"You can never be too rich, too thin, or
have too much RAM and Disk storage"[1]

Terse, Lean, Slim, Zair – Small is Beautiful

Joseph (Yossi) Gil
Department of Computer Science
Technion—Israel Institute of Technology
Technion City, Haifa 32000
ISRAEL

February 8, 1994

[1]Ancient American proverb, CA. 1980 As quoted by the *QEDIT* manual.)

From the manual:

Terse is a tiny (only 4096 bytes) but amazingly powerful full-screen editor for files of up to 64K in length. Terse runs on all PC compatible machines. Its command keys are very similar to those of the famous BRIEF editor (by UnderWare Inc.). Terse can edit both UNIX and MS-DOS style text files as well as binary files. No hacker's disk is complete without it. No disk, be it hard or floppy, is too full to include it.

...

Terse is free (but copyrighted) to private users. However, if you use Terse for any commercial application, i.e., in your company, at work, etc., or make money from selling software or distributing shareware, you are *required* to register your copy of Terse. Registration cost is only $15 and will buy you a personalized version, configuration utilities and grant your institution a site license. For an additional payment of only $5, you will get the source code of Terse.

Abstract

Terse is a powerful full-screen editor running on the IBM-PC. Terse code size is only 4096 bytes, yet it features many capabilities one would expect only from "big editors". These include unlimited line length, search and search and replace, cut, copy and paste, help screen and status line, configurable key bindings, variable tab setting and more. Terse can edit both UNIX and MS-DOS style text files as well as binary files. Some more advanced features, such as applying an external filter to a text block are included with an extended (in the 5KB range) version of Terse called Slim. Zair is a bidirectional version of Slim intended for entering mixed right-to-left and left-to-right text such as input for TEX--XET.

Contents

Chapter 1

Introduction

Terse is an amazingly powerful full-screen editor running on all IBM-PC compatible computers under the MS-DOS operating system. Terse is a very tiny editor. The entire program occupies only 4096 bytes of disk space. No hacker's disk is complete without it. No disk, be it hard or floppy, is too full to include it.

A conscious effort was made to reduce the program size to an absolute minimum. The code was written in 8086 assembly using a style that breaks many of the rules of structure programming. This space economy does not come on expense of features or of power. Full screen mode-less editing is offered to the user with the usual cursor movements, insertion, overwriting and deletion of text and prime editing operations: search and search and replace, cut, copy and paste. In addition, Terse provides advanced features including unlimited line length and horizontal scrolling, help screen and status line, configurable key bindings and variable tab setting, support for many screen types, editing of binary and UNIX and MS-DOS style text files and more. Terse default command keys binding are very similar to those of the famous *brief* editor (by UnderWare Inc.).

Terse main weakness is that it is restricted to editing one file viewed from one window at at time. This deficiency can be circumvented by using a multi-tasking windowing environment such as DESQview. Indeed, one may argue that it is the responsibility of such an external shell to furnish a good and consistent windowing and multi-file operation support. An editor should then concentrate on carrying out a a coherent duty of file editing.

Yet another deficiency Terse is that it is limits file size to 65,536 bytes. This limitation follows from the segmented architecture of the 8086, its restricted address space and the lack of virtual memory or any other sophisticated memory management in the MS-DOS operating system.

Despite the numerous dirty assembly tricks employed in the implementation, Terse code is compatible, re-usable and extendable. Terse works flawlessly on all MS-DOS/PC-DOS versions greater or equal to 2, and all MS-DOS clones and replacements known to the author including Windows DOS shell, 4DOS, DR-DOS, DESQview and OS/2 DOS compatibility box. In addition there exists a version of Terse for the HP95lx palmtop. This version is especially adapted to some of this computer size limitations.

An extended (in the 5KB range) version of Terse called Slim has even more editing functions. One particularly interesting function allows the user to apply an arbitrary external filter to a text block. A scaled down (in the 3KB range) version of Terse Lean is basically Terse without the editing code. Naturally, Lean could never compete with the excellent LIST utility of Vernon D. Buerg, and other fine tiny file-listers. However so, Lean contributes to the purpose of demonstrating the reusability of the editor code.

The editor display engine was modified in Zair which is a bidirectional version of Slim intended for entering mixed right-to-left and left-to-right text such as input for TEX--XET.

Terse is free (but copyrighted) to private users. If you are a private user who uses Terse "for pleasure", you are not required to register. However, if you use Terse for any commercial application, i.e., in your company, at work, etc., or make money from selling software or distributing shareware, you are *required* to register your copy of Terse. Registration cost is only $15 and buys you a personalized version, and grant your institution a site license. For an additional payment of only $5, you will receive the source code of Terse.

Disclaimer Terse and Slim are written in highly optimized assembly language. The code is probably the worst spaghetti you have ever seen! This means that the risk of Terse and Slim having bugs is especially high. The author exterminated all the bugs he identified, and he did his very best to test the program thoroughly. However, there might still be other unidentified bugs lurking there. These bugs by nature are unpredictable. Usage of the program acknowledges the following disclaimer of warranty: "This program is supplied as is!" All warranties, expressed or implied, including, without limitation, the fitness of this program for any purpose, are hereby explicitly disclaimed.

