

Eject コマンドユーザー会を勝手に応援する企画

L10Ns' Commentary on Eject Command with Source Code

本書について

構成

第1部では、Linux Eject コマンドのソースの eject.c を、コメントを翻訳した形で全部載せています。2011 年 8 月 16 日時点の CVS HEAD のソースを使っています。最新リリース版である 2.1.5 に対して、多少の機能追加がおこなわれています。

第2部では、Eject コマンドのソースへの解説を載せていますが、元ネタのようなありがたい解説ではありません。

附属 CD-R には、Eject コマンドのオリジナルのソースコード、eject.c のコメントを翻訳したもの、日本語メッセージカタログを更新したものと、本書の .odt ファイル、.pdf ファイルを収めてあります。

元ネタ

説明するまでもなさげですが、Unix v6 のコードと解説を加えた Lions' Commentary on UNIX 6th Edition, with Source Code が元ネタです。

日本語訳はアスキーから出版されています。

原書の元ネタは以下の web サイトで読むことができます。

- Source code listing for the Lions' Commentary in PDF and PostScript
<http://www.lemis.com/grog/Documentation/Lions/>
- Commentary on the Sixth Edition UNIX Operating System
<http://v6.cuzuco.com/>

第1部 Eject コマンドソースコード

Aug 20 10:00 2011 eject/eject.c Page 1

```
0100 /*****
0101 *
0102 *           L I N U X   E J E C T       C O M M A N D
0103 *
0104 *           by Jeff Tranter (tranter@pobox.com)
0105 *
0106 *****/
0107 *
0108 * Copyright (C) 1994-2001 Jeff Tranter (tranter@pobox.com)
0109 * Copyright (C) 2004, 2005 Frank Lichtenheld (djpig@debian.org)
0110 *
0111 * This program is free software; you can redistribute it and/or modify
0112 * it under the terms of the GNU General Public License as published by
0113 * the Free Software Foundation; either version 2 of the License, or
0114 * (at your option) any later version.
0115 *
0116 * This program is distributed in the hope that it will be useful,
0117 * but WITHOUT ANY WARRANTY; without even the implied warranty of
0118 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
0119 * GNU General Public License for more details.
0120 *
0121 * You should have received a copy of the GNU General Public License
0122 * along with this program; if not, write to the Free Software
0123 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
0124 *
0125 *****/
0126 *
0127 * このプログラムが何をするものかとか、動かすために何が必要かとかの説明は、
0128 * マニュアルページを参照のこと。
0129 *
0130 */
0131
0132 #include "config.h"
0133 #include "i18n.h"
0134
0135 #ifndef DEFAULTDEVICE
0136 #error DEFAULTDEVICE not set, check Makefile
0137 #endif
0138
0139 #include <unistd.h>
0140 #include <stdlib.h>
0141 #include <stdio.h>
0142 #include <string.h>
0143 #include <fcntl.h>
0144 #include <limits.h>
0145
0146 #ifdef GETOPTLONG
0147 #include <getopt.h>
0148 #endif /* GETOPTLONG */
0149 #include <errno.h>
```

```

0150 #include <regex.h>
0151 #include <sys/types.h>
0152 #include <sys/stat.h>
0153 #include <sys/ioctl.h>
0154 #include <sys/wait.h>
0155 #include <sys/mtio.h>
0156 #include <sys/mount.h>
0157
0158 #if defined(__linux__)
0159 #include <linux/version.h>
0160 /* 2.1 カーネル (訳註: 2.1.90 以降) にはこの便利なマクロが定義されているが、古いカーネルにはない
    */
0161 #ifndef KERNEL_VERSION
0162 #define KERNEL_VERSION(a,b,c) (((a) << 16) + ((b) << 8) + (c))
0163 #endif
0164 #include <linux/types.h>
0165 #include <linux/cdrom.h>
0166 #if LINUX_VERSION_CODE < KERNEL_VERSION(2,1,0)
0167 #include <linux/ucdrom.h>
0168 #endif
0169 #include <linux/fd.h>
0170 #include <scsi/scsi.h>
0171 #include <scsi/sg.h>
0172 #include <scsi/scsi_ioctl.h>
0173 #include <sys/time.h>
0174
0175 /* ToggleTray() 関数で使用する値。トレイのイジェクトに要した時間がこの値以下だった場合、
0176  * おそらくトレイは最初からイジェクト済みなので、
0177  * その場合はトレイを閉じることになる。
0178  */
0179 #define TRAY_WAS_ALREADY_OPEN_USECS 200000 /* 約 0.2 秒 */
0180
0181
0182 #define CLOSE(fd) if (close(fd)==-1) { \
0183     perror(programName); \
0184     exit(1); \
0185 }
0186
0187 #define FCLOSE(fd) if (fclose(fd)==-1) { \
0188     perror(programName); \
0189     exit(1); \
0190 }
0191
0192 #define HAVE_EJECT_SCSI
0193 #define HAVE_EJECT_FLOPPY
0194 #define HAVE_EJECT_TAPE
0195
0196 #elif defined(__FreeBSD_kernel__)
0197 #include <sys/cdio.h>
0198 #endif /* if defined(__linux__) */
0199

```

```

0200
0201 #define CLOSE(fd) if (close(fd)==-1) { \
0202     perror(programName); \
0203     exit(1); \
0204 }
0205
0206 #define FCLOSE(fd) if (fclose(fd)==-1) { \
0207     perror(programName); \
0208     exit(1); \
0209 }
0210
0211 /* グローバル変数 */
0212 static const char *version = VERSION; /* プログラムのバージョン */
0213 int a_option = 0; /* コマンドラインオプションと引数 */
0214 int c_option = 0;
0215 int d_option = 0;
0216 int f_option = 0;
0217 int h_option = 0;
0218 int n_option = 0;
0219 int i_option = 0;
0220 int q_option = 0;
0221 int r_option = 0;
0222 int s_option = 0;
0223 int t_option = 0;
0224 int T_option = 0;
0225 int X_option = 0;
0226 int v_option = 0;
0227 int x_option = 0;
0228 int p_option = 0;
0229 int m_option = 0;
0230 int a_arg = 0;
0231 int c_arg = 0;
0232 int x_arg = 0;
0233 int i_arg = 0;
0234 static char *programName; /* エラーメッセージで使う */
0235
0236 /*
0237  * 複数のパーティションを持ち得るデバイスの、
0238  * デバイスノードの基本名。
0239  */
0240 static const char *partitionDevice[] = {
0241     "hd",
0242     "sd",
0243     "xd",
0244     "dos_hd",
0245     "mfm",
0246     "ad",
0247     "ed",
0248     "ftl",
0249     "pd",

```

```

0250 0};
0251
0252
0253 /* コマンドの使い方を標準エラー出力に表示し、終了する。 */
0254 static void usage()
0255 {
0256     fprintf(stderr,_(
0257 "Eject version %s by Jeff Tranter (tranter@pobox.com)\n"
0258 "Usage:\n"
0259 "    eject -h                -- display command usage and exit\n"
0260 "    eject -V                -- display program version and exit\n"
0261 "    eject [-vnrsfqpm] [<name>]    -- eject device\n"
0262 "    eject [-vn] -d          -- display default device\n"
0263 "    eject [-vn] -a on|off|1|0 [<name>] -- turn auto-eject feature on or off\n"
0264 "    eject [-vn] -c <slot> [<name>]    -- switch discs on a CD-ROM changer\n"
0265 "    eject [-vn] -t [<name>]          -- close tray\n"
0266 "    eject [-vn] -T [<name>]          -- toggle tray\n"
0267 "    eject [-vn] -i on|off|1|0 [<name>] -- toggle manual eject protection on/off\n"
0268 "    eject [-vn] -x <speed> [<name>]   -- set CD-ROM max speed\n"
0269 "    eject [-vn] -X [<name>]          -- list CD-ROM available speeds\n"
0270 "Options:\n"
0271 "    -v\t-- enable verbose output\n"
0272 "    -n\t-- don't eject, just show device found\n"
0273 "    -r\t-- eject CD-ROM\n"
0274 #ifdef HAVE_EJECT_SCSI
0275 "    -s\t-- eject SCSI device\n"
0276 #endif
0277 #ifdef HAVE_EJECT_FLOPPY
0278 "    -f\t-- eject floppy\n"
0279 #endif
0280 #ifdef HAVE_EJECT_TAPE
0281 "    -q\t-- eject tape\n"
0282 #endif
0283 "    -p\t-- use /proc/mounts instead of /etc/mtab\n"
0284 "    -m\t-- do not unmount device even if it is mounted\n"
0285 )
0286 , version);
0287 #ifdef GETOPTLONG
0288     fprintf(stderr,_(
0289 "Long options:\n"
0290 "    -h --help      -v --verbose      -d --default\n"
0291 "    -a --auto      -c --changerslot  -t --trayclose  -x --cdspeed\n"
0292 "    -r --cdrom\n"
0293 #ifdef HAVE_EJECT_SCSI
0294 "    -s --scsi\n"
0295 #endif
0296 #ifdef HAVE_EJECT_FLOPPY
0297 "    -f --floppy\n"
0298 #endif
0299 "    -X --listspeed"

