List of Variables which are not inplemented to Demacs 16
List of Functions added to Demacs 16
List of Functions added to Demacs 16

Appendix C Index 17

Functions 17
Variables 17

Table of Contents

1 Demacs version 1.2.0 1
- 1.1 Current Version 1
- 1.2 Platforms 1
- 1.3 Bug Report 1

2 Installation 2
- 2.1 Contents of This Distribution Kit 2
- 2.2 How to Install 2
- 2.3 Sample Session to Install 3

3 Invocation 4
- 3.1 Environment Variables 4
- 3.2 Options of DOS Extender `demacs.exe` 4
- 3.3 Options of Demacs 5

4 Features 6
- 4.1 File Type: Text or Binary Translation 6
  - 4.1.1 Translation Mode 6
  - 4.1.2 Demacs Buffer Management 6
    - 4.1.2.1 File Type of Each Buffer 6
    - 4.1.2.2 Buffer Creation 7
  - 4.1.3 Reading Files into Demacs Buffers 7
    - 4.1.3.1 Inserting Files into Buffers 7
    - 4.1.3.2 Visiting Files and Reading 7
    - 4.1.3.3 Creating Files and Visiting 8
  - 4.1.4 Writing Buffers into Files 8
  - 4.1.5 Process I/O 8
- 4.2 "8bit clean" Display Mode 8
- 4.3 8086 Software Interrupt 8
- 4.4 Machine Specified Features 9
  - 4.4.1 IBM PC 10
  - 4.4.2 Toshiba J-3100 10
  - 4.4.3 NEC PC-9801, EPSON PC-386 10
- 4.5 Other Demacs Features 10

5 Restrictions 12

6 Compilation 13

Appendix A Terminal Specified Features 14

Appendix B Differences from UNIX Version 16

List of Functions which are not inplemented to Demacs 16

Appendix C. Index

Functions

D

define-file-name-file-type 7
demacs-version 1

F

find-file-type-from-file-name 7

I

int86 9

M

make-register 9

R

register-value 9

S

set-default-file-type 7
set-file-type 6
set-register-value 9

T

toggle-pass8-mode 8

Variables

D

default-file-type 7
dos-machine-type 9

F

file-type 6

P

pass8-mode 8

Appendix B. Differences from UNIX Version

List of Functions which are not inplemented to Demacs

```
accept-process-output
continue-process
delete-process
file-locked-p
get-buffer-process
get-process
interrupt-process
kill-process
list-processes
make-symbolic-link
open-network-stream
process-buffer
process-command
process-exit-status
process-filter
process-id
process-kanji-code
process-kill-without-query
process-list
process-mark
process-name
process-send-eof
process-send-region
process-send-string
process-sentinel
process-status
processp
quit-process
set-process-buffer
set-process-filter
set-process-kanji-code
set-process-sentinel
start-process
stop-process
waiting-for-user-input-p
```

List of Variables which are not inplemented to Demacs

```
delete-exited-processes
process-connection-type
```

List of Functions added to Demacs

```
bdos
define-file-name-file-type
demacs-version
disk-free-space
disk-total-space
file-type-internal
file-type-p
find-file-not-found-set-file-type
find-file-type-from-file-name
get-cursor-mode
get-screen-mode
int86
intdos
invoke-find-file-type
make-register
register-value
set-cursor-mode
set-default-file-type
set-file-type
set-keyclick
set-register-value
set-screen-mode
toggle-pass8-mode
```

List of Functions added to Demacs

```
default-file-type
demacs-version
demacs-version-date
dos-inhibit-setdisk
dos-machine-type
file-name-file-type-alist
file-type
file-type-alist
find-file-type
pass8-mode
register-name-by-byte-alist
register-name-by-word-alist
```

`Shift-F8`	`save-buffer`	Save current buffer
`Ctrl-F8`	`write-file`	Write current buffer into file
`F9`	`dired`	'Edit' directory
`Shift-F9`	`compile`	Compile the program
`Ctrl-F9`	`repeat-complex-command`	Edit and re-evaluate last command
`F10`	`shell-command`	Execute command
`Shift-F10`	`suspend-emacs`	Call DOS command interpreter
`Ctrl-F10`	`save-buffers-kill-emacs`	Quit Emacs

Summary:

	F.1	F.2	F.3	F.4	F.5	F.6	F.7	F.8	F.9	F.10
	help	nextW	delOW	splitWH	splitWV	copy	goto	read	dired	command
s	mode	switchW	delW	largeWH	largeWV	kill	search	save	make	suspend
c	key	buffers	scrlOW	shrnkWH	shrnkWV	yank	replace	write	repeat	quit

If you want to append and/or change the function, use lisp variable `term-setup-hook`. See '`lisp/term/ibmpc.el`'.

