Debian GNU/Linux as a Japanese language teaching tool

Janusz S. Bień
Warsaw University
jsbien@uw.edu.pl

12th July 2003

1 Introduction

The present paper has two purposes. First, to provide some basic information about Debian, together with some background information concerning free software, for the readers who never heard about this distribution. Secondly, to provide some useful hints for a Debian user who want to set up his system for Japanese learning either for himself or for a Debian novice.

In consequence the reader of the first type is kindly requested to skip the details which are too technical for him. Analogically, the reader of the second kind may skip the introductory material.

2 Free software and Linux

We use the expression *free software* in the meaning assigned to it by Richard Stallman and Free Software Foundation (cf. [4]). Let's quote the most essential fragment:

> Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:
>
> - The freedom to run the program, for any purpose (freedom 0).
> - The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
> - The freedom to redistribute copies so you can help your neighbor (freedom 2).

- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

Due to the ambiguity of the word *free* some people prefer to refer to free software with other terms, such as *open source*[1]; we will not go here into some nuances of these terms.

Since the establishment of Free Software Foundation and the beginning of its GNU project more and more free programs become available; they are distributed on principles written down in one of several GNU project licenses. The most popular of them is GPL, i.e. *General Public License*, so sometimes free programs are described shortly as GPLed.

The most famous GPLed program is now Linux, in a narrow sense just an operating system kernel, and in a larger sense (deprecated by Free Software Foundation) the complete GNU/Linux operating system, incorporating various utilities and applications developed by the GNU project.

For years GPLed programs were difficult to install and sometimes difficult to use. However, recently they reached such a maturity that they are a viable alternative to commercial software even from the point of view of a novice user. In consequence they are now a real competition for the commercial software developers, who apply various countermeasures. The most dangerous ones for the free software movement are software patents, about to be introduced in Europe[2].

It should be noted that patents protect ideas, contrary to the copyright which protect texts. If a text is original, its author cannot be sued even if somebody else wrote earlier a similar text. On the other hand, an author of a program can be sued for a software patent infringement even if he is completely unaware of the patent in question. In consequence, this leaves on the market only large corporations which can afford costly monitoring of all patents granted or other ways to protect themselves against legal dangers (you can find more arguments against software patents e.g. in [16]).

Fortunately software patents do not mean an immediate death of free software, as it is exemplified by the USA (to the best of my knowledge, the only country with software patents at the very moment). Software patents are not only fought back directly, but also indirectly, sometimes in quite an original way; cf. for example *Patent Violating Programming Contest*[3] or the idea to treat free software as world cultural heritage protected by Unesco[4].

Last but not least, the advantages of free software become more and more often appreciated by politicians. For example, Associated Press on November 20, 2002 announced[5]

[1] Cf. e.g. `http://www.opensource.org/docs/definition.html`.
[2] Cf. `http://swpat.ffii.org/news/index.en.html`.
[3] Cf. `http://www.elug.de/projekte/patent-party/contest.html`.
[4] Cf. `http://www.fsfeurope.org/projects/mankind/`.
[5] Cf. e.g. `http://www.myneweconomy.com/articles/201102/JAPAN.htm`.

> Eager to catch up with nations switching to computer systems other than Microsoft Windows, Japan will study the possibility of using open-source software such as Linux at the government level.
>
> The public management ministry is earmarking 50 million yen ($410,000) for a panel of scholars and computer experts, including Microsoft officials, to finish the study by March 2004, Tatsuya Kawachi, a ministry deputy director, said Wednesday.

Amongst many examples from Europe there is the local government of Extremadura (a region in Spain), who distributed over 150 000 copies of Linux to schools and other institutions[6].

To conclude this introduction, there are quite high chances that a student of Japanese will have access to a Linux-based computer system in his office, school or home.

3 Debian project

The free software gives you freedom, but that does not mean that it cannot be sold for money. GNU/Linux is a large complicated system and arranging it into a set of CD-ROMs and making it easy to install requires a substantial effort. Such a ready to use version of the system is called a *distribution*; most of distributions are available on commercial basis.

