長音を表す“う”を“長音符(ー)”を用いて書くことである。なお、“わ行”の“ゐ”と“ゑ”は、原則として用いない。

2.5 点字の分かち書き

点字と墨字間の変換プログラムを作成する場合に大きな問題となるのが、本節の「分かち書き」である。本節では、その分かち書きの問題点について説明する。

分かち書きは、本来、点字使用者にとって、意味がわかりやすく、早く読むことができるために必要なものである。したがって、そのルールのかなりの部分は、個人的、または地域的なものであった。

前述の「日本点字記法」では、ルールを統一するために、分かち書きの詳細な規則を定めている。その原則は、「自立語」は文の単位となるから、その前で区切り、「助詞」や「助動詞」は自立語に続ける、である。Table 6 に、その例を示した。ただし、Table 中の「□」は、区切り目を表す (Table 7 も同様)。

分かち書きがこの原則だけですむのであれば、辞書を作ることで(それだけでも、かなり大きな辞書になるだろうが)対処可能である。しかしながら、Table 7 のように、かなり微妙な表現もある(ここに挙げたものは、その一部にすぎない)。現時点では、この対処方法の方針ができておらず、したがって、本稿では、分かち書きの処理に関して基本的に未対応である。

しかし、分かち書きに全く対応できないというのでは、試験的にさえ使用不能であるので、現実的な対応策として、墨字を入力する時点で、分かち書きしておくという方法をとることとした。その使用例は、4章に示す。

Table 6 分かち書きの例 1

- 涼しい風が木の葉を揺らせた
→ スズシイ□カゼガ□コノハヲ□ユラセタ
- 我輩わ猫である
→ ワガハイワ□ネコデ□アル

Table 7 分かち書きの例 2

- 彼わ一人寂しく亡くなった
→ カレワ□ヒトリ□サビシク□ナクナッタ
- 君のせいで寂しくなくなった
→ キミノ□セイデ□サビシク□ナク□ナッタ
- 全力投球する
→ ゼンリョク□トーキュー□スル
- 断固反対する
→ ダンコ□ハンタイスル

3. データ構造

3.1 点字の8進数表示

前章でみたように、点字は縦3点が横に並んだ形になっており、その各点が凸面(●)または平らな面(-)で表されている。そこで、‘●’をON, ‘-’をOFFとみて、点字の縦3点を8進数として表示することは自然なことと思われる。

この方式では、たとえば、一マスの“あ”は‘10’, “か”は‘14’と表される。同様に、二マスの“か”は‘0214’, “ば”は‘0454’である。

3.2 構造体

プログラミング言語C⁽³⁾には、異なった型をもついくつかのデータを、一つの単位として扱うことができる構造体(struct)がある。

本稿では、この構造体を用いて“墨字”と“8進数表示の点字”を対応させた辞書を作成した。ただし、構造体は、今後の拡張が容易なように、マスの数毎に構造体タグを付けて別々に作成した。これは、辞書の検索に時間はかかるマイナス面はあるものの、現在未対応の符合(たとえば、“矢印”には六マスのものがある)を取り入れることが簡単になると判断したためである。

4. プログラムの構成と使用例

“墨字”と“点字”の間の変換をおこなうため、本稿では、機能別に (1) 墨字を点字に変換 - 点訳 - する場合は、直接、変換する、(2) 点字を墨字に変換 - 墨訳 - する場合は、(2-1) 点字を8進数に変換する、(2-2) 8進数を墨字に変換する、のような3構成でプログラムを作成した。

以下では、構成別に、各々の概略と使用法について説明する。ただし、これらのデータ入力部は全て、実行の際、引き数のない場合は“標準入力”より入力を受け付け、“ファイル”より入力する場合は、ファイル名を引き数とすることとした。また、出力部は“標準出力”への出力とした。

なお、プログラム中の表示では、点字の凸面は‘o’で、平らな面は‘-’で表している。

4.1 墨字を点字に変換 (実行名は s2t)

本プログラムは、最初に、入力された文字列を、英数字・清音・拗音等毎に分類する。次に、各文字について、3章で述べた辞書を検索し、対応する8進数に置き換えた後、‘o’と‘-’のパターンとして出力する。Fig. 2は、標準入力より入力した例である (図は画面のハードコピーである。以下の図も同様)。