```

```

0300 #ifdef HAVE_EJECT_TAPE
0301 "      -q --tape"
0302 #endif
0303 "\n"
0304 " -n --noop    -V --version\n"
0305 " -p --proc    -m --no-unmount    -T --traytoggle\n"));
0306 #endif /* GETOPTLONG */
0307 fprintf(stderr,_(
0308 "Parameter <name> can be a device file or a mount point.\n"
0309 "If omitted, name defaults to `s'.\n"
0310 "By default tries -r, -s, -f, and -q in order until success.\n"),
0311         DEFAULTDEVICE);
0312     exit(1);
0313 }
0314
0315
0316 /* コマンドラインオプションの処理。 */
0317 static void parse_args(int argc, char **argv, char **device)
0318 {
0319     const char *flags = "a:c:x:i:dfhnqrstTXvVpm";
0320 #ifdef GETOPTLONG
0321     static struct option long_options[] =
0322     {
0323         {"help",      no_argument,      NULL, 'h'},
0324         {"verbose",   no_argument,      NULL, 'v'},
0325         {"default",   no_argument,      NULL, 'd'},
0326         {"auto",      required_argument, NULL, 'a'},
0327         {"changerslot", required_argument, NULL, 'c'},
0328         {"manualeject", required_argument, NULL, 'i'},
0329         {"trayclose", no_argument,      NULL, 't'},
0330         {"traytoggle", no_argument,      NULL, 'T'},
0331         {"cdspeed",   required_argument, NULL, 'x'},
0332         {"listspeed", no_argument,      NULL, 'X'},
0333         {"noop",      no_argument,      NULL, 'n'},
0334         {"cdrom",     no_argument,      NULL, 'r'},
0335         {"scsi",      no_argument,      NULL, 's'},
0336         {"floppy",    no_argument,      NULL, 'f'},
0337         {"tape",      no_argument,      NULL, 'q'},
0338         {"version",   no_argument,      NULL, 'V'},
0339         {"proc",      no_argument,      NULL, 'p'},
0340         {"no-unmount", no_argument,      NULL, 'm'},
0341         {0, 0, 0, 0}
0342     };
0343     int option_index;
0344 #endif /* GETOPTLONG */
0345     int c;
0346
0347 #ifdef GETOPTLONG
0348     while ((c = getopt_long(argc, argv, flags, long_options, &option_index)) != EOF) {
0349 #else

```

```

0350 while ((c = getopt(argc, argv, flags)) != EOF) {
0351 #endif /* GETOPTLONG */
0352     switch (c) {
0353         case 'a':
0354             a_option = 1;
0355             if (!strcmp(optarg, "0"))
0356                 a_arg = 0;
0357             else if (!strcmp(optarg, "off"))
0358                 a_arg = 0;
0359             else if (!strcmp(optarg, "1"))
0360                 a_arg = 1;
0361             else if (!strcmp(optarg, "on"))
0362                 a_arg = 1;
0363             else {
0364                 fprintf(stderr, _("%s: invalid argument to --auto/-a
option\n"), programName);
0365                 exit(1);
0366             }
0367             break;
0368         case 'c':
0369             c_option = 1;
0370             /* atoi() はエラーで 0 を返すので、元の値が "0" の場合用の処理が別に必要 */
0371             if (!strcmp(optarg, "0"))
0372                 c_arg = 0;
0373             else {
0374                 c_arg = atoi(optarg);
0375                 if (c_arg <= 0) {
0376                     fprintf(stderr, _("%s: invalid argument to
--changerslot/-c option\n"), programName);
0377                     exit(1);
0378                 }
0379             }
0380             break;
0381         case 'x':
0382             x_option = 1;
0383             if (!strcmp(optarg, "0"))
0384                 x_arg = 0;
0385             else {
0386                 x_arg = atoi(optarg);
0387                 if (x_arg <= 0) {
0388                     fprintf(stderr, _("%s: invalid argument to --cdspeed/-x
option\n"), programName);
0389                     exit(1);
0390                 }
0391             }
0392             break;
0393         case 'd':
0394             d_option = 1;
0395             break;
0396         case 'f':
0397             f_option = 1;
0398             break;
0399         case 'h':

```



```

0400         usage();
0401         exit(0);
0402         break;
0403     case 'i':
0404         i_option = 1;
0405         if (!strcmp(optarg, "0"))
0406             i_arg = 0;
0407         else if (!strcmp(optarg, "off"))
0408             i_arg = 0;
0409         else if (!strcmp(optarg, "1"))
0410             i_arg = 1;
0411         else if (!strcmp(optarg, "on"))
0412             i_arg = 1;
0413         else {
0414             fprintf(stderr, _("%s: invalid argument to -i option\n"),
programName);
0415             exit(1);
0416         }
0417         break;
0418     case 'm':
0419         m_option = 1;
0420         break;
0421     case 'n':
0422         n_option = 1;
0423         break;
0424     case 'p':
0425         p_option = 1;
0426         break;
0427     case 'q':
0428         q_option = 1;
0429         break;
0430     case 'r':
0431         r_option = 1;
0432         break;
0433     case 's':
0434         s_option = 1;
0435         break;
0436     case 't':
0437         t_option = 1;
0438         break;
0439     case 'X':
0440         X_option = 1;
0441         break;
0442     case 'T':
0443         T_option = 1;
0444         break;
0445     case 'v':
0446         v_option = 1;
0447         break;
0448     case 'V':
0449         printf(_("eject version %s by Jeff Tranter (tranter@pobox.com)\n"),
version);

```

```

0450             exit(0);
0451             break;
0452         case '?':
0453             exit(1);
0454             break;
0455     }
0456 }
0457 /* 追加引数が単一であることの確認 */
0458 if ((argc - optind) > 1) {
0459     fprintf(stderr, _("%s: too many arguments\n"), programName);
0460     exit(1);
0461 }
0462 if ((argc - optind) == 1) { /* 引数は一つ */
0463     *device = strdup(argv[optind]);
0464 }
0465 }
0466
0467
0468 /* ファイルやデバイスが存在すれば 1 を返し、そうでなければ 0 を返す。 */
0469 static int FileExists(const char *name, const int try, int *found)
0470 {
0471
0472     if (!found) return -1;
0473     /*
0474      * access() は EUID ではなく UID を使う。この方法では、
0475      * 一般ユーザーは、たとえ eject が SUID root されていたとしても、
0476      * ファイル (たとえば /root/fubar) の有無を知ることができない。
0477      */
0478     if (access (name, F_OK) == 0) {
0479         (*found)++;
0480         if (try <= (*found))
0481             return 1;
0482         else
0483             return 0;
0484     } else {
0485         return 0;
0486     }
0487 }
0488
0489
0490 /*
0491  * Linux は、マウントポイントに含まれる空白を八進表現の文字列 '\040' に変換してしまう。
0492  * そこで、マウントポイントに含まれる '\0##' をすべて探し出して、ASCII 値 0##
0493  * の文字に置換する。このような修正をした文字列を入力することが必要となるからである。
0494  * この処理の一部は、util-linux パッケージ (訳註: util-linux 2.9 以降の
0495  * mount/mntent.c; 2.18 以降は lib/mangle.c) からパクった。
0496  */
0497 #define octalify(a) ((a) & 7)
0498 #define tooctal(s) (64*octalify(s[1]) + 8*octalify(s[2]) + octalify(s[3]))
0499 #define isoctal(a) (((a) & ~7) == '0')