Chapter 2

In Depth

2.1 Highlights

- Very small: fits in 4096 bytes = 4 clusters on 5.25"/3.5" floppies.
- Minimal free memory requirements: 137KB.
- Uses current screen size, including the most common $80x25$ screen dimensions but also unusual sizes such as 40×16, 40×25, 40×40, 40×50, 132×25, 132×43, 96×33, 80×30, 80×33, etc.
- Fully DESQview (version 2.0 and above) aware.
- Edit binary files. Terse is ideal for a quick and dirty job of editing and patching strings in small programs.
- Supports editing of UNIX style files (lines terminated with '`\n`').
- Supports both '`/`' and '`\`' as directory separator.
- Edit lines of unlimited length (limited only by file size).
- Automatic unlimited horizontal scrolling.
- *Brief*-like keyboard configuration.
- Support for tabs, and configurable tab positions.
- Prevents snow when run on a CGA screen.
- Cut, Copy and Paste commands.
- Search, Search again, and Translate commands.

- Safe: confirm exits when text is not saved.
- Edit files whose name contains spaces and other weird names.
- Extended version (the Slim editor) of size~5KB with many more features, including support (via external filters) for word count, sorting, character codes translations, quoting, tab expansion and un-expansion, and other complex editing tasks.
- Status line indicating: current column, current line, horizontal scrolling, total number of lines, file name, modified status, insert/overwrite status, the current offset in file and total file size in bytes are displayed in the status line. The extended version also displays the hexadecimal value of the current char.

2.2 Requirements

- PC-DOS/MS-DOS version 2.0 or higher
- 137KB free memory
- 64KB free disk space

2.3 Problems, Questions, Suggestions?

Please do not hesitate to communicate all your thoughts and comments. I answer all my E-mail usually in less than 24 hours. If you do not get a response within a reasonable time, it is probably because I didn't get your message, or because I wasn't able to reply. Kindly try sending again using another route, and include more alternate return paths.

Any comments you may care to make are valuable to me. The editors and the included software would not have been what they are today without all of the generous aid users provided: requests, complaints, thoughts and even gripe are always welcomed.

2.4 Compatibility

Terse uses only pure 8088 instructions and assumes the bare minimum of the underlying operating system. It would run on your good old original 4.77MHz IBM-PC running under DOS 2.0 as well as on a fast 80486 66MHz machines running DR DOS 6.0 + 4DOS 5.0, or OS/2 in DOS compatibility mode. More than once during the development, I had a chance to save a few bytes by using a more advanced instruction, or by assuming a more advanced operating system version. I managed to resist this temptation. Other than that every dirty assembly trick I could think of was used. It is conceivable that some of the tricks could prevent Terse from running on some DOS-like environments and simulators, although I don't know of any concrete example.

2.5 UNIX style files

Terse edits UNIX style files directly. That is, Terse uses both '`\n`' (aka ⟨`LF`⟩, aka Line-Feed) and a '`\r\n`' (aka CR-LF, aka Carriage-return + Line-feed) combo for line ends. Hitting Enter puts '`\r\n`' into the file, and Ctrl-Enter puts '`\n`'. (Configuration utilities allow swapping these 2 keys.) Thus, you can transfer UNIX text files to your PC, edit them on the PC, and then transfer them back without need for file conversion.

2.6 Binary files editing

Terse can edit binary files as well. To aid in this task, the status line displays the file size in bytes and the current offset in bytes in the file. The status line of the Slim editor displays the hexadecimal value of the current char. (This feature may be added in the future, space permitting to Terse.) A difficulty in editing binary files is that the following characters: space (ASCII 32), Null (ASCII 0), and Meta-Del (ASCII 255) all look the same. If you have an EGA or VGA screen you may want to use the font loaders WITHFONT or LOADFONT together with the binary editing font and circumvent this difficulty.

2.7 Configuration Utilities

The configuration utilities allow you to: swap the Enter and Ctrl-Enter keys, change the tab size to 2/4/8/16 (I personally prefer tabs set every 4 characters), and modify the key assignments (with some minor restrictions). In addition, you can eliminate the snow check done automatically on CGA screens, and you get better support (as described above) for editing binary files on EGA/VGA screens.

2.8 Future

The main constraint in adding features to Terse is that its total size must be exactly 4096 bytes. All additions are dependent on my ability to squeeze the current code further. I usually work several hours, and sometimes days, for each byte squeezed. And this task is getting harder and harder all the time. (A few versions had a slack of very few bytes. In these versions, a few dummy bytes were appended to the program to keep its size 4096 bytes exactly.)

My current efforts are concentrated on the developing of the Terse library. This is an object code program which could be invoked from a C or any other high level language.

2.9 How to obtain Terse

Terse is available from Simtel, Garbo, Wustl, their mirrors and from other fine FTP archive sites. Some of these sites have mail-servers for those who do not have access to FTP. Some of these sites carry only an abridged distribution.

In addition, the latest version of Terse is regularly deposited in

`FTP.Technion.AC.IL:/pub/supported/cs/msdos/ters[ XXXaf].zip`

Where XXX denotes the three digit version number, and the letter a or f denotes the version type: abridged or full. For example, the full distribution of version 1.42 should be in a file named trs142f.zip.

If you are experiencing problems downloading these files (`FTP.Technion.AC.IL` is picky for no real reason sometimes), or if you don't have FTP access, please drop me a line and I will e-mail to you whichever distribution you are interested in. The full distribution is sent out in 5 parts of about 1000 uuencoded lines each. The abridged distribution (basically just the Terse editor and the manual) goes in one part of about 600 uuencoded lines. You can use the UUDECODE program enclosed with this archive to decode this mail.

2.10 Genealogy

Terse is an exercise of editor writing inspired by TED, the famous *T*iny *ED*itor. TED was written by Tom Kihlken and published on the November 1988 issue of PC Magazine (Copyright 1988 Ziff Communications Co.).

Some features of Terse are borrowed from the fine descendants of TED called TED+ (apparently also written by Tom Kihlken in November 1988) and TED2 (written by James E. Galbraith, 1201 Chase St., Novato CA, 94945, USA).