Debian GNU/Linux is a distribution available at no cost on the Internet, developed by almost a thousand of volunteers. It is also the largest GNU/Linux distribution, consisting of over 8 000 software packages. Contrary to most commercial distributions which are targeted exclusively for PC computers, Debian is available for various architectures, ranging from palmtops through Motorola- and PowerPC-based computers to IBM mainframes.

Debian project started in 1983. Its home page is `http://www.debian.org/`, mirrored in many countries and often translated, at least partially, into other languages (including Polish and Japanese). The goals of the project are formulated in *Debian Social Contract*, which has to be accepted by every developer. An essential part of the contract is called *The Debian Free Software Guidelines*; the Debian understanding of free software is more limited then that of Free Software Foundation, so to avoid confusion the term *DFSG-free* is used when appropriate. The organisation of the project is described in *Debian Constitution*.

The primary means of communication both between the developers and the users are mailing lists hosted at `http://lists.debian.org/`; for example, the `debian-announce` list has over 32 000 subscribers. The postings are archived and can be searched for required information.

More general information about the project can be found in particular in [8].

[6] Cf. `http://inquirerinside.com/?article=6048`.

3.1 Development and users feedback

To assure the quality of the system, Debian project accepts for distribution only the software packages provided by the official Debian developers, so every package has its Debian maintainer who is responsible for it; although everybody can apply to be a Debian developer, the applications are reviewed quite carefully. Quite often the Debian maintainer is not the author of the software, but only tests and adapts to Debian requirement some software developed and released by somebody else; the original author is called then the upstream maintainer.

The list of all packages is available at `http://www.debian.org/distrib/packages`. Entry for each package contains links to Debian Bug Tracking System (BTS) and Package Tracking System (PTS). Bugs can be reported via e-mail and are stored in BTS, together with the comments by other users; when the reported problem is solved, the bug is closed by the developer. The bugs are tagged with such labels as *Critical*, *Wishlist* etc.

Package Tracking System allows, among others, every interested user to enter a subscription allowing to receive various information about the very package, such as bug reports or the announcements of new version uploads.

The capabilities of BTS and PTS are used also for slightly different purposes, thanks to the introduction of the so called pseudo-packages. From a user point of view the most important of them is `wnpp`, i.e. *Work-Needing and Prospective Packages list*. Every user who is aware of a useful piece of software not yet available in Debian may submit a bug report against `wnpp` tagged RFP (*Request for package*) and drawing this way the attention of Debian developers to his needs.

3.2 Debian packages

As it was said above, the comprehensive information about Debian packages is available on the Debian WWW site. A subset of it is available also on every computer running Debian, in the file `/var/lib/dpkg/available`. Here are some sample entries from the file (slightly edited to fit the page width):

```
Package: xshodo
Priority: optional
Section: non-free/graphics
Installed-Size: 45
Maintainer: Kenshi Muto <kmuto@debian.org>
Architecture: i386
Version: 2.0-4
Depends: libc6 (>= 2.1), xlib6g (>= 3.3-5), xlib6g (>= 3.3.5-1)
Filename: dists/potato/non-free/binary-i386/graphics/xshodo_2.0-4.deb
Size: 13314
MD5sum: 11961164dfe611bf6641ce20927c0f18
Description: a virtual "SHODO - Japanese calligraphy" tool on X.
 XShodo, formerly known as XBakuzan, is a program to enjoy a virtual
 "SHODO" on X. Without writing brushes 'fude' and Indian ink 'sumi',
 XShodo may help you produce the fine art of KANJI writing or painting.
```

```
Package: chasen
Priority: extra
Section: misc
Installed-Size: 532
Maintainer: NOKUBI Takatsugu <knok@daionet.gr.jp>
Architecture: i386
Version: 2.2.9-3
Depends: libc6 (>= 2.2.4-4), libchasen0, ipadic
Filename: pool/main/c/chasen/chasen_2.2.9-3_i386.deb
Size: 369424
MD5sum: 2bae9dd122d2349fae450c876aaf4f11
Description: a Japanese Morphological Analysis System
 ChaSen is a morphological analysys system. It can segment and
 tokenize Japanese text string, and can output with many additional
 informations (pronunciation, semantic information, and others).
 It will print the result of such an operation to the standard output,
 so that it can either written to a file or further processed.
```

This file is used by the package management programs (the user has a choice of several of them) used primarily to install the packages, which involves in particular fullfiling the prerequisites specified in the `Depends` field.