```

```

0500 static char *DeMangleMount(char *s)
0501 {
0502 char *tmp = s;
0503 while ((tmp = strchr(tmp, '\\')) != NULL) {
0504     if (isoctal(tmp[1]) && isoctal(tmp[2]) && isoctal(tmp[3])) {
0505         tmp[0] = tooctal(tmp);
0506         memmove(tmp+1, tmp+4, strlen(tmp)-3);
0507     }
0508     ++tmp;
0509 }
0510 return s;
0511 }
0512
0513
0514 /*
0515  * 与えられた名前 (foo) に対して、以下のいずれかが存在するかを調べる:
0516  *
0517  * foo (foo が '.' または '/' で始まる場合)
0518  * /dev/foo
0519  * /media/foo
0520  * /mnt/foo
0521  * /dev/cdroms/foo
0522  * /dev/cdroms/foo0
0523  * /dev/dsk/foo (訳註: /dev/rdisk/foo との順序が逆)
0524  * /dev/rdisk/foo
0525  * ./foo
0526  *
0527  * 見つかった場合は、そのフルパスを返す。いずれも見つからない場合は 0 を返す。
0528  * 返り値は、動的に割り当てられる文字列へのポインターである。
0529  */
0530 static char *FindDevice(const char *name)
0531 {
0532 char *buf;
0533 static int try = 0;
0534 int found = 0;
0535
0536 buf = (char *) malloc(strlen(name)+14); /* "/dev/cdroms/" + "0" + null の文字数分 */
0537 if (buf==NULL) {
0538     fprintf(stderr, _("%s: could not allocate memory\n"), programName);
0539     exit(1);
0540 }
0541
0542 if (try == INT_MAX) {
0543     fprintf(stderr, _("%s: FindDevice called too often\n"), programName );
0544     exit(1);
0545 } else
0546     try++;
0547
0548 if ((name[0] == '.') || (name[0] == '/')) {
0549     strcpy(buf, name);

```

```
0550         if (FileExists(buf, try, &found))
0551             return buf;
0552     }
0553
0554     strcpy(buf, "/dev/");
0555     strcat(buf, name);
0556     if (FileExists(buf, try, &found))
0557         return buf;
0558
0559     strcpy(buf, "/media/");
0560     strcat(buf, name);
0561     if (FileExists(buf, try, &found))
0562         return buf;
0563
0564     strcpy(buf, "/mnt/");
0565     strcat(buf, name);
0566     if (FileExists(buf, try, &found))
0567         return buf;
0568
0569     /* Linux の devfs 用 */
0570     strcpy(buf, "/dev/cdroms/");
0571     strcat(buf, name);
0572     if (FileExists(buf, try, &found))
0573         return buf;
0574
0575     strcpy(buf, "/dev/cdroms/");
0576     strcat(buf, name);
0577     strcat(buf, "0");
0578     if (FileExists(buf, try, &found))
0579         return buf;
0580
0581     /* Solaris の devfs 用 */
0582     strcpy(buf, "/dev/rdisk/");
0583     strcat(buf, name);
0584     if (FileExists(buf, try, &found))
0585         return buf;
0586
0587     strcpy(buf, "/dev/dsk/");
0588     strcat(buf, name);
0589     if (FileExists(buf, try, &found))
0590         return buf;
0591
0592     strcpy(buf, "./");
0593     strcat(buf, name);
0594     if (FileExists(buf, try, &found))
0595         return buf;
0596
0597     free(buf);
0598     buf = 0;
0599     return 0;
```

```

0600 }
0601
0602
0603 /*
0604  * CD-ROM のボタンを押して手でイジェクトしようとしても、開かないようにする。
0605  * この機能は、ラップトップマシンの電源を入れたまま、
0606  * ドライブに CD を入れない状態で、鞆に入れて持ち歩く時に便利である。
0607  * Linux カーネル附属文書 Documentation/ioctl/cdrom.txt の記載のとおり実装した。
0608  *
0609  * TODO: 以下のエラーも確認したほうがよいかもしれない:
0610  * EDRIVE_CANT_DO_THIS 扉の固定機能に対応していない。
0611  * EBUSY 複数ユーザーがドライブをオープンしており、かつケーパビリティ
0612  * CAP_SYS_ADMIN がない状態で、ロックを解除しようとした。
0613 */
0614 static void ManualEject(int fd, int onOff)
0615 {
0616     if (ioctl(fd, CDROM_LOCKDOOR, onOff) < 0) {
0617         perror("ioctl on CDROM_LOCKDOOR");
0618     } else {
0619         if (onOff)
0620             printf("CD-Drive may NOT be ejected with device button\n");
0621         else
0622             printf("CD-Drive may be ejected with device button\n");
0623     }
0624 }
0625
0626
0627 /* 自動イジェクトモードの設定および解除。 */
0628 static void AutoEject(int fd, int onOff)
0629 {
0630     int status = -1;
0631
0632     #if defined(CDROM_SET_OPTIONS) && defined(CDROM_CLEAR_OPTIONS)
0633     if (onOff)
0634         status = ioctl(fd, CDROM_SET_OPTIONS, CDO_AUTO_EJECT);
0635     else
0636         status = ioctl(fd, CDROM_CLEAR_OPTIONS, CDO_AUTO_EJECT);
0637     #else
0638     errno = ENOSYS;
0639     #endif
0640     if (status < 0) {
0641         fprintf(stderr, _("%s: CD-ROM auto-eject command failed: %s\n"), programName,
0642             strerror(errno));
0643         exit(1);
0644     }
0645
0646
0647 /*
0648  * チェンジャーの選択。CDROM_SELECT_DISC が望ましいが、古いカーネルでは
0649  * CDROMLOADFROMSLOT を使っていた。

```

```

0650 */
0651 static void ChangerSelect(int fd, int slot)
0652 {
0653     int status;
0654
0655     #ifdef CDROM_SELECT_DISC
0656     status = ioctl(fd, CDROM_SELECT_DISC, slot);
0657     if (status < 0) {
0658         fprintf(stderr, _("%s: CD-ROM select disc command failed: %s\n"),
0659             programName, strerror(errno));
0659         exit(1);
0660     }
0661     #elif defined CDROMLOADFROMSLOT
0662     status = ioctl(fd, CDROMLOADFROMSLOT, slot);
0663     if (status != 0) {
0664         fprintf(stderr, _("%s: CD-ROM load from slot command failed: %s\n"),
0665             programName, strerror(errno));
0665         exit(1);
0666     }
0667     #else
0668     fprintf(stderr, _("%s: IDE/ATAPI CD-ROM changer not supported by this
0669         kernel\n"), programName);
0669     #endif
0670 }
0671
0672
0673 /*
0674  * トレイを閉じる。古いカーネルではこれに対応していない。
0675  */
0676 static void CloseTray(int fd)
0677 {
0678     int status;
0679
0680     #if defined(CDROMCLOSETRAY) || defined(CDIOCCLOSE)
0681     #if defined(CDROMCLOSETRAY)
0682     status = ioctl(fd, CDROMCLOSETRAY);
0683     #elif defined(CDIOCCLOSE)
0684     status = ioctl(fd, CDIOCCLOSE);
0685     #endif
0686     if (status != 0) {
0687         fprintf(stderr, _("%s: CD-ROM tray close command failed: %s\n"), programName,
0688             strerror(errno));
0688         exit(1);
0689     }
0690     #else
0691     fprintf(stderr, _("%s: CD-ROM tray close command not supported by this
0692         kernel\n"), programName);
0692     #endif
0693 }
0694
0695 /*
0696  * トレイの開閉を切替える。
0697  *
0698  * このコードは Benjamin Schwenk <benjaminschwenk@yahoo.de> および
0699  * Sybren Stuveld <sybren@thirdtower.com> によるものである。

