```
/*
  --- out2uni.c ---

out2uni: Change string in foo.out into Unicode.
          foo.out is a file made by hyperref.

Usage 1 : out2uni foo (Shift-JIS case)
Usage 2 : out2uni -e foo (EUC-JP case).

if compiled by -DEUC
Usage 1 : out2uni foo (EUC-JP case)
Usage 2 : out2uni -s foo (Shift-JIS case).


usage of hyperref:

\usepackage[dvipdfm]{color}
\usepackage[dvipdfm,bookmarks=true,bookmarksnumbered=true,%
bookmarkstype=toc]{hyperref}

Example:

platex foo
platex foo
out2uni foo
platex foo
dvipdfm foo

2001 --ak
*/

/*
この C ソースは、c++2platex で処理して文書を作成できるように
書いてあります。

out2uni は、hyperref パッケージを使用して、dvipdfm によって
PDF を作成する場合、日本語しおり、日本語文書情報、日本語 text annotation
作成を可能にするツールです。


o 日本語しおり作成法

日本語しおりを作成するには、殆ど何も意識する必要はありません。
```

hyperref パッケージを

```
\usepackage[dvipdfm]{color}
\usepackage[dvipdfm,bookmarks=true,bookmarksnumbered=true,%
colorlinks=true,bookmarkstype=toc]{hyperref}
```

のような感じで *pLaTeX* ソースに記述しておいて下さい。
そうして

```
platex foo
platex foo
out2uni foo
platex foo
dvipdfm foo
```

とすると、出来上がった *foo.pdf* には日本語しおりが入っているでしょう。

o 日本語文書情報作成法*:*

Acrobat Reader において *Control D* で表示される
文書情報 *(pdftitle, pdfsubject, pdfauthor, pdfkeywords)*
に日本語を含めるには次のようにします。

[1]
hyperref パッケージは次のようにして読み込みます*:*

```
... ...
\input docinfo.out
\usepackage[dvipdfm,bookmarks=true,bookmarksnumbered=true,%
colorlinks=true,bookmarkstype=toc,%
pdftitle=\PDFTITLE,%
pdfsubject=\PDFSUBJECT,%
pdfauthor=\PDFAUTHOR,%
pdfkeywords=\PDFKEYWORDS]{hyperref}
... ...
```

[2]
別ファイル *docinfo.out* は、次のように作っておき、
あらかじめ一回だけ *out2uni docinfo* としておきます。
*(*これによって *Unicode* に変換する。*)*

ファイル名は *docinfo.out* である必要はありませんが、サフィックス
は必ず *.out* でなければいけません。 *out2uni* はサフィックスが *.out* である
ことを仮定しているからです。また一項目は一行で終わる必要があります。
ただし、行の最後のバックスラッシュ \ によって行を継続することができます。
プログラムはこの \ と、行末コードを取り除いて行を構成します。
\ の後にはすぐに *Enter* キーを打たないと、継続行とは解釈されないことに
注意して下さい。こうして作る一行の長さ制限は *16* キロバイトほど可能です
から、通常長さを気にする必要はありません。*docinfo.out* は *out2uni docinfo*
とすると書き換えられますから、内容を変更しながらくりかえす場合には、
編集するファイルを例えば *docinfo.txt* としておき、
copy docinfo.txt docinfo.out としてから *out2uni docinfo* とすれば良い
でしょう。*docinfo.out* に対しては *out2uni* を実行するのは一回だけである
ことに注意して下さい。訂正するには *docinfo.txt* を訂正してから
copy docinfo.txt docinfo.out とします。

```
%
% a sample source of docinfo.out
% this is overwritten by out2uni
%
\catcode`\@=0
@catcode`@\=12
@def@PDFTITLE{TeX による日本語組版技術}
@def@PDFSUBJECT{TeX と PDF の連携について}
@def@PDFAUTHOR{中村 一郎}
@def@PDFKEYWORDS{TeX、メタフォント、メタポスト、ポストスクリプト、PDF}
@catcode`@\=0
\catcode`\@=12
```

あとは通常どおり

```
platex foo
platex foo
out2uni foo
platex foo
dvipdfm foo
```

です。

o 日本語 *Text annotation* 作成法

Text annotation に日本語を含めるには、上で述べた *docinfo.out* のソース
ファイルに次のような行を付加しておきます*:*

```
@def@ANNOT{これは Text annotaion の例です。}
```