If you don't need cursor and function keys, append next line to '$*HOME*`/_emacs`':

```
(setq term-file-prefix nil)
```

Appendix A. Terminal Specified Features

If your machine is IBM PC or its compatibles, set environment variable *TERM* with `ibmpc`, then you can use cursor keys and functon keys.

The functions which is assigned to those keys are as follows:

`PgUp`	`scroll-up`	Scroll text upward by page
`PgDn`	`scroll-down`	Scroll text downward by page
`Insert`	`set-mark-command`	Set mark current cursor position
`Del`	`delete-backward`	Delete previous character
`UpArrow`	`previous-line`	Move cursor vertically up
`LeftArrow`	`backward-char`	Move cursor left
`RightArrow`	`forward-char`	Move cursor right
`DownArrow`	`next-line`	Move cursor vertically down
`Home`	`beginning-of-buffer`	Move cursor beginning of buffer
`End`	`end-of-buffer`	Move cursor end of buffer

`F1`	`help-for-help`	Show help for help
`Shift-F1`	`describe-mode`	Show help of current mode
`Ctrl-F1`	`describe-key`	Show help of key
`F2`	`other-window`	Select different window
`Shift-F2`	`switch-to-buffer`	Select buffer by name
`Ctrl-F2`	`list-buffers`	Display a list of buffers
`F3`	`delete-other-windows`	Make current window fill the screen
`Shift-F3`	`delete-window`	Remove current window
`Ctrl-F3`	`scroll-other-window`	Scroll text of other window upward by page
`F4`	`split-window-vertically`	Split current window vertically
`Shift-F4`	`enlarge-window`	Make current window bigger
`Ctrl-F4`	`shrink-window`	Make current window smaller
`F5`	`split-window-horizontally`	Split current window horizontally
`Shift-F5`	`enlarge-window-horizontally`	Make current window wider
`Ctrl-F5`	`shrink-window-horizontally`	Make current window narrower
`F6`	`copy-region`	Copy from mark till cursor to yank-buffer
`Shift-F6`	`kill-region`	Cut from mark till cursor to yank-buffer
`Ctrl-F6`	`yank`	Paste yank-buffer to current cursor position
`F7`	`goto-line`	Goto line
`Shift-F7`	`isearch-forward`	Do incremental search forward
`Ctrl-F7`	`query-replace`	Do query replace
`F8`	`find-file`	Edit file

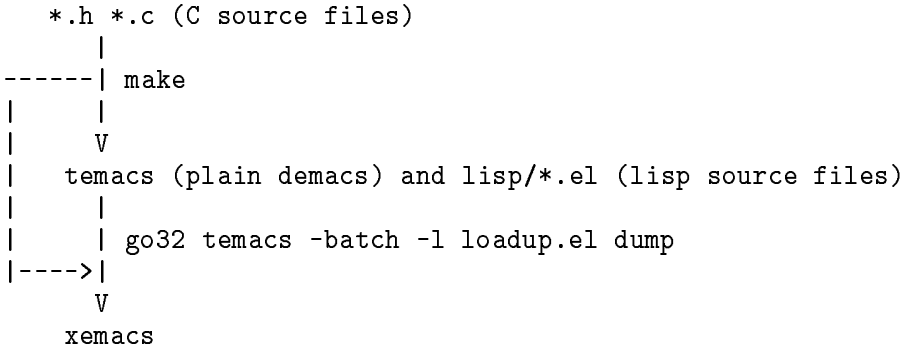
6. Compilation

If you wish to recompile Demacs, 'GJ's GCC/G++' (GNU C compiler for DOS) and original Emacs source are required.

Diffs to the original Emacs source files are included in '`src\diffs.lzh`'. Unpack the archive. The files in '`diffs.lzh`' are not `diff` form but they are full source files, so you need not to apply patch. You may need to alter makefile.

Then run '`make -f xmakefile all`' command. Make generates `temacs` and `demacs`. Caution! just `make` will destroy '`xmakefile`'. I prepared '`xmake.bak`' for you.

