

〔非公開〕

TR-C-0045

U l t r i x コマンド利用者マニュアル

佐藤 隆
TAKASHI SATO

1 9 9 0 . 3 . 2

A T R 通信システム研究所

Ultrix コマンド利用者マニュアル

通信ソフトウェア研究室 佐藤 隆

1990年3月3日

このマニュアルは、Ultrix 上に作成したコマンドの使用方法について説明したものである。コマンドは、`/usr/local/bin` に登録してあります。

1. `add-new-schedule` - ファイル中のカレンダー情報をカレンダーに登録する
2. `args` - パラメータを単語ごとに切り分ける
3. `bell` - ベルを鳴らせることのできる `echo` コマンド
4. `chmod-move-to-tmp` - ファイルを他人に見えないようにして `/tmp` に移動する
5. `intex` - インテリジェント・テフ・プロセッサ
6. `jis-print` - JIS コード・ファイルを CS01 のプリンタに出力する
7. `jis-to-shift-jis` - JIS コードをシフト JIS コードに変換する
8. `jis-translator` - JIS コードにより作成されたファイルを変換する
9. `remove-temporary-files` - 作業用ファイルの撤去
10. `rmail-deliver` - `rmail` 形式のファイルを分割する
11. `shift-jis-to-jis` - シフト JIS コードを JIS コードに変換する
12. `shift-jis-to-jis-replace` - シフト JIS コード・ファイルの JIS コードへの変換 (ファイルのリブレース版)

add-new-schedule - ファイル中のカレンダー情報をカレンダーに登録する

SYNTAX

```
add-new-schedule [files]
```

DESCRIPTION

電子メールで送られてきたスケジュールを取り出し、ホーム・ディレクトリのファイル “calendar” に追加する。電子メール中のスケジュールは、“.cal” で始まる行に、次のような形式で記載する。

```
.cal mm/dd 曜日、時刻、場所、内容
```

記載例は、次のようになります。

```
.cal 12/26 Sat 18:00 新潟県三条市にて佐藤隆生まれる。
```

相手にどうしても出席してもらいたいならば、電子メールを出す人は本文中に上のような行を追加しておきましょう。

ファイル (files が無指定の場合は標準入力) 中のカレンダー情報を取り出し、ホーム・ディレクトリ上のカレンダー・ファイル “calendar” に追加する。カレンダー情報は、次の形式でなければならない。

- 1 カレンダー情報は、1 行に記入する。
- 1 行には、1 カレンダー情報しか記入できない。
- 行頭の 5 文字は、“.cal” または、“.cal タブ” でなければならない。
- 行頭の 6 文字目以降は、calendar コマンドで許されている形式でなければならない。

OPTIONS

なし

EXAMPLES

Add-new-schedule 受け付けるカレンダー情報の例は、次のようなものである。

```
.cal Dec. 26 Sato's birthday
```

```
.cal december 26 Happy birthday to you
```

```
.cal 12/26 My birthday
```

カレンダー情報の受け入れは、次のような手順で行ないます。

(1) 電子メールを読む。

```
atr-sw:/usr1/takashi/{56} mail
```

```
Mail version 2.18 5/19/83. Type ? for help.
```

```
"/usr/spool/mail/takashi": 1 message 1 new
```

```
>N 1 yokota Fri Dec 4 19:40 12/388 "Koshien Bowl"
```

```
& 1
```

```
Message 1:
```

```
From yokota Fri Dec 4 19:40:59 1987
```

```
Received: by atr-sw.atr.junet (4.12/4.7)
```

```
id AA23183; Fri, 4 Dec 87 19:40:58+0900
```

```
Date: Fri, 4 Dec 87 19:40:58+0900
```

From: yokota (Yokota Masanori)
Message-Id: <8712041040.AA23183@atr-sw.atr.junet>
To: takashi
Subject: Koshien Bowl
Status: R

甲子園ボールを見にゆきませんか?

.cal 12/13 Sun 8:30 甲子園球場 甲子園ボール (京大 vs 日大)

&

(2) そのスケジュールを受け入れるならば、メールをファイルにセーブし、終了する。受け入れるメールが複数個の場合は、ファイルの後ろに追加しますので気にせずセーブしてください。

& s cal

"cal" [New file] 12/388

& q

(3) セーブしたメール中のスケジュールを calendar に追加する。

atr-sw:/usr1/takashi/{57} add-new-schedule cal

.cal 12/13 Sun 8:30 甲子園球場 甲子園ボール (京大 vs 日大)

atr-sw:/usr1/takashi/{58}

RESTRICTIONS

FILES

~/calendar - カレンダー・ファイル

SEE ALSO

calendar (1), mail (1)

args - パラメータを単語ごとに切り分ける

SYNTAX

```
args [parameters]
```

DESCRIPTION

与えられたパラメータを単語ごとに切り分け、改行を付加して標準出力に出力する。単語は、英数字の並び、または空白・英数字を除く図形文字の並びである。

このコマンドは、シェル・スクリプトのパラメータで範囲指定を行なわせるために作成した。使用例は、intex コマンドにあります。

OPTIONS

なし

EXAMPLES

英数字列、図形文字列ごとに分けて、1行ずつ出力する。

```
args 1-2
```

```
1  
-  
2
```

英数字、特殊文字が複数個並んでいるときは、1個の単語として取り扱う。

```
args abc--123
```

```
abc  
--  
123
```

パラメータが特殊文字列で始まっている場合、終わっている場合も構わない。

```
args --123abc++
```

```
--  
123abc  
++
```

RESTRICTIONS

FILES

SEE ALSO

csh (1), intex (1)

bell - ベルを鳴らせることのできる echo コマンド

SYNTAX

```
bell [parameters]
```

DESCRIPTION

パラメータを解釈して、標準出力に出力する。パラメータの先頭文字がハイフン“-”でないときは、パラメータをそのまま出力する。パラメータの先頭文字がハイフンのときは、それ以降の文字をオプションと解釈し、それぞれの意味にしたがって出力する。

OPTIONS

-b - 空白 (040) を出力する。

-B - ベル (007) を出力する。

-N - 改行 (012) を出力する。

-T - タブ (011) を出力する。

-other - 文字をそのまま出力する。

もし、パラメータをすべて省略されたときには、“-B” が指定されたものと解釈し、標準出力にベルを出力する。

EXAMPLES

パラメータを指定しなければ、ベルが出力される。

```
bell
```

```
^G
```

文字列の途中で改行を挿入して出力できる。

```
bell Hello -N World.
```

```
Hello
```

```
World
```

複数個のオプションをまとめて指定することもできる。

```
bell I -b am -b cat. -T He -bb is not -bb dog.
```

```
I am cat. He is not dog.
```

RESTRICTIONS

FILES

SEE ALSO

echo (1), ascii (7)

chmod-move-to-tmp - ファイルを他人に見えないようにして /tmp に移動する

SYNTAX

```
chmod-move-to-tmp [files]
```

DESCRIPTION

パラメータで指定されたファイルの自分以外の参照許可フラグをオフにし、作業ディレクトリ “/tmp” に移動する。作業ディレクトリに同名のファイルが存在する場合は、可能ならばオーバーライトします。

OPTIONS

なし

EXAMPLES

RESTRICTIONS

同名のファイルが存在しているとオーバーライトする。

FILES

/tmp - 作業ディレクトリ

SEE ALSO

chmod (1)

intex - インテリジェント・テフ・プロセッサ

SYNTAX

`intex [options—files]`

DESCRIPTION

J_TE_X または JI_AT_EX の文法で記述されたソース・ファイルを J_TE_X プロセッサにより処理し、imagen プリンタに出力する。

intex は、ソース・ファイルの先頭行が “%- ” の場合、4 桁目以降をオプションとして解釈します。Emacs などを使っている人には便利な機能だと思います。