There is not a single piece of code in Terse which is the same as in TED, TEDPLUS or TED2. Terse represents almost three years of work on space optimization of the code, and on added functionality. Some of the newly added features include help screen, unlimited line length, space optimization of the code, and on added functionality. Some of the newly added features include the help screen, unlimited line length, status line, smooth horizontal scrolling, previous and next word, UNIX files editing, search and replace, safety features and support for insertion of all 256 characters of the IBM-PC extended ASCII set (including ⟨`Nul`⟩, ⟨`BS`⟩, ⟨`CR`⟩, and ⟨`LF`⟩). Some old features include scrolling, cut, copy, paste, and printing operations.

Some of the documentation for Terse is based on the documentation provided with TED2. See the article "The tiniest editor you'll ever need" by Tom Kihlken, in the November 15, 1988 issue of PC Magazine for further details concerning TED, TED, TEDPLUS and TED2 are available from SIMTEL20:

- TED
  `PD1:⟨MSDOS.PCMAG⟩VOL7N19.ARC`
- TED+
  `PD1:⟨MSDOS.EDITOR⟩TEDPLUS.ARC`
- TED2
  `PD1:⟨MSDOS.EDITOR⟩TED2.ZIP`

Terse is not related in any way to TED3.

Chapter 3

License and Copying Policy

The following policy is intended to promote free private use and distribution, and to collect money from those who can afford to pay for this software. In particular, those institutions which make money from private users are requested to pass on part of their income to whoever is responsible for generating it.

3.1 Private Users:

You are free to use, copy and distribute Terse for non-commercial private use provided that:

1. No fee is charged for use, copying or distribution.
2. It is not modified in any way.
3. It is distributed as a package. All files in this package must be included, none should be added, and none should be changed.

Thus if you are a private user, you can *use* Terse freely, and you are under no obligation to register. (The only restriction is on its further distribution.)

3.2 Registration (Private users)

If you find Terse a useful, easy, and nifty editor you may show your appreciation in one of two ways:

1. Send the author the $15 US dollars registration fee. Terse, the configuration utilities and, if you wish, a personalized version. In addition the author will do his best to update them as soon as a new version comes out.

Please, do register if you can afford it, and if you would like to support financially the production of this high quality software.

2. Devote some time and effort, beyond the usual media service, to learn about Israel and its problems. You may wish to download some information files from the `Israel.nysernet.ORG` FTP site or contact the author for directions on gaining access to other electronically available information.

3.3 Non-Private Use

Institutions and companies *must/* register this package after an evaluation period of 21 days. Registration of one copy per company site is sufficient.

3.4 Non-Private Distribution (Shareware distributors)

You may distribute this package as long as no more than a nominal fee (up to $10) per disk is collected per disk.

3.5 No-Fee Distribution

Simtel, Garbo and other public access anonymous FTP servers: Obviously, you may store this package and offer it to your users free of any restrictions.

3.6 Author's Address

E-mail: (try addresses in this order):

1. `yogi@cs.Technion.AC.IL`
   Technion – Israel's institute of technology, Haifa, Israel.

2. `yogi\@cs.UBC.CA`
   University of British Columbia, Vancouver, British Columbia, Canada.

3. `yogi\@UMIACS.UMD.EDU`
   University of Maryland, Maryland, USA.

4. Permanent mailing address is: Joseph Gil, P.O. Box 3148, Jerusalem, Israel. This is guaranteed to arrive, but may incur a long delay.

3.7 License for Terse source code

If you choose to register the the source code of Terse, you will be abided by the following:

> This is a legal agreement which allows you, the end user, to use Terse source under certain terms and conditions.
>
> Satisfaction Guarantee. If you are dissatisfied with Terse source code or with its licensing terms, you may return it at any time up to 90 days after purchase for a refund. If you decide to do so, you should erase all copies of the source code you have made and destroy all printouts. With your return, you should enclose a signed handwritten statement, indicating that yo have done so. Refunds will be based on the price you paid, shipping costs excluded.
>
> Registering Terse source codes grants you the right to make copies of it necessary for normal backup purposes. You agree not to install and use this source code on more than one computer at a time.
>
> You agree not to give this source in full or in part to others for any purpose whatsoever. This means that you may not sell, give away or allow access to others (by means of negligence or any other form) copies of the source code or any part of it in any way, including bundling or re-selling with your own software, or placing such a copy of the source code in any disk library, shareware distribution service, bulletin board, usenet or any other electronic service.
>
> You may modify the source program for creating editors or any other programs for your own personal usage. You may not distribute programms or other products which are based on or use Terse source or parts of without a written permit from the author of the source (Yossi Gil).

Chapter 4

User Manual

Synopsis

`t` [filename]

To invoke Terse, just type `T` from the DOS command line, optionally followed by a filename.

4.1 File

Terse opens and reads a file whose name (and path, if required) is initially supplied on the command line. If no file name is provided, Terse opens a new file. A name may be supplied later during the editing session using the Output command invoked by $\boxed{\text{Alt O}}$. Terse expects a legal DOS filename, which may include a path name. Terse can also edit files whose names contain spaces. These names are legal, although most DOS commands cannot manipulate them.

While editing, you can save your work by pressing $\boxed{\text{Alt-W}}$. Upon save, the original of the modified file is saved with the extension `.bak`. To exit the editor and save your work press $\boxed{\text{Alt-X}}$.

To exit the editor without saving changes made to the file press $\boxed{\text{Alt-Q}}$. Hitting $\boxed{\text{Alt-Q}}$ causes the Terse to exit immediately if the file was not modified since last saved. Otherwise, Terse presents the following prompt:

```
Lose changes (Y/N)?_
```

Hit `Y` only if you are certain that you don't want recent changes to be saved.

