The Linux Emacspeak HOWTO

Table of Contents

Table of Contents

The Linux Emacspeak HOWTO

Jim Van Zandt, `jrv@vanzandt.mv.com`

v. 1.7, 16 Dec 2000

This document describes how a blind user can use Linux with a speech synthesizer to replace the video display. It describes how to get Linux running on your own PC, and how to set it up for speech output. It suggests how to learn about Unix.

---

7. Frequently Asked Questions (FAQ)

8. Legalese

---

1. Introduction

Emacspeak is an Emacs subsystem that allows the user to get feedback using synthesized speech.

Screen reading programs allow a visually impaired user to get feedback using synthesized speech. Such programs have been commercially available for well over a decade. Most of them run on PC's under DOS, and there are now a few screen−readers for the Windows platform. However, screen−readers for the UNIX environment have been conspicuous in their absence.

This means that most visually impaired computer users face the additional handicap of being DOS−impaired :−)

Emacspeak is an emacs subsystem that provides basic speech access. Emacspeak will always have the shortcoming that it will only work under Emacs. This said, there is very little that cannot be done inside Emacs, so it's not a real shortcoming:−) Within Emacs, you can open a "shell window" where you can run commands and examine their output, even output which has scrolled out of the window. Emacs provides special modes for running certain commands. For example, it can parse error messages printed by a compiler and open a separate edit window with the cursor at the point of the error. It can also run a debugger and keep a separate edit window open at the point in the source code corresponding to the program counter.

Emacspeak does have a significant advantage: since it runs inside Emacs, a structure−sensitive, fully customizable editor, Emacspeak often has more context−specific information about what it is speaking than its commercial counterparts. In this sense, Emacspeak is not a "screenreader", it is a subsystem that produces speech output. A traditional screen−reader speaks the content of the screen, leaving it to the user to interpret the visually laid−out information. Emacspeak, on the other hand, treats speech as a first−class output mode; it speaks the information in a manner that is easy to comprehend when listening.

This initial version provides a basic speech subsystem for Emacs; using Emacs' power and flexibility, it has proven straightforward to add modules that customize how things are spoken, e.g. depending on the major/minor mode of a given buffer. Note that the basic speech functionality provided by Emacspeak is sufficient to use most Emacs packages effectively; adding package−specific customizations makes the interaction much smoother. This is because package−specific extensions can take advantage of the current context.

Emacspeak will only work with emacs. However, emacs can be used to run any program that has a command−line interface (ls, cd, rm, adduser, etc.). In general use M−x shell to run such programs.

You can even run those like less or lynx which use escape sequences to control the appearance of the screen. The key to this is term mode, which you get with the emacs command M−x term.

Let me insert here some explanation, because the word "terminal" is used in several senses. The original terminal was of course an electromechanical device that translated keystrokes into character codes and received the same codes and printed the characters. Eventually these were replaced by entirely electronic "dumb terminals" which also implemented escape sequences to clear the screen, move the cursor, etc. Unix adapted to these by introducing an abstract notion of a "terminal" with very general capabilities. A program like a text editor issues generic commands, which are translated into the commands suitable for your terminal. The choice of translation is controlled by the variable "TERM" in the environment. One common value for this variable is "vt100", since many terminals implement the same commands as the DEC VT100. The Emacs command `M−x term' starts a virtual terminal with its own set of cursor commands. It notifies the operating system of this by setting the environment variable "TERM" to the value "eterm" (which is why it is sometimes called eterm mode).

Note that this Unix "terminal" is concerned only with escape commands for the display. Let me just mention a couple of other kinds of "terminal" to distinguish them from the Unix concept: For DOS, there are "terminal emulators" like COMMO and TELIX that simulate a terminal, and can be used to connect to a remote computer via a serial port (and usually a modem). That is, they accept keystrokes and send the character codes out a serial port, and receive character codes from the serial port and display them. Unix has similar programs, such as minicom and seyon. Both DOS and Unix have another program called "telnet" which can establish a similar connection to a separate computer using the Internet (that is, a TCP/IP connection rather than a serial connection). The program "kermit" is available for both DOS and Unix, and can establish either kind of connection. However, none of these should be confused with the eterm mode of emacs. There is nothing about eterm mode that implies a connection to a remote computer.

Note that in most cases there is an emacs mode that works better than the separate program in an eterm buffer. That is, instead of "less", just open a buffer on the file. Instead of "lynx", use w3. Instead of "man", use `M−x man', and type the name of the program in the minibuffer.

Remember, though: ordinary command−line oriented programs are best run in shell mode, which starts with the command `M−x shell'.

Emacs is a large program, but it does not all have to be in RAM, because Linux has virtual memory. You can designate a swap partition, so that programs (or parts of programs) can be swapped out when they are not being used. You can comfortably run emacs with 8 MB of ram plus 8 MB of swap space.

This document is limited to the following:

- Linux (not Free BSD)
- The Slackware distribution (not Red Hat, Debian, etc.)
- Speech output only (not Braille − see the Access HOWTO)
- DECtalk (Dectalk Express and MultiVoice), DoubleTalk, LiteTalk, Braille 'n Speak, Type 'n Speak,

Braille Lite, Apollo, and Accent synthesizers (−−not the SmarTalk, a sound card, etc.)
* Use of Emacs, with T. V. Raman's Emacspeak package, to drive the synthesizer.

The use of adaptive technology with Linux, and in particular, using adaptive technology to make Linux accessible to those who could not use it otherwise, is covered in the Linux Access HOWTO.

If you would like to help extend this document to cover one or more of the other alternatives, or point me to a discussion somewhere else, please contact me.

The most recent version of this document should be available in several formats at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/ and its mirrors (see below). The plain text version should be available at ftp://leb.net/pub/blinux/doc/.

Emacspeak was written by T. V. Raman `raman@adobe.com`. Emacspeak has a Web page at http://www.cs.cornell.edu/Info/People/raman/emacspeak/emacspeak.html.

Emacspeak supports several speech synthesizers. The software required depends on which you have.

If you have a DECtalk Express or MultiVoice, you need the basic Emacspeak package, tcl (an interpreter), and tclx (extensions for tcl). You can get the source package for Emacspeak from the Emacspeak web page. You can also get a binary package for a Red Hat distribution, either from the Emacspeak web page or one prepared by Matt Campbell. I maintain binary packages for the Debian distribution. At this writing, the most recent release of Emacspeak is version 13.0. Here are some URLs:

```
http://www.pobox.com/~mattcampbell/emacspeak.html http://leb.net/pub/blinux/emacspeak/cornell.mirror ftp://ftp.cs.cornell.edu/pub/raman/emacspeak/ ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-6.0-2.i386.rpm http://www.mv.com/ipusers/vanzandt/
```

Note: there are European mirrors of the blinux site which you should use if they closer:

```
ftp://ftp.uni-muenster.de/share/public6/ ftp://ftp.gwdg.de/pub/linux/misc/blinux/
```

For the internal DECtalk PC, you also need a kernel driver which is available from ftp://leb.net/pub/blinux/dectalk_pc−0.91.tgz

Extract it with the command: `tar xfz dectalk_pc−0.91.tgz' and then follow the installation instructions found in the README file. This driver apparently works with both the original DECTALK−pc and the new DECTALK−PC II.