OPTIONS

- help - パラメータの一覧を表示する。
- copy *n* - プリント時のコピー枚数を *n* で指定する。
- nocopy - プリントしない。(-noprnt と同じ)
- noprnt - プリントしない。(-nocopy と同じ)
- save - dvi ファイルをカレントディレクトリにセーブする。
- nosave - dvi ファイルをセーブしない。
- prev[iewer] - -save、-nocopy を指定したときと同じ意味である。
- ref[er] - J_TE_X(または JI_AT_EX) を 2 回実行し、リファレンスを作成する。
- noref[er] - J_TE_X(または JI_AT_EX) を 1 回だけ実行する。リファレンスは考慮しない。
- ver[sion] {c 1.01 new pascal 0.9 old} - J_TE_X のバージョンを指定する。c、1.01、new は同じ意味であり、C 言語版の T_EX をベースとした J_TE_X(version 1.01)、JI_AT_EX により処理されます。pascal、0.9、old は同じ意味であり、pascal 言語版の T_EX をベースにした J_TE_X(version 0.9)、JI_AT_EX により処理されます。
- latex - JI_AT_EX により処理する。
- tex - J_TE_X により処理する。
- tty - ソースをそのままプリントする。漢字が混じっていても大丈夫。
- lisp - Lisp ソースをプリントする。
- slide - Sli_TE_X を実行する (OHP やスライド用に便利です)。
- sheet {a4 b4} {portrait port tate landscape land yoko} - シートのサイズ、置き方を指定するオプションです。a4、b4 は用紙のサイズを指定します。portrait などは、用紙の向きを指定するオプションです。portrait、port、tate は、同じ意味であり、用紙を縦に置きます。landscape、land、yoko は、同じ意味であり、用紙を横に置きます。

`-page s[-[e]]` - 印刷するページを指定する。s は印刷開始ページを表し、e は印刷終了ページを表す。e を省略した場合は、s ページから最後まで印刷する。-e を省略した場合は、s ページだけを印刷する。

`files` - 処理対象のファイル名を指定する。拡張子を省略すると、“.tex”、“.lisp” のうち適当なものが選ばれる。

オプションの省略時の値は、以下のとおりです。

```
intex -version 1.01 -copy 1 -nosave -noreferrer -sheet a4 portrait -page 1-
```

EXAMPLES

ファイル (foo.tex) を JIAT_EX し、imagen プリンタに出力する。

```
intex foo
```

ファイル (foo.tex) を JT_EX し、2～5 ページを imagen プリンタに出力する。

```
intex -tex -page 2-5 foo.tex
```

ファイル (foo.tex) を JIAT_EX し、imagen プリンタに出力する。JIAT_EX は、2 回行なわれ、参照部分 (`\cite` `\ref` `\pageref`) を自動的に埋めてくれる。

```
intex -refer ipsj-sig-ai.tex
```

RESTRICTIONS

1. ATR-SW 上では、オプション “-version 0.9” は使用できません。
2. JIS コード以外は処理していません。
3. -tty オプションにバグがあります。
4. SliT_EX のファイルが 2 個になって不細工だ。

FILES

`/usr/local/bin/intex-body` - JT_EX(または JIAT_EX) の本処理を制御するシェル・スクリプトである。

`/usr/local/etc/intex.man` - ヘルプ機能のためのファイルである。

`/utility/nttjtex/local/lib/tex/inputs` `/usr/lib/tex/macros` - JT_EX のスタイル・ファイルの格納場所である。シェルの環境変数 `TEXINPUTS` に値が設定されない場合にデフォルトとして選ばれる。

SEE ALSO

JT_EX, JIAT_EX

`jis-print` - JIS コード・ファイルを CS01 のプリンタに出力する

SYNTAX

```
jis-print [files]
```

DESCRIPTION

JIS コードで作成されたファイルを CS01 のプリンタに出力する。ファイル名が指定されない場合、標準入力を CS01 のプリンタに出力する。

OPTIONS

なし

EXAMPLES

~/News/Fj.jokes をプリントする。

```
jis-print ~/News/Fj.jokes
```

コマンド `grep` の結果をプリントする。

```
grep abc *.man | jis-print
```

RESTRICTIONS

FILES

SEE ALSO

`lbp` (1), `pr` (1)

jis-to-shift-jis - JIS コードをシフト JIS コードに変換する

SYNTAX

jis-to-shift-jis

DESCRIPTION

stdin から JIS コードファイルを入力し、シフト JIS コードに変換して標準出力に出力します。漢字 in/out のシーケンスとして、次のものをサポートしています。

漢字 in 033 \$ B または 033 \$

漢字 out 033 (B または 033 (J

OPTIONS

なし

EXAMPLES

RESTRICTIONS

歴史的な理由により、033 (H を漢字 out シーケンスとして認めている。ただし、警告メッセージは表示する。

FILES

SEE ALSO

shif-jis-to-jis (1)

jis-translator - JIS コードにより作成されたファイルを変換する

SYNTAX

```
jis-translator [options]
```

DESCRIPTION

オプションに従って、ファイルを変換する。ファイルは、JIS コードにより作成されていると解釈する。

OPTIONS

[+/-]kutoten - 句読点 (。、) を ピリオド空白 (.u)、カンマ空白 (.u) に変換する (+) しない (-)。

[+/-]h-kana - 半角カナコード (1 byte) を全角カナコード (2 byte) に変換する (+) しない (-)。半角カナ文字の後ろに濁点、半濁点が続く場合、濁点、半濁点付の全角カナコードに変換する。カナの空白 (0xA0) は、英字の空白 (0x20) に変換する。

[+/-]ascii-z-h - 全角 ASCII コードを半角 ASCII コード (0x20-0x7F) に変換する (+) しない (-)。全角から半角への変換は、JIS コード表に従って行なり。全角の空白は、半角の空白に変換する。しかし、全角の \ は半角の
に変換されるが、全角の ¥ マークは半角の ¥ マークに変換されない。

[+/-]ascii-h-z - 半角 ASCII コード (0x20-0x7F) を全角 ASCII コードに変換する (+) しない (-)。半角から全角への変換は、JIS コード表に従って行なり。半角の空白は、全角の空白に変換する。しかし、半角のは全角に変換されるが、半角の ¥ マークは全角の ¥ マークに変換されない。

[+/-]TeX - J_TEX、J_AT_EX モードを適用する (+) しない (-)。行の先頭文字が、'%', '、', '{', '}', '[', ']', '&', '\$' ならば、その行を変換対象としない。

-stdout - 変換結果を標準出力に出力する。

-stderr - 変換結果を標準エラー出力に出力する。

-output *file* - 変換結果をファイルに出力する。

files - 入力ファイルを指定する。入力ファイルが1つも指定されなかった場合、stdin から入力される。

[+/-]z-ascii - ascii-z-h オプションと同じ意味である。将来、削除する予定である。

[+/-]h-ascii - ascii-h-z オプションと同じ意味である。将来、削除する予定である。

これらのオプションは、出現順に処理される。

EXAMPLES

このコマンドでは、ファイル *f1* の半角カナコードが全角に変換され、標準出力に出力される。また、ファイル *f2* の半角コードは、全角に変換され (TeX モードで)、ファイル *f3* に出力される。

```
jis-translator +h-kana f1 -h-kana +TeX +ascii-h-z -output f3 f2
```

RESTRICTIONS

FILES

SEE ALSO

remove-temporary-files - 作業用ファイルの撤去

SYNTAX

```
remove-temporary-files [-TeX | -tex]
```

DESCRIPTION

カレント・ディレクトリより下にある作業用ファイルを撤去する。撤去したファイルは、同じ名前で "/tmp" に移動させられる。同名のファイルがあるときには、最後に撤去されたものが残る。このコマンドをログインファイル "/.login" に入れておくと、古い作業用ファイルが自動的に削除され、ディスクの節約になると思います。撤去対象ファイルおよび撤去条件は、次の通りである。

```
core    無条件に撤去する。
a.out   7日間以上アクセスしていないファイルを撤去する。
#*      "
*~      "
.*~     "
*.o     "
*.dvi   7日間以上アクセスしていないファイルを撤去する。(-TeX オプション指定時)
*.log   "
*.aux   "
*.bbl   "
*.glo   "
*.idx   "
*.lof   "
*.lot   "
*.toc   "
```

OPTIONS

TeX, tex - TeX 関連の作業ファイルを撤去する。

EXAMPLES

作業用ファイルを撤去する。

```
remove-temporary-files -TeX
```

RESTRICTIONS

FILES

core a.out *.o #* *~ .*~ *.dvi *.log *.aux *.bbl *.glo *.idx *.lof *.lot *.toc - 撤去対象ファイル名

SEE ALSO

rmail-deliver - rmail 形式のファイルを分割する

SYNTAX

```
rmail-deliver [-v] [files]
```

DESCRIPTION

Files で指定する Emacs の rmail 形式のファイルを入力し、各メールを1つのファイルにする。入力ファイル名を指定しない場合は、標準入力から入力する。