There is however a problem with locating packages of interest for a particular user, as assigning a package to exactly one *section* is evidently not sufficient. The solution to this problem will be provided by *Debian Package Tags* ([20], cf. also `http://deb-usability.alioth.debian.org/debtags/`), which at the moment of this writing are in the experimental ("unstable") stage.

A single package can have several tags describing its different aspects. The tags can be then used by the Debian Package Browser[7]) to help the user to find a package with requested functionality. The list of tags, called *normative vocabulary*, is included in the `debtag` package, which contains also program to update it when needed. The vocabulary contains such tags as

```
Tag: language::japanese
Implies: language
Description: Japanese localization
```

and

```
Tag: laptop
Implies: hardware
Description: Laptop
```

The existing packages are now being tagged by some volunteers.

Although introducing such tags as `language::japanese` is a step in the right direction, I don't think it is sufficient. As I wrote in [3], the tags should be used to distinguish clearly between such cases as exemplified by the following packages:

[7] Cf. `http://debian.vitavonni.de/packagebrowser/`.

- `chasen` is the package containing a Japanese morphological analyser, documented in Japanese.

- `lookup-el` is the package containing a general purpose dictionary lookup tool (actually an extension of Emacs editor), documented only in Japanese.

In other words, the tags should distinguish the language of the application domain and the language of the documentation.

So how to find Debian packages related to Japanese? For the time being one of the best ways consists in browsing the `/var/lib/dpkg/available` file with such tools as the *occur* command of the Emacs editor, looking for such words as *Japanese*, *kanji*, *kana* etc. This will allow us to identify, in particular, the *edict* package

```
Package: edict
Priority: optional
Section: non-free/text
Installed-Size: 5660
Maintainer: Hayao Nakahara <nakahara@debian.org>
Architecture: all
Version: 2001.12.04-1
Suggests: lookup | xjdic | sdic-edict
Filename: pool/non-free/e/edict/edict_2001.12.04-1_all.deb
Size: 2141966
MD5sum: 92b307f9676462896645f67671692dfb
Description: English/Japanese dictionary.
 The EDICT file is the outcome of a  voluntary  project  to  produce a freely
 available Japanese/English  Dictionary  in  machine-readable  form.
 This package also contains compdic and jddict dictionaries.
```

containing the famous and ubiquitous Japanese-English dictionaries developed by Jim Breen and his colleagues[8].

Of course, several tools to use these dictionaries are also available.

3.3 Official and unofficial package repositories

Although Debian can be installed from a set of CDs (the recent release consists of 7 CDs), it can be also installed through a network, and a typical Debian users regularly update the system using Internet and `APT` (*Advanced Package Tool*); for computers without Internet connectivity there is a workaround provided by the `apt-zip` program, which prepares for another computer a script which downloads the required packages and stores them on removable media. Package manipulation (installation, deletion, upgrading etc.) can be done either with a command-line program such as `apt-get` or with full-screen or graphic interfaces such as `aptitude` or `synaptic-debtags` (unstable). By default a Debian system is configured to use only the packages available in the official Debian repositories or its official mirrors. The addresses of the repositories are stored in the `/etc/`

[8] Cf. `http://www.csse.monash.edu.au/groups/edrdg/`.

apt/sources.list file, which can be edited by hand or modified using such programs as apt-spy.

Besides the official repositories there are also unofficial ones, sometimes containing only a single package. Let us discuss a Japanese-related example.