```

```

0700 *
0701 *  トレイを閉じる際には CloseTray() を使うことになるので、
0702 *  古いカーネルはこれに対応していない。
0703 *
0704 */
0705 static void ToggleTray(int fd)
0706 {
0707     struct timeval time_start, time_stop;
0708     int time_elapsed;
0709
0710 #ifdef CDROMCLOSETRAY
0711
0712     /* CDROM トレイを開けようとして、開けるのに必要な時間を計測する。
0713      *  経験上、すでにトレイが開いている場合は、この操作に要する時間は 0.05 秒未満となる。
0714      *  また、閉まっているトレイを開く場合は、この操作に要する時間は
0715      *  1.5 秒以上となる。
0716      */
0717     gettimeofday(&time_start, NULL);
0718
0719     /* デバイスに CDROMEJECT コマンドを送る。 */
0720     if (ioctl(fd, CDROMEJECT, 0) < 0) {
0721         perror("ioctl");
0722         exit(1);
0723     }
0724
0725     /* トレイを開くのに要した時間を計測するため、
0726      *  時刻を再度取得する。 */
0727     gettimeofday(&time_stop, NULL);
0728
0729     time_elapsed = (time_stop.tv_sec * 1000000 + time_stop.tv_usec) -
0730         (time_start.tv_sec * 1000000 + time_start.tv_usec);
0731
0732     /* トレイが「開いた」のが速過ぎる場合は、トレイが元から開いていたことがほぼ確実である。
0733      *  この場合は、ここでトレイを閉じる。それ以外の場合はトレイが元は閉じていたことになるが、
0734      *  （その場合に必要となるトレイを開く操作は）すでに完了している。 */
0735     if (time_elapsed < TRAY_WAS_ALREADY_OPEN_USECS)
0736         CloseTray(fd);
0737
0738 #else
0739     fprintf(stderr, _("%s: CD-ROM tray toggle command not supported by this
0740 kernel\n"), programName);
0741 #endif
0742 }
0743
0744 /*
0745  * CD-ROM ドライブの速度を選択する。
0746  * このコードは Roland Krivanek (krivanek@fmph.uniba.sk) によるものである
0747  * http://dmph.dbp.fmph.uniba.sk/~krivanek/cdrom\_speed/
0748  */
0749 static void SelectSpeedCdrom(int fd, int speed)

```

```

0750 {
0751     unsigned long rw_size;
0752     unsigned char buffer[28];
0753     struct cdrom_generic_command cgc;
0754     struct request_sense sense;
0755
0756 #ifdef CDROM_SELECT_SPEED
0757     memset(&cgc, 0, sizeof(cgc));
0758     memset(&sense, 0, sizeof(sense));
0759     memset(&buffer, 0, sizeof(buffer));
0760
0761     /* SET STREAMING コマンド */
0762     cgc.cmd[0] = 0xB6;
0763     /* パラメーターリスト長: 28 バイト */
0764     cgc.cmd[10] = 28;
0765
0766     cgc.sense = &sense;
0767     cgc.buffer = buffer;
0768     cgc.buflen = sizeof(buffer);
0769     cgc.data_direction = CGC_DATA_WRITE;
0770     cgc.quiet = 1;
0771
0772     if(speed == 0) {
0773         /* Restore Drive Defaults ビットを立てる */
0774         buffer[0] = 4;
0775     }
0776
0777     buffer[8] = 0xFF;
0778     buffer[9] = 0xFF;
0779     buffer[10] = 0xFF;
0780     buffer[11] = 0xFF;
0781
0782     rw_size = 177 * speed;
0783
0784     /* 読み取りサイズ */
0785     buffer[12] = (rw_size >> 24) & 0xFF;
0786     buffer[13] = (rw_size >> 16) & 0xFF;
0787     buffer[14] = (rw_size >> 8) & 0xFF;
0788     buffer[15] = rw_size & 0xFF;
0789
0790     /* 読み取り時間: 1 秒 */
0791     buffer[18] = 0x03;
0792     buffer[19] = 0xE8;
0793
0794     /* 書き込みサイズ */
0795     buffer[20] = (rw_size >> 24) & 0xFF;
0796     buffer[21] = (rw_size >> 16) & 0xFF;
0797     buffer[22] = (rw_size >> 8) & 0xFF;
0798     buffer[23] = rw_size & 0xFF;
0799

```



```

0800      /* 書き込み時間: 1 秒 */
0801      buffer[26] = 0x03;
0802      buffer[27] = 0xE8;
0803
0804      if(ioctl(fd, CDROM_SEND_PACKET, &cgc) != 0) {
0805          if(ioctl(fd, CDROM_SELECT_SPEED, speed) != 0) {
0806              fprintf(stderr, _("%s: CD-ROM select speed command failed: %s\n"),
programName, strerror(errno));
0807              exit(1);
0808          }
0809      }
0810      #else
0811          fprintf(stderr, _("%s: CD-ROM select speed command not supported by this
kernel\n"), programName);
0812      #endif
0813  }
0814
0815  /*
0816   * CD-ROM ドライブの速度を読み取る。Linux 2.6.13 以降で、現在の速度が正しく報告される
0817   */
0818  static int ReadSpeedCdrom(const char *shortName)
0819  {
0820      char line[512];
0821      char *str_speed, *str_name;
0822      int drive_number = -1, i;
0823      FILE *f = fopen("/proc/sys/dev/cdrom/info", "r");
0824
0825      if (f == NULL) {
0826          fprintf(stderr, _("%s: unable to read the speed from
/proc/sys/dev/cdrom/info\n"), programName);
0827          exit(1);
0828      }
0829
0830      while (!feof(f)) {
0831          fgets(line, sizeof(line), f);
0832
0833          /* "drive name" 行の shortName の位置からドライブ番号を求める */
0834          if (drive_number == -1) {
0835              if (strncmp(line, "drive name:", 11) == 0) {
0836                  str_name = strtok(&line[11], "\t ");
0837                  drive_number = 0;
0838                  while (strncmp(shortName, str_name, strlen(shortName)) != 0) {
0839                      drive_number++;
0840                      str_name = strtok(NULL, "\t ");
0841                      if (str_name == NULL) {
0842                          fprintf(stderr, _("%s: error while finding CD-ROM
name\n"), programName);
0843                          exit(1);
0844                      }
0845                  }
0846              }
0847          /* "drive speed" 行を探して、適切な速度を読みとる */
0848          } else {
0849              if (strncmp(line, "drive speed:", 12) == 0) {

```

```

0850         str_speed = strtok(&line[12], "\\t ");
0851         for (i = 1; i < drive_number; i++)
0852             str_speed = strtok(NULL, "\\t ");
0853
0854         if (str_speed == NULL) {
0855             fprintf(stderr, _("%s: error while reading speed\\n"),
programName);
0856                 exit(1);
0857             }
0858             return atoi(str_speed);
0859         }
0860     }
0861 }
0862
0863 fprintf(stderr, _("%s: error while reading speed\\n"), programName);
0864 exit(1);
0865 return -1;
0866 }
0867
0868
0869 /*
0870  * CD-ROM ドライブの速度を列挙する。
0871  */
0872 static void ListSpeedCdrom(const char *fullName, int fd)
0873 {
0874     #ifdef CDROM_SELECT_SPEED
0875     int max_speed, curr_speed = 0, prev_speed;
0876     char *shortName = rindex(fullName, '/') + 1;
0877
0878     SelectSpeedCdrom(fd, 0);
0879     max_speed = ReadSpeedCdrom(shortName);
0880     while (curr_speed < max_speed) {
0881         prev_speed = curr_speed;
0882         SelectSpeedCdrom(fd, prev_speed + 1);
0883         curr_speed = ReadSpeedCdrom(shortName);
0884         if (curr_speed > prev_speed)
0885             printf("%d ", curr_speed);
0886         else
0887             curr_speed = prev_speed + 1;
0888     }
0889
0890     printf("\\n");
0891     #else
0892     fprintf(stderr, _("%s: CD-ROM select speed command not supported by this
kernel\\n"), programName);
0893     #endif
0894 }
0895
0896 /*
0897  * CDROMEJECT ioctl を使ってイジェクトする。成功すれば 1 を返し、そうでなければ 0 を返す。
0898  */
0899 static int EjectCdrom(int fd)