メールを出力するファイル名は、“発信日-発信者名.rmail”である。発信日は YYMMDD である。同一日に同一発信者から電子メールが発信されたならば、それらのメールは1つのファイルに格納される。Rmail 形式ファイルのヘッダ部は、ファイル “000000-BABYL.rmail” に出力される。

個々のファイルから rmail 形式のファイルを復元するには、cat コマンドを使用する。

OPTIONS

-v - 饒舌モードにする。

EXAMPLES

ファイル RMAIL を分割する。

```
rmail-deliver ~RMAIL
```

RESTRICTIONS

FILES

yymmdd-name.rmail - Rmail 形式ファイルの各メール部

000000-BABYL.rmail - Rmail 形式ファイルのヘッダ部

SEE ALSO

rmail (emacs)

shift-jis-to-jis - シフト JIS コードを JIS コードに変換する

SYNTAX

```
shift-jis-to-jis [-stdout] [-stderr]
```

DESCRIPTION

標準出力からのシフト JIS コード入力を JIS コードに変換して標準出力 (または標準エラー出力) に出力します。標準出力への出力は、バッファリングされますが、標準エラー出力への出力はバッファリングされません。漢字 in/out は、次のシーケンスで出力されます。

漢字 in 033 \$ B

漢字 out 033 (B

OPTIONS

-stdout - 標準出力に出力する。

-stderr - 標準エラー出力に出力する。バッファリングしない。

EXAMPLES

RESTRICTIONS

入力ファイル名を指定できませんので、あしからず。(タ～)

FILES

SEE ALSO

jis-to-shift-jis (1)

shift-jis-to-jis-replace - シフト JIS コード・ファイルの JIS コードへの変換 (ファイルのリプレース版)

SYNTAX

```
shift-jis-to-jis-replace files
```

DESCRIPTION

Files で指定されたファイルのシフト JIS コードを JIS コードに変換し、同一名のファイルに上書きする。

OPTIONS

なし

EXAMPLES

カレント・ディレクトリ上のファイル (*.jxw) のコード変換を行なう。

```
shift-jis-to-jis-replace *.jxw
```

RESTRICTIONS

拡張子が “.jis” であるようなファイルは、ヌル・ファイルになります。

FILES

*.jis - 作業用ファイル名

SEE ALSO

shift-jis-to-jis (1)

APPENDIX: ソース・リスト

- add-new-schedule.sh
- args.c
- bell.c
- chmod-move-to-tmp.sh
- intex.sh
- intex.man
- intex-body.sh
- jis-print.sh
- jis-to-shift-jis.c
- jis-translator
 - Makefile
 - translator.h
 - variables.h
 - changer.c
 - translator.c
 - variables.c
- remove-temporary-files.sh
- rmail-deliver.c
- shift-jis-to-jis.c
- shift-jis-to-jis-replace.sh

```
#!/bin/sh
if [ $# = 0 ]
then
    grep "^%.cal[ ]" | sed -e "s/%.cal[ ]*/" | tee -a ${HOME}/calendar
else
    for fil in $*
    do
        grep "^%.cal[ ]" ${fil} | sed -e "s/%.cal[ ]*/" | tee -a ${HOME}/calendar
    done
fi
```

```
#include <stdio.h>
#include <ctype.h>
#define S 0
#define W 1
#define B 2
#define I 3
main(argc, argv)
    int argc;
    char *argv[];
{
    int i, f;
    char *p;

    for (i = 1; i < argc; i++) {
        for (f = I, p = argv[i]; *p != '\0'; p++) {
            if (isdigit(*p) || isalpha(*p)) {
                if (f == S || f == B)
                    putchar('\n');
                f = W;
                putchar(*p);
            } else {
                if (f == W)
                    putchar('\n');
                if (*p == ',' || *p == '\t') {
                    switch (f) {
                        case W:      f = B;      break;
                        case B: case I:      break;
                        case S:      putchar(*p); break;
                    }
                } else {
                    f = S;
                    putchar(*p);
                }
            }
        }
        putchar('\n');
    }
    exit(0);
}
```

```

#include <stdio.h>

main(ac, av)
int ac;
char *av[];
{
    int chputed = 0;
    int nlputed = 1;
    int spputed = 1;
    int i, j;
    char *p;

    for (i = 1; i < ac; i++) {
        if (*av[i] == '-') {
            for (p = av[i] + 1; *p; p++) {
                switch (*p) {
                    case 'b':
                        putchar(' ');
                        chputed = 1;
                        nlputed = 0;
                        spputed = 1;
                        break;
                    case 'B':
                        putchar('%07');
                        chputed = 1;
                        break;
                    case 'N':
                        putchar('%n');
                        chputed = 1;
                        nlputed = 1;
                        spputed = 1;
                        break;
                    case 'T':
                        putchar('%t');
                        chputed = 1;
                        nlputed = 0;
                        spputed = 1;
                        break;
                    default:
                        putchar(*p);
                        chputed = 1;
                        nlputed = 0;
                        spputed = 0;
                        break;
                }
            }
        } else {
            if (!spputed)
                putchar(' ');
            fputs(av[i], stdout);
            chputed = 1;
            nlputed = 0;
            spputed = 0;
        }
        fflush(stdout);
    }
    if (!chputed)
        putchar('%07');
    if (!nlputed)
        putchar('%n');
    exit(0);
}

```

```
/bin/chmod go-rwx $*  
/bin/mv $* /tmp
```

```

#!/bin/csh
onintr rmfiles
#set source=cat
set source=source
#set body=/soft/takashi/bin/intex-body.sh
set body=/usr/local/bin/intex-body
set cont=
set script=
set options="/tmp/#intex$$options"
set command="/tmp/#intex$$command"
echo -n "" >! ${options}
#
foreach opt (${argv[*]})
  switch (${opt})
  case -help:
    cat /soft/takashi/bin/intex.man
    exit 0
  case -sheet:
    set cont=sheet1
    echo -n " ${opt}" >> ${options}
    breaksw
  case -copy:
  case -page:
  case -version:
  case -ver:
    set cont=yes
    echo -n " ${opt}" >> ${options}
    breaksw
  case -latex:
  case -tex:
  case -slide:
    set script=
    set cont=
    echo -n " ${opt}" >> ${options}
    breaksw
  case -tty:
  case -lisp:
    set script=yes
    set cont=
    echo -n " ${opt}" >> ${options}
    breaksw
  case -*:
    set cont=
    echo -n " ${opt}" >> ${options}
    breaksw
  case *:
    switch (${cont})
    case yes:
      set cont=
      echo -n " ${opt}" >> ${options}
      breaksw
    case sheet1:
      set cont=sheet2
      echo -n " ${opt}" >> ${options}
      breaksw
    case sheet2:
      set cont=
      if (${opt} == "a4" || ${opt} == "b4") then
        echo -n " ${opt}" >> ${options}
        breaksw
      endif
      if (${opt} == "tate" || ${opt} == "portrait" || ${opt} == "port") then
        echo -n " portrait" >> ${options}
        breaksw
      endif
      if (${opt} == "yoko" || ${opt} == "landscape" || ${opt} == "land") then
        echo -n " landscape" >> ${options}
        breaksw
      endif
    thru.
  default:
    set cont=
    if (${script} == "") then
      set bar="${opt}"
      if (${opt:r} == ${opt}) then
        set bar="${opt}.tex"
      endif
      if (! -e ${bar}) then
        echo "Can not read the file ${bar}."
        exit 1
      endif
      echo -n "${body}" >! ${command}
      set top=`head -l ${bar} | cut -c4-`
      set sep=`head -l ${bar} | cut -c1-3`
      if ("${sep}" == "%-") then
        echo -n " ${top}" >> ${command}
      endif
    else
      echo -n "${body}" >! ${command}