One of the convenient tools to access Jim Breen's dictionaries is Gjiten, designed for the GNOME graphical desktop environment but usable also with other ones such as KDE (a Linux user has a large choice of them). For technical reasons we do not include here its screenshots, but the reader can have a look e.g. at http://www.mimuw.edu.pl/~jsbien/slajdy/JSB-EAJS03-s.pdf On Gjiten's home page (http://gjiten.sourceforge.net) in the download section we find in particular the following information:

```
Debian: gjiten_2.1-1_i386.deb

   Gjiten is apt-get-able. Put this into your /etc/apt/sources.list:
         deb http://gjiten.sourceforge.net/ ./
         deb-src http://gjiten.sourceforge.net/ ./

         You are welcome to become a sponsor to make gjiten an
         offcial debian package if you are a debian developer!
```

It means that Gjiten is available as a Debian package consisting of the file gjiten_2.1-1_i386.deb. However, the package is not an official one as its author is not a Debian developer and therefore is not entitled to include the package in the official distribution. Despite this, adding the entries quoted above to the /etc/apt/sources.list causes the package to be treated exactly in the same way as official packages. In particular, the available file will contain the entry:

```
Package: gjiten
Priority: optional
Section: x11
Installed-Size: 1882
Maintainer: Botond Botyanszki <b0ti@users.sourceforge.net>
Architecture: i386
Version: 2.1-1
Depends: bonobo-activation (>= 1:2.2.1.1), libart-2.0-2 (>= 2.3.8),
...
Suggests: im-ja
Size: 213824
Description: Japanese dictionary for GNOME
  gjiten is a Japanese dictionary for GNOME with advanced word and
  kanji lookup features.  Requires dictionary files (edict, kanjidic)
  to function.  See http://gjiten.sourceforge.net for dictionary files
  and updates.
```

It is also possible to download the gjiten_2.1-1_i386.deb file and install it directly using the appropriate Debian program; although this way may seem simpler, it is actually more complicated because it does not take into account the prerequisites of the package, which happens to depend on 28 official Debian packages.

The author of `gjiten` is looking for a *sponsor* of this package. A sponsor is a Debian developer who, after testing an unofficial package, agrees to become its Debian maintainer and to upload it to the official repository. Hence there is a chance that after some time *gjiten* package will change its status to official.

Gjiten has an interface to `kanjipad`, a program allowing to draw a kanji with a mouse (unfortunately, its abilities to recognise a hand-drawn kanji are very limited). It will serve us as an example of another problem: what to do if a needed program is not available as a Debian package at all?

`kanjipad` is distributed as the `kanjipad-1.2.1.tar.gz` file, which is a compressed archive containing about 40 files. A typical well-written program is installed on Linux (actually, on any Unix-like system) by three commands issued in the directory obtained by uncompressing and unpacking the program archive (usually the name has the form `*.tar.gz` or `*.tgz`). As `kanjipad` is a simple program, it does not need the first one (`./configure`) and requires only the two remaining ones:

```
make
make install
```

The last command moves all the files belonging to the program to the various directories of the system, which makes difficult to keep track of them e.g. for the purpose of uninstalling the program. The solution is to replace the last command by

```
checkinstall make install
```

A new Debian package `kanjipad_1.2.1-1_i386.deb` is then created on the fly, with some information provided by the program (e.g. the maintainer field is set to the user who issued the `checkinstall` command) or by the user (the description field). The package is installed and can be later manipulated (upgraded, deleted) in the standard way.

3.4 The Knoppix revolution

Despite many unquestionable advantages Debian had for years a very important drawback: it has been very difficult to install. The size of the distribution is also a disadvantage for a novice user, who is confused by the large number of programs with similar functionality. The situation changed drastically with the introduction of Knoppix (`http://www.knopper.net/knoppix/`), a Debian-based system bootable from a single CD, developed by Hans Knopper ([7]).

The idea of a mini- or midi-distributions bootable from a floppy or a CD is quite old, but usually their purposes were very specialised, like a rescue system. The first CD-bootable Linux version for general purpose seems DemoLinux[9]. It featured already some hardware recognition procedures, but (as the name suggests) it was not designed for regular use.