```

```

0900 {
0901 int status = -1;
0902
0903 #if defined(CDROMEJECT)
0904 status = ioctl(fd, CDROMEJECT);
0905 #elif defined(CDIOCEJECT)
0906 status = ioctl(fd, CDIOCEJECT);
0907 #else
0908 /* cdrom のみのイジェクトを実装したカーネルはあるが、これとは逆に、cdrom 以外のドライブのみのイ
    ジェクトを
0909 実装したカーネルなんてものがあるとは思えない。というわけで、(cdrom に非対応ならばほかにも非対応だろ
    うから) これで終了。 */
0910 # error
0911 #endif
0912 return (status == 0);
0913 }
0914
0915 #ifdef HAVE_EJECT SCSI
0916 /*
0917  * SCSI SG_IO コマンドを使ってイジェクトする。成功すれば 1 を返し、そうでなければ 0 を返す。
0918  */
0919 static int EjectScsi(int fd)
0920 {
0921 int status, k;
0922 sg_io_hdr_t io_hdr;
0923 unsigned char allowRmBlk[6] = {ALLOW_MEDIUM_REMOVAL, 0, 0, 0, 0, 0};
0924 unsigned char startStop1Blk[6] = {START_STOP, 0, 0, 0, 1, 0};
0925 unsigned char startStop2Blk[6] = {START_STOP, 0, 0, 0, 2, 0};
0926 unsigned char inqBuff[2];
0927 unsigned char sense_buffer[32];
0928
0929 if ((ioctl(fd, SG_GET_VERSION_NUM, &k) < 0) || (k < 30000)) {
0930     printf("not an sg device, or old sg driver\n");
0931     return 0;
0932 }
0933
0934 memset(&io_hdr, 0, sizeof(sg_io_hdr_t));
0935 io_hdr.interface_id = 'S';
0936 io_hdr.cmd_len = 6;
0937 io_hdr.mx_sb_len = sizeof(sense_buffer);
0938 io_hdr.dxfer_direction = SG_DXFER_NONE;
0939 io_hdr.dxfer_len = 0;
0940 io_hdr.dxferp = inqBuff;
0941 io_hdr.sbp = sense_buffer;
0942 io_hdr.timeout = 10000;
0943
0944 io_hdr.cmdp = allowRmBlk;
0945 status = ioctl(fd, SG_IO, (void *)&io_hdr);
0946 if (status < 0)
0947     return 0;
0948
0949 io_hdr.cmdp = startStop1Blk;

```

```

0950 status = ioctl(fd, SG_IO, (void *)&io_hdr);
0951 if (status < 0)
0952     return 0;
0953
0954 io_hdr.cmdp = startStop2Blk;
0955 status = ioctl(fd, SG_IO, (void *)&io_hdr);
0956 if (status < 0)
0957     return 0;
0958
0959 /* 新たにディスクが挿入された際に、カーネルにパーティションテーブルを再読み込みさせる */
0960 status = ioctl(fd, BLKRRPART);
0961 return 1;
0962 }
0963 #endif
0964
0965
0966 #ifdef HAVE_EJECT_FLOPPY
0967 /*
0968  * FDEJECT ioctl を使ってイジェクトする。成功すれば 1 を返し、そうでなければ 0 を返す。
0969  */
0970 static int EjectFloppy(int fd)
0971 {
0972     int status;
0973
0974     status = ioctl(fd, FDEJECT);
0975     return (status >= 0);
0976 }
0977 #endif
0978
0979
0980 #ifdef HAVE_EJECT_TAPE
0981 /*
0982  * tape ioctl を使ってイジェクトする。成功すれば 1 を返し、そうでなければ 0 を返す。
0983  */
0984 static int EjectTape(int fd)
0985 {
0986     int status;
0987     struct mtop op;
0988
0989     op.mt_op = MTOFFL; /* リワインドしてイジェクトする */
0990     op.mt_count = 0; /* 使用されない */
0991     status = ioctl(fd, MTIOCTOP, &op);
0992     return (status >= 0);
0993 }
0994 #endif
0995
0996
0997 /* デバイスをアンマウントする。 */
0998 static void Unmount(const char *fullName)
0999 {

```

```

1000 int status;
1001
1002 switch (fork()) {
1003     case 0: /* 子プロセス */
1004         setuid(getuid()); /* setuid されて実行されている場合のセキュリティーホールの可能性を
減らす */
1005         if(p_option) {
1006             execlp("pumount", "pumount", fullName, "-n", NULL);
1007             execlp("umount", "umount", fullName, "-n", NULL);
1008         } else {
1009             execlp("pumount", "pumount", fullName, NULL);
1010             execlp("umount", "umount", fullName, NULL);
1011         }
1012         fprintf(stderr, _("%s: unable to exec umount of `%s': %s\n"),
1013             programName, fullName, strerror(errno));
1014         exit(1);
1015         break;
1016     case -1:
1017         fprintf(stderr, _("%s: unable to fork: %s\n"), programName,
strerror(errno));
1018         break;
1019     default: /* 親プロセス */
1020         wait(&status);
1021         if (WIFEXITED(status) == 0) {
1022             fprintf(stderr, _("%s: unmount of `%s' did not exit normally\n"),
programName, fullName);
1023             exit(1);
1024         }
1025         if (WEXITSTATUS(status) != 0) {
1026             fprintf(stderr, _("%s: unmount of `%s' failed\n"), programName,
fullName);
1027             exit(1);
1028         }
1029         break;
1030 }
1031 }
1032
1033
1034 /* デバイスファイルをオープンする。まず読み書き両方で、それが失敗した場合は読み込み専用でのオープン
を試みる。 */
1035 static int OpenDevice(const char *fullName)
1036 {
1037     int fd;
1038
1039     fd = open(fullName, O_RDWR|O_NONBLOCK);
1040     if (fd != -1) {
1041         return fd;
1042     }
1043
1044     fd = open(fullName, O_RDONLY|O_NONBLOCK);
1045     if (fd == -1) {
1046         fprintf(stderr, _("%s: unable to open `%s'\n"), programName, fullName);
1047         exit(1);
1048     }
1049     return fd;

```

```

1050 }
1051
1052
1053 /*
1054  * 与えられたデバイスファイル名の、デバイスメジャー番号とマイナー番号を取得する。
1055  * これにより、デバイスノードの重複をチェックすることができる。
1056  */
1057 static int GetMajorMinor(const char *name, int *maj, int *min)
1058 {
1059     struct stat sstat;
1060     if (maj) *maj = -1;
1061     if (min) *min = -1;
1062     if (stat(name, &sstat) == -1)
1063         return -1;
1064     if (! S_ISBLK(sstat.st_mode) && ! S_ISCHR(sstat.st_mode))
1065         return -1;
1066     if (maj) *maj = major(sstat.st_rdev);
1067     if (min) *min = minor(sstat.st_rdev);
1068     return 0;
1069 }
1070
1071
1072 /*
1073  * デバイスがマウントされているかどうかを、マウントテーブルを見て調べる。マウントされている場合は、
1074  * デバイス名とマウントポイント名を設定して 1 を返し、そうでない場合は 0 を返す。
1075  */
1076 static int MountedDevice(const char *name, char **mountName, char **deviceName)
1077 {
1078     FILE *fp;
1079     char line[1024];
1080     char s1[1024];
1081     char s2[1024];
1082     int rc;
1083
1084     int maj;
1085     int min;
1086
1087     GetMajorMinor(name, &maj, &min);
1088
1089     fp = fopen((p_option ? "/proc/mounts" : "/etc/mtab"), "r");
1090     if (fp == NULL)
1091     {
1092         fprintf(stderr, _("unable to open %s: %s\n"), (p_option ? "/proc/mounts" :
1093             "/etc/mtab"), strerror(errno));
1094         exit(1);
1095     }
1096     while (fgets(line, sizeof(line), fp) != 0) {
1097         rc = sscanf(line, "%1023s %1023s", s1, s2);
1098         if (rc >= 2) {
1099             int mtabmaj, mtabmin;