追加する場所は下記の完全な例を見て下さい*:*

```
%
% a sample source of docinfo.out
% this is overwritten by out2uni
%
\catcode`\@=0
@catcode`@\=12
@def@PDFTITLE{TeX による日本語組版技術}
@def@PDFSUBJECT{TeX と PDF の連携について}
@def@PDFAUTHOR{中村 一郎}
@def@PDFKEYWORDS{TeX、メタフォント、メタポスト、ポストスクリプト、PDF}
@def@ANNOT{これは Text annotaion の例です。}
@catcode`@\=0
\catcode`\@=12
```

そうして、*TeX* ソースに次のように *dvipdfm special* を書いておきます*:*

```
\special{pdf: ann width 7cm height 5cm
<< /Type /Annot /Subtype /Text /Open true
/Contents (\ANNOT) >>}
```

この *special* を覚えるのは大変なので、

```
\def\textannot#1#2#3#4{%
\special{pdf: ann width #1 height #2
<< /Type /Annot /Subtype /Text /Open #3
/Contents (#4) >>}
}
```

のようなマクロを作っておくと便利でしょう。そうすると、

```
\textannot{7cm}{5cm}{false}{\ANNOT}
```

のように書くことができます。

docinfo.out では、一項目は必ず一行でなければいけないことに注意して
下さい。ただし、行の最後の文字 \ によって行を継続できます。
プログラムはこの \ と、行末文字を取り除いて一行を構成します。

このようにして作られた一行の長さ制限は約 *16* キロバイトです。
通常気にする必要は無いでしょう。改行文字は上述のように取り除かれます
から、改行文字を陽に入れたい場合は *\012* としておきます。
このディレクトリにあるサンプル *web737w32.pdf* には、上のようにして
作成した、しおり、文書情報、*text annotation* が入れてあります。
*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/*
Tables for sjis ---> Unicode
*/

/*
以下に、Shift-JIS ---> Unicode テーブルがありますが、
TEX 出力には出さないようにします。
定義されている配列は
unsigned int in_sjis_a[];
unsigned int in_sjis_b[];
unsigned int in_sjis_c[];
unsigned int in_sjis_d[];
です。
再び TEX 出力をします。
*/
#define KFS1(a) ((0x81<=(a)) && ((a)<=0x9f))
#define KFS2(a) ((0xe0<=(a)) && ((a)<=0xfc))
#define KSC1(a) ((0x40<=(a)) && ((a)<=0x7e))
#define KSC2(a) ((0x80<=(a)) && ((a)<=0xfc))

/*
16 KB will be big enough.
*/
#define RDBUF 16384

static int euc = 0;
FILE *Fout;
FILE *Fin;

void usage(void)
{
  printf("Usage 1 :  out2uni foo\n");
```

```
#ifndef EUC
  printf("Usage 2 :  out2uni −e foo\n"
         "Default Japanese encoding is Shift−JIS.\n"
         "Option −e changes to EUC−JP.\n");
#else
  printf("Usage 2 :  out2uni −s foo\n"
         "Default Japanese encoding is EUC−JP.\n"
         "Option −s changes to Shift−JIS.\n");
#endif
}

/*
euc−japan
*/
int iskanji(int c)
{
  c &= 0xff;
  return(c>=0xa1 && c<=0xfe);
}

/*
shift−jis
*/
int iskanji1(int c)
{
  c &= 0xff;
  return((c>=0x81 && c<=0x9f) || (c>=0xe0 && c<=0xfc));
}

int iskanji2(int c)
{
  c &= 0xff;
  return(c>=0x40 && c<=0xfc && c!=0x7f);
}

/*
EUCtoSJIS
*/
unsigned int EUCtoSJIS(unsigned int k)
{
  unsigned int u;
  unsigned int high, low;
  unsigned int nh, nl;
```

```
  u = k & 0x7f7f;
  high = (u ≫ 8) & 0xff;
  low = u & 0xff;
  nh = ((high − 0x21) ≫ 1) + 0x81;
  if (nh > 0x9f) nh += 0x40;
  if (high & 1) {
    nl = low + 0x1f;
    if (low > 0x5f) nl++;
  } else
    nl = low + 0x7e;
  if(iskanji1(nh) && iskanji2(nl))
    return((nh ≪ 8) | nl);
  else
    return(0x813f);
}