```

```
endif
cat ${options} >> ${command}
echo " ${opt}" >> ${command}
${source} ${command}
breaksw
endsw
breaksw
endsw
end
rmfiles:
/bin/rm -f ${options} ${command}
```

intexの使い方

形式

`/usr/local/bin/intex [オプション | ファイル名] ...`

機能

JLaTeXソース（当然JISコード）をJLaTeXし、imagenプリンタに出力する。

オプション

<code>-help</code>	パラメータの一覧を表示する。
<code>-copy n</code>	プリント時のコピー枚数をnで指定する。
<code>-nocopy</code>	プリントしない。(<code>-noprint</code> と同じ)
<code>-noprint</code>	プリントしない。(<code>-nocopy</code> と同じ)
<code>-save</code>	dviファイルをカレントディレクトリにセーブする。
<code>-nosave</code>	dviファイルをセーブしない。
<code>-prev[iewer]</code>	<code>-save</code> 、 <code>-nocopy</code> を指定したときと同じ意味である。
<code>-ref[er]</code>	LaTeXを2回実行し、リファレンスを作成する。
<code>-noref[er]</code>	LaTeXを1回だけ実行する。リファレンスは考慮しない。
<code>-ver[ision]</code> (c 1.01 new pascal 0.9 old)	TeXのバージョンを指定する。c、1.01、newは同じ意味であり、C言語版のTeXをベースにしたJTeX (version 1.01)、JLaTeXにより処理されます。pascal、0.9、oldは同じ意味であり、pascal言語版のTeXをベースにしたJTeX (version 0.9)、JLaTeXにより処理されます。
<code>-latex</code>	JLaTeXを実行する。
<code>-tex</code>	JTeXを実行する。
<code>-tty</code>	ソースをそのままプリントする。漢字が混じっていても大丈夫。
<code>-lisp</code>	Lispソースをプリントする。
<code>-slide</code>	Slitexを実行する。(OHPやスライド用に便利です)
<code>-sheet (a4 b4)</code> (portrait port tate landscape land yoko)	シートのサイズ、置き方を指定する。a4、b4は用紙のサイズを指定します。portraitなどは、用紙の向きを指定するオプションです。portrait、port、tateは、同じ意味であり、用紙を縦に置きます。landscape、land、yokoは、同じ意味であり、用紙を横に置きます。
<code>-page s[-[e]]</code>	印刷するページを指定する。sは印刷開始ページを表し、eは印刷終了ページを表す。eを省略した場合は、sページから最後まで印刷する。-eを省略した場合は、sページだけを印刷する。

ファイル名 処理対象のファイルを指定する。拡張子を省略すると、tex、lispのうち適当なものが選ばれる。

オプションの省略時の値は、以下のとおりです。

```
intex -version 0.9 -copy 1 -nosave -norefer -sheet a4 portrait -page 1-
```

intexは、ファイルの先頭行が“%-”の場合、4桁目以降をオプションとして解釈します。emacsなどを使っている人には便利な機能だと思います。

```

#!/bin/csh
set version="1.01"
set bell=/usr/local/bin/bell
#set jdviimp=echo
set jdviimp=/utility/nttjtex/local/bin/jdviimp
set tex=jlatex
set script=
set cont=
set copy="-c1"
set refer=
set sheet="-Dpaper a4"
set place=
set page=
set save=
set tmpfile="/tmp/${user}$$$.TeX-log"
#
foreach opt (${argv[*]})
  switch (${opt})
  case -[0-9]*:
  case -[0-9]*-:
  case -[0-9]*-[0-9]*:
  case -[0-9]*--[0-9]*:
    breaksw
  case -*:
    if (${cont} != "" && ${cont} != "sheet2") then
      echo "No following value after -${cont}."
      exit 1
    endif
    breaksw
  endsw
#
  switch (${opt})
  case -copy:
    set cont=copy
    breaksw
  case -nocopy:
  case -noprint:
    set copy="-c0"
    breaksw
  case -save:
    set save=yes
    breaksw
  case -previewer:
  case -prev:
    set save=yes
    set copy="-c0"
    breaksw
  case -nosave:
    set save=
    breaksw
  case -refer:
  case -ref:
    set refer=yes
    breaksw
  case -norefer:
  case -noref:
    set refer=
    breaksw
  case -version:
  case -ver:
    set cont=version
    breaksw
  case -latex:
    set tex=jlatex
    set script=
    breaksw
  case -slide:
    set tex=jslitex
    set script=
    breaksw
  case -tex:
    set tex=jtex
    set script=
    breaksw
  case -tty:
    set tex=jlatex
    set script=tty
    breaksw
  case -lisp:
    set tex=jlatex
    set script=lisp
    breaksw
  case -sheet:
    set cont=sheet1
    breaksw
  case -page:
    set cont=page
    breaksw
  case -[0-9]*:

```

```

case -[0-9]*-:
case -[0-9]*-[0-9]*:
case -[0-9]*--[0-9]*:
    goto contproc
case -*:
    echo "Invalid option ${opt}."
    exit 1
case *:
contproc:
    switch (${cont})
    case version:
        set cont=
        switch (${opt})
        case c:
        case 1.01:
        case new:
            set version="1.01"
            breaksw
        case pascal:
        case 0.9:
        case old:
            set version="0.9"
            breaksw
        default:
            echo "Invalid keyword ${opt} after -version."
            exit 1
        endsw
    breaksw
case copy:
    set cont=
    if (${opt} < 1) then
        echo "Invalid keyword ${opt} after -copy."
        exit 1
    endif
    set copy="-c${opt}"
    breaksw
case page:
    set cont=
    set foo=`args ${opt}`
    switch (${#foo})
#   [0-9]
    case 1:
        if (${foo[1]} >= 0) then
            set page="-B${foo[1]} -N1"
            breaksw
        endif
        echo "Invalid keyword ${opt} after -page."
        exit 1
#   -[0-9] or [0-9]-
    case 2:
        if (${foo[1]} == - && ${foo[2]} >= 0) then
            set page="-B-${foo[2]} -N1"
            breaksw
        endif
        if (${foo[2]} == - && ${foo[1]} >= 0) then
            set page="-B${foo[1]}"
            breaksw
        endif
        echo "Invalid keyword ${opt} after -page."
        exit 1
#   -[0-9]- or [0-9]-[0-9]
    case 3:
        if (${foo[1]} == - && ${foo[3]} == - && ${foo[2]} >= 0) then
            set page="-B-${foo[2]}"
            breaksw
        endif
        if (${foo[2]} == - && ${foo[1]} <= ${foo[3]}) then
            set page="-B${foo[1]} -N`expr ${foo[3]} - ${foo[1]} + 1`"
            breaksw
        endif
        echo "Invalid keyword ${opt} after -page."
        exit 1
#   -[0-9]-[0-9] or -[0-9]--[0-9]
    case 4:
        if (${foo[1]} == - && ${foo[3]} == - && ${foo[2]} >= 0 && ${foo[4]} >= 0) then
            set page="-B-${foo[2]} -N`expr ${foo[4]} + ${foo[2]} + 1`"
            breaksw
        endif
        if (${foo[1]} == - && ${foo[3]} == -- && ${foo[2]} >= ${foo[4]}) then
            set page="-B-${foo[2]} -N`expr ${foo[2]} - ${foo[4]} + 1`"
            breaksw
        endif
        echo "Invalid keyword ${opt} after -page."
        exit 1
    endsw
    breaksw
case sheet1:
    set cont=sheet2
    if (${opt} == "a4" || ${opt} == "b4") then