```

```

1100         GetMajorMinor(s1, &mtabmaj, &mtabmin);
1101         if (((strcmp(s1, name) == 0) || (strcmp(s2, name) == 0)) ||
1102             ((maj != -1) && (maj == mtabmaj) && (min == mtabmin))) {
1103             FCLOSE(fp);
1104             *deviceName = DeMangleMount(strdup(s1));
1105             *mountName = DeMangleMount(strdup(s2));
1106             return 1;
1107         }
1108     }
1109 }
1110 *deviceName = 0;
1111 *mountName = 0;
1112 FCLOSE(fp);
1113 return 0;
1114 }
1115
1116
1117 /*
1118  * /etc/fstab をもとにデバイスがマウント可能かどうか調べる。
1119  * マウント可能な場合は、デバイス名とマウントポイント名を設定して 1 を返し、
1120  * そうでない場合は 0 を返す。
1121  */
1122 static int MountableDevice(const char *name, char **mountName, char **deviceName)
1123 {
1124     FILE *fp;
1125     char line[1024];
1126     char s1[1024];
1127     char s2[1024];
1128     int rc;
1129
1130     fp = fopen("/etc/fstab", "r");
1131
1132     /* /etc/fstab ファイルは、中にパスワードが書かれているといった理由で読み取り可能でない場合もある。
1133      */
1134     if (fp == NULL) {
1135         if (v_option) {
1136             printf( _("%s: unable to open /etc/fstab: %s\n"), programName,
1137                 strerror(errno));
1138         }
1139         return -1;
1140     }
1141     while (fgets(line, sizeof(line), fp) != 0) {
1142         rc = sscanf(line, "%1023s %1023s", s1, s2);
1143         if (rc >= 2 && s1[0] != '#' && strcmp(s2, name) == 0) {
1144             if (strncasecmp(s1, "UUID=", 5) == 0) {
1145                 char *realDeviceName = malloc(strlen(s1) + 32);
1146                 sprintf(realDeviceName, "/dev/disk/by-uuid/%s", s1 + 5);
1147                 strncpy(s1, realDeviceName, sizeof(s2)-1);
1148             }
1149             FCLOSE(fp);
1150             *deviceName = DeMangleMount(strdup(s1));

```

```

1150         *mountName = DeMangleMount(strdup(s2));
1151         return 1;
1152     }
1153 }
1154 FCLOSE(fp);
1155 return 0;
1156 }
1157
1158
1159 /*
1160  * マウントテーブルの各項目を調べ、
1161  * 正規表現にマッチするデバイスをすべてアンマウントする。
1162  */
1163 static void UnmountDevices(const char *pattern)
1164 {
1165     regex_t preg;
1166     FILE *fp;
1167     char s1[1024];
1168     char s2[1024];
1169     char line[1024];
1170     int status;
1171
1172     if (regcomp(&preg, pattern, REG_EXTENDED)!=0) {
1173         perror(programName);
1174         exit(1);
1175     }
1176
1177     fp = fopen((p_option ? "/proc/mounts" : "/etc/mtab"), "r");
1178     if (fp == NULL)
1179     {
1180         fprintf(stderr, _("unable to open %s: %s\n"), (p_option ? "/proc/mounts" :
1181         "/etc/mtab"), strerror(errno));
1182         exit(1);
1183     }
1184     while (fgets(line, sizeof(line), fp) != 0) {
1185         status = sscanf(line, "%1023s %1023s", s1, s2);
1186         if (status >= 2) {
1187             status = regexec(&preg, s1, 0, 0, 0);
1188             if (status == 0) {
1189                 if (v_option)
1190                     printf(_("%s: unmounting `%s'\n"), programName, s2);
1191                 Unmount(s2);
1192             }
1193         }
1194     }
1195     regfree(&preg);
1196     FCLOSE(fp);
1197 }
1198
1199

```



```

1200 /* name がシンボリックリンクかどうか確認する。シンボリックリンクである場合はリンク先を返す。 */
1201 static char *SymLink(const char *name)
1202 {
1203     int status;
1204     char s1[PATH_MAX];
1205     char s2[PATH_MAX];
1206     char s4[PATH_MAX];
1207     char result[PATH_MAX];
1208     char *s3;
1209
1210     memset(s1, 0, sizeof(s1));
1211     memset(s2, 0, sizeof(s2));
1212     memset(s4, 0, sizeof(s4));
1213     memset(result, 0, sizeof(result));
1214
1215     status = readlink(name, s1, sizeof(s1) - 1);
1216
1217     if (status == -1)
1218         return 0;
1219
1220     s1[status] = 0;
1221     if (s1[0] == '/') { /* 絶対パスを指すリンクの場合 */
1222         return strdup(s1);
1223     } else { /* 相対パスを指すリンクの場合、パスの基準名を追加 */
1224         strncpy(s2, name, sizeof(s2)-1);
1225         s3 = strrchr(s2, '/');
1226         if (s3 != 0) {
1227             s3[1] = 0;
1228             snprintf(result, sizeof(result)-1, "%s%s", s2, s1);
1229         }
1230     }
1231     realpath(result, s4);
1232     return strdup(s4);
1233 }
1234
1235
1236 /*
1237  * 与えられた name が、複数パーティションを持ちうるデバイスのパターンにマッチするかを調べる。
1238  * マッチする場合はそのデバイスに含まれる各パーティションにマッチする正規表現を返し、
1239  * マッチしない場合は 0 を返す。
1240  */
1241 static char *MultiplePartitions(const char *name)
1242 {
1243     int i = 0;
1244     int status;
1245     regex_t preg;
1246     char pattern[256];
1247     char *result = 0;
1248
1249     for (i = 0; partitionDevice[i] != 0; i++) {

```

```

1250      /* ^/dev/foo[a-z]([0-9]?[0-9])?$ を探す。たとえば /dev/hda1 */
1251      strcpy(pattern, "^/dev/");
1252      strcat(pattern, partitionDevice[i]);
1253      strcat(pattern, "[a-z]([0-9]?[0-9])?$");
1254      if (regcomp(&preg, pattern, REG_EXTENDED|REG_NOSUB) != 0) {
1255          perror(programName);
1256          exit(1);
1257      }
1258      status = regexec(&preg, name, 1, 0, 0);
1259      regfree(&preg);
1260      if (status == 0) {
1261          result = (char *) malloc(strlen(name) + 25);
1262          if (result == NULL) {
1263              fprintf(stderr, _("%s: could not allocate memory\n"),
programName);
1264              exit(1);
1265          }
1266          strcpy(result, name);
1267          result[strlen(partitionDevice[i]) + 6] = 0;
1268          strcat(result, "([0-9]?[0-9])?$");
1269          if (v_option)
1270              printf(_("%s: `%s' is a multipartition device\n"), programName,
name);
1271          return result;
1272      }
1273  }
1274  if (v_option)
1275      printf(_("%s: `%s' is not a multipartition device\n"), programName, name);
1276  return 0;
1277  }
1278
1279
1280  /* -x オプションの処理 */
1281  static void HandleXOption(char *deviceName)
1282  {
1283      int fd;          /* デバイスのファイル記述子 */
1284      if (x_option) {
1285          if (v_option)
1286          {
1287              if (x_arg == 0)
1288                  printf(_("%s: setting CD-ROM speed to auto\n"), programName);
1289              else
1290                  printf(_("%s: setting CD-ROM speed to %dX\n"), programName,
x_arg);
1291          }
1292          fd = OpenDevice(deviceName);
1293          SelectSpeedCdrom(fd, x_arg);
1294          exit(0);
1295      }
1296  }
1297
1298
1299  /* メインプログラム */

```

```

1300 int main(int argc, char **argv)
1301 {
1302     const char *defaultDevice = DEFAULTDEVICE; /* ユーザーが name を指定しなかった場合の既定
        値 */
1303     int worked = 0; /* イジェクトに成功した場合は 1 にセットされる */
1304     char *device = 0; /* ユーザーが指定した name */
1305     char *fullName; /* 展開後の name */
1306     char *fullNameOrig; /* 展開後の name (シンボリックリンクは解決しない) */
1307     char *deviceName; /* デバイス名 */
1308     char *linkName; /* デバイスのシンボリックリンクの名前 */
1309     char *mountName; /* デバイスのマウントポイントの名前 */
1310     int fd; /* デバイスのファイル記述子 */
1311     int mounted = 0; /* デバイスがマウントされている場合は、真 */
1312     int mountable = 0; /* デバイスが /etc/fstab に記述されている場合は、真 */
1313     int result = 0; /* 操作の結果を格納する */
1314     char *pattern; /* デバイスが複数パーティションを持つ場合の正規表現 */
1315     int ld = 6; /* シンボリックリンクの最大の深さ */
1316
1317     I18NCODE
1318
1319     /* プログラム名は、他の処理でも使われるグローバル変数 */
1320     programName = strdup(argv[0]);
1321
1322     /* コマンドライン引数の解析 */
1323     parse_args(argc, argv, &device);
1324
1325
1326     /* -d オプションの処理 */
1327     if (d_option) {
1328         printf(_("%s: default device: `%s'\n"), programName, defaultDevice);
1329         exit(0);
1330     }
1331
1332     /* デバイスが指定されなかった場合は、既定値を使う */
1333     if (device == 0) {
1334         device = strdup(defaultDevice);
1335         if (v_option)
1336             printf(_("%s: using default device `%s'\n"), programName, device);
1337     }
1338
1339     /* ユーザーが bash/tcsh のようなファイル名補完を使った場合
        (例: /mnt/cdrom/ ) に備え、末尾のスラッシュを除去する */
1340     if (device[strlen(device)-1] == '/')
1341         device[strlen(device)-1] = 0;
1342
1343     if (v_option)
1344         printf(_("%s: device name is `%s'\n"), programName, device);
1345
1346     do {
1347         /* デバイスまたはマウントポイントのフルパス名を調べる */
1348         fullName = FindDevice(device);