For the external DoubleTalk, LiteTalk, Accent, or Apollo synthesizers, or for the Braille 'n Speak, Type 'n Speak, or Braille Lite devices in "speech box" mode, you need the Emacspeak package and a separate "speech server" which comes in the emacspeak−ss package:

```
ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-ss-1.7.tar.gz
```

For the internal DoubleTalk, you need three pieces of software: the basic Emacspeak package, the emacspeak−ss package, and a device driver. The device driver is now part of the official Linux kernel sources. Use version 2.2.17 or later, and configure with "Double Talk PC internal speech card support" either compiled into the kernel or installed as a module (dtlk.o).

There is a server for the Braille 'n Speak, Braille Lite, and Type 'n Speak devices used in "speech box" mode:

```
ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-tcl-1.2.tar.gz ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-tcl_1.2-1_all.deb ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-1.2-1.i386.rpm
```

Once again, the emacspeak−ss server handles all the synthesizers other than the DECtalk. You should use that one if you don't mind building and installing a source package.

Computer hardware, Unix user commands, Unix system administration, Emacs, and Emacspeak are each substantial subjects. Attempting to learn all of them at once is likely to lead to frustration. Instead, I suggest that the new user go through a sequence of stages, learning about only one system at a time.

2. Stage 1. DOS with speech

Most blind computer users have DOS or Windows, a speech synthesizer, and a screen reader program like JAWS [JAWS]. (References in this format refer to entries in the "Footnotes and References" section below.) Using this setup, install and become familiar with some terminal emulator like Telix [TELIX] or Commo [COMMO], which are available from the SimTel archive [SimTel] among others.

2.1 Getting Linux on CDROM

If you have or can borrow a CDROM drive, I recommend you get one of the many good distributions of Linux on that medium. The instructions below are for the Slackware distribution. I am most familiar with disks from Linux Central [Linux Central]. Other sources are InfoMagic [InfoMagic] and Walnut Creek [Walnut Creek] (where the whole idea of inexpensive CDROMs full of programs from Internet archives got its start). Linux distributions are also available from Red Hat [Red Hat], Craftwork [Craftwork], and Yggdrasil [Yggdrasil]. As a rule, these CDROMs use the "ISO 9660" format, which can be read under DOS. (They also use the "Rock Ridge extensions" which add extra files in each directory. Linux uses the extra information to give you long filenames, both upper and lower case characters in filenames, and file permissions.)

2.2 Getting Linux by FTP

Another way to get Linux and its documentation is by FTP over the Internet. The home site for the Slackware distribution is Walnut Creek [Walnut Creek]. It is also carried by metalab (formerly known as sunsite) and many of it mirror sites. Here is a partial list:

- USA (home site) ftp://ftp.cdrom.com/pub/linux/slackware
- UK/Europe ftp://src.doc.ic.ac.uk/public/Mirrors/ftp.cdrom.com/pub/linux/slackware−3.1
- Japan ftp://ftp.cs.titech.ac.jp/pub/os/linux/slackware
- Taiwan ftp://NCTUCCCA.edu.tw/OS/Linux/Slackware
- Hong Kong ftp://ftp.cs.cuhk.hk/pub/slackware
- USA ftp://metalab.unc.edu/pub/Linux/distributions/slackware
- USA ftp://uiarchive.cso.uiuc.edu/pub/systems/linux/sunsite/distributions/slackware

More sites are listed in the INFO−SHEET:

```
http://metalab.unc.edu/pub/Linux/welcome.html ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/INFO-SHEET ftp://uiarchive.cso.uiuc.edu/pub/systems/linux/sunsite/docs/HOWTO/INFO-SHEET
```

2.3 Linux Documentation

Read the Linux documentation. I will quote here the file names and locations on the first disk of InfoMagic's December 1996 "Developer's Resource" set of six CDROMs, as seen under DOS. Other CDROM sets should have similar information, though perhaps differently arranged. The Slackware distribution is on disk 2 of the set. Matt Welsh's step by step guide to installing Slackware is in `\doc\install-\install-.002`. (This is a 245 page book!)

More general information is in the Linux "Frequently Asked Questions" list in `\docs\linux.faq\linux-fa.asc`. Longer descriptions are in "HOWTO" documents (of which this is one). They are found in `\docs`. Note particularly `\docs\hardware`, which lists which kinds of hardware are supported by Linux, `\docs\meta-faq`, which points to sources of information (that is, a more extensive version of this paragraph), and `\help\index`, which is a list of the HOWTO documents with short descriptions. The Linux installation HOWTO, `\docs\installation`, is another (much shorter, somewhat older) version of Matt Welsh's installation instructions.

One note on reading the documentation. You may run into files with ASCII highlighting, where character−backspace−character stands for "bold", and underscore−backspace−character stands for "italics". One way to handle this is to use the `less` program, which displays these sequences in alternate colors. A DOS screenreader can, for example, search for such highlighted text. A DOS version of `less` can be obtained by FTP from the SimTel archive [SimTel]. Within the SimTel collection, look for directory msdos/textutil. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/textutil.

I will suggest four alternatives for learning Emacs commands (see section [Learning Emacs]). The first option is to install Emacs under DOS and learn it while using the DOS screen reader. Where to get Emacs for DOS is a "frequently asked question" [Emacs for DOS].

The source code for Emacs (about 10 MB) can be gotten from ftp://prep.ai.mit.edu/pub/gnu/ (look for emacs−19.34b.tar.gz or similar), or from one of many mirrors of the GNU collection [Gnu Mirrors].

---

3. Stage 2. Terminal to remote UNIX system

Arrange for what is called a "shell account" on some Unix system. Most Internet Service Providers (ISPs) can provide this service. Use the terminal emulator program and a modem to dial in. Learn the basic Unix commands. If the system has Emacs installed, or you can persuade the system administrator to install it, this is your second chance to learn it. It is probably best to learn it at this point, because administering a Unix system (the next stage) will call for you to edit files. Therefore, I include here my suggestions for learning both Unix and Emacs.

3.1 Learning Unix

When you arrange for a shell account, or set up a new account on your own machine, you will have to decide on a username and a password. Your username will also be used in your email address, so try to find something short and memorable. Your password is important, and should be hard to guess. That usually means at least six characters, including at least one non−alphanumeric character.

When a Unix system is ready for you to log in, it normally displays a prompt ending with "login:". At this point you should type in your username. It will then prompt you for your password, and will turn off command echoing while you type it in.

The command to finish a terminal session is `logout`.

There are many kinds of documentation available on a Linux system. Traditionally, each command has a manual page which can be displayed by the `man` command. For example, learn more about the `cp` command by typing `man cp`. Of course, this helps only if you know or can guess the command name. However, each man page has a line near the beginning with the command name and a short description of what the command does. You can search a database of these lines using the command `apropos`. Thus, typing `apropos working` will list lines that include the word "working". Here's what that command prints here:

```
cd (n)               - Change working directory
pwd (n)              - Return the current working directory
Cwd (3pm)            - get pathname of current working directory
chdir (2)            - change working directory
fchdir (2)           - change working directory
get_current_dir_name (3) - Get current working directory
getcwd (3)           - Get current working directory
getwd (3)            - Get current working directory
pwd (1)              - print name of current/working directory
rcsclean (1)         - clean up working files
```

The numbers in parentheses are sections of the manual. User commands are in section 1. Functions called from within programs are in sections 2 and 3. Commands used mostly by the system administrator are in section 8. You can find out more about the on−line manual with the command `man man`.