漢字交じりの文の場合、漢字をひら仮名に直す必要がある。Linuxには“kakasi”というツールがあり、これを使用することにより、漢字交じりの文の点訳が可能である。その例をFig. 3に示す。

2.5節で述べた分かち書きの対策は、Fig. 4のように、入力時点で分かち書きしておくことである。

4.2 点字を8進数に変換 (実行名は t2e)

本プログラムは、3行を一度に読み込み、3.1節で述べたように、‘o’と‘-’のパターンを8進数に変換するものである。Fig. 5は、前述のFig. 2に示したs2tの出力を、t2eの入力として与えたものである。

4.3 8進数を墨字に変換 (実行名は e2s)

本プログラムは、原則的には、前節のt2eの出力である8進数を受けとり、辞書を検索することにより、墨字に変換する。Fig. 6は、Fig. 5の出力を、入力とした例である。

5. GUIを用いた使用例

前章の使用例は、実行する際、全てコマンドラインから入力したものであるが、これではコマンドを覚える必要があり、Linuxに慣れてないと使いにくい面がある。そこで、本章では、コマンドを知らなくても使うことができるようなTcl/Tk⁽⁴⁾による

```
(algol)% s2t
きゃ。b9
-00-----0--0-0
----00-----00--00-
---0-0-----0--00--
```

Fig. 2 点訳の出力例1

```
(algol)% kakasi -JH | s2t
我輩わ猫である
----0-0-0-0--00-0--000-00
---0-----0---0-0--000---0
0-----000--0-0--0--0-----
```

Fig. 3 点訳の出力例2 (漢字交じり文)

```
(algol)% kakasi -JH | s2t
我輩わ 猫で ある
----0-0-0-0--00-0--00--0-00
---0-----0---0-0--000---0
0-----000--0-0--0--0-----
```

Fig. 4 点訳の出力例3 (分かち書き)

```
(algol)% s2t | t2e
きゃ。b9
011426000006304721
```

Fig. 5 8進数変換の出力例

```
(algol)% s2t | t2e | e2s
きゃ。b9
きゃ。b9
```

Fig. 6 墨字に変換した出力例

GUI (Graphical User Interface) を用いた例を紹介する。ただし、現段階では実用に耐えるようなものにはなっておらず、練習用程度にしか使えないものであることを、はじめにお断りしておく。