```

```

1350     if (fullName == 0) {
1351         fprintf(stderr, _("%s: unable to find or open device for: `%s'\n"),
1352             programName, device);
1353         exit(1);
1354     }
1355     if (v_option)
1356         printf(_("%s: expanded name is `%s'\n"), programName, fullName);
1357
1358     /* /proc/mounts はシンボリックリンクの解決をしないので、 */
1359     /* シンボリックリンクについて確認する */
1360     fullNameOrig = strdup(fullName);
1361     linkName = strdup(fullName); /* linkName を確実に初期化しておく */
1362     if (!p_option) {
1363         while ((linkName = SymLink(fullName)) && (ld > 0)) {
1364             if (v_option)
1365                 printf(_("%s: `%s' is a link to `%s'\n"), programName,
1366                     fullName, linkName);
1367             free(fullName);
1368             fullName = strdup(linkName);
1369             free(linkName);
1370             linkName = 0;
1371             ld--;
1372         }
1373     }
1374     /* リンクが深すぎた場合の処理 */
1375     if (ld <= 0) {
1376         printf(_("%s: maximum symbolic link depth exceeded: `%s'\n"), programName,
1377             fullName);
1378         exit(1);
1379     }
1380     /* (fullName の指すパスが) マウントポイントである場合は、デバイス名を取得する */
1381     mounted = MountedDevice(fullName, &mountName, &deviceName);
1382     if (v_option) {
1383         if (mounted)
1384             printf(_("%s: `%s' is mounted at `%s'\n"), programName,
1385                 deviceName, mountName);
1386         else
1387             printf(_("%s: `%s' is not mounted\n"), programName, fullName);
1388     }
1389     if (!mounted) {
1390         deviceName = strdup(fullName);
1391     }
1392
1393     /* (fullName の指すパスに) 何もマウントされていない場合は、そこがマウントポイントとなりうるかを
1394     調べる */
1394     if (!mounted) {
1395         mountable = MountableDevice(fullName, &mountName, &deviceName);
1396         /* 戻り値が -1 の場合は、fstab が読めなかったということ */
1397         if (v_option && mountable >= 0) {
1398             if (mountable)
1399                 printf(_("%s: `%s' can be mounted at `%s'\n"), programName, deviceName,
1400                     mountName);

```

```

1400     else
1401         printf(_("%s: `%s' is not a mount point\n"), programName, fullName);
1402     }
1403 }
1404
1405 result = GetMajorMinor(deviceName, NULL, NULL);
1406 if (result == -1) {
1407     fprintf(stderr,
1408         _("%s: tried to use `%s' as device name but it is no block device\n"),
1409         programName, deviceName);
1410 }
1411
1412 } while (result == -1);
1413
1414 /* -n オプションの処理 */
1415 if (n_option) {
1416     printf(_("%s: device is `%s'\n"), programName, deviceName);
1417     if (v_option)
1418         printf(_("%s: exiting due to -n/--noop option\n"), programName);
1419     exit(0);
1420 }
1421
1422 /* -i オプションの処理 */
1423 if (i_option) {
1424     fd = OpenDevice(deviceName);
1425     ManualEject(fd, i_arg);
1426     exit(0);
1427 }
1428
1429 /* -a オプションの処理 */
1430 if (a_option) {
1431     if (v_option) {
1432         if (a_arg)
1433             printf(_("%s: enabling auto-eject mode for `%s'\n"),
1434                 programName, deviceName);
1434         else
1435             printf(_("%s: disabling auto-eject mode for `%s'\n"),
1436                 programName, deviceName);
1437     }
1438     fd = OpenDevice(deviceName);
1439     AutoEject(fd, a_arg);
1440     exit(0);
1441 }
1442
1443 /* -t オプションの処理 */
1444 if (t_option) {
1445     if (v_option)
1446         printf(_("%s: closing tray\n"), programName);
1447     fd = OpenDevice(deviceName);
1448     CloseTray(fd);
1449     HandleXOption(deviceName);
1450     exit(0);

```

```

1450 }
1451
1452 /* -x オプションの処理 */
1453 if (X_option) {
1454     if (v_option)
1455         printf(_("%s: listing CD-ROM speed\n"), programName);
1456     fd = OpenDevice(deviceName);
1457     ListSpeedCdrom(deviceName, fd);
1458     exit(0);
1459 }
1460
1461 /* -x オプションのみを処理する */
1462 if (!c_option) HandleXOption(deviceName);
1463
1464 /* デバイスがマウントされている場合はアンマウントする */
1465 if ((m_option != 1) && mounted) {
1466     if (v_option)
1467         printf(_("%s: unmounting device '%s' from '%s'\n"), programName,
deviceName, mountName);
1468     Unmount(mountName);
1469 }
1470
1471 /* 対象デバイスが複数パーティションを持つ場合、
1472     そのデバイス上のパーティションをすべてアンマウントする */
1473 pattern = MultiplePartitions(deviceName);
1474 if ((m_option != 1) && (pattern != 0))
1475     UnmountDevices(pattern);
1476
1477 /* -T オプションの処理 */
1478 if (T_option) {
1479     if (v_option)
1480         printf(_("%s: toggling tray\n"), programName);
1481     fd = OpenDevice(deviceName);
1482     ToggleTray(fd);
1483     HandleXOption(deviceName);
1484     exit(0);
1485 }
1486
1487 /* -c オプションの処理 */
1488 if (c_option) {
1489     if (v_option)
1490         printf(_("%s: selecting CD-ROM disc #%d\n"), programName, c_arg);
1491     fd = OpenDevice(deviceName);
1492     ChangerSelect(fd, c_arg);
1493     HandleXOption(deviceName);
1494     exit(0);
1495 }
1496
1497 /* ユーザーがイジェクト方式を指定しなかった場合は、4 種類の方式をすべて試みる */
1498 if ((r_option + s_option + f_option + q_option) == 0) {
1499     r_option = s_option = f_option = q_option = 1;