The program that interprets your command is a "shell". Under DOS, COMMAND.COM is the shell. Most Unix shells are descendents of either the Bourne shell `sh` or the C shell `csh`. The shell most commonly used with Linux is the "Bourne again shell", or `bash`. Many Unix commands that were once separate programs are built into bash. Help for these is also built in. For example, typing `help cd` will display a short description of the `cd` command. Typing `help` by itself will display a list of all the builtin commands. Unfortunately, it is shown in two columns. If you find this confusing, you can pipe the list through `colrm` to eliminate one column. Type `help|colrm 37` to show just the first column, and `help|colrm 1 36` to show just the second column.

Many Linux programs from the Free Software Foundation are best documented in info pages. For example, the C compiler documentation can be displayed by typing `info gcc`.

Under Unix, commands normally accept options starting with a minus sign rather than the forward slash used under DOS. In a path, directory names are separated by forward slashes rather than backward slashes. Both operating systems have a "standard input", by default the keyboard, and a "standard output", by default the display screen. You can redirect the standard input using "<", and redirect the output using ">". You can use the output from one command as the input of another by separating the two commands with "|". This is called the "pipe" symbol.

If a program gets "stuck", here is a sequence of keystrokes to try:

- Control−Q. You may have sent a control−S, which halts all output, without realizing it. The control−Q will restart it.
- Control−D, which signals "end of file" under Unix (similar to control−Z under DOS), in case the program expects input which you are not prepared to supply.
- Control−C is an interrupt, which may halt the program.
- Control−Z puts the program in the background. At this point you may simply log out, although you will be warned about the background process and will have to repeat the `logout` command. You can instead kill the process, as follows: Run `ps` with no arguments. It will list a header line, then one line for each of your processes. The first item on each line is the process id number, or PID. The command used to start the process (or at least the beginning of it) appears at the end of the line. If the PID were 117, you would kill the process with the command `kill -9 117`.

- If running Linux from the console, alt−2, or some other alt−number combination, will switch to a different virtual console. You can log in there just as if you had sat down to a different terminal.
- If your machine is connected to a network, you can log in from another machine.
- Control−alt−del should reboot the computer nondestructively.
- As a last resort, you can hit "reset" or cycle the power. This will leave the filesystems in an invalid state, since some buffers will not have been written to disk. The kernel will discover this while booting, and will take time to check and repair the filesystems. Actual data loss is unlikely unless you had something else going on at the time.

Guido Gonzato `Guido@ibogfs.cineca.it` has written an excellent guide to Linux for (former) DOS users, the DOS2Linux mini−HOWTO. You can probably find it where you found this document, or else at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/mini/.

You can find general Unix information, including manual pages for several systems at http://www.cis.ohio−state.edu/hypertext/man_pages.html

There is a tutorial entitled "Beginning Unix and the C Shell" at http://www.eng.hawaii.edu:80/Courses/C.unix/page−03.html.

You can get general help from http://www.nova.edu/Inter−Links/UNIXhelp/TOP_.html or http://www.eecs.nwu.edu/unix.html

You can find a list of books on UNIX at http://www.eskimo.com/~cher/eskimospace/booklist.html.

3.2 Learning Emacs

When you start Emacs, you will normally list on the command line one or more files which you will be editing. To edit a file named "foobar" with Emacs, you would enter the command `emacs foobar`. If you enter the command `emacs` with no arguments, GNU Emacs will print out an introduction which includes the first five commands you need to learn, approximately as follows:

```
Type C-h for help;   (`C-' means use CTRL key.)
Type C-x u to undo changes.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.
To kill the Emacs job, type C-x C-c.
```

Note the way Emacs documentation refers to key combinations. C−h means hold the control key down while typing "h". You will also run into key combinations like M−v, which is pronounced "meta v". The tutorial suggests holding down the key labeled "edit" or "meta" then typing "v". I have never run across a keyboard with those keys, so I always use the escape key instead: typing "Esc" then "v" (two separate keystrokes). After using Emacs for a long time, I discovered that under Linux, the left "Alt" key works like a "meta" key. You may want to use this. On the other hand, some of these key combinations may conflict with your screen reader or communications program under DOS. Using the escape key is more reliable.

Three of the above commands start with C−h, which may be treated as a backspace by your communications program. In that case, you may access the help command using the long form M−x help. Conversely, you may find that pressing the backspace key starts the help command. This issue is treated in the Emacs FAQ, which is available within Emacs using C−h F or M−x help F. Look for the question "Why does the `Backspace' key invoke help?". In the mean time, you can end the help session with the command C−g. (This is the keyboard−quit command, which cancels any prefix keys you have typed.)

You may also find that C−s and C−q are unavailable because they are used for flow control (XON and XOFF). You should look at the question "How do I handle C−s and C−q being used for flow control?" in the FAQ. For the particular command C−x C−s (save buffer), you may substitute the command C−x s (save−some−buffers). The former command saves the current buffer, while the latter asks the user about each of the modified buffers.

Note in particular the command "C−h t" to start the Emacs tutorial. That is one the first things you will want to try. I will only make a couple of comments on the tutorial. To move the cursor, it gives the four commands C−f, C−b, C−p, and C−n (for forward, back, previous line, and next line). These commands always work. However, with a properly installed Emacs, the regular arrow keys should also work. Try them out and use them if you are more comfortable with them. Similarly, you may be able to use home, end, page down, and page up keys in place of the standard commands C−a, C−e, C−v, and M−v. Finally, all Emacspeak commands begin with C−e. Once you start using Emacspeak, you will have to type it twice to get the end of line function. (The "End" key should be unaffected by Emacspeak.)

4. Stage 3. Terminal to local Linux system

This arrangement again requires a DOS machine with a speech synthesizer and a terminal emulator program. However, instead of dialing up a remote computer, it is used as a terminal to a local computer running Linux. To get to this point, you need to install Linux on a machine. You may be able to prevail on a knowledgeable friend to help you with this. However, it is also possible to install it yourself with speech feedback for almost the whole procedure.

4.1 Installing Linux

First, some background. Even the simplest Unix system requires a program called the kernel and a root file system. The kernel has all the device drivers and resource management functions. One normally thinks of a "file system" as residing on a hard disk or floppy disk, but during an installation it is usually in ram. Linux is normally installed by writing a kernel image to a floppy disk, called the "boot floppy", configuring it to reserve a section of RAM for a ramdisk, then filling that ramdisk with data from a second floppy disk, called the "root floppy". As soon as both floppies have been read in, the user can log in as "root" and complete the installation. The sighted user logs in on the "system console", that is, the computer's own keyboard and video display. However, remember that Unix has been a multiprocessing operating system from the very beginning. Even this very primitive Unix system, running out of a small ramdisk, also supports logins from a terminal connected to a serial port. This is what a blind user can use.