4.2 Basic Editing

A help screen is available upon pressing F1 or Alt-H. When the help screen is displayed, pressing any key returns you to the text.

In Terse, the cursor movement keys (Left, Right, Up, Down, PgUp, PgDn, etc.) allow you to move around in the text. Text is entered in insert mode by default. Text can be deleted using the Del and Back-Space keys.

Pressing the Alt-I key will toggle between the *insert* and *overstrike* modes. The current mode, insert or overwrite, is indicated by a display of an `Ins` or `Ovr` mode indicator in the status line.

Text can be deleted using the Del and Back-Space keys.

The Alt-U key restores up to 255 characters deleted by the Del key or overwritten while in the Overwrite mode. The Alt-U key only restores the last character deleted with the Backspace key.

Terse text buffer is limited to 65535 characters. An attempt to extend the buffer beyond that results in a short beep.

4.3 Lines

Lines may be of any length. Terse recognizes both the UNIX and DOS conventions for line termination, i.e., both the ⟨CR⟩⟨LF⟩ pair and a single ⟨LF⟩ denote end of line. Off screen characters may be viewed by moving the cursor toward them. Lines may be broken by pressing Enter (insert ⟨CR⟩⟨LF⟩ pair) or Ctrl Enter (insert ⟨CR⟩ character) at any point, in either Insert or Overstrike mode. Lines may be joined by pressing Del at the line end or Backspace at the beginning. Pressing Alt-D deletes the entire line and closes the gap. Pressing Alt-K deletes from the cursor position to the end of the present line. Pressing Alt-G restores the most recent line deletion. The line delete buffer remains intact until another line is deleted. The line delete buffer has room for 255 characters.

4.4 Block

A Block is some portion of the text file which has been specifically delineated for later manipulation. One end of a *block* of text is defined by hitting Alt-A key (anchor drop/raise). The other end of the block is where the cursor is located. Move the cursor with the cursor control keys to the block end. As you move the cursor, note that the blocked area is shown in reverse video. The Alt-A is a toggle key, so hitting it again raises the anchor dropped at the block end, and thus cancel the blocking operation.

Pressing the Keypad- key (keypad grey minus key) removes the block to a paste buffer; this operation is known as Cut. The Keypad+ key (keypad grey plus) copies the block to the paste buffer without removing it from the text; this operation is known as Copy. The Ins key copies the text stored in the paste buffer at any point where the cursor is located; this operation is known a Paste. The paste buffer remains intact until another section is marked and cut or copied. The paste buffer has room for 64K bytes.

When a block is marked, pressing Alt-P prints the marked text (directly from the file buffer). After the print, the block remains marked.

4.5 Search and Replace

Terse has a string Search and Find Next functions. When the Alt-S key is pressed, the prompt `Search for:` appears on the bottom line. Enter the desired text string and press the Enter key. If the string is found, it appears blinking in inverse video on the screen. The blinking inverse video is extinguished when any key is pressed. If the string is not found in the area following the cursor, a short beep sounds. To search for the next occurrence of the string, press Alt-N. Terse cannot search backward, and it doesn't recognize regular expressions.

Terse also has a Translate aka Search and Replace function which is invoked by pressing the Alt-T key. You are prompted for the string to search for, and then for the replacement string. The translate function is best explained by an example. To change all occurrences of the acronym `TLA` in your file to the three-word phrase `Three Letters Acronym`, you would move to the beginning of the file (Press Ctrl-PgUp) and then press Alt-T. Terse erases the status line (the last screen line) and displays in its place a prompt:

```
Search for: _
```

(The underscore in the above indicates the cursor position). Type in the word: `TLA`. The Back-Space key can be used for erasing the previous character. The last line should now read:

```
Search for: TLA_
```

Hit Enter. (Hitting ESC at this stage ends the translate operation.) Terse now prompts you for the replacement string in the last screen line:

```
Replace with: _
```

Type in the phrase: `Three Letters Acronym`. The last screen line now reads:

```
Replace with: Three Letters Acronym_
```

Hit Enter. (Again, hitting Esc at this stage ends the translate operation.) Terse searches for the first occurrence of `TLA` in your file, moves the cursor to this point, displays "TLA" in blinking inverse video, and then prompts you in the last screen line:

```
Replace with: Three Letters Acronym [Yes/No/All]?
```

Your response could be to hit `Y` or `y`, which means Yes, make this replacement; or `N` or `n` which means No, do not make this replacement. In both cases, Terse continues by moving to the next occurrence of the string `TLA` and prompts you again. You may also answer the prompt by hitting Esc, which aborts the translate operation. If you wish to replace unconditionally all occurrences of the string `TLA` with `Three Letters Acronym` (be warned that it may take a considerable time for a long file), respond with `A` or `a`. Terse marches through the file and makes these replacements for you.

Search and Search and Replace are by default case insensitive. This means that the search catches all of the following strings: `tla`, `TLA`, `tLa`, etc. Pressing Alt-C changes both Search and Search and Replace to be case sensitive. Pressing Alt-C again returns you to the case insensitive Search and Search and Replace. Note that a short message appears in the status line whenever case sensitivity is toggled. This message is erased, and the normal status line reappears, when you go on with your work.

4.6 Character Set and Binary file editing

Any of the characters in the IBM extended ASCII set can be entered and manipulated by Terse. To enter control characters (ASCII 00-31 and 127), use Ctrl-@, Ctrl-A,..., Ctrl-Z, Ctrl-[, Ctrl- Ctrl-], Ctrl-, Ctrl-_ and Ctrl-Backspace. A more general method is to press the Alt key and, without releasing it, type a character ASCII decimal value on the numeric keypad. The PC-BIOS does not support entry of the ⟨`Nul`⟩ character (00h) in this fashion. Instead, you may enter the ⟨`Nul`⟩ character by hitting the Ctrl @ key.