Note that all modifications are embeded by `#ifdef MSDOS` and `#endif MSDOS`, so you may follow my modifications easily. Machine specified codes are embedded in similar flags.



Provided '`bin/temacs`' enables you to reconfigure demacs without recompilation. After editing lisp sources which are loaded up by '`loadup.el`', run '`go32 temacs -batch -l loadup.el dump`'.

5. Restrictions

- Demacs does not support asyncronous process. Therefore you can not use standard '`compile.el`', '`shell-mode.el`' and etc. I think it's too painful, so, customized '`compiled.el`' and '`dired.el`' are provided. See lisp directory.
- `Ctrl-G` can't generate inturrupt signal. So you can't quit lisp-form like this:

  ```
  (while t ())
  ```

  You shoud use the following form instead of it.

  ```
  (while (not (input-pending-p)) ())
  ```

- Demacs backup filename convension is following.

  Base backup filename is made from following rule.

  ```
  foo      ->  foo.~
  foo.c    ->  foo.c~
  foo.tex  ->  foo.te~
  ```

  Since DOS filename rule (8.3) is too tight, numeric backup such like '`foo.c.~2~`' can not be used.

  Similarly auto save filename is made from following rule.

  ```
  abcdefgh ->  #abcdefg.#
  foo.c    ->  #foo.c#
  foo.tex  ->  #foo.te#
  ```

  Similarly, auto-save file name is created.

- If you don't configure XMS or VCPI driver and invoke Demacs with `-keep` option, child process Demacs (or go32 application) will *not* work correctly.
- If you use VCPI manager and you have less extended memory than 300KB, Demacs will hang up or being to work *VERY* slowly. Plese give Demacs much memory if you met this.

If you have '`ls.exe`', append

```
(setq dired-load-hook
      '(lambda ()
         (setq dired-ls-program "ls")))
```

to your startup file. If you use Demacs with `-keep` option, this configration works faster than former one.

You can get more infomation from '`dired.README`' and its document string.

4.4.1 IBM PC

If you use IBM PC, evaluate `(setq dos-machine-type 'ibmpc)`, then `Ctrl-SPACE` will generate null code and `meta-flag` will works.

4.4.2 Toshiba J-3100

If you have J-3100, evaluate `(setq dos-machine-type 'j3100)`, then following functions will be abailable in addition to IBM PC features.

```
set-cursor-mode
get-screen-mode
set-cursor-mode
set-keyclick
set-screen-mode
```

See '`lisp/dos-fns.el`' for more information.

4.4.3 NEC PC-9801, EPSON PC-386

If you have PC-9801 or EPSON PC-386, evaluate `(setq dos-machine-type 'pc98)`, then following functions will be abailable.

```
pc98-assign-special-key
pc98-cancel-special-key
```

See '`src/dosfns.c`' for more information.

4.5 Other Demacs Features

1. Filename Completion

   Demacs provides filename completion feature. Filename may include drive name in its front. For example, we assume the current directory is '`c:/tools/emacs`' and there is a file '`d:/config.sys`', under this condition `C-x C-f` invokes `find-file` function then the function prompts like

   ```
   Find file: c:/tools/emacs/
   ```

   at this point your typing of `d:/confi` and `TAB` key causes:

   ```
   a:/tools/emacs/d:confi^I  -> d:/config.sys [sole complete]
   ```

2. Child Process

   Function `suspend-emacs` calls sub shell. Try `C-z` and a new command shell is invoked.

   Function `call-process` calls sub shell like `suspend-emacs` but its stdout is redirected to a temporary file and it will be displayed after completion of the process. Try `M-!`. But don't call programs which require input from keyboard, because Demacs can't response to them.

3. Enhanced Dired

   This package includes enhanced dired-mode by Sebastian Kremer and their dos ported codes which works without UNIX compatible '`ls.exe`'.

   To use this, extract '`edired.lzh`' to your lisp directory and put lines

   ```
   (setq dired-mode-map nil)
   (load "emacs-19")
   (autoload 'dired "dired" nil t)
   ```

   to your startup file ('`direddos.elc`' is loaded from '`dired.elc`').

```
  int86 (0x21 /* 33 */, &regs, &regs);
  return regs.h.al;
}
```

C funtion to get current disk number may write in Demacs as follows:

```
(defun get-disk ()
  (let ((regs (make-register)))
    (set-register-value regs 'ah 25)    ; 0x19
    (int86 33 regs)                     ; 0x21
    (register-value regs 'al)))
```

make-register Function

Generate instance of register type which is passwd to `int86` function.

register-value *REGISTER NAME* Function

Get the value of *REGISTER*'s *NAME*. *NAME* is one of followings.