To connect the two computers, you can use a "null modem", a serial cable that connects ground to ground, and transmit on each end to receive on the other. The cable that comes with the DOS application LapLink will work fine. It is particularly handy, in fact, because it has both a 9 pin and a 25 pin connector on each end. If you want to check a cable or have one made, here are the required connections:

For two 9 pin connectors, connect pin 2 (receive data) to pin 3, pin 3 (transmit data) to pin 2, and pin 5 (signal ground) to pin 5.

For two 25 pin connectors, connect pin 2 (receive data) to pin 3, pin 3 (transmit data) to pin 2, and pin 7 (signal ground) to pin 7.

For a 9 pin connector (first) to a 25 pin connector (second), connect pin 2 (receive data) to pin 2 (transmit data), pin 3 (transmit data) to pin 3 (receive data), and pin 5 (signal ground) to pin 7 (signal ground).

You may have noted that I have included no connections for the "handshaking" signals. During login, the serial port is handled by the program `agetty`. Recent versions of this program accept a −L switch which tells it not to expect modem control signals. The version in Slackware 3.0 does, but the one on the 3.0 (and earlier) installation root disks does not. However, Pat Volkerding has assured me that the root disks in the next release of Slackware will have the updated version of `agetty`. It is also possible to use the earlier root disks [Emacspeak with Earlier Slackware Releases].

Consult the documentation on your CDROM, or downloaded from an FTP site, and choose a boot disk with the proper kernel features for your hardware (IDE or SCSI, CDROM driver, etc.). I have the InfoMagic September 1996 "Developer's Resource" set of six CDROMs. Slackware 3.1 is on disk 1 of that set, mostly in the two directories slackwar and slakware. (Note the difference in spelling. You will access them in alphabetical order: first slackwar, then slakware.)

Documentation on the boot floppies is in \bootdsks.144\which.one. A copy of the DOS program for writing boot images to a floppy, rawrite.exe, is in the same directory. Assuming the CDROM is the M drive under DOS, one might use these commands to write to a floppy disk in the A drive:

```
C>m:
M>cd \bootdsks.144
M>rawrite scsinet.s a:
```

Similarly, to write the "text" root disk:

```
C>m:
M>cd \rootdsks
M>rawrite text.gz a:
```

If you install from floppies, you should also copy the Emacspeak package onto a floppy with a command like this:

```
C>copy m:\contrib\emacspea.tgz a:
```

For the actual installation, proceed as follows: Use the null modem to connect the computer running DOS and equipped with speech output (which I will call the "DOS machine") to the computer into which you want to install Linux (the "Linux machine").

Boot the DOS machine, and start your terminal emulation program. Set it up for 9600 baud, no parity, eight data bits, 1 stop bit.

On the Linux machine, insert the "boot" disk and boot (power up, cntl−alt−del, or hit the reset switch). It should read the disk for five seconds or so, beep, and stop with the following text:

(Note: in the following, the large blocks of text quoted from the installation disks are preceded by "−− begin quote" and followed by "−− end quote". To skip to the end of a quote, you may search for two dashes starting in the first column. I have word wrapped some sections to limit the line lengths.)

```
-- begin quote

      Welcome to the Slackware96 Linux (v. 3.1.0) bootkernel disk!

      If you have any extra parameters to pass to the kernel, enter them at
      the prompt below after one of the valid configuration names (ramdisk,
      mount, drive2)
```

```
       Here are some examples (and more can be found in the BOOTING file):

         ramdisk hd=cyl,hds,secs    (Where "cyl", "hds", and "secs" are the
                                    number of cylinders, sectors, and heads
                                    on the drive.  Most machines won't need
                                    this.)

       In a pinch, you can boot your system with a command like:
         mount root=/dev/hda1

       On machines with low memory, you can use mount root=/dev/fd1 or mount
       root=/dev/fd0 to install without a ramdisk.  See LOWMEM.TXT for
       details.

       If you would rather load the root/install disk from your second
       floppy drive: drive2 (or even this: ramdisk root=/dev/fd1)

       DON'T SWITCH ANY DISKS YET!  This prompt is just for entering extra
       parameters.  If you don't need to enter any parameters, hit ENTER to
       continue.

       boot:

-- end quote
```

I have almost always been able to just hit "enter" at this point.

After your entry, the Linux machine should read the floppy for another twenty seconds or so, then boot the kernel. The first thing it prints is "Loading ramdisk...", which is somewhat misleading. In this case, "ramdisk" is actually the name of the kernel configuration.

Each device driver in the kernel displays a line or two. The particular disk I'm using (the "bare.i" bootdisk) displays more than one screen's worth. It is possible to type shift−page up to scroll the text back. On my machine, the boot messages are as follows:

```
-- begin quote

       Loading ramdisk.....
       Uncompressing Linux...done.
       Now booting the kernel
       Console: colour VGA+ 80x25, 1 virtual console (max 63)
       Calibrating delay loop.. ok - 35.94 BogoMIPS
       Memory: 23028k/24768k available (688k kernel code, 384k reserved,
        668k data)
       Swansea University Computer Society NET3.035 for Linux 2.0
       NET3: Unix domain sockets 0.12 for Linux NET3.035.
       Swansea University Computer Society TCP/IP for NET3.034
       IP Protocols: ICMP, UDP, TCP
       VFS: Diskquotas version dquot_5.6.0 initialized
       Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
       Checking 'hlt' instruction... Ok.
       Linux version 2.0.0 (root@darkstar) (gcc version 2.7.2) #1 Mon Jun 10
       21:11:56 CDT 1996
       Serial driver version 4.13 with no serial options enabled
       tty00 at 0x03f8 (irq = 4) is a 16550A
       PS/2 auxiliary pointing device detected -- driver installed.
       Ramdisk driver initialized : 16 ramdisks of 49152K size
       hda: IBM-DBOA-2720, 689MB w/64KB Cache, LBA, CHS=700/32/63
```

```
ide0: at 0x1f0-0x1f7,0x3f6 on irq 14
Floppy drive(s): fd0 is 1.44M
Started kswapd v 1.4.2.2
FDC 0 is a 8272A
Partition check:
  hda: hda1 hda2 hda3
VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER
```

-- end quote

Some messages will of course be different on a machine with different hardware. Now, insert the "text" rootdisk and press ENTER. After it is read, the following is displayed on the console:

-- begin quote

```
RAMDISK: Compressed image found at block 0
JAVA Binary support v1.01 for Linux 1.3.98 (C)1996 Brian A. Lantz
VFS: Mounted root (minix filesystem).
INIT: version 2.60 booting
none on /proc type proc (rw)
INIT: Entering runlevel: 4


Welcome to the Slackware Linux installation disk ,version 3.1.0-text!
### READ THE INSTRUCTIONS BELOW CAREFULLY! ###

You will need one or more partitions of type "Linux native"
prepared. It is also recommended that you create a swap partition
(type "Linux swap") prior to installation. Most users can use the
Linux "fdisk" utility to create and tag the types of all these
partitions. OS/2 Boot Manager users, however, should create their
Linux partitions with OS/2 "fdisk", add the bootable (root) partition
to the Boot Manager menu, and then use the Linux "fdisk" to tag the
partitions as type "Linux native".

If you have 4 megabytes or less of RAM, you MUST ACTIVATE a swap
partition before running setup. After making the partition with fdisk,
use:

mkswap /dev/<partition> <number of blocks> ; swapon /dev/<partition>

Once you have prepared the disk partitions for Linux, type "setup" to
begin the installation process.

You may now login as "root".

slackware login:
```

-- end quote

The program that prints the login prompt is called `agetty`. The Slackware 3.1 root disks are set up to allow logins only from the computer's own keyboard. You will have to reconfigure it to also allow logins from a serial port. This requires typing four lines on the Linux machine keyboard, with no voice feedback. If you realize you have made a mistake before hitting the carriage return, you can erase it with the backspace key. You can also discard what you have typed on a line with control−C. Here is what you type:

```
root
cat >>/etc/inittab
s1:45:respawn:/sbin/agetty 9600 ttyS0
```

```
control-D
init q
```

I will repeat that with explanations of what is going on.