```

```

    set sheet="-Dpaper ${opt}"
    breaksw
  endif
  if (${opt} == "tate" || ${opt} == "portrait" || ${opt} == "port") then
    set place=
    breaksw
  endif
  if (${opt} == "yoko" || ${opt} == "landscape" || ${opt} == "land") then
    set place="-L"
    breaksw
  endif
  echo "Invalid keyword ${opt} after -sheet."
  exit 1
case sheet2:
  set cont=
  if (${opt} == "a4" || ${opt} == "b4") then
    set sheet="-Dpaper ${opt}"
    breaksw
  endif
  if (${opt} == "tate" || ${opt} == "portrait" || ${opt} == "port") then
    set place=
    breaksw
  endif
  if (${opt} == "yoko" || ${opt} == "landscape" || ${opt} == "land") then
    set place="-L"
    breaksw
  endif
# thru.
default:
  if (${script} == "") then
    set cont=
    set foo=${opt:r}
    set bar="${opt}"
    if (${foo} == ${opt}) then
      set bar="${opt}.tex"
    endif
    if (! -e ${bar}) then
      echo "Can not read the file ${bar}."
      exit 1
    endif
    /bin/rm -f ${foo}.aux ${foo}.dvi ${foo}.log
    /bin/rm -f ${foo}.bbl ${foo}.glo ${foo}.idx ${foo}.lof ${foo}.lot ${foo}.toc
    onintr rmfiles
    if (${version} == "0.9") then
      set bin=/usr/local/bin
      echo Sorry, I do not support old version TeX from Feb. 1, 1990.
      exit 1
    else
      set bin=/utility/nttjtex/local/bin
      if (${TEXINPUTS} == "0") then
        setenv TEXINPUTS ".:/utility/nttjtex/local/lib/tex/inputs:/usr/lib/tex/macros"
      endif
    endif
    ${bin}/${tex} ${bar}
    if (${status} != 0) goto rmfiles
    if (-e ${foo}.aux && ${refer} == yes) then
      ${bin}/${tex} ${bar}
      if (${status} != 0) goto rmfiles
    endif
    if (-e ${foo}.dvi && ${copy} != "-c0") then
      ${bell} ... Loading Font ${bar} ... -N
      ${jdviimp} -Pimagen ${copy} "${sheet}, file ${bar}" ${place} ${page} ${foo}.dvi >&! ${tmpfile}
  ile)
    if (${status} == 0) then
      ${bell} -B ... Sending, ${foo}.dvi ... -N
    else
      ${bell} -B @@@ Error Occurred, ${foo}.dvi @@@ -N
      cat ${tmpfile}
    endif
    /bin/rm -f ${tmpfile}
    if (${save} != yes) /bin/rm -f ${foo}.dvi
  endif
  /bin/rm -f ${foo}.log ${foo}.aux
  /bin/rm -f ${foo}.bbl ${foo}.glo ${foo}.idx ${foo}.lof ${foo}.lot ${foo}.toc
else
  set bar="${opt}"
  if (! -e ${opt}) then
    set foo=${opt:r}
    if (${foo} == ${opt}) then
      set bar="${opt}.${script}"
    endif
    if (! -e ${bar}) then
      echo "Can not read the file ${bar}."
      exit 1
    endif
  endif
  /bin/rm -f ${script}.dvi ${script}.log ${script}.aux
  /bin/rm -f ${script}.bbl ${script}.glo ${script}.idx ${script}.lof ${script}.lot ${script}.t
oc

```

```

onintr rmscripts
  if (${version} == "0.9") then
    set bin=/usr/local/bin
    set mac=/usr/lib/tex/macros
  else
    set bin=/utility/nttjtex/local/bin
    set mac=/utility/nttjtex/local/lib/tex/inputs
    if (${?TEXINPUTS} == "0") then
      setenv TEXINPUTS ".:/utility/nttjtex/local/lib/tex/inputs"
    endif
  endif
  set name="#intex$.nl"
  echo "${bar}" >! ${name}
  ${bin}/${tex} ${mac}/${script}.scr < ${name}
  if (${status} != 0) goto rmscripts
  if (-e ${script}.dvi && ${copy} != "-c0") then
    ${bell} ... Loading Font ${bar} ... -N
    ${jdviimp} -Pimagen ${copy} "${sheet}, file ${bar}" ${place} ${page} ${script}.dvi >&! ${t
mpfile}
    if (${status} == 0) then
      ${bell} -B ... Sending, ${script}.dvi ... -N
    else
      ${bell} -B @@@ Error Occurred, ${script}.dvi @@@ -N
      cat ${tmpfile}
    endif
    if (${save} != yes) /bin/rm -f ${script}.dvi
  endif
  /bin/rm -f ${script}.log ${script}.aux ${name}
  /bin/rm -f ${script}.bbl ${script}.glo ${script}.idx ${script}.lof ${script}.lot ${script}.t
oc
  endif
  breaksw
endsw
breaksw
endsw
end
exit 0
rmfiles:
/bin/rm -f ${foo}.dvi ${foo}.log ${foo}.aux
/bin/rm -f ${foo}.bbl ${foo}.glo ${foo}.idx ${foo}.lof ${foo}.lot ${foo}.toc
/bin/rm -f ${tmpfile}
exit 1
rmscripts:
/bin/rm -f ${script}.dvi ${script}.log ${script}.aux ${name}
/bin/rm -f ${script}.bbl ${script}.glo ${script}.idx ${script}.lof ${script}.lot ${script}.toc
/bin/rm -f ${tmpfile}
exit 1

```

```
#!/bin/csh
if ({#argv} == 0) then
    jis-to-shift-jis | kcode +m | fold -80 | pr -w80 -l66 | kcode -m | lbp -r
    echo LBPed.
else
    foreach opt ({argv[*]})
        jis-to-shift-jis < {opt} | kcode +m | fold -80 | pr -w80 -l66 -h " {opt}" | kcode
    -m | lbp -r
        echo {opt} LBPed.
    end
endif
```

```

#include <stdio.h>
#define ON 1
#define OFF 0

#define M_ASCII 0 /* ASCII mode */
#define M_ESCAP 1 /* Escaped mode */
#define M_ESCKO 2 /* ESC-KO mode */
#define M_ESCKI 3 /* ESC-KI mode */
#define M_KANJI 4 /* KANJI mode */
#define M_IBM 5 /* IBM mode */
#define M_KANA 6 /* KANA mode */

#define INPUT stdin
#define OUTPUT stdout

#define ESC '\033'
#define KI2 '$'
#define KI3_1 'B' /* to ASCII */
#define KI3_2 'e' /* to JIS Roman */
#define K02 '('
#define K03_1 'B' /* to Kanji (1978) */
#define K03_2 'J' /* to Kanji (1983) */
#define SWEDEN3 'H' /* to Sweden */
#define S0 0x0E
#define SI 0x0F

#define KUTEN1 0x81
#define KUTEN2 0x42
#define TOTEN1 0x81
#define TOTEN2 0x41

int texmode = OFF; /* TeX mode (Kutou-ten -> ".",") */

main(argc, argv)
int argc;
char *argv[];
{
    int stat;
    int i;
    register unsigned c;
    register unsigned c1;
    register unsigned c2;
    unsigned s1;
    unsigned s2;

    for (i = 1; i < argc; i++) {
        if (strcmp(argv[i], "-TeX") == 0) {
            texmode = ON;
            continue;
        }
    }
    stat = M_ASCII;
    while ((c = getc(INPUT)) != EOF) {
        switch (stat) {
            case M_ASCII: /* ? */
                if (c == ESC)
                    stat = M_ESCAP;
                else if (c == S0)
                    stat = M_KANA;
                else
                    putc(c, OUTPUT);
                break;
            case M_ESCAP: /* ESC ----> ? */
                switch (c) {
                    case K02:
                        stat = M_ESCKO;
                        break;
                    case KI2:
                        stat = M_ESCKI;
                        break;
                    case SI:
                        putc(ESC, OUTPUT);
                        stat = M_ASCII;
                        break;
                    case S0:
                        putc(ESC, OUTPUT);
                        stat = M_KANA;
                        break;
                    case ESC:
                        putc(ESC, OUTPUT);
                        break;
                    default:
                        putc(ESC, OUTPUT);
                        putc(c, OUTPUT);
                        /* stat = M_ASCII; */
                        break;
                }
                break;
            case M_ESCKO: /* ESC --> K02 --> ? */

```

```

switch (c) {
case SWEDEN3:
    fprintf(stderr, "Warnig: A input use the swedish caraceter set.¥n");
case K03_1:
case K03_2:
    stat = M_ASCII;
    break;
case S0:
    putc(ESC, OUTPUT);
    putc(K02, OUTPUT);
    stat = M_KANA;
    break;
case S1:
    putc(ESC, OUTPUT);
    putc(K02, OUTPUT);
    stat = M_ASCII;
    break;
case ESC:
    putc(ESC, OUTPUT);
    putc(K02, OUTPUT);
    stat = M_ESCAP;
    break;
default:
    putc(ESC, OUTPUT);
    putc(K02, OUTPUT);
    putc(c, OUTPUT);
    /* stat = M_ASCII; */
    break;
}
break;
case M_ESCKI: /* ESC --> KI2 --> ? */
    switch (c) {
    case KI3_1:
    case KI3_2:
        stat = M_KANJI;
        break;
    case S0:
        putc(ESC, OUTPUT);
        putc(KI2, OUTPUT);
        stat = M_KANA;
        break;
    case S1:
        putc(ESC, OUTPUT);
        putc(KI2, OUTPUT);
        stat = M_ASCII;
        break;
    case ESC:
        putc(ESC, OUTPUT);
        putc(KI2, OUTPUT);
        stat = M_ESCAP;
        break;
    default:
        putc(ESC, OUTPUT);
        putc(KI2, OUTPUT);
        putc(c, OUTPUT);
        stat = M_ASCII;
        break;
    }
    break;
case M_KANJI:
    if (c == ESC)
        stat = M_ESCAP;
    else if (c == S0)
        ;
    else {
        c1 = c;
        stat = M_IBM;
    }
    break;
case M_IBM:
    c2 = c;
    JIS_IBM(c1, c2, &s1, &s2);
    if (texmode) {
        if (s1 == KUTEN1 && s2 == KUTEN2) {
            s1 = (int) ' ';
            s2 = (int) ' ';
        } else if (s1 == TOTEN1 && s2 == TOTEN2) {
            s1 = (int) ' ';
            s2 = (int) ' ';
        }
    }
    putc(s1, OUTPUT);
    putc(s2, OUTPUT);
    stat = M_KANJI;
    break;
case M_KANA:
    if (c == S0);
    else if (c == S1)
        stat = M_ASCII;