なお、ここで使用したXのウィンドウ・マネージャーはtwm, ターミナル・タイプはktermである。

5.1 墨訳の例

Fig. 7に、変換前(a)と変換後(b)の図を示した。(a)図のように、中央下部にあるのが点字の入力部

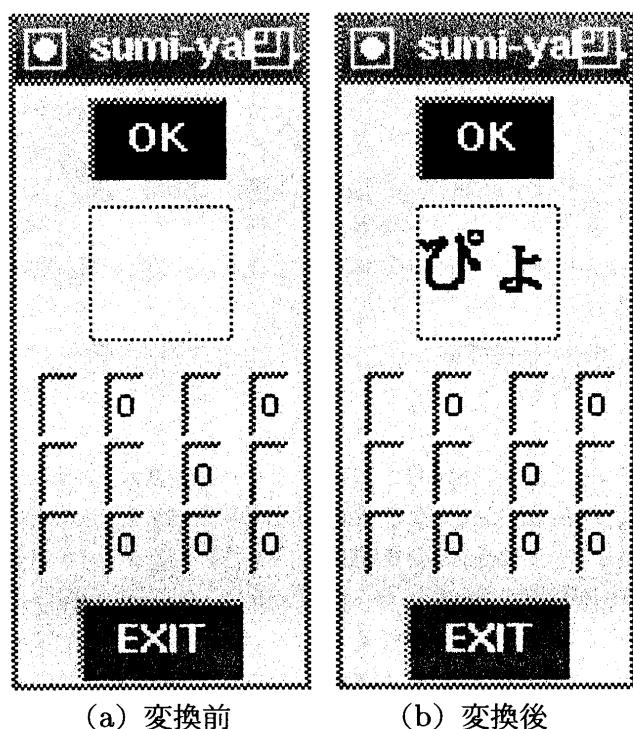


Fig. 7 GUIを用いた墨訳の例

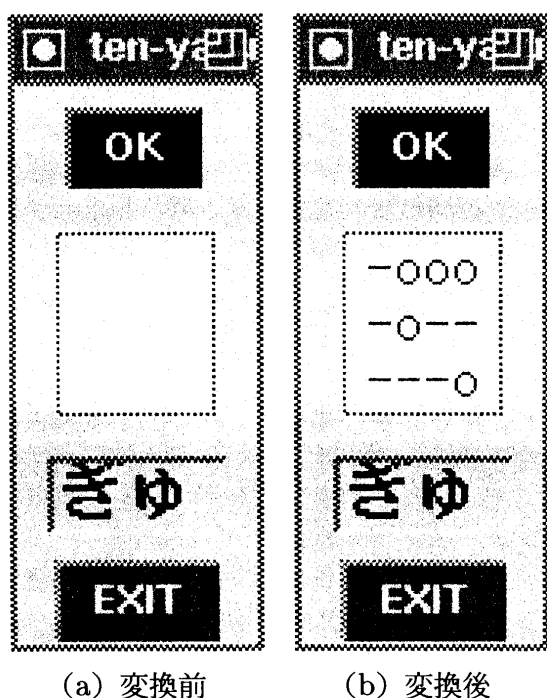


Fig. 8 GUIを用いた点訳の例

Fig. 8の(a)図の下部の欄に墨字を入力する。その後、“OK”ボタンを押すと、(b)図のように、点字が表示される。

6. おわりに

本稿では、点字・墨字間の変換に関して、未対応の部分がかかなり残っている。ここでは、それらの問題について整理しておく。

6.1 未対応の符合

2.2節や2.3節であげた、開拗音、カッコ等の未対応については、数は多いものの、辞書に登録するだけで対応可能である。しかし、辞書そのものの作成方法には、検索の効率の面から考えると、再考の余地はある。

6.2 分かち書き

分かち書きについては、2.5節で述べたように、現時点では手を付けていない状態である。基本的には、自立語の辞書を作成することから始める必要があるが、完成までには時間がかかると思われる。

6.3 Tcl/TkによるGUI

5章の例は、墨訳・点訳どちらの場合も使いにくい。今後の方向としては、メニューの中のファイル名をクリックしただけで、点字のファイルであれば墨訳、墨字のファイルであれば点訳するようなものが考えられる。また、点字の入力を簡単にするためには、ある特定の6個のキーを6点点字の各ビットに割り当てて、キーボードから使うことができるようにすることも考えられる。

6.4 点字の音読化

これは、本稿では取り上げていないことであるが、今後の重点課題であるので、説明しておく。

墨字の入力時もそうではあるが、特に点字を入力したときは、入力ミスのチェックに気をを使うことになる。そこで、点字のファイルの音読化や、点字入力時の音読化は、入力ミス発見の有効な手段になると思われる。

謝 辞

本研究をすすめるにあたり、教育文化学部・障害児教育講座 大城英名助教授には、点字に関する資料を貸していただいた。また、原稿にも目を通していただき、有益なご指導・ご助言をいただいた。ここに記して、感謝の意を表する次第である。

参考文献

- (1) ピエール・アンリ (1984) : 点字発明者の生

である。ここに、点字の凸面を表す‘o’（小文字のオー）を入力する。その後、“OK”ボタンを押すと、(b)図のように墨字が表示される。

この例では、二マスの点字しか表示できない上に、‘o’を入力するのが煩わしいという問題がある。

5.2 点訳の例

涯（奥寺百合子訳），朝日新聞社.

(2) 日本点字委員会（1990）：日本点字表記法
1990年版，日本点字委員会.

(3) B.W. カーニハン，D.M. リッチー（2001）：
プログラミング言語 C 第2版（石田晴久訳），共
立出版株式会社.

(4) B.B. ウェルチ（1999）：Tcl/Tk 入門 第2
版（西中芳幸，石曾根信訳），株式会社ピアソン・
エデュケーション.