First, type "root" and a single carriage return to log in (no password is needed). Next, you need to append one line to `/etc/inittab`. Type the following two lines:

```
cat >>/etc/inittab
s1:45:respawn:/sbin/agetty 9600 ttyS0
```

Finish each line with the "enter" key. Then type a control−D, which signals end of file to a Unix program. (Note: In the second line, the next to last character is an upper case "S". Everything else is in lower case.) This adds a line to the configuration file of the program `init`, to instruct it to use `agetty` to watch for logins on the first serial port on the Linux machine, called "COM1" under DOS, or "/dev/ttyS0" under Linux. To use the second port instead, change the last item on the above line to "ttyS1".

Then type

```
init q
```

which causes `init` to reread `/etc/inittab`. At this point the DOS machine should display the login prompt (the third of the blocks of text quoted above). On the DOS machine, type `root`, and finish the installation. (The next thing you should do is create and enable a swap partition.)

If you don't get the Slackware installation disk prompt, try the following:

- Type a single carriage return on the DOS machine.
- Recheck the terminal setup (9600 baud, no parity, eight data bits, 1 stop bit)
- Disconnect the null modem from the DOS machine. In its place, connect a modem which supports the Hayes "AT" commands. Type `AT` and a carriage return. You should get a reply of "OK" from the modem.

Once you get the above prompt on the DOS machine, you may type `root` and a carriage return to log in, and complete the installation like any other user. Of course, you must remember to include these packages: emacs, tcl, and tclX.

The installation script will offer to prepare a boot floppy. You should do this, since it is the most foolproof way to boot Linux. You will probably also want to install `lilo` (which is an abbreviation for "Linux loader") and/or `loadlin` (which is an abbreviation for "load Linux"). The installation script can install lilo. Loadlin is a DOS program that will let you boot from DOS to Linux. Install it on a DOS partition, and copy a compressed kernel file (usually named `zImage`) to the same partition. While running DOS, you may boot Linux with a command like `loadlin zimage root=/dev/hda3 ro/`. (I have assumed here that the kernel image is in the same directory as the loadlin program. You may find it more convenient to store kernel images in subdirectories named for the kernel version.)

After the Slackware setup script finishes the main installation, it will tell you to restart by pressing cntl−alt−del. Before doing that, you should install emacspeak. It can be found with the other "contributed" software. In the InfoMagic set, it is in slackwar/contrib. Assuming you are installing Linux directly from a CDROM, the setup script will mount the CDROM under /CDROM, and you may install emacspeak with the following command:

```
# installpkg /CDROM/slackwar/contrib/emacspeak.tgz
```

If you install from floppies, insert the floppy you made earlier and type this:

```
# mount -tmsdos /dev/fd0 /floppy
# cp /floppy/emacspea.tgz /tmp/emacspeak.tgz
# installpkg /tmp/emacspeak.tgz
```

You should not install the package directly off the floppy disk, because the DOS filesystem will not allow the full filename, so the installpkg program will think the package name is "emacspea" and will store its records under that name.

If you have a DoubleTalk or LiteTalk speech synthesizer, you should also install the emacspeak−dt package.

Reboot the Linux machine with the new boot floppy, with the DOS machine still connected. You should get a login prompt on the DOS machine. Celebrate! After getting this system working, you need to learn emacs (third option) and Unix system administration.

4.2 Learning Unix System Administration

Mostly you will learn system administration as the need arises. First adding a user (yourself), then installing programs, and so forth. The exception to this is making backups, which you should learn **before** you need them.

Among the many programs you will need to learn are these:

adduser

Register a new user, including creating a home directory and adding an entry in /etc/passwd.

tar

Create and unpack `.tar` files, which are collections of files (something like `.zip` files). To list the contents of an archive, use `tar -tf foobar.tar`. For a more verbose listing, use `tar -tvf foobar.tar`. To unpack an archive, use `tar -xf foobar.tar`.

chmod

Change permissions of a file or directory.

chown

Change ownership of a file or directory.

find

Search directories recursively. For example, the command `find . -name '*alpha*' -print` means: search starting in the current directory (`.`) for a file whose name contains the string "alpha" (`-name '*alpha*'`), and print its path and name (`-print`). (With GNU find, the `-print` is optional.)

du

Display the amount of space occupied by files or subdirectories. For a file with "holes", this may be much less than the length of the file.

df

Display filesystem capacities, free space, and where they are mounted.

mount

Display filesystems, where they are mounted, and the mount flags.

ifconfig

Configure and check internet protocol (IP) network interfaces, including Ethernet cards, SLIP links, and PLIP links.

route

Configure and check IP network routing, after the interface is configured.

ping

Check IP network connectivity, after the interfaces and routes are configured.

ftp

Transfer files across the Internet.

Here are some programs you may want to install:

agrep

Approximate grep searches for approximate, not exact, string matches (also called "fuzzy string searches").

archie

Search Internet archives for files.

flip

Convert text files between Unix and DOS formats.

glimpse

Fuzzy string searches in large collection of files (uses agrep).

lynx

Text mode web browser.

Here are some Web pages related to Unix system administration:

General information http://www.ensta.fr/internet/unix/sys_admin/ or http://www.sai.msu.su/sysadm.html

There is a Unix system administration tutorial at http://www.iem.ac.ru/sysadm.html

UnixWorld Online Magazine Home Page http://www.wcmh.com/uworld/

Internet Essentials for UNIX System Administrators Tutorial http://www.greatcircle.com/tutorials/ieusa.html

Pointers to Unix goodies available on the Internet http://www.ensta.fr/internet/unix/

Pointers to Unix system administration "goodies" available on the Internet http://www.ensta.fr/internet/unix/sys_admin/

5. Stage 4. Emacspeak under Linux

The Slackware setup script for Emacspeak should create the needed environment variables and install a script `emacspeak` that starts emacs with emacspeak. This is your fourth option for learning Emacs. This is the first time you will be able to actually use Emacspeak. A short tutorial appears below. Within Emacs, you may type C−h C−e to get a list of the commands. To search for a command, use C−h a. To get an explanation for a key sequence, use C−h k. There is also an info file which is part of the Emacspeak distribution. Within emacs, you may type C−h i to open the directory to the info pages. Search for the emacspeak menu item by typing C−s emacspeak, then two carriage returns (one to terminate the search, and a second one to go to the info page). If you have the standalone info program installed, you can consult the info file with the command `info Emacspeak`.