```

```
        else {
            if ((c >= 0x21) && (c <= 0x5F))
                c |= 0x80;
            putchar(c, OUTPUT);
        }
        break;
    }
}
exit(0);
}

JIS_IBM(jis1, jis2, ibm1, ibm2)
register unsigned jis1;
register unsigned jis2;
register unsigned *ibm1;
register unsigned *ibm2;
{
    if (jis1 >= 0x5F)
        jis1 += 0x80;
    if ((jis1 % 2) == 0) {
        *ibm1 = (jis1 - 0x30) / 2 + 0x88;
        *ibm2 = jis2 + 0x7E;
    } else {
        if (jis2 >= 0x60)
            jis2 = jis2 + 0x01;
        *ibm1 = (jis1 - 0x31) / 2 + 0x89;
        *ibm2 = jis2 + 0x1F;
    }
    *ibm1 &= 0xFF;
    *ibm2 &= 0xFF;
    return;
}
```

```
CFLAGS=-I/usr1/takashi/tools/lib/  
LIB=/usr1/takashi/tools/lib/jlib.ar  
HEAD= translator.h variables.h  
OBJS= translator.o variables.o changer.o  
jis-translator: ${OBJS}  
        cc ${OBJS} ${LIB} -o jis-translator  
translator.o: translator.h variables.h  
variables.o: translator.h variables.h  
changer.o: translator.h variables.h
```

```
#define ER_OPT "%s: Unkown option '%s'.%n"
#define ER_NGET "%s: Can not read '%s'.%n"
#define ER_NPUT "%s: Can not write '%s'.%n"
#define ER_NPAR "%s: No argument.%n"

#define KUTEN 0x2123
#define TOTEN 0x2122

#define C_DAK 0xDE /* Hankaku dakuten code */
#define C_HAN 0xDF /* Hankaku handakuten code */
#define KANABEG 0xA0 /* Hankaku katakana kaishi ichi */
#define KANAEND 0xDF /* Hankaku katakana shuryo ichi */
#define HZSIZE (KANAEND-KANABEG+1)
#define CVTAB struct _cvtab_
CVTAB {
    int cv_code; /* Zenkaku code */
    int cv_dak; /* Dakuten tsuki */
    int cv_han; /* Handakuten tsuki */
};

#define BUPPER 0x2341 /* Zenkaku UPPER kaishi ichi */
#define EUPPER 0x235A /* Zenkaku UPPER shuryo ichi */
#define BLOWER 0x2361 /* Zenkaku LOWER kaishi ichi */
#define ELOWER 0x237A /* Zenkaku LOWER shuryo ichi */
#define BDIGIT 0x2330 /* Zenkaku DIGIT kaishi ichi */
#define EDIGIT 0x2339 /* Zenkaku DIGIT shuryo ichi */
#define ALPSIZ 33 /* Zenkaku tokushu-moji kosu */
#define ALPTAB struct _alptab_
ALPTAB {
    int at_zen; /* Zenkaku code */
    int at_han; /* Hankaku code */
};

#define EPRINT(f, p) fprintf(stderr, f, argv[0], p)
```

```
extern int    kutoten;      /* Kutoten -> ".," */
extern int    hankaku;     /* Hankaku-kana -> Zenkaku */
extern int    z_ascii;     /* Zenkaku-ascii -> Hankaku */
extern int    h_ascii;     /* Hankaku-ascii -> Zenkaku */
extern int    tex_mode;    /* TeX Mode (ignore after ¥{}[!$&%) */
extern char   *replace;    /* Replace file name */

extern CVTAB  h2z[HZSIZE]; /* Hankaku to zenkaku table */
extern ALPTAB alp[ALPSIZ]; /* Zenkaku ASCII CODE table */
```

```

#include <jstdio.h>
#include "translator.h"
#include "variables.h"

changer(ijfp, ojfp)
JFILE *ijfp;
JFILE *ojfp;
{
    int c;
    int old_c = '\n';
    int c_sw = ON;
    int d;
    int k;
    int osw; /* Output switch */

    while ((c = jgetc(ijfp)) != EOF) {
        osw = ON;
        if (tex_mode && old_c == '\n')
            c_sw = (c != '{' && c != '}' && c != '[' && c != ']' &&
                    c != '¥' && c != '&' && c != '$' && c != '%');
        old_c = c;
        if (c_sw) {
            if (hankaku && KANABEG <= c && c <= KANAEND) {
                k = c - KANABEG;
                c = h2z[k].cv_code;
                if (h2z[k].cv_dak) {
                    d = jgetc(ijfp);
                    if (d == C_DAK)
                        c = h2z[k].cv_dak;
                    else if (d == C_HAN && h2z[k].cv_han)
                        c = h2z[k].cv_han;
                    else
                        jungetc(d, ijfp);
                }
            }
            if (kutoten) {
                if (c == KUTEN) {
                    jputc('.', ojfp);
                    jputc('.', ojfp);
                    osw = OFF;
                } else if (c == TOTEN) {
                    jputc(',', ojfp);
                    jputc(',', ojfp);
                    osw = OFF;
                }
            }
            if (z_ascii) {
                if (BUPPER <= c && c <= EUPPER)
                    c = c + ('A' - BUPPER);
                else if (BLOWER <= c && c <= ELLOWER)
                    c = c + ('a' - BLOWER);
                else if (BDIGIT <= c && c <= EDIGIT)
                    c = c + ('0' - BDIGIT);
                else {
                    for (k = 0; k < ALPSIZ; k++) {
                        if (c == alp[k].at_zen) {
                            c = alp[k].at_han;
                            break;
                        }
                    }
                }
            }
            if (h_ascii) {
                if ('A' <= c && c <= 'Z')
                    c = c + (BUPPER - 'A');
                else if ('a' <= c && c <= 'z')
                    c = c + (BLOWER - 'a');
                else if ('0' <= c && c <= '9')
                    c = c + (BDIGIT - '0');
                else {
                    for (k = 0; k < ALPSIZ; k++) {
                        if (c == alp[k].at_han) {
                            c = alp[k].at_zen;
                            break;
                        }
                    }
                }
            }
        }
        if (osw)
            jputc(c, ojfp);
    }
    return;
}