[9] Cf. `http://www.demolinux.org`.)

Knoppix appeared to be a great success, due to several factors. One of them is the extremely high quality of the harware recognition programs developed by Knopper, which allow Knoppix to be fully functional on almost every computer (for example, in an experiment of mine Knoppix had no problem to recognize an USB scanner and a TV tuner, although configuring these devices by hand appeared quite difficult even for an experienced user of Debian).

Knoppix is a full-fledged Debian system. After installing on the hard disk it can be updated and modified using the standard Debian tools. Moreover, the Knoppix-specific part of the system has also the form of Debian packages, which allows for easy creation of one's own versions of Knoppix; for example, the instruction how to make Knoppix more Polish-oriented can be found on Polish Knoppix site at `http://knoppix.7thguard.net/`.

Knoppix-based distributions become now abundant. They are oriented at different user groups and serve different purposes, for example Oralux[10] is intended for English and French speaking blind and visually-impaired users.

There are already two Japanese versions of Knoppix, available at `http://sourceforge.jp/projects/ya-knoppix-jp/` and `http://unit.aist.go.jp/it/knoppix/`; cf. also [21]. An advanced learner of Japanese may use any of them, getting the full power of the computer environment adapted to the needs of a typical Japanese user.

3.5 Towards a Knoppix version for learners of Japanese

THIS SECTION TO BE COMPLETELY REWRITTEN!!!

Although Debian provides two versions of TEX fully adapted to Japanese language (cf. eg. [18], p. 329), for a Polish user an extension of LATEX called CJKTEX ([9]) is much more convenient although less efficient. The quality of CJKTEX output can be judged e.g. by consulting the papers [1] and [14] available on Internet. Although there is a Debian package, maintained by Anthony Fok, theoretically equivalent to CJKTEX as available from Comprehensive TEX Archive Network and distributed on TEXLive (`http://www.tug.org/texlive/`), its support for Japanese seems to be broken from the very beginning[11] (the maintainer probably uses the package only for Chinese).

Knoppix by default starts the KDE graphical desktop; at the boot time the user has a chance to switch to another one, e.g. IceWM. In any case using a graphical user interface is resource consuming. KDE requires at least 96 MB RAM as minimum and at least 128 MB is recommended to run such applications as OpenOffice. On the other hand even 20 MB is sufficient to start Knoppix in text mode. Although the standard text mode limits drastically the number of different characters which can be displayed simultaneously, the so called framebuffer console does not have this limitation.

Unicode is a universal character set intended to cover all the living languages of the world and the most important dead languages. Detailed information is

[10] Cf. `http://oralux.org/`.

[11] Cf. e.g. the bug report at `http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=165350`.

available on the Web site of the Unicode Consortium (`http://www.unicode.org`); the version 4.0 is available in the book form ([19]). Unicode is now quite widely used; its application for ideographic scripts like Japanese is discussed in [10].

`bterm` is a program belonging to the `bogl-bterm` package. Its installation is slightly cumbersome, as it needs a Unicode font in a special format. The font is provided by the `bf-utf-source` package, but before use it must be converted with the `bdftobogl` program included in the *libbogl-dev* package. Running GNU Emacs in the `bterm` environment allows to make almost full use of the multilingual features of Emacs, in particular to input and edit Japanese texts. Unfortunately, there is also an annoying drawback: `bterm` does not allow to make use of colors, in particular to differentiate different parts of text (markup versus the text proper etc.).

Colors are supported by another framebuffer Unicode terminal, namely `fbiterm` available in the experimental ("unstable") version of Debian. Unfortunately, it comes with no documentation[12] which makes experimenting with it quite difficult (the maintainer of this package is Anthony Fok mentioned earlier).

The advantages and importance of Knoppix has been almost immediately recognised by a student of mine, Tomasz Starosta. In his M.Sc. thesis ([17]) he used it as a basis for a version of Knoppix focused on the multilingual GNU Emacs editor used as the frontend to the TEX and LATEX typesetting systems ([18]).