/*
write unicode into the file Fout
*/
void putucode(unsigned int u)
{
  unsigned int a, b;

  a = u & 0xff;
  b = (u ≫ 8) & 0xff;

  fprintf(Fout,"%c%03o%c%03o",'\\',b,'\\',a);
}



/*
{string} or {}
 ∧∧∧∧∧∧∧ ∧
*/
void dopar(unsigned char **q)
{
  unsigned int a, b, u, eu, ea, eb;

  if(**q != '}' && **q) {
    putucode(0xfeff);
  }
```

```
while(**q != '}' && **q) {
  if(euc) {
    if(iskanji(**q)) {
      eu = EUCtoSJIS((**q) * 256 + *(*q + 1));
      ea = (eu >> 8) & 0xff;
      eb = eu & 0xff;
      **q = ea;
      *(*q + 1) = eb;
    }
  }
  a = **q; *q += 1;
  if(KFS1(a) || KFS2(a)) {
    b = **q; *q += 1;
    if (KFS1(a) && KSC1(b)) {
      u = in_sjis_a[(a - 0x81) * 63 + (b - 0x40)];
    }
    else if (KFS1(a) && KSC2(b)) {
      u = in_sjis_b[(a - 0x81) * 125 + (b - 0x80)];
    }
    else if (KFS2(a) && KSC1(b)) {
      u = in_sjis_c[(a - 0xe0) * 63 + (b - 0x40)];
    }
    else if (KFS2(a) && KSC2(b)) {
      u = in_sjis_d[(a - 0xe0) * 125 + (b - 0x80)];
    }
  }
  else if(a == 0x5c) {
    if('0' <= **q && **q <= '7') {
      u = **q - '0';
      *q += 1;
      if('0' <= **q && **q <= '7') {
        u = (u << 3) + **q - '0';
        *q += 1;
        if('0' <= **q && **q <= '7') {
          u = (u << 3) + **q - '0';
          *q += 1;
        }
      }
    }
    else {
      u = **q;
      *q += 1;
    }
```

```
      }
      else
        u = a;
      putucode(u);
    }
    putc('}', Fout);
    *q += 1;
}

int main(int argc, char **argv)
{
    unsigned int a, b;
    unsigned char *q;
    int i, len;
    char inname[256];
    char outname[256];
    char *p;
    unsigned char *b_in;

#ifdef WIN32
    strcpy(argv[0], "out2uni");
#endif

    if((argc == 1) || ((argc == 2) && !strncmp(argv[1], "-h", 2))) {
      usage();
      return 0;
    }
#ifndef EUC
    else if((argc == 3) && !strncmp(argv[1], "-e", 2)) {
      euc = 1;
#else
    else if((argc == 3) && !strncmp(argv[1], "-s", 2)) {
      euc = 0;
#endif
      strcpy(inname, argv[2]);
    }
    else if((argc == 2) && strncmp(argv[1], "-h", 2)) {
#ifndef EUC
      euc = 0;
#else
      euc = 1;
#endif
      strcpy(inname, argv[1]);
```

```
  }
  else {
    usage();
    return 1;
  }

  b_in = (unsigned char *)malloc(RDBUF);
  if(!b_in) {
    fprintf(stderr, "Memory allocation error.\n");
    exit (2);
  }

  p = strrchr(inname, '.');
#ifdef WIN32
  if((p == NULL) || stricmp(p, ".out"))
#else
  if((p == NULL) || strcmp(p, ".out"))
#endif
    strcat(inname, ".out");
  strcpy(outname, inname);
  strcat(outname, ".tmp");
  Fin = fopen(inname, "r");
  if(!Fin) {
    fprintf(stderr, "Cannot open %s to read.\n", inname);
    exit(1);
  }
  Fout = fopen(outname, "wb");
  if(!Fout) {
    fprintf(stderr, "Cannot open %s to write.\n", outname);
    exit(1);
  }

  while(fgets(b_in, RDBUF, Fin)) {
    if(*b_in == '\n') continue;
    len = strlen(b_in);
    if((b_in[len-2] == '\\') && (b_in[len-1] == '\n')) {
      do {
        b_in[len-2] = b_in[len-1] = '\0';
        if(!fgets(b_in + len - 2, RDBUF - len + 2, Fin)) {
          fprintf(stderr, "Syntax error.\n");
          exit (3);
        }
        len = strlen(b_in);
```

```
      } while((b_in[len-2] == '\\') && (b_in[len-1] == '\n'));
    }

    q = b_in;
    while(*q == ' ' || *q == '\t') q++;
    if(!strncmp(q,"@def",4)) {
      while(*q && *q != '{')
        putc(*q++, Fout);
    }
    else {
      if(*q) putc(*q++, Fout);
      while(*q) {
        if(*q == '}' && *(q-1) != '\\') {
          putc(*q++, Fout);
          break;
        }
        putc(*q++, Fout);
      }
      while(*q == ' ' || *q == '\t') q++;
    }
    if(*q) putc(*q++, Fout);
    if(*q) dopar(&q);
    while(*q)
      putc(*q++, Fout);
  }
  fclose(Fin);
  fclose(Fout);
  remove(inname);
  rename(outname,inname);
  return 0;
}
```