```

```

#include <jstdio.h>
#include "translator.h"
#include "variables.h"

main(argc, argv)
int    argc;
char   *argv[];
{
    int    i;
    JFILE *sijfp = mkjfp(stdin);
    JFILE *sojfp = mkjfp(stdout);
    JFILE *sejfp = mkjfp(stderr);
    JFILE *ijfp;
    JFILE *ojfp;

    ijfp = JFNULL;
    ojfp = sojfp;
    for (i = 1; i < argc; i++) {
        if (argv[i][0] == '-' || argv[i][0] == '+') {
            if (strcmp(argv[i], "+kutoten") == 0) {
                kutoten = ON;
                continue;
            }
            if (strcmp(argv[i], "-kutoten") == 0) {
                kutoten = OFF;
                continue;
            }
            if (strcmp(argv[i], "+h-kana") == 0) {
                hankaku = ON;
                continue;
            }
            if (strcmp(argv[i], "-h-kana") == 0) {
                hankaku = OFF;
                continue;
            }
            if (strcmp(argv[i], "+TeX") == 0) {
                tex_mode = ON;
                continue;
            }
            if (strcmp(argv[i], "-TeX") == 0) {
                tex_mode = OFF;
                continue;
            }
            if (strcmp(argv[i], "+z-ascii") == 0 ||
                strcmp(argv[i], "+ascii-z-h") == 0) {
                z_ascii = ON;
                continue;
            }
            if (strcmp(argv[i], "-z-ascii") == 0 ||
                strcmp(argv[i], "-ascii-z-h") == 0) {
                z_ascii = OFF;
                continue;
            }
            if (strcmp(argv[i], "+h-ascii") == 0 ||
                strcmp(argv[i], "+ascii-h-z") == 0) {
                h_ascii = ON;
                continue;
            }
            if (strcmp(argv[i], "-h-ascii") == 0 ||
                strcmp(argv[i], "-ascii-h-z") == 0) {
                h_ascii = OFF;
                continue;
            }
            if (strcmp(argv[i], "-stdout") == 0) {
                jflush(ojfp);
                if (ojfp != sojfp && ojfp != sejfp)
                    jclose(ojfp);
                ojfp = sojfp;
                continue;
            }
            if (strcmp(argv[i], "-stderr") == 0) {
                jflush(ojfp);
                if (ojfp != sojfp && ojfp != sejfp)
                    jclose(ojfp);
                ojfp = sejfp;
                continue;
            }
            if (strcmp(argv[i], "-output") == 0) {
                jflush(ojfp);
                if (ojfp != sojfp && ojfp != sejfp)
                    jclose(ojfp);
                if (i + 1 >= argc) {
                    EPRINT(ER_NPAR, argv[i]);
                    exit(1);
                }
                ojfp = jopen(argv[++i], "w");
                if (ojfp == JFNULL) {
                    EPRINT(ER_NPUT, argv[i]);
                    exit(1);
                }
            }
        }
    }
}

```

```
        }
        continue;
    }
    EPRINT(ER_OPT, argv[i]);
    exit(1);
}
if ((ijfp = jopen(argv[i], "r")) == JFNULL) {
    EPRINT(ER_NGET, argv[i]);
    exit(1);
}
changer(ijfp, ojfp);
jclose(ijfp);
}
if (ijfp == JFNULL)
    changer(sijfp, ojfp);
jclose(ojfp);
exit(0);
}
```

```

#include <jstdio.h>
#include "translator.h"

int    kutoten = OFF;      /* Kutoten -> ".," */
int    hankaku = OFF;     /* Hankaku-kana -> Zenkaku */
int    z_ascii = OFF;     /* Zenkaku-ASCII -> Hankaku */
int    h_ascii = OFF;     /* Hankaku-ASCII -> Zenkaku */
int    tex_mode = OFF;    /* Tex Mode (ignore after %{}[]$&%) */
char   *replace = CPNULL; /* Replace file name */

CVTAB  h2z(HZSIZE) = {
    {0x0020, 0x0000, 0x0000}, /* <SP> */
    {0x2123, 0x0000, 0x0000}, /* <Ku-ten> */
    {0x2156, 0x0000, 0x0000}, /* <L-Kagi> */
    {0x2157, 0x0000, 0x0000}, /* <R-Kagi> */
    {0x2122, 0x0000, 0x0000}, /* <To-ten> */
    {0x2126, 0x0000, 0x0000}, /* <Chu-ten> */
    {0x2572, 0x0000, 0x0000}, /* WO */
    {0x2521, 0x0000, 0x0000}, /* a */
    {0x2523, 0x0000, 0x0000}, /* i */
    {0x2525, 0x0000, 0x0000}, /* u */
    {0x2527, 0x0000, 0x0000}, /* e */
    {0x2529, 0x0000, 0x0000}, /* o */
    {0x2563, 0x0000, 0x0000}, /* ya */
    {0x2565, 0x0000, 0x0000}, /* yu */
    {0x2567, 0x0000, 0x0000}, /* yo */
    {0x2543, 0x0000, 0x0000}, /* tsu */
    {0x213C, 0x0000, 0x0000}, /* <Cho-on> */
    {0x2522, 0x0000, 0x0000}, /* A */
    {0x2524, 0x0000, 0x0000}, /* I */
    {0x2526, 0x2574, 0x0000}, /* U */
    {0x2528, 0x0000, 0x0000}, /* E */
    {0x252A, 0x0000, 0x0000}, /* O */
    {0x252B, 0x252C, 0x0000}, /* KA */
    {0x252D, 0x252E, 0x0000}, /* KI */
    {0x252F, 0x2530, 0x0000}, /* KU */
    {0x2531, 0x2532, 0x0000}, /* KE */
    {0x2533, 0x2534, 0x0000}, /* KO */
    {0x2535, 0x2536, 0x0000}, /* SA */
    {0x2537, 0x2538, 0x0000}, /* SHI */
    {0x2539, 0x253A, 0x0000}, /* SU */
    {0x253B, 0x253C, 0x0000}, /* SE */
    {0x253D, 0x253E, 0x0000}, /* SO */
    {0x253F, 0x2540, 0x0000}, /* TA */
    {0x2541, 0x2542, 0x0000}, /* CHI */
    {0x2544, 0x2545, 0x0000}, /* TU */
    {0x2546, 0x2547, 0x0000}, /* TE */
    {0x2548, 0x2549, 0x0000}, /* TO */
    {0x254A, 0x0000, 0x0000}, /* NA */
    {0x254B, 0x0000, 0x0000}, /* NI */
    {0x254C, 0x0000, 0x0000}, /* NU */
    {0x254D, 0x0000, 0x0000}, /* NE */
    {0x254E, 0x0000, 0x0000}, /* NO */
    {0x254F, 0x2550, 0x2551}, /* HA */
    {0x2552, 0x2553, 0x2554}, /* HI */
    {0x2555, 0x2556, 0x2557}, /* FU */
    {0x2558, 0x2559, 0x255A}, /* HE */
    {0x255B, 0x255C, 0x255D}, /* HO */
    {0x255E, 0x0000, 0x0000}, /* MA */
    {0x255F, 0x0000, 0x0000}, /* MI */
    {0x2560, 0x0000, 0x0000}, /* MU */
    {0x2561, 0x0000, 0x0000}, /* ME */
    {0x2562, 0x0000, 0x0000}, /* MO */
    {0x2564, 0x0000, 0x0000}, /* YA */
    {0x2566, 0x0000, 0x0000}, /* YU */
    {0x2568, 0x0000, 0x0000}, /* YO */
    {0x2569, 0x0000, 0x0000}, /* RA */
    {0x256A, 0x0000, 0x0000}, /* RI */
    {0x256B, 0x0000, 0x0000}, /* RU */
    {0x256C, 0x0000, 0x0000}, /* RE */
    {0x256D, 0x0000, 0x0000}, /* RO */
    {0x256F, 0x0000, 0x0000}, /* WA */
    {0x2573, 0x0000, 0x0000}, /* N */
    {0x212B, 0x0000, 0x0000}, /* <Daku-ten> */
    {0x212C, 0x0000, 0x0000}, /* <Handaku-ten> */
};

ALPTAB alp(ALPSIZ) = {
    0x2121, ' ',
    0x212A, '!',
    0x2149, '"',
    0x2174, '#',
    0x2170, '$',
    0x2173, '%',
    0x2175, '&',
    0x2147, '¥',
    0x214A, '(',
    0x214B, ')',
    0x2176, '*'
};

```

```
0x215C, '+',
0x2124, ',,',
0x215D, '-,',
0x2125, '.,',
0x213F, '/,',
0x2127, ':,',
0x2128, ';,',
0x2163, '<,',
0x2161, '|,',
0x2164, '>,',
0x2129, '?',
0x2177, '@',
0x214E, '|',
0x2140, '##',
0x214F, '|',
0x2130, '>',
0x2132, '|',
0x2146, '|',
0x2150, '{',
0x2143, '|',
0x2151, '|',
0x2131, '|',
```