`'ax`	`ax` register
`'bx`	`bx` register
`'cx`	`cx` register
`'dx`	`dx` register
`'si`	`si` register
`'di`	`di` register
`'cflag`	carry flag
`'flags`	flag register

or

`'al`	lower byte of `ax` register
`'ah`	upper byte of `ax` register
`'bl`	lower byte of `bx` register
`'bh`	upper byte of `bx` register
`'cl`	lower byte of `cx` register
`'ch`	upper byte of `cx` register
`'dl`	lower byte of `dx` register
`'dh`	upper byte of `dx` register

set-register-value *REGISTER NAME VALUE* Function

Set *REGISTER*'s *NAME* to *VALUE*. *VALUE* is **unsigned integer**.

int86 *INTNO REGISTER* Function

Generate 8086 software interrupt of number *INTNO* with *REGISTER*. This returns set of register value.

4.4 Machine Specified Features

New lisp variable `dos-machine-type` is introduced for support various machines.

dos-machine-type Global Variable

Set one of `ibmpc`, `j3100`, `pc98` to select machine specific code.

4.1.3.3 Creating Files and Visiting

The `find-file-not-found-set-file-type` function is appended to the `find-file-not-found-hooks` variable. This means that `find-file-not-found-set-file-type` is called when the visiting file not exists.

This function calls internally the `find-file-type-from-file-name` function with **target name** as its argument and set return value to file type of the buffer.

4.1.4 Writing Buffers into Files

When writing a buffer into a file, Demacs set translation mode to file type of the buffer.

4.1.5 Process I/O

Demacs communicate with external process (child process) using temporary file. The `default-file-type` variable is used as file type of the buffer which displays result.

4.2 "8bit clean" Display Mode

Demacs supports so called "8bit clean" display mode, which displays 8bit dirty characters as it is instead of backslashed or arrowed form.

If you want use this mode, execute command `toggle-pass8-mode`. You can find in mode line whether the buffer is in "8bit clean" display mode.

```
--**-Emacs: foo.bar      (T:Fundamental Pass8)--28%---------------
                                       ^^^^^
```

To return ordinary display mode, again execute command `toggle-pass8-mode`.

If you want create a buffer always in "8bit clean" display mode, place

```
(setq-default pass8-mode t)
```

in your startup file.

pass8-mode Local Variable

If this variable is `*Non-nil`, Demacs displays 8bit dirty characters as it is. Default value is `nil`.

toggle-pass8-mode Command

Set/Unset the buffer local variable `pass8-mode`.

4.3 8086 Software Interrupt

Function `int86` generate 8086 software interrupt. DOS Extender `demacs.exe` not handle all interrupts, so some interrupts are treated as exception and may cause violation. Use this carefully.

Use this function like this.

```
int
GetDisk ()
{
  union REGS regs;
  regs.h.ah = 0x19;     /* 25 */
```

`1` or `'binary` or `"binary"`
: Specify the buffer's file type to binary mode.

default-file-type Global Variable
: The value of this grobal bariable is the default value of buffer-local `file-type` variable.

set-default-file-type *TYPE* Command
: This function sets the value of `default-file-type` variable to *TYPE*. The value of *TYPE* is one of followings.

`0` or `'text` or `"text"`
: Specify the buffer's file type to text mode, by default.

`1` or `'binary` or `"binary"`
: Specify the buffer's file type to binary mode, by default.

4.1.2.2 Buffer Creation

When creating a new buffer, Demacs set the buffer's local variable `file-type` to the value of the `default-file-type` variable.

4.1.3 Reading Files into Demacs Buffers

Using the `define-file-name-file-type` function, you can define file type associated with file name.

For example,

```
(define-file-name-file-type "\\.mem$" 'binary)
```

defines that file type of files having extention '`.mem`' is binary.