5.1 Emacspeak Introduction − Speech Enabled Normal Commands

All of the normal Emacs movement commands will speak the relevant information after moving. Here are some of the cursor movement functions that have been speech enabled. Note that this list only enumerates a few of these speech enabled commands; the purpose of emacspeak is to speech−enable all of emacs and provide you spoken feedback as you work. Thus, this list is here only as a representative example of the kind of speech−enabling extensions Emacspeak provides.

`C−n' or `M−x next−line' or `down'

Moves the cursor to the next line and speaks it.

`C−p' or `M−x previous−line' or `up'

Moves the cursor to the previous line and speaks it.

`M−f' or `M−x forward−word' or

Moves the cursor to the next word and speaks it. Places point on the first character of the next work, rather than on the space preceding it (This is my personal preference).

`M−b' or `M−x backward−word'

Moves the cursor to the previous word and speaks it.

`M−C−b' or `M−x backward−sexp'

Moves the cursor to the previous sexp and speaks it. If the sexp spans more than a line, only the first line is spoken.

`M−<' or `M−x beginning−of−buffer'

Speaks line moved to.

`M−>' or `M−x end−of−buffer'

Speaks line moved to.

`M−m' or `M−x back−to−indentation'

Speaks entire current line. A useful way of hearing the current line.

5.2 Emacspeak Introduction − New Commands

Emacspeak provides a number of commands for reading portions of the current buffer, getting status information, and modifying Emacspeak's state.

All of the commands are documented in the subsequent sections. They can be classified into types:

The first group of commands are for listening to chunks of information. The names of these commands all start with the common prefix `emacspeak−'. All Emacspeak commands are bound to the keymap EMACSPEAK−KEYMAP and are accessed with the key `Control e'. Thus, the Emacspeak command "emacspeak−speak−line" is bound to `l' in keymap EMACSPEAK−KEYMAP and can be accessed with the keystroke `Control−e l'.

Here are some of the commands for reading text:

`C−e c' or `M−x emacspeak−speak−char'

Speak current character, using the phonetic alphabet.

`C−e w' or `M−x emacspeak−speak−word'

Speak current word.

`C−e l' or `M−x emacspeak−speak−line'

Speak current line. With prefix `C−u', speaks the rest of the line from point. With negative prefix

`C−u −', speaks from start of line to point. Voicifies if voice−lock−mode is on. Indicates indentation with a tone if audio indentation is in use. Indicates position of point with an aural highlight if option emacspeak−show−point is turned on −−see command `M−x emacspeak−show−point'.

`C−e .' or `M−x emacspeak−speak−sentence'

Speak the current sentence.

The second category of commands provided by Emacspeak report status of various kinds.

`C−e k' or `M−x emacspeak−speak−current−kill'

Speak the current kill entry (which would be yanked by the next C−y).

`C−e =' or `M−x emacspeak−speak−current−column'

State the column where point is.

`C−e C−l' or `M−x emacspeak−speak−line−number'

State the line where point is.

`C−e m' or `M−x emacspeak−speak−line−number'

Speak the mode−line: the name of the buffer, how far point is into the current buffer as a percentage, and the major mode.

`C−e M' or `M−x emacspeak−speak−minor−mode−line'

Speak the minor modes that are in effect.

`C−e v' or `M−x emacspeak−speak−version'

Announce the emacspeak version.

The third category of commands provided by Emacspeak manipulate the state of the speech device. The names of these commands start with the common prefix `dtk−'. Many of these commands take the prefix `Control−e d'. Thus, the command "dtk−set−rate" is bound to `r' in keymap EMACSPEAK−DTK−SUBMAP and can be executed by pressing `Control e d r'.

`C−e s' or `M−x dtk−stop'

Stop speech now. In addition, any command that causes speech output will discard anything in the speech buffer.

`C−e d V' or `M−x emacspeak−dtk−speak−version'

Use this to find out which version of the Dectalk firmware you have.

`C−e d I' or `M−x dtk−toggle−stop−immediately−while−typing'

Toggle state of variable dtk−stop−immediately−while−typing. As the name implies, if true then speech flushes immediately as you type.

`C−e d i' or `M−x emacspeak−toggle−audio−indentation'

Toggle state of Emacspeak audio indentation. Specifying the method of indentation as `tone' results in the DECtalk producing a tone whose length is a function of the line's indentation. Specifying `speak' results in the number of initial spaces being spoken.

`C−e d k' or `M−x emacspeak−toggle−character−echo'

Toggle state of Emacspeak character echo (that is, whether typed characters are echoed).

`C−e d w' or `M−x emacspeak−toggle−word−echo'

Toggle state of Emacspeak word echo (initially on).

`C−e d l' or `M−x emacspeak−toggle−line−echo'

Toggle state of Emacspeak line echo (that is, whether typed text is echoed after typing enter).

`C−e d p' or `M−x dtk−set−punctuations'

Set punctuation state. Possible values are `some', `all', or `none'.

`C−e d q' or `M−x dtk−toggle−quiet'

Toggle state of the speech device between being quiet and talkative. Useful if you want to continue using an emacs session that has emacspeak loaded but wish to make the speech shut up.

`C−e d R' or `M−x dtk−reset−state'

Restore sanity to the Dectalk. Typically used after the Dectalk has been power cycled.

`C−e d SPC' or `M−x dtk−toggle−splitting−on−white−space'

Toggle state of emacspeak that decides if we split text purely by clause boundaries, or also include whitespace.

`C−e d r' or `M−x dtk−set−rate'

Set speaking rate for the dectalk.

`C−e d s' or `M−x dtk−toggle−split−caps'

Toggle split caps mode. In split caps mode, a transition from lower case to upper case is treated like the beginning of a new word. This is useful when reading Hungarian notation in program source code.

`C−e d v' or `M−x voice−lock−mode'

Toggle Voice Lock mode (initially off). When Voice Lock mode is enabled, text is voiceified as you type it, as follows:

- Comments are spoken in `voice-lock-comment-personality`; (That is a variable whose value should be a personality name.)
- Strings are spoken in `voice-lock-string-personality`.
- Documentation strings are spoken in `voice-lock-doc-string-personality`.
- Function and variable names in their defining forms are spoken in `voice-lock-function-name-personality`.
- Certain other expressions are spoken in other personalities according to the value of the variable `voice-lock-keywords`.

5.3 Emacspeak Introduction − Using the Help System

When you press C−h to get the help index, the screen will appear, but Emacspeak will not speak the window. The only thing spoken is "Type one of the options listed or Space to scroll:".

Here is the menu that Emacspeak is not speaking:

```
-- begin quote
```

```
You have typed C-h, the help character.  Type a Help option:
(Use SPC or DEL to scroll through this text.  Type q to exit the Help
command.)

a  command-apropos.  Give a substring, and see a list of commands
        (functions interactively callable) that contain
        that substring.  See also the  apropos  command.
b  describe-bindings.  Display table of all key bindings.
c  describe-key-briefly.  Type a command key sequence;
        it prints the function name that sequence runs.
f  describe-function.  Type a function name and get documentation of
it.
C-f Info-goto-emacs-command-node.  Type a function name;
        it takes you to the Info node for that command.
F  view-emacs-FAQ.  Shows emacs frequently asked questions file.
i  info. The  info  documentation reader.
k  describe-key.  Type a command key sequence;
        it displays the full documentation.
C-k Info-goto-emacs-key-command-node.  Type a command key sequence;
        it takes you to the Info node for the command bound to that
key.
l  view-lossage.  Shows last 100 characters you typed.
m  describe-mode.  Print documentation of current major mode,
        which describes the commands peculiar to it.
n  view-emacs-news.  Shows emacs news file.
p  finder-by-keyword. Find packages matching a given topic keyword.
s  describe-syntax.  Display contents of syntax table, plus
explanations
t  help-with-tutorial.  Select the Emacs learn-by-doing tutorial.
v  describe-variable.  Type name of a variable;
        it displays the variable's documentation and value.
w  where-is.  Type command name; it prints which keystrokes
        invoke that command.
C-c print Emacs copying permission (General Public License).
C-d print Emacs ordering information.
```

```
        C-n print news of recent Emacs changes.
        C-p print information about the GNU project.
        C-w print information on absence of warranty for GNU Emacs.

-- end quote
```

Suppose you type "a", for command−apropos.

