

## 1. Function and Use.

This small program will convert SJIS encoded Japanese characters into a ‘preprocessed’ form. The need of this program arises from the fact that this encoding uses the characters ‘\’, ‘{’, and ‘}’ which have special meanings in  $\text{\TeX}$ .

Use this program as a filter:

```
sjisconv < input_file > output_file
```

## 2. The program.

The only function of this program is to replace all occurrences of SJIS encoded two byte characters  $XY$  with  $\text{XZZ}$  ( $X$  and  $Y$  are the first and the second byte of the character;  $ZZ$  represents the second byte as a decimal number).

Additionally we define a  $\text{T}\text{E}\text{X}$  macro at the very beginning to signal a preprocessed file.

The following code is very simple. No error detection is done because  $\text{T}\text{E}\text{X}$  which will see the output of `sjisconv` complains loudly if something is wrong.

Note that the user-defined character area of SJIS (with the first bytes in the range 0xF0-0xFC) is not supported because it is not portable.

```
#define banner "sjisconv(CJKver.4.8.4)"
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{int ch;
  fprintf(stdout, "\\def\\CJKpreproc{%s}", banner);
  ch = fgetc(stdin);
  while (!feof(stdin))
  {if ((ch >= #81 & ch <= #9F) ∨ (ch >= #E0 & ch <= #EF))
    {fprintf(stdout, "\\177%c\\177", ch);
     ch = fgetc(stdin);
     if (!feof(stdin))
       fprintf(stdout, "%d\\177", ch);
    }
    else
      fputc(ch, stdout);
    ch = fgetc(stdin);
  }
  exit(EXIT_SUCCESS);
  return 0;
}
```

/\* never reached \*/