The ASCII Backspace (BS) code can be entered as Ctrl-H, and the Backspace key deletes the character to the left of the cursor. Similarly, ASCII Carriage-Return (CR) and Line-Feed (LF) character codes can be entered directly as Ctrl-M and Ctrl-J. Terse interprets the ⟨`LF`⟩ character as an end-of-line. A ⟨`CR`⟩ is considered an ordinary text character except when it is immediately followed by a ⟨`LF`⟩, in which case it is considered part of the end-of-line marker. If ⟨`CR`⟩⟨`LF`⟩ are entered together, or ever become adjacent, they cannot be separated. The Enter key inserts an end-of-line marker (CR)(LF).

Terse does not use the CP/M style End-of-file marker, ⟨`Ctrl-Z`⟩, to indicate the last character in a text file. If a ⟨`Ctrl-Z`⟩ character appears in the file, it is shown as is.

4.7 The Screen

Terse configures itself to the display adapter in use, and supports text modes other than the standard 80 columns by 25 rows, e.g., EGA 43 lines and CGA 40 columns mode. PgUp and PgDn scroll the file by the number of rows displayed. Terse does not work properly if the screen is initially in a graphics mode.

Terse does not alter screen attributes or colors. Inverse video is used in the status line and to indicate a *mark*ed block. Terse uses blinking inverse video to indicate that the search string has been found. It automatically handles *de-snow* on a CGA display.

Because Terse makes BIOS calls and writes directly to the screen buffer, it would probably not work properly on MS-DOS computers that are not sufficiently IBM-PC compatible.

4.8 The Status Line

At the bottom of the editing screen is the Status Line. The Status Line constantly displays information regarding the file you are currently editing. The Status Line is shown below with a description of the information displayed.

```
+---------------------------------------------------------------------------+
:F1 Help | Line 137  Col 25    =282  +0  @893  #10324 Ins letter.doc*      :
+---+----------+--------+-------+-----+---+------+-----+-----+------+------+
    :          :        :       :     :   :      :     :     : File modified
    :          :        :       :     :   :      :     :     : since last
    :          :  The current   :     :   :      :     :     : written
    :          :  cursor column :     :   :      :     :     :
    :          :  number        :     :   :      :     : The name of file
    :          :                :     :   :      :     : you are editing
    :     The current           :     :   :      :     :
    :     cursor line           :     :   :      :    +Insert mode is ON
    :     number                :     :   :      :
    :                           :     :   :     +File length in characters
 A reminder that                :     :   :
 Help screen is                 :     :  +Current offset in file
 available upon                 :     :
 pressing the                   :    +The number of columns scrolled off
 \Key{F1} key                       :      to the left of the screen
                                :
                                +The total number of lines in the file
```

Chapter 5

Advanced Features

This chapter reviews some of the advanced features which incorporated into Slim, Terse's big brother.

5.1 Edit Another File

If while editing one file, you need to edit another files, you can do so by pressing $\boxed{\text{Alt-E}}$. If you have saved all changes to the buffer, then Slim will prompt you for the name of the file you wish to edit next. Type in the name of this file and press $\boxed{\text{Enter}}$. Slim will load this file and you can edit it.

If you haven't saved the buffer since you last changed it, Slim will respond to $\boxed{\text{Alt-E}}$ by the following prompt:

```
Lose changes (Y/N)?
```

If you press anything but 'Y' or 'y', Slim will just beep and give a warning message that the file was not saved. If you do press 'Y' or 'y' then all the changes you have made to the buffer since you last saved it are lost. You are then prompted for the name of file you wish to edit next.

5.2 Word wrap

Slim can do automatic word wrap as you type. Upon pressing $\boxed{\text{Alt-M}}$, Slim prompts you for a right margin:

```
Margin:_
```

Type in your preferred right margin setting (72 is pretty common and useful selection). If a right margin is set, then whenever you finish typing a word (i.e., when you hit a Space or Tab key) Slim checks if this margin was exceeded, and if so, the last word typed is wrapped to the next line.

If the right margin is set to 0, then word wrapping is disabled will occur. By default, the right margin is 0. This default can be changed with the sconfig configuration program.

5.3 Swap Cursor and Mark

An easy way to go to the beginning or the end of a marked block the "Swap Cursor and Mark" command. Upon pressing Shift-Tab, the cursor position is swapped with mark position.

5.4 Exchange Marked Block with Paste Buffer

Many users are familiar with the "cut, copy and paste" paradigm for manipulating chunks of texts. In Terse you block a piece of text by dropping a mark (Alt-A key) at its beginning and then moving the cursor to its end. You can then Copy (Keypad + key) the blocked text to a paste buffer, or Cut (Keypad+ key) it to the paste buffer. Later, you can move to a different location and Paste the text, i.e., insert the text in the paste buffer.

Slim adds to these three basic operations yet another one: "Exchange Marked Block with Paste Buffer", or for short Exchange. By default, this command is bound to Keypad* key (keypad grey asterisk). Upon pressing this key, the contents of the currently marked block is exchanged with the current contents of the paste buffer. Hitting Keypad* again effectively undoes this operation.

The Exchange operation should come in handy in situations where you need to shuffle a lot text around. An important application of this operation is discussed next.

5.5 Pump Block Through Filter

The term "filter" is borrowed from UNIX. It refers to a program, which takes input from the standard input, processes it, and then produces output to the standard output. Several filters are supplied with the full distribution. For example, the WC filter produces to the standard output a count of the number of lines, words and characters in the standard input.