};

```
#!/bin/csh
find . -name core -print -exec chmod-move-to-tmp {} \;;
find . \!( -name a.out -o -name '*.o' -o -name '#*' -o -name '*-' -o -name '.*' \;) -atime +7 -print
-exec chmod-move-to-tmp {} \;;
foreach opt (${argv[*]})
  switch ($opt)
  case -TeX:
  case -tex:
    find . \!( -name "*.dvi" -o -name "*.log" -o -name "*.aux" -o -name "*.bbl" -o -name "*.glo" -o -
name "*.idx" -o -name "*.lof" -o -name "*.lot" -o -name "*.toc" \;) -atime +7 -print -exec chmod-move
-to-tmp {} \;;
    breaksw
  default:
    echo "remove-temporary-files: Usage [-tex -TeX]"
    exit 1
  endsw
end
exit 0
```

```

#include <stdio.h>
#include <sys/file.h>

#define FPNUL ((FILE *) 0)
#define ROPEN_E "%s: Can not read %s.\n", comname
#define WOPEN_E "%s: Can not write %s.\n", comname
#define F_OUTED "%d %s\n"
#define F_BABYL "000000-BABYL.rmail"
#define F_MAIL "%s-%s.rmail"
#define F_TEMP "/tmp/RMAIL%09d"
#define M_BABYL "BABYL OPTIONS:%n"
#define M_EOOH "*** EOOH ***\n"
#define M_DATE "Message-Id: <"
#define M_FROM "From: "
#define C_EOM '\037'
#define C_BOM '\014'
#define S_EOM1 "\037\014\n"
#define S_EOM2 "\037"
#define MAXLIN 1024

char *comname;
int verbose;

main(ac, av)
int ac;
char *av[];
{
    int outsw = 0;
    int i;

    comname = av[0];
    for (i = 1; i < ac; i++) {
        if (strcmp(av[i], "-v") == 0) {
            verbose = 1;
        } else {
            FILE *ifp;
            if ((ifp = fopen(av[i], "r")) == FPNUL) {
                fprintf(stderr, ROPEN_E, av[i]);
                exit(1);
            }
            outsw = 1;
            deliver(ifp);
            fclose(ifp);
        }
    }
    if (!outsw)
        deliver(stdin);
    exit(0);
}

char inbuf[MAXLIN];
char sdate[MAXLIN];
char sfrom[MAXLIN];

deliver(ifp)
FILE *ifp;
{
    char wf[20];
    int rcd;
    int nmail = 0;

    sprintf(wf, F_TEMP, getpid());
    while (!getmail(ifp, wf)) {
        if (sdate[0] == '\0' && sfrom[0] == '\0') {
            FILE *ofp, *wfp;
            int c;
            if (access(F_BABYL, F_OK) != 0) {
                wfp = fopen(wf, "r");
                ofp = fopen(F_BABYL, "w");
                if (ofp == FPNUL) {
                    fprintf(WOPEN_E, F_BABYL);
                    exit(1);
                }
                while ((c = getc(wfp)) != EOF)
                    putc(c, ofp);
                putc(C_EOM, ofp);
                fclose(ofp);
                fclose(wfp);
            }
        } else {
            FILE *ofp, *wfp;
            char mf[MAXLIN];
            int c;
            sprintf(mf, F_MAIL, sdate, sfrom);
            wfp = fopen(wf, "r");
            ofp = fopen(mf, "a");
            if (ofp == FPNUL) {
                fprintf(WOPEN_E, F_BABYL);
                exit(1);
            }

```

```

        }
        putc(C_BOM, ofp);
        putc('\n', ofp);
        while ((c = getc(wfp)) != EOF)
            putc(c, ofp);
        putc(C_EOM, ofp);
        fclose(ofp);
        fclose(wfp);
        if (verbose)
            printf(F_OUTED, ++nmail, mf);
    }
}
return;
}
}

int getmail(ifp, wf)
FILE *ifp;
char *wf;
{
    FILE *ofp;

    if ((ofp = fopen(wf, "w")) == FPNUL) {
        fprintf(WOPEN_E, "work file");
        exit(1);
    }
    strcpy(sdate, "");
    strcpy(sfrom, "");
    strcpy(inbuf, "");
    while (fgets(inbuf, MAXLIN, ifp) &&
           strcmp(S_EOM1, inbuf) != 0 &&
           strcmp(S_EOM2, inbuf) != 0) {
        fputs(inbuf, ofp);
        if (sdate[0] == '\0' && hedcmp(M_DATE, inbuf) == 0) {
            char *p, *q;
            int i;
            strcpy(sdate, inbuf);
            for (p = sdate, q = sdate+13, i = 0; *q && i < 6; i++) {
                *(p++) = *(q++);
            }
            *p = '\0';
        }
        if (sfrom[0] == '\0' && hedcmp(M_FROM, inbuf) == 0) {
            char *p, *q;
            strcpy(sfrom, inbuf);
            for (p = sfrom, q = sfrom+6; *q && *q != ' ' && *q != '\n'; )
                *(p++) = *(q++);
            *p = '\0';
        }
    }
    fclose(ofp);
    return (inbuf[0] == '\0');
}

int hedcmp(s1, s2)
char *s1;
char *s2;
{
    while (*s1 && *s2 && *s1 == *s2)
        s1++, s2++;
    return (*s1 == '\0' || *s2 == '\0') ? 0 : ((*s1 > *s2) ? 1 : -1);
}

```

```

#include <stdio.h>

#define ESC '\033'
#define KI1 '$'
#define KI2 'B'
#define KO1 '('
#define KO2 'B'
#define KANAIN 0x0e
#define KANAOUT 0x0f

FILE * input = stdin;
FILE * output = stdout;

main (argc, argv)
int   argc;
char  *argv[];
{
    register int     stat;
    register unsigned c,
                   c1,
                   c2;

    unsigned  s1,
              s2;
    int       i;

    for (i = 1; i < argc; i++) {
        if (*argv[i] == '-') {
            if (strcmp (argv[i], "--stderr") == 0) {
                output = stderr;
                continue;
            }
            if (strcmp (argv[i], "--stdout") == 0) {
                output = stdout;
                continue;
            }
        }
    }
    stat = 0;
    while ((c = getc (input)) != EOF) {
        switch (stat) {
            case 0:
                if ((c >= 0x81 && c <= 0x9f) || (c >= 0xe0 && c <= 0xfc)) {
                    c1 = c;
                    stat = 1;
                    putc (ESC, output); /* put KANJI-IN */
                    putc (KI1, output);
                    putc (KI2, output);
                }
                else
                    if ((0xa1 <= c) && (c <= 0xdf)) /* hankaku kana */
                        putc (KANAIN, output);
                        c &= 0x7f;
                        putc (c, output);
                        stat = 3;
                    }
                else
                    putc (c, output); /* put ASCII code */
                break;
            case 1:
                stat = 2;
                c2 = c;
                IBM_JIS (c1, c2, &s1, &s2);
                /* change shift-JIS into JIS */
                putc (s1, output); /* put KANJI code */
                putc (s2, output);
                break;
            case 2:
                if ((c >= 0x81 && c <= 0x9f) || (c >= 0xe0 && c <= 0xfc)) {
                    c1 = c;
                    stat = 1;
                }
                else {
                    putc (ESC, output); /* put KANJI-OUT */
                    putc (KO1, output);
                    putc (KO2, output);
                    ungetc (c, input);
                    stat = 0;
                }
                break;
            case 3:
                /* hankaku kana */
                if ((0xa1 <= c) && (c <= 0xdf)) {
                    c &= 0x7f;
                    putc (c, output);
                }
                else {
                    putc (KANAOUT, output);
                    ungetc (c, input);
                    stat = 0;
                }
        }
    }
}

```

```
        break;
    }
}
if (stat == 1 || stat == 2) {
/* put KANJI-OUT */
   putc (ESC, output);
   putc (K01, output);
   putc (K02, output);
}
else
    if (stat == 3)
        putc (KANAOUT, output);
}

IBM_JIS (ibm1, ibm2, jis1, jis2)
register unsigned    ibm1,
                   ibm2,
                   *jis1,
                   *jis2;
{
    if (ibm1 >= 0xe0)          /* The 2nd standard of JIS */
        ibm1 -= 0x40;

    if (ibm2 >= 0x9f) {
        *jis1 = (ibm1 - 0x88) * 2 + 0x30;
        *jis2 = ibm2 - 0x7e;
    }
    else {
        if (ibm2 >= 0x7f)
            ibm2 -= 0x01;
        *jis1 = (ibm1 - 0x89) * 2 + 0x31;
        *jis2 = ibm2 - 0x1f;
    }
}
}
```

```
#!/bin/csh
foreach opt (${argv[*]})
  shift-jis-to-jis < ${opt} | jis-translator +h-kana +ascii-z-h > ${opt:r}.jis
  mv ${opt:r}.jis ${opt}
end
```