By default, the files ending with '`.elc`', '`.obj`', '`.exe`', '`.com`', '`.lib`', '`.sys`' (except '`config.sys`'), '`.chk`', '`.o`', '`.a`' and '`.out`' are defined as binary file type. This is defined in '`demacs.el`'.

define-file-name-file-type *FILENAME TYPE* Function
: This function defines file type associate with file name. *FILENAME* is regular expression or `nil`. `nil` matches any file anme. *TYPE* is file type.

find-file-type-from-file-name *FILENAME* Function
: This function returns file type which associate with *FILENAME* defined by `define-file-name-file-type` function. If no file type is defined, this returns a value of `default-file-type`.

4.1.3.1 Inserting Files into Buffers

When inserting a file already visiting buffer, Demacs calls the `find-file-type-from-file-name` function with **file name of target file** as its argument, and set return value to file type of the buffer.

4.1.3.2 Visiting Files and Reading

When visiting a file and reading into a buffer, Demacs calls the `find-file-type-from-file-name` function with **file name of target file** as its argument, and set return value to file type of the buffer.

4. Features

4.1 File Type: Text or Binary Translation

4.1.1 Translation Mode

On MS-DOS file system, a line is ended with `CR` (`0x0d`)/`LF` (`0x0a`) characters and a file is ended with `Ctrl-Z` character. But on UNIX file system, a line is ended with only `LF` character, and end of data meas end of file.

Treating "binary files" which contains `Ctrl-Z` characters and for compatibility to UNIX file system, C I/O library on MS-DOS has following two translation mode which control these characters.

- Text mode translation
- Binary mode translation

With **text mode translation**, on reading, `CR`/`LF` code are translated to `CR` and reading is terminated as soon as `Ctrl-Z` appears. On writing, `LF` is added to `CR` and `Ctrl-Z` is added to end of file.

With **binary mode translation**, no code is translated.

Demacs supports these two translation mode, and manage them on each buffer. You can find current translation mode of the buffer from file type mnemonic on the mode line.

```
--**-Emacs: demacs.tex   (T:Texinfo)--42%---------------
                          ^
```

File type mnemonic means

`T` Text mode translation.

`B` Binary mode translation.

We call this translation mode which is managed on each buffer "**file type**".

4.1.2 Demacs Buffer Management

4.1.2.1 File Type of Each Buffer

File type of each buffer is stored on buffer-local `file-type` variable. To set the value of `file-type` use the `set-file-type` function.

The default value of `file-type` is the value of `default-file-type` variable. To set its value, use the `set-default-file-type` function.

file-type Local Variable

`0` Text mode translation.

`1` Binary mode translation.

set-file-type *TYPE* &optional *BUFFER* Command

This function sets buffer-local `file-type` variable of *BUFFER* to *TYPE*. The argument *BUFFER* defaults to the current buffer. The value of *TYPE* is one of followings.

`0` or `'text` or `"text"`

Specify the buffer's file type to text mode.

In order to eliminate this swapping time, our `demacs.exe` can take `-keep` option. `-keep` option requests to `demacs.exe` NOT to swap out contents of extended memory (conventional memory (less than 640KB) is always swapped out).

In case you use **XMS** driver such as '`himem.sys`', since `demacs.exe` obtains all of free extended memory at startup, a child process can not use any extended memory (Remember GCC/G++ requires extended memory). You can reserve extended memory for the child process by following *num* to `-keep`, e.g. '`-keep` *1024*'. *num* which is killo byte unit number limits amount of usage of extended memory by `demacs.exe`. If less amount of extended memory is available than *num* at startup, `demacs.exe` uses all of extended memory.

Otherwise in case you use **VCPI** driver such as `QEMM` or `EMM386`, `demacs.exe` gets extended memory on demand basis. '`-keep` *num*' limits amount of extended memory used by `demacs.exe` to *num* killo bytes.

The following table shows a brief summary.

	swap out extended memory at fork child	extended memory for child
`without -keep`	YES	all of extended memory at startup
`-keep`	NO	nothing
`-keep` *num*	NO	extended memory at startup − *num*

Too small *num* makes Demacs slower. I recommend to set *num* at least 512.

Example (Demacs uses 1024KB extended memory):

```
demacs.exe -keep 1024
```

num can be also specified by *DEMACS_KEEP* environment variable. If both environment variable and option are specified, option has priority. If you don't need to specify *num* with `-keep` option, specify the value `0` with this environment variable.