The "pump block through filter" command, by default bound to the Alt-F key, runs a filter while supplying the currently marked block as input to the block and capturing the filter's output. When the filter finishes, its output will show as the currently marked block. The previous contents of the marked block are moved to the paste buffer. By pressing Keypad* you can exchange the contents of the block and of the paste buffer, effectively undoing the filter operation.

Thus, to count the number of words in a document, follow these easy steps:

1. Hit Ctrl-PgUp to move to the top of the file.
2. Hit Alt-A to drop an anchor.
3. Hit Ctrl-PgDn to go to the end of the file. The whole file should be marked now.
4. Hit Alt-F to invoke the "pump block through filter" command. Upon doing so, Slim prompts you for the name of the filter

   ```
   Filter: _
   ```

5. Type in `wc -w` and hit Enter. (The `-w` flag tells the command WC to produce word count only)
6. The number of words in the document now shows, while the document itself is erased.
7. Hit Keypad* to restore the documents contents, and erase the word count. Later you can hit Ins to paste the word count wherever you may need it.

The full distribution includes also the nfmt filter, which reformats text using smart line breaking algorithm. All of Terse documentation was formatted using this filter. To reformat a piece of text, I simply mark it (usually the whole paragraph), and invoke the nfmt filter. Some other filter included with the full distribution are the DOS equivalents of the some famous UNIX filters: expand, unexpand, uudecode, and cat. The quote filter is useful in composing e-mail messages. The rot13 filter allows you to decrypt and encrypt spoilers etc.

Many more new features can be easily added to Slim by virtue of this command. A few examples are: date and time stamping, sorting, column summation, character sets conversions, directory contents listings etc. Sophisticated users should probably get copy of a DOS port of the famous UNIX `sed` and `awk` filters, and harness their power to enhance Slim.

The "pump block thru filter" command does not use temporary files, and can be run with even when the disk is full. Programmers may enjoy contemplating how this little wonder was implemented. (If you really cannot figure this out, write to me and ask for a clue...)

The "pump block thru filter" command does captures also the standard error of the executed filter. You can thus run your favourite compiler from within Slim.

5.6 Go to Line

If you use the "pump block thru filter" to invoke a compiler, you may find it useful to be able to find a line by its number. To invoke the "go to line" command press $\boxed{\text{Alt-G}}$. Upon doing so, Slim presents the following prompt:

```
Go to line: _
```

Type in the desired line number and hit $\boxed{\text{Enter}}$.

Chapter 6

Keyboard Summary

The following is a description of all Terse and Slim commands and their key binding. To get a help screen with keyboard usage summary, hit F1 or Alt-H.

6.1 Cursor movements

Left Arrow Moves cursor left one column.

Right Arrow Moves cursor right one column.

Ctrl Left Arrow Moves cursor to the beginning of the next word.

Ctrl Right Arrow Moves cursor to the beginning of previous word.

Home Moves cursor to start of line.

End Moves cursor to end of line.

Up Arrow Moves cursor up one row.

Down Arrow Moves cursor down one row.

PgUp Moves text window up one page.

PgDn Moves text window down one page.

Ctrl PgUp Moves cursor to top of file.

Ctrl PgDn Moves cursor to bottom of file.

6.2 Basic editing operations

Any keyboard char Insert/Overwrite character into file

Tab Insert tab character

Enter Start new line, add ⟨CR⟩⟨LF⟩ pair to file.

Backspace Delete character to left of cursor

Del Delete character under cursor (to right)

Alt-U Undo recent Del operation/Overwritten characters.

Alt-I Toggle between Insert/Overwrite mode (Ins/Ovr)

Alt-K Delete to end of line

Alt-D Delete current line

Alt-Y Restore deleted line (up to 255 characters)

6.3 Block Operations

Alt-A drop/raise Anchor. Block is between anchor and cursor.

Keypad-Plus Copy block to clipboard buffer

Keypad-Minus Delete block to clipboard buffer

Ins Paste from clipboard buffer

Alt-P Print block.

6.4 Search

Alt-S Search for string

Alt-C Toggle search case sensitivity.

Alt-N Find next occurrence of string

Alt-T Search and Translate

6.5 File Operations

$\boxed{\textbf{Alt-Q}}$ Quit edit: no-save exit (asks for confirmation)

$\boxed{\textbf{Alt-X}}$ eXit and save: same as Quit if file name is unknown

$\boxed{\textbf{Alt-O}}$ Change output file name. Supply name if not given.

$\boxed{\textbf{Alt-W}}$ Write the current buffer into its file (if filename known)

$\boxed{\textbf{Alt-R}}$ (in Slim only) Read a file to buffer.

$\boxed{\textbf{Alt-E}}$ (in Slim only) Edit another file.

6.6 Scrolling

$\boxed{\textbf{Alt -}}$ Scroll window horizontally to the left

$\boxed{\textbf{Alt +}}$ Scroll window horizontally to the right

$\boxed{\textbf{Ctrl Home}}$ Scroll window up

$\boxed{\textbf{Ctrl End}}$ Scroll window down

6.7 Miscellaneous

$\boxed{\textbf{Alt-I}}$ Insert/Overwrite toggle.

$\boxed{\textbf{Alt-Z}}$ Push to DOS.

$\boxed{\textbf{F1}}$ Display the help screen.

$\boxed{\textbf{Alt-H}}$ Display the help screen.

$\boxed{\textbf{Alt-M}}$ (in Slim only) Set right margin for word wrap.

$\boxed{\textbf{Alt-G}}$ (in Slim only) Goto line.