Starosta modified the startup procedure to allow the user to choose a pseudo-desktop called "emacs". This choice results in starting a Unicode console `bterm` with Emacs configured to use Unicode.

Starosta prepared an alternative Debian package which unfortunately in the meantime became obsolete.

Starosta's work was just a feasibility study. The task of making a Knoppix version oriented at the learners of Japanese still waits for volunteers.

4 GNU Emacs as a Debian migration tool

If a user has a MS Windows computer, he can prepare himself for the transition to Debian by starting to use GNU software under MS Windows. One of the most useful GNU programs is GNU Emacs.

GNU Emacs is more then just an text editor. It is a so powerful tool that it is difficult to provide for it a compact definitions. Its author described it in [15] as *The Extensible, Customizable, Self-Documenting, Display Editor*[13]; the Emacs-related news groups describe it as *the Swiss army knife of text editors*. My own definition is a *universal editor of textual data*. I mean by this that you can use Emacs to edit any kind of information which can be represented in the text form; natural language texts (plain or enriched, i.e. tagged for LATEX,

[12] Cf. `http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=160921`.

[13] As you can see from the date of this publication, Emacs is one of the oldest programs still in use.

SGML, XML etc) are only an example, the list includes even binary files and various other kinds of data. It can be also described as an editing environment. It is available on several platform, including Linux, MS Windows and, with some limitations, even DOS.

Since version 20 (of 15 September 1997) the editor has so called multibyte mode, allowing to mix in a single text various alphabets and scripts ([5]). It is possible due to integration of MULE (*MULti-lingual Enhancement to GNU Emacs*), which was earlier available as a separate program. Now this acronim is resolved as *MULtilingual Environment*. GNU Emacs 21, released on 22 October 2001, offers a lot of new features, including some improvements of the multibyte mode and better documentation. Although Emacs features now menus and toolbars, by design all the Emacs functions can be accessed from keyboard in a very efficient way.

One of the jokes about Emacs says that its name means *Escape Meta Alt Control Shift*[14], as most of the commands has keyboard shortcuts involving the keys mentioned above. The one most used is *Meta*, which is usually not available on present day keyboards, so is implemented with the help of *Alt* or *ESC* key. The notation `C-h C-t` means holding *Control* while typing *h* and *t*; this is actually the shortcut for the `describe-copying` command, which displayes the GNU license mentioned above. The notation `M-f` means either pressing the *Esc* key, releasing it and pressing the *f* key or, which is usually more convenient, holding the key assigned the *Meta* function (on a PC keyboard this is usually `Alt`) while pressing *f* key; this shortcut, as you can check with `C-h k` (`describe-key`) is for `forward-word` command. Many functions useful for multilingual texts have shortcuts starting with *M-x Ret*.

You can learn basics of Emacs with the tutorial, available with the `help-with-tutorial` command (`C-h t`). In my opinion the tutorial does not stress sufficiently an extremely convenient feature of Emacs, so called `completion`. When you are prompted for choosing from a limited (although often quite large) number of possibilities, pressing `TAB` once completes your input as much as possible; pressing `TAB` again shows you all the remaining possibilities (you can also use `SPACE` for completion, but its function is slightly different).

The tutorial explains how to provide a function with a numeric argument, but does not mention that many functions accept so called *prefix argument*, consisting just of `C-u` typed before the command. You can use it e.g. to obtain the tutorial in a language of your choice instead of the default one (which depends on your locale setting).

You can test the multilingual features of Emacs with the `view-hello-file` command (`C-h h`). If you are interested in larger samples of texts in various languages, use `help-with-tutorial` command with the prefix argument; you will be prompted for the choice between Czech, Dutch, English, German, Japanese, Korean, Polish, Romanian, Slovak, Slovenian, Spanish and Thai (cf. [11] and [12]). Due to the completion, it is sufficient to type `C-u C-h t p TAB` for the tutorial in Polish and `C-u C-h t j TAB`) for the Japanese version.

[14] Cf. `JOKES` file distributed with Emacs.