3.3 Options of Demacs

Type `demacs [-keep [`*num*`]] [options]` to invoke Demacs. If you need more information, see "GNU Emacs manual".

3. Invocation

3.1 Environment Variables

`C>set GO32TMP=c:/tmp`
: directory where swapped files are placed (see DJGCC's readme).

`C>set TERM=ibmpc`
: entry name of '`termcap`'.

`C>set TERMCAP=c:/etc/termcap`
: directory where '`termcap`' file are placed.

`C>set HOME=c:/home`
: directory where startup file '`_emacs`' are placed.

`C>set TZ=JST-9`
: time zone.

`C>set USER=manabu`
: your name up to 8 characters.

`C>set EMACSLOADPATH=d:/emacs/lisp`
: set lisp load path if you don't use standard path '`/lib/eamcs/lisp`'.

`C>set EMACSEXECPATH=d:/emacs/etc`
: set etc load path if you don't use standard path '`/lib/emacs/etc`'.

`C>set DEMACS_KEEP=1024`
: amount of extended memory to keep (detail, see below).

`C>set SHELL=c:\command.com`
: your command interpreter path.

3.2 Options of DOS Extender `demacs.exe`

`-keep [`*num*`]`
: When `demacs.exe` invokes a child process, `demacs.exe` does not swap out contents of the extended memory to a disk. If *num* is specified, `demacs.exe` uses up to *num* killo bytes of extended memory.

  You should use this option with XMS or VCPI memory manager. If not, may cause violation.

More about `-keep` opiton:

`demacs.exe` provides virtual memory mechanism. If an application (e.g. Demacs) requires more memory than equipped real memory, `demacs.exe` write out a portion of real memory to a disk in order to make a free memory, then `demacs.exe` assigns the new free memory to the application. This is called "swapping".

Swapping does not only happen when real memory shortage, but also happens when `demacs.exe` calls a child process. That is, before execution of the child process, `demacs.exe` swapps out all of allocated memory to a disk so that the child process can use memory. Unfortunatly, swapping takes very long time because Demacs is a huge program.

Installation procedure is quite easy.

1. If your system is IBM PC variant (including AX and J-3100 series), copy '`bin/demacspc.exe`' to the directory in *PATH* with its name '`demacs.exe`'.

   If your system is NEC PC-9801 or EPSON PC-386 series, copy '`bin/demacs98.exe`' to the directory in *PATH* with its name '`demacs.exe`'.
2. Copy '`bin\demacs`' in the same directory with '`demacs.exe`'.
3. Create '`lisp`' and '`etc`' directory in '`\lib\emacs`' directory.
4. If you want to use lisp library, get original lisp files from somewhere and copy them in '`\lib\emacs\lisp`' directory. Then unpack '`lisp.lzh`' in '`\lib\emacs`' directory.
5. Unpack '`etc.lzh`' in '`\lib\emacs`' directory.
6. Set *TERM* environment variable for your system (See following example).
7. Set *TERMCAP* variable if '`termcap`' file is not stay under '`\etc`'.
8. Prepare your '`_emacs`' ('`.emacs`' on Unix system) under your *HOME* directory if you need it.

2.3 Sample Session to Install

```
C> copy bin\demacspc.exe \tools\bin\demacs.exe (IBM PC, AX, J-3100)
C> copy bin\demacs \tools\bin
     or
C> copy bin\demacs98.exe \tools\bin\demacs.exe (PC-9801, PC-386)
C> copy bin\demacs \tools\bin

C> mkdir \lib\emacs
C> mkdir \lib\emacs\lisp
C> mkdir \lib\emacs\etc
C> copy lisp.lzh \lib\emacs
C> copy etc.lzh \lib\emacs
C> cd \lib\emacs
C> lharc x /mx lisp.lzh
C> lharc x /mx etc.lzh
C> del lisp.lzh
C> del etc.lzh

C> set TERM=ibmpc  (IBM PC)
      or
   set TERM=j3100  (J-3100)
      or
   set TERM=pc98   (PC-9801, PC-386)

C> set TERMCAP=c:\etc\termcap   (IBM PC, J-3100)
     or
   set TERMCAP=a:\etc\termcap   (PC-9801, PC-386)

C> set HOME=c:\home (your startup file '_emacs' is placed)
```

2. Installation

2.1 Contents of This Distribution Kit

The package consists of two archives; '`dem120e.lzh`' and '`dem120s.lzh`'. You need **LHarc** or **LHa** utility to unpack these archives. Don't forget to use `/mx` switches with `-x` option to unpack them. For example do as following.

```
lharc x /mx dem120e.lzh
lharc x /mx dem120s.lzh
```

And similarly apply these options to the archives which are contained by them.