6.8 Entering Special Characters

Ctrl @ Enter the [NUL] character

Ctrl A Enter the [SOH] character

... All control characters can be inserted this way. Special control characters are described below

Ctrl C Enter the [ETX] character (DOS abort disabled)

... ...

Ctrl H Enter the [BS] character

Ctrl I Enter the [TAB] character

Ctrl J Enter the [LF] character

... ...

Ctrl M Enter the [CR] character

... ...

Ctrl Z Enter the [SUB] character (No CP/M style EOF)

Ctrl [Enter the [ESC] character

Esc Enter the [ESC] character

Ctrl Enter the [FS] character

Ctrl] Enter the [GS] character

Ctrl Enter the [RS] character

Ctrl _ Enter the [US] character

Ctrl Backspace Enter the [DEL] character

Chapter 7

Version History

7.1 Version 1.5

1. Cleanup and reorganization. Moved all filters to a separate archives.
2. Documentation is updated now and is in LaTeX format.
3. Added a configuration option to disable the creation of backup files.
4. Slim now asks "Lose Changes (Y/N)?" when you try to

7.2 Version 1.43

An intermediate version with some cleanups.

7.3 Version 1.42

1. The "pump block thru external filter" command now outfoxes filters that check if their standard input or output is redirected.
2. The following flags where added to nfmt.exe:

   `-m` Manuscript mode. Compress all intra-word spaces into a single space. Make all sentences end with two spaces.

   `-h` Print help information to stderr.

7.4 Version 1.41

1. The following UNIX like filters were introduced: uudecode, fold, expand, and unexpand. [Some of these filters include software developed by the University of California, Berkeley and its contributors.] The cat filter recognizes wild cards now.

2. The "pump block thru external filter" command was improved: Slim displays an error message if this command is attempted when no block was marked. It beeps if an external filter cannot be executed. The standard error output of a filter is also inserted into the buffer. A bug related to passing command line arguments to a filter on non-4dos systems was eliminated.

3. Several bugs in the configuration utilities were fixed. An option for setting a default right margin (for word wrap) was introduced.

4. Corrected a bugs which garbled line numbering after a translate command which had LF characters in the replaced string or in the new string. (18 bytes squeezed in)

5. The documentation incorporated now many thoughtful comments and restructuring suggestions of Larry Bennett `usr2059a@tso.uc.EDU`.

6. `H.COM` now clears the screen when exiting. Thanks to George Chow from the computing services of the University of British Columbia for allowing me to try myself `H.COM` for the first time on an HP95LX palmtop and for his help in understanding the screen clearing problem so many users complained about.

7.5 Version 1.4

1. The source code of Terse is now made available to the public for a very modest registration fee of $20: $15 for the basic registration, and an additional $5 for the source code.

2. The pre-release of Slim was greatly enhanced. Slim can do every thing Terse can, and has the following additional features:

   - "read file into buffer" command Alt-R.
   - "switch to another file" command Alt-E,
   - "go to line number" command Alt-G,
   - "set right margin" command Alt-M,
   - "swap cursor and mark" command Shift-Tab,

- “pump block thru external filter” command Alt-F,
- “exchange marked block with paste buffer” command Keypad*, and
- display of the hexadecimal value of the current char in the status line.

The current char display uses a special notation for the ⟨TAB⟩, ⟨EOF⟩ and ⟨CR⟩ characters as well as for the ⟨CR⟩LF combination.

My favourite new command is ”pump block thru external filter” (Alt-F key), which can be undone (and then redone) by the ”exchange block with paste buffer” command (Keypad* key). Think of the possibilities: formatting, spelling, word counting, sorting, sed and awk, DOS and 4DOS commands etc.

3. Whenever Terse prompts you for an input it also offers your previous input to the same prompt as a default response string. For example, upon pressing Alt-S, Terse displays the prompt

```
Search for:
```

in the status line, and offers your previous response to the prompt `Search for:`. You can use this previous response unchanged (hit Enter) or you can delete characters from it (use the Backspace key) or you can add characters to it (hit any arrow key, and start adding characters. If you immediately start typing a new entry, the default string is deleted and replaced by what you type. Hitting the Ctrl-U key deletes all of the currently entered response. The configuration utilities allow changing Ctrl-U to any other control letter. (13 bytes squeezed in.)

4. Several external filters are now included in the unabridged distribution. These utilities are intended to be used in conjunction with the ”pump block thru external filter” command of Slim. These filters include among others: cat, wc, lf, crlf, quote and nfmt. With the exception of nfmt, the registration of Terse includes registration of all of these programs.

   Ross Paterson, `rap@doc.ic.ac.uk` the author of NFMT, recently posted the sources in the alt.sources usenet newsgroup. Porting to the PC and some packaging was done by myself.

5. Several useful utilities are now included in the unabridged distribution. These utilities include among others the following programs: esc, lptstat, bigc, litc. The registration of Terse includes registration of all of these utilities.

6. A pre-release of Terse for the HP 95lx palmtop (named h.com) is now included in the unabridged distribution. This program, together with the option in the configuration utilities to set the file size replaces Wee. The key binding of `H.COM` is slightly different than that of Terse and Slim, to support the keyboard of the HP 95lx.

7. The font loader and the fonts used for editing binary files are now included in the unabridged distribution.

8. The configuration utilities have been enhanced. Among other things, more control of the status line attributes was added and more options of automatic horizontal scroll. Thanks to Mike Bessy (from JP Software) `75300.1215@compuserve.com` and to Terry Chan `echan@chinug.acns.nwu.edu` for suggesting some of these changes.