GNU Emacs comes with his own version of input method for many languages including Japanese. Its name, *quail*, is a joke. As quail egg is smaller than chicken egg, so the *quail* input method is a simplified version of *tamago* (卵)、which means a (chicken) egg, but is also a popular input method for Japanese.

The documentation for the quail input method exists in English, but is dispersed in the so called documentation strings of individual commands. My paper [1] contains the only, to the best of my knowledge, systematic presentation of its use for Japanese (at least in a language other than Japanese). It is actually a kind of hands-on tutorial using as an example a sentence from the Emacs documentation, namely

あなたが現在見ているのはEmacs 入門ガイドです。

GNU Emacs is a big piece of software. Besides the standard distribution there are also many extensions contributed by its users. The packaging of Emacs in Debian is to be changed because of legal consideration; the elements considered to be not free according to the *Debian Free Software Guidelines* are to be put into separte packages. To make the full use of Emacs for Polish and Japanese one needs the Emacs proper (now at the `emacs21` package), the fonts for display (the default fonts should work, but the fonts originally used with Emacs are available in packages such as `xfonts-intl-japanese`) and fonts for printing (for historical reasons Emacs uses different font format for printing; the fonts are available in the `emacs-intl-fonts` package). To handle Unicode an extension `mule-ucs` is needed. If Emacs is to be used for TEX or LATEX, `auctex` is a must.

Using multilingual features of Emacs on MS Windows is possible, but requires rather sophisticated configuration. Therefore I prepered several versions of a live Emacs CD which uses the system autostart feature: after inserting the CD to a drive GNU Emacs starts ready for editing Polish, Japanese and other texts. The recent version of the CD is available for download as [2]. However, as it was intended for the students attending my lectures, its documentation is almost not existent.

5 Conclusion

Appropriately configured Debian GNU/Linux system, such as Knoppix (bootable from a single CD) or its variant, provides facilities needed for writing and editing Japanese texts at no cost. A Japanese language learner can use it even on a low-end PC, while advanced learners can switch to Japanese versions of Debian with Japanese graphical user interfaces. Migration to Debian can be done smoothly, by starting to use the GNU Emacs editor on a MS Windows-based computer.

6 Acknowledgements

First of all I would like to thank Prof. Wiesław Kotański for his attempt (in late seventies) to teach me (and several my Computer Science colleagues) Japanese

language. I would like also to thank Toho Mitsuhiko (東保 光彦) for his encouragement to renew my interest in Japanese, Katarzyna Ganderska for her effort to make Polish computer users familiar with Mule for Windows (a Japanese version of Emacs, cf. [6]) and Michał Piskorski for his work on using CJKTEX with Polish texts. Last but not least, on behalf of Department of Applied Computer Science (established in 1994, since October 1998 to February 1993 headed by me, now in liquidation) of Institute of Oriental Studies, I would like to thank Cezary Żemis and Seweryn Karłowicz, who for several years maintained for the Institute the network server without any renumeration.

This paper was prepared on a Knoppix-based Debian system with GNU Emacs and CJKTEX. To fullfil the requirements of the editors it was converted to HTML using the `latex2html` program and imported to MS Word.

References

[1] Janusz S. Bień: Teksty wielojęzyczne w edytorze GNU Emacs [Multilingual texts in GNU Emacs editor; in Polish]. *Biuletyn GUST* z. 16, r. 2001, s. 3–13. `http://www.mimuw.edu.pl/~jsbien/publikacje/JSB-Bach01.pdf`.

[2] Janusz S. Bień (red.). Wybrane narzędzia przetwarzania tekstów wielojęzycznych dla Windows 95/98/NT/2000 i komputerów PC [Selected tools for multilingual text processing for Windows 95/98/NT/2000 and PC computers]. CD-ROM image available at `http://www.mimuw.edu.pl/~jsbien/wnptw/wnptw093p.iso.bz2`.

[3] Janusz S. Bień. Package metadata. `deb-usability-list@lists.alioth.debian.org`, 16 Apr 2003 06:41:49 +0200. `http://lists.alioth.debian.org/pipermail/deb-usability-list/2003-April/000100.html`.