After unpacking, following files will appear.

'`dem120e.lzh`' (executables)

`readme`	this file
`demacs.tex`	texinfo version of `readme`
`demacs.dvi`	compiled `demacs.tex` by TEX
`copying`	GNU General Public Licence
`copying.dj`	DJGCC copyright notice
`termcap`	sample termcap file
`_emacs`	sample startup file
`bin\demacspc.exe`	DOS extender for IBM PC and its compatible machines
`bin\demacs98.exe`	DOS extender for PC-9801, PC-386 series
`bin\demacs`	demacs object file (stripped)
`bin\temacs`	'plain' demacs object file (stripped)
`etc.lzh`	emacs documentation string file and etc.
`lisp.lzh`	includes lisp files which are necessary to run
`edired.lzh`	enhanced dired-mode

'`dem120s.lzh`' (source files)

`src\diffs.lzh`	includes only different files from the original
`lisp\diffs.lzh`	includes only different files from the original
`go32_98\source.lzh`	PC-9801 version go32 DOS extender from DJGCC
`go32_pc\source.lzh`	IBM PC version go32 DOS extender from DJGCC

NOTE:

This package only includes lisp files which are modified for Demacs. Original and/or other useful lisp stuff can be obtained from various FTP sites (for example `ftp.sigmath.osaka-u.ac.jp`) or from:

Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

2.2 How to Install

Demacs requires special version of '`go32.exe`' DOS extender. This package includes it with its name '`bin\demacspc.exe`' for IBM PC or '`bin\demacs98.exe`' for PC-9801 series. You should use one of them with renameing to '`demacs.exe`'.

Since `go32.exe` which came along with the package of DJ's GCC/G++ version 1.05 can not handle `Ctrl-C` properlly, we modified it to work correctly. And we made some changes to be able to handle more DOS function call, and enhanced its memory management strategy (detail, see below).

1. Demacs version 1.2.0

Demacs—386/486 DOS version of GNU Emacs.

1.1 Current Version

Demacs is currently based on GNU Emacs version **18.55** (**partly 18.57**).

Demacs version itself is **1.2.0** (**1991/12/12**).

demacs-version Command

Return string describing the version of Demacs that is running.

1.2 Platforms

`HARDWARE` **386/486 based DOS machines**—IBM PC and its compatible, AX, J-3100, or NEC PC-9801, High resolusion PC-98 and its compatible EPSON PC-386

`SOFTWARE` **MSDOS 3.0 or later**

XMS manager (‘`HIMEM.SYS`’) and **VCPI** memory manager (‘`EMM386.EXE`’, ‘`QEMM`’, etc.) compatible.

Demacs does not support **DPMI**, thus it can not be used with extended mode on Windows.

1.3 Bug Report

When you have troubles with using Demacs, please follow the following steps before inquire us.

First, read this README and the online-manuals (with Info Browser) to find answers. Next, consult to Emacs experts around you or on the networks. In case you decide to send a mail to the authors, you must read "Bugs" section in the Emacs manual before sending the mail.

Suggestions and bug fixes are welcome with next address.

`demacs-bugs@sigmath.osaka-u.ac.jp`

If you want to know current status about Demacs, send empty mail to next address.

`demacs@sigmath.osaka-u.ac.jp`

We will reply your mail with latest information, automatically.

Special Thanks to

Morgan Schweers
Shigeki Yoshida
Junn Ohta
Toshihiko Shimokawa
Makoto Matsushita

Copyright © 1991 HIRANO Satoshi, Halca Computer Science Laboratory
Copyright © 1991 Manabu Higashida

Demacs is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 1 as published by the Free Software Foundation.

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 (this package includes it as file '`copying`') for more details.

Using Demacs

A Guide to the 386/486 DOS version of GNU Emacs

Version 1.2.0
December 12, 1991

Manabu Higshida
manabu@sigmath.osaka-u.ac.jp
Osaka University, JAPAN

and

HIRANO Satoshi
hirano@tkl.iis.u-tokyo.ac.jp
University of Tokyo, Japan