The next spoken prompt is "Apropos command (regexp):"

Now you type some word you think is part of an emacs command, like "visit".

The help system will display the first section of the help, but will leave the cursor in the other window. The spoken text is "Type C−x 1 to remove help window. M−C−v to scroll the help." At this point, I think it's more helpful to move point to the other window with C−x o, then you can use regular navigation commands to speak the help text. You can delete the help window with C−x 0, which will also put point back where it was.

The complete menu displayed by help−for−help is also visible if you do a describe function on help−for−help. In a future version of Emacspeak, Raman plans to add a message to that effect when the user presses C−h ?

6. Footnotes and References

6.1 JAWS

Job Access With Speech (JAWS) is a screen reader which runs under Microsoft MSDOS. JAWS for Windows supports Windows 95, Windows 98 and Windows NT. Both are products of Henter−Joyce, Inc., 2100 62nd Avenue North, St. Petersburg, FL 33702, telephone: 800−336−5658. For details, see http://www.hj.com.

6.2 TELIX

TELIX is a shareware terminal emulator for MSDOS. It can be obtained by FTP from the SimTel archive [SimTel]. Within the SimTel collection, look for directory msdos/telix. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/telix. The latest version of the program itself is in the four files tlx322−1.zip, tlx322−2.zip, tlx322−3.zip, and tlx322−4.zip.

6.3 COMMO

COMMO is another shareware terminal emulator for DOS. In the SimTel archive [SimTel], it is in directory msdos/commprog, file `commo66.zip`. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/commprog/commo66.zip.

6.4 SimTel

The SimTel archive is maintained by Keith Petersen `w8sdz@Simtel.Net`. CD−ROM copies of Simtel.Net collections are available from Walnut Creek CDROM [Walnut Creek]. The primary ftp sites are ftp://ftp.simtel.net/pub/simtelnet, and oak.oakland.edu://pub/simtelnet. There are many mirror sites, as listed in the following table:

- US, ALL (primary) ftp://ftp.simtel.net/pub/simtelnet
- US, California ftp://ftp.cdrom.com/pub/simtelnet
- US, California ftp://ftp.digital.com/pub/micro/pc/simtelnet
- US, California ftp://ftp.lib.sonoma.edu/pub/simtelnet
- US, Illinois ftp://uiarchive.cso.uiuc.edu/pub/systems/pc/simtelnet
- US, Massachusetts ftp://ftp.bu.edu/pub/mirrors/simtelnet
- US, Michigan ftp://oak.oakland.edu/pub/simtelnet
- US, New York ftp://ftp.rge.com/pub/systems/simtelnet
- US, Oklahoma ftp://ftp.ou.edu/pub/simtelnet
- US, Oregon ftp://ftp.orst.edu/pub/simtelnet
- US, Pennsylvania ftp://ftp.epix.net/pub/simtelnet
- US, Utah ftp://ftp.cyber−naut.com/pub/simtelnet
- US, Virginia ftp://mirrors.aol.com/pub/simtelnet
- Argentina ftp://ftp.satlink.com/pub/mirrors/simtelnet
- Australia ftp://ftp.iniaccess.net.au/pub/simtelnet
- Australia ftp://sunsite.anu.edu.au/pub/pc/simtelnet
- Austria, Vienna ftp://ftp.univie.ac.at/mirror/simtelnet
- Belgium ftp://ftp.linkline.be/mirror/simtelnet
- Belgium ftp://ftp.tornado.be/pub/simtelnet
- Bulgaria ftp://ftp.eunet.bg/pub/simtelnet
- Brazil ftp://ftp.iis.com.br/pub/simtelnet
- Brazil ftp://ftp.unicamp.br/pub/simtelnet
- Canada, Ottawa ftp://ftp.crc.doc.ca/systems/ibmpc/simtelnet
- Canada, Vancouver ftp://ftp.direct.ca/pub/simtelnet
- Chile ftp://sunsite.dcc.uchile.cl/pub/Mirror/simtelnet
- China ftp://ftp.pku.edu.cn/pub/simtelnet
- Czech Republic ftp://ftp.eunet.cz/pub/simtelnet
- Czech Republic ftp://ftp.zcu.cz/pub/simtelnet
- Czech Republic ftp://pub.vse.cz/pub/simtelnet
- Finland ftp://ftp.funet.fi/mirrors/ftp.simtel.net/pub/simtelnet
- France ftp://ftp.grolier.fr/pub/simtelnet
- France ftp://ftp.ibp.fr/pub/simtelnet
- Germany ftp://ftp.mpi−sb.mpg.de/pub/simtelnet
- Germany ftp://ftp.rz.ruhr−uni−bochum.de/pub/simtelnet
- Germany ftp://ftp.tu−chemnitz.de/pub/simtelnet
- Germany ftp://ftp.uni−heidelberg.de/pub/simtelnet
- Germany ftp://ftp.uni−magdeburg.de/pub/mirrors/simtelnet
- Germany ftp://ftp.uni−paderborn.de/pub/simtelnet
- Germany ftp://ftp.uni−trier.de/pub/pc/mirrors/Simtel.net
- Germany ftp://ftp.rz.uni−wuerzburg.de/pub/pc/simtelnet
- Greece ftp://ftp.ntua.gr/pub/pc/simtelnet
- Hong Kong ftp://ftp.cs.cuhk.hk/pub/simtelnet
- Hong Kong ftp://ftp.hkstar.com/pub/simtelnet
- Hong Kong ftp://sunsite.ust.hk/pub/simtelnet
- Ireland ftp://ftp.iol.ie/pub/simtelnet
- Israel ftp://ftp.huji.ac.il/pub/simtelnet
- Italy ftp://cis.utovrm.it/simtelnet
- Italy ftp://ftp.flashnet.it/pub/simtelnet
- Italy ftp://ftp.unina.it/pub/simtelnet