9. The distribution of Terse now includes a batch file to assist in invoking the editor on several files. An `.BTM` file for setting appropriate aliases in 4DOS is included as well.

10. Many of the documentation and configuration utilities typos have been corrected. Special thanks to Joan Abarbanel `abraban@taunivm.tau.ac.il` for her careful proofreading of a previous version of this file. Terry Chan `echan@chinug.acns.nwu.edu` did a great job of pointing out to me errors and typos in the configuration utilities as well as in the documentation. [Note that I still take the credit and blame for all remaining errors which have not yet been found and corrected.]

11. The following scenario used to create a file of length 0:

    `T` newfile

    $\boxed{\text{Alt-Q}}$

    This bug has been fixed. Thanks to the many kind users who brought this bug to my attention. (4 bytes of code added.)

12. Terse recognizes and parses correctly file names which include dots in them (e.g., `..\..\file.dat`). Previously, such files would not be saved. (20 bytes code squeezed in.)

13. Terse can now read files of size 65,535 exactly. Previously, files of this size could be created by Terse, but not read by it. (4 bytes code squeezed in.)

14. Eliminated several bugs related to the display of the help screen, including the misalignment when ANSI console driver was present, eliminated. (12 bytes gained.)

15. Eliminated a bug which could have caused Terse to hang after pushing to DOS version 2.x. (4 bytes code squeezed in.)

16. Terse now beeps whenever a save file attempt fails. Previously, the only indication of such an error was that the "Write successful" message didn't appear. (2 bytes code squeezed in.)

17. Next/Pret/Previous line now recognize all of the following characters as word separators: space, tab, vertical tab, carriage return, line feed, and form feed.

7.6 Version 1.3

1. Simplified and unified the configuration utilities.
2. Added a pre-release of Lean, Wee and Slim.
3. Included some of the configuration utilities in the shareware distribution. The font loaders required for binary file editing are still part of the registration package.
4. Fixed a bug, introduced in version 1.2, that prevented the display of the status line.

7.7 Version 1.2

1. Code size was shrunk by another 9 (!) bytes.
2. The restriction on distribution by fee shareware distributors has been removed.
3. After a successful print (Alt-P), the marked block is unmarked.
4. Pre-Release versions of Slim, Lean, Wee and Terse for HP 95 LX palmtop computer are available upon request.

7.8 Version 1.1

1. A bug in the snow prevention routine which caused Terse to hang on some CGA systems has been corrected. Terse should run now with no problems on all adapters. Many thanks to Andrea Omodeo (E-mail address: `ele9037@cdc835.cdc.polimi.it`) for identifying the bug and for suggesting the correction.
2. After considerable work, the 27 bytes code fragment required for DESQview awareness was squeezed in. Terse can now run in a DESQview window, and it is fully DESQview aware.

7.9 Version 1.0

First version released to the public.

Chapter 8

Utilities

`BIGC.COM` A program to set a big cursor (for PALMTOPS/Notebooks.)

`LITC.COM` A program to set a little cursor (for PALMTOPS and Notebooks).

`ESC.COM` A program to send the escape sequences. Examples of usage are:

```
ESC [7m
ESC 0 > prn
```

`LF.COM` Filter. DOS to UNIX text file translation program. Run `lf -h` to see a manual page.

`CRLF.COM` Filter. UNIX to DOS text file translation program. Run `crlf -h` to see a manual page.

`CAT.COM` Filter. Concatenate files. Like UNIX cat.

`QUOTE.COM` Filter. Quote text by prepending the string "`> `" to lines.

`WC.COM` Filter. Count lines, words, chars. Like UNIX wc.

`UUDECODE.COM` Filter. A program to convert uudecoded files back to binary. Like UNIX uuedecode

`EXPAND.COM` UNIX-like filter. Expand tabs into spaces. Run `expand -h` to see a manual page.

`UNEXPAND.COM` UNIX-like filter. Compress spaces into tabs Run `unexpand -h` to see a manual page.

`FOLD.COM` UNIX-like filter. Fold long lines for finite width output device. Run `fold -h` to see a manual page.

`NFMT.EXE` UNIX fmt-like formatter. `NFMT` is a simple text formatter, which is very similar to the BSD program fmt(1). However NFMT uses a best-fit line breaking algorithm, by a simple version of "Breaking Paragraphs into Lines", Donald E. Knuth and Michael F. Plass, *Software – Practice and Experience* 11 (1981) pp. 1119-1184. Author: Ross Paterson `<rap@doc.ic.ac.uk>`, porting to Turbo-C by Yossi Gil ¡yogi@cs.ubc.ca¿. Run `nfmt -h` to view its help screen.

`NFMT.HLP` Help screen extracted from `NFMT.EXE`

Chapter 9

Known Bugs and Future Directions

- Horizontal scrolling does not move the cursor position, so by pressing $\boxed{\text{Alt-+}}$ and $\boxed{\text{Alt–}}$ you can scroll to the right or to the left but only to the extent which leaves the current cursor position visible. For example when the cursor is in the left most column, no horizontal scrolling is possible. There are no plans for modifying this behavior as the author feels that it is very reasonable. Strong user pressure may change this.
- $\boxed{\text{Alt-U}}$ only restores the last character erased by $\boxed{\text{Backspace}}$. (Future plan is to push deleted characters (by $\boxed{\text{DEL}}$, $\boxed{\text{Backspace}}$ and overwritten characters) to a stack. The stack will have limited depth, and as characters are pushed in, old characters stored deeply in the stack disappear. The $\boxed{\text{Alt-U}}$ will pop characters from the stack and insert/append them.)
- Editing can get extremely slow if lines are excessively long. Lines longer than 20,000 characters cause a noticeable delay after every editing operation on these lines.