[4] Free Software Foundation. The Free Software Definition. `http://www.gnu.org/philosophy/free-sw.html`

[5] Kenichi Handa (半田 剣一), Mikiko Nishikimi (錦見 美貴子), Satoru Tomura (戸村 哲), Naoto Takahashi (高橋 直人). Unified and Extensible Mechanism for Multilingual Text Processing. *The Fourth Pacific Rim Conference on Artificial Intelligence* PRICAI '96, 26–30 August 1996, Cairns, North Queensland, Australia. `http://www.m17n.org/mule/pricai96/`.

[6] Miyashita Hisashi (宮下 尚). Mule for Windows 活用入門. カットシステム 1997.

[7] Klaus Knopper. Building a self-contained auto-configuring Linux system on an iso9660 filesystem. Proceedings of the 4th Annual Linux Showcase & Conference, Atlanta, Georgia, USA, October 10–14, 2000, pp. 373-376. Available online at `http://www.usenix.org/publications/library/proceedings/als00/2000papers/papers/knopper.html` mirrored at `http://www.knopper.net/knoppix/`.

[8] Christoph Lameter. Debian/GNU Linux: The Past, the Present and the Future. Free Software Symposium 2002. University of Tokyo. http://u-os.org/tokyo/.

[9] Werner Lemberg. The CJK package for LaTeX 2ε — Multilingual support beyond babel. http://www.tug.org/TUGboat/Articles/tb18-3/cjkintro600.pdf.

[10] Ken Lunde. *CJKV Information Processing*. O'Reilly 1999.

[11] Naoto Takahashi (高橋 直人), Kenichi Handa (半田 剣一), Mikiko Nishikimi (錦見 美貴子), Satoru Tomura (戸村 哲). Thai handling in Mule/Emacs. Orchid Corpus Technical Report. http://www.links.nectec.or.th/orchid/pub/Paper3.html.

[12] Naoto Takahashi (高 橋 直 人). Enhanced Thai handling in Mule-2.3. Orchid Corpus Technical Report. http://www.links.nectec.or.th/orchid/pub/Paper4.html.

[13] Naoto Takahashi (高橋 直人), Daniel Yacob. Adding Ethiopic Features to Emacs. *First International Conference on Information Technology in Ethiopia and Computational Ethiopics Symposium*, Addis Abeba, 1997. http://m17n.org/ntakahas/abstracts/mule-paper.ps.gz.

[14] Michał Piskorski: Języki dalekowschodnie i pakiet CJK [Far East languages and CJK package; in Polish]. *Biuletyn GUST* z. 16, r. 2001, s. 14–19. http://www.mimuw.edu.pl/~jsbien/MP/MP-Bach01.pdf.

[15] Richard Stallman. Emacs: The Extensible, Customizable, Self-Documenting, Display Editor. Cambridge, Massachusetts: MIT Artificial Intelligence Laboratory, AI Memo #519, June 1979. Also AI Memo #519A, March 1981, ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-519A.pdf

[16] Richard Stallman. Software patents victimise developers. http://news.zdnet.co.uk/story/0,,t269-s2107481,00.html.

[17] Tomasz Starosta. Tworzenie bootowalnych płyt CDROM z systemem Linux na potrzeby przetwarzania tekstów [Creation of bootable CDs with Linux system customized for text processing; in Polish]. M.Sc. thesis, Warsaw University, Computer Science Institute, December 2002.

[18] Apostolos Syropoulos, Antonis Tsolomitis, Nick Sofroniou. *Digital Typography Using LaTeX*. Springer Verlag 2002.

[19] The Unicode Standard, Version 4.0. Addison-Wesley 2003.

[20] Enrico Zini. Announcing Debian Package Tags. deb-usability-list@lists.alioth.debian.org, Mon, 28 Apr 2003 13:55:22 +0200. http://lists.alioth.debian.org/pipermail/deb-usability-list/2003-April/000103.html.

[21] 三原弘一著. KNOPPIXではじめよう！かんたん体験Linux. 初版発行2003.09.