```

```
1500 }
1501
1502 /* デバイスをオープンする */
1503 fd = OpenDevice(deviceName);
1504
1505 /* 各種イジェクト方式を、成功するまで順に試していく */
1506 if (r_option) {
1507     if (v_option)
1508         printf(_("%s: trying to eject `%s' using CD-ROM eject command\n"),
1509             programName, deviceName);
1509     worked = EjectCdrom(fd);
1510     if (v_option) {
1511         if (worked)
1512             printf(_("%s: CD-ROM eject command succeeded\n"), programName);
1513         else
1514             printf(_("%s: CD-ROM eject command failed\n"), programName);
1515     }
1516 }
1517
1518 #ifdef HAVE_EJECT SCSI
1519 if (s_option && !worked) {
1520     if (v_option)
1521         printf(_("%s: trying to eject `%s' using SCSI commands\n"),
1522             programName, deviceName);
1523     worked = EjectScsi(fd);
1524     if (v_option) {
1525         if (worked)
1526             printf(_("%s: SCSI eject succeeded\n"), programName);
1527         else
1528             printf(_("%s: SCSI eject failed\n"), programName);
1529     }
1530 }
1531 #endif
1532
1533 #ifdef HAVE_EJECT_FLOPPY
1534 if (f_option && !worked) {
1535     if (v_option)
1536         printf(_("%s: trying to eject `%s' using floppy eject command\n"),
1537             programName, deviceName);
1538     worked = EjectFloppy(fd);
1539     if (v_option) {
1540         if (worked)
1541             printf(_("%s: floppy eject command succeeded\n"), programName);
1542         else
1543             printf(_("%s: floppy eject command failed\n"), programName);
1544     }
1545 }
1546 #endif
1547
1548 #ifdef HAVE_EJECT_TAPE
1549 if (q_option && !worked) {
1550     if (v_option)
1551         printf(_("%s: trying to eject `%s' using tape offline command\n"),
1552             programName, deviceName);
1553     worked = EjectTape(fd);
1554     if (v_option) {
1555         if (worked)
1556             printf(_("%s: tape offline command succeeded\n"), programName);
1557         else
1558             printf(_("%s: tape offline command failed\n"), programName);
1559     }
1560 }
1561 #endif
1562 }
```

```
1550         worked = EjectTape(fd);
1551         if (v_option) {
1552             if (worked)
1553                 printf(_("%s: tape offline command succeeded\n"), programName);
1554             else
1555                 printf(_("%s: tape offline command failed\n"), programName);
1556         }
1557     }
1558 #endif
1559
1560 if (!worked) {
1561     fprintf(stderr, _("%s: unable to eject, last error: %s\n"), programName,
1562             strerror(errno));
1563     exit(1);
1564 }
1565
1566 /* 後片づけ */
1567 CLOSE(fd);
1568 free(device);
1569 free(deviceName);
1570 free(fullName);
1571 free(fullNameOrig);
1572 free(linkName);
1573 free(mountName);
1574 free(pattern);
1575 exit(0);
1576 }
```


第 2 部 Eject コマンド解説

eject.c の usage() で使い方を出力しているが、そのなかにこんなものがある。

```
0256 fprintf(stderr,_(
0257 "Eject version %s by Jeff Tranter (tranter@pobox.com)\n"
0258 "Usage:\n"
0259 "  eject -h                -- display command usage and exit\n"
0260 "  eject -v                -- display program version and exit\n"
0261 "  eject [-vnrsfqpm] [<name>]    -- eject device\n"
0262 "  eject [-vn] -d          -- display default device\n"
0263 "  eject [-vn] -a on|off|1|0 [<name>] -- turn auto-eject feature on or off\n"
0264 "  eject [-vn] -c <slot> [<name>]    -- switch discs on a CD-ROM changer\n"
0265 "  eject [-vn] -t [<name>]    -- close tray\n"
0266 "  eject [-vn] -T [<name>]    -- toggle tray\n"
0267 "  eject [-vn] -i on|off|1|0 [<name>] -- toggle manual eject protection on/off\n"
0268 "  eject [-vn] -x <speed> [<name>]    -- set CD-ROM max speed\n"
0269 "  eject [-vn] -X [<name>]    -- list CD-ROM available speeds\n"
0270 "Options:\n"
0271 "  -v\t-- enable verbose output\n"
0272 "  -n\t-- don't eject, just show device found\n"
0273 "  -r\t-- eject CD-ROM\n"
0274 #ifdef HAVE_EJECT_SCSI
0275 "  -s\t-- eject SCSI device\n"
0276 #endif
0277 #ifdef HAVE_EJECT_FLOPPY
0278 "  -f\t-- eject floppy\n"
0279 #endif
0280 #ifdef HAVE_EJECT_TAPE
0281 "  -q\t-- eject tape\n"
0282 #endif
0283 "  -p\t-- use /proc/mounts instead of /etc/mstab\n"
0284 "  -m\t-- do not unmount device even if it is mounted\n"
0285 )
0286 , version);
```

ひとつのメッセージが、# ifdef で細切れにされている (この次にある、長いオプションを説明するメッセージも同様に細切れにされている)。

これらのマクロは、以下のように定義されている。

```
0158 #if defined(__linux__)
(snip)
0192 #define HAVE_EJECT_SCSI
0193 #define HAVE_EJECT_FLOPPY
0194 #define HAVE_EJECT_TAPE
0195
0196 #elif defined(__FreeBSD_kernel__)
0197 #include <sys/cdio.h>
0198 #endif /* if defined(__linux__) */
```

これらは Debian GNU/kFreeBSD に対応するための措置であり、Debian 側のパッチ¹をそのまま取り込んだ結果こうなっている。

メッセージカタログ (eject/po/ja_JP.eucJP.po²) は以下のようになり、Debian GNU/kFreeBSD

1 <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=324569>

2 CVS HEAD のものだが、PO ファイルは古いまま更新されていないので、-T, -i, -m オプションの説明がない。

ではこのメッセージが常に原文で表示されることになる³。xgettext(1) は #ifdef 等を解釈せずに無視するので、Debian GNU/kFreeBSD 上で POT ファイルを作り直しても同じことである。

```
0116 #: ../eject.c:112
0117 #, c-format
0118 msgid ""
0119 "Eject version %s by Jeff Tranter (tranter@pobox.com)\n"
0120 "Usage:\n"
0121 "  eject -h\t\t\t\t-- display command usage and exit\n"
0122 "  eject -V\t\t\t\t-- display program version and exit\n"
0123 "  eject [-vnrsfq] [<name>]\t\t-- eject device\n"
0124 "  eject [-vn] -d\t\t\t-- display default device\n"
0125 "  eject [-vn] -a on|off|1|0 [<name>]\t-- turn auto-eject feature on or off\n"
0126 "  eject [-vn] -c <slot> [<name>]\t-- switch discs on a CD-ROM changer\n"
0127 "  eject [-vn] -t [<name>]\t\t-- close tray\n"
0128 "  eject [-vn] -x <speed> [<name>]\t-- set CD-ROM max speed\n"
0129 "Options:\n"
0130 "  -v\t\t-- enable verbose output\n"
0131 "  -n\t\t-- don't eject, just show device found\n"
0132 "  -r\t\t-- eject CD-ROM\n"
0133 "  -s\t\t-- eject SCSI device\n"
0134 "  -f\t\t-- eject floppy\n"
0135 "  -q\t\t-- eject tape\n"
0136 "  -p\t\t-- use /proc/mounts instead of /etc/mtab\n"
0137 msgstr ""
0138 "Eject バージョン %s by Jeff Tranter (tranter@pobox.com)\n"
0139 "使い方:\n"
0140 "  eject -h\t\t\t\t-- コマンドの使い方を表示して終了\n"
0141 "  eject -V\t\t\t\t-- プログラムのバージョンを表示して終了\n"
0142 "  eject [-vnrsfq] [<名前>]\t\t-- デバイスのイジェクト\n"
0143 "  eject [-vn] -d\t\t\t-- デフォルトのデバイスを表示\n"
0144 "  eject [-vn] -a on|off|1|0 [<名前>]\t-- 自動イジェクト機能をオン/オフ\n"
0145 "  eject [-vn] -c <スロット> [<名前>]\t-- CD-ROM チェンジャのディスクを切り替え\n"
0146 "  eject [-vn] -t [<名前>]\t\t-- トレーを閉じる\n"
0147 "  eject [-vn] -x <速度> [<名前>]\t-- CD-ROM の最大速度を設定\n"
0148 "オプション:\n"
0149 "  -v\t\t-- 詳細表示を有効にする\n"
0150 "  -n\t\t-- イジェクトを行わず、見つけたデバイスの表示のみを行う\n"
0151 "  -r\t\t-- CD-ROM のイジェクト\n"
0152 "  -s\t\t-- SCSI デバイスのイジェクト\n"
0153 "  -f\t\t-- フロッピーのイジェクト\n"
0154 "  -q\t\t-- テープのイジェクト\n"
0155 "  -p\t\t-- /etc/mtab の代わりに /proc/mounts を使う
```

3 メッセージが更新されているので、このままだと Linux でも原文が表示されることになるが、たとえ PO ファイルがコードに同期できていても kFreeBSD では原文が表示されることになる。

奥付と著作権

Doc-ja Archive Project

2011 年 8 月 20 日発行

文責: 岡野孝悌

ライセンス: 岡野が書いた部分に権利を主張するつもりはないので、自由に使ってちょうだい。Eject コマンドのライセンスは GPL2 or later なので、コードと翻訳は元のライセンスに従ってね。

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and

(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections

1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the

Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.