- Italy ftp://mcftp.mclink.it/pub/simtelnet
- Japan ftp://ftp.iij.ad.jp/pub/simtelnet
- Japan ftp://ftp.riken.go.jp/pub/simtelnet
- Japan ftp://ftp.saitama−u.ac.jp/pub/simtelnet
- Japan ftp://ftp.u−aizu.ac.jp/pub/PC/simtelnet
- Japan ftp://ftp.web.ad.jp/pub/simtelnet
- Japan ftp://ring.aist.go.jp/pub/simtelnet
- Japan ftp://ring.asahi−net.or.jp/pub/simtelnet
- Latvia ftp://ftp.lanet.lv/pub/mirror/simtelnet
- Malaysia ftp://ftp.jaring.my/pub/simtelnet
- Malaysia ftp://ftp.mimos.my/pub/simtelnet
- Mexico ftp://ftp.gdl.iteso.mx/pub/simtelnet
- Netherlands ftp://ftp.euro.net/d5/simtelnet
- Netherlands ftp://ftp.nic.surfnet.nl/mirror−archive/software/simtelnet
- New Zealand ftp://ftp.vuw.ac.nz/pub/simtelnet
- Norway ftp://ftp.bitcon.no/pub/simtelnet
- Poland ftp://ftp.cyf−kr.edu.pl/pub/mirror/Simtel.Net
- Poland ftp://ftp.icm.edu.pl/pub/simtelnet
- Poland ftp://ftp.man.poznan.pl/pub/simtelnet
- Portugal ftp://ftp.ip.pt/pub/simtelnet
- Portugal ftp://ftp.ua.pt/pub/simtelnet
- Romania ftp://ftp.sorostm.ro/pub/simtelnet
- Singapore ftp://ftp.nus.sg/pub/simtelnet
- Slovakia ftp://ftp.uakom.sk/pub/simtelnet
- Slovenia ftp://ftp.arnes.si/software/simtelnet
- South Africa ftp://ftp.is.co.za/pub/simtelnet
- South Africa ftp://ftp.sun.ac.za/pub/simtelnet
- South Korea ftp://ftp.nuri.net/pub/simtelnet
- South Korea ftp://ftp.sogang.ac.kr/pub/simtelnet
- South Korea ftp://sunsite.snu.ac.kr/pub/simtelnet
- Spain ftp://ftp.rediris.es/mirror/simtelnet
- Sweden ftp://ftp.sunet.se/pub/simtelnet
- Switzerland ftp://sunsite.cnlab−switch.ch/mirror/simtelnet
- Taiwan ftp://ftp.ncu.edu.tw/Packages/simtelnet
- Taiwan ftp://nctuccca.edu.tw/mirror/simtelnet
- Thailand ftp://ftp.nectec.or.th/pub/mirrors/simtelnet
- UK, Edinburgh ftp://emwac.ed.ac.uk/mirrors/simtelnet
- UK, London ftp://ftp.demon.co.uk/pub/simtelnet
- UK, Lancaster ftp://micros.hensa.ac.uk/pub/simtelnet
- UK, London ftp://sunsite.doc.ic.ac.uk/packages/simtelnet

6.5 InfoMagic

InfoMagic is at 11950 N. Highway 89, Flagstaff AZ 86004, telephone 800−800−6613 or 520−526−9565, fax 520−526−9573, email: info@infomagic.com, web: http://www.infomagic.com.

6.6 Linux Central

Linux Central is at Linux Central Inc., 37060 Garfield Ste. T2, Clinton Township MI 48036, telephone 810−226−8200, fax 810−226−8600, email: sales@linuxcentral.com, web: http://linuxcentral.com.

6.7 Walnut Creek

Walnut Creek CDROM has many useful CDROMs. They are at 4041 Pike , Concord, CA 94520, USA. Telephone (800) 786−9907 or (510) 674−0783, or FAX (510) 674−0821. email: orders@cdrom.com. Web: http://www.cdrom.com/

6.8 Red Hat

Red Hat Software: telephone 800−454−5502 or 203−454−5500, fax: 203−454−2582, email: sales@redhat.com. Web: http://www.redhat.com.

6.9 Craftwork

CraftWork Solutions, 4320 Stevens Creek Blvd, Suite 170, San Jose CA 95129, telephone 800−985−1878, email: info@craftwork.com, web: http://www.craftwork.com.

6.10 Yggdrasil

Yggdrasil Computing, 4880 Stevens Creek Blvd., Suite 205, San Jose CA 95129−1024, telephone 800−261−6630 or 408−261−6630, fax: 408−261−6631, email: info@yggdrasil.com, web: http://www.yggdrasil.com.

6.11 Emacs for DOS

From the Emacs FAQ of November 11, 1996:

```
--begin quote
```

93: Where can I get Emacs for my PC running MS−DOS?

A pre−built binary distribution of Emacs 19.34 should be available by the beginning of November 1996 from the Simtel archives, the main site of which is at

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp/v2gnu/

If you prefer to compile Emacs for yourself, you will need a 386 (or better) processor, and are running MS−DOS 3.0 or later. According to Eli Zaretskii `eliz@is.elta.co.il` and Darrel Hankerson `hankedr@dms.auburn.edu`, you will need the following:

Compiler: djgpp version 1.12 maint 1 or later. Djgpp 2.0 or later is recommended, since 1.x is being phased out. Djgpp 2 supports long filenames under Windows 95.

You can get the latest release of djgpp by retrieving all of the files in

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp

Gunzip and tar:

The easiest way is to use "djtar" which comes with djgpp v2.x, because it can open gzip'ed tarfiles (i.e., those ending with ".tar.gz") in one step. Djtar comes in "djdev201.zip", from the URL mentioned above.

Utilities: make, mv, sed, rm.

All of these utilities are available at

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp/v2gnu

16−bit utilities can be found in GNUish:

ftp://ftp.simtel.net/pub/simtelnet/gnu/gnuish

The file INSTALL in the top−level directory of the Emacs source contains some additional information regarding Emacs under MS−DOS. In addition, the file etc/MSDOS contains some information on the differences between the Unix and MS−DOS versions of Emacs.

For the most comprehensive information on running GNU Emacs on a PC, see the file prepared by Michael Ernst `mernst@theory.lcs.mit.edu` at

ftp://theory.lcs.mit.edu/pub/emacs/pc−emacs.gz

For a list of other MS−DOS implementations of Emacs (and Emacs look−alikes), consult the list of "Emacs implementations and literature," available at

ftp://rtfm.mit.edu/pub/usenet/comp.emacs/

Note that while many of these programs look similar to Emacs, they often lack certain features, such as the Emacs Lisp extension language.

```
--end quote
```

6.12 GNU Mirror Sites

The GNU collection at ftp://prep.ai.mit.edu/pub/gnu is mirrored at many sites. Mirrors in USA include these:

- ftp://labrea.stanford.edu/pub/gnu
- ftp://wuarchive.wustl.edu/systems/gnu
- ftp://ftp.kpc.com/pub/mirror/gnu
- ftp://f.ms.uky.edu/pub3/gnu
- ftp://jaguar.utah.edu/gnustuff
- ftp://ftp.hawaii.edu/mirrors/gnu
- ftp://uiarchive.cso.uiuc.edu/gnu
- ftp://uiarchive.cso.uiuc.edu/pub/gnu
- ftp://ftp.cs.columbia.edu/archives/gnu/prep
- ftp://gatekeeper.dec.com/pub/GNU
- ftp://ftp.uu.net/systems/gnu

6.13 Emacspeak with Earlier Slackware Releases

If you want to install Slackware 3.0 or earlier, you will need to prepare a full null modem cable, including modem control signals.

For two DB25 (25 pin) connectors, the required connections are:

- 1 (Frame Ground) – 1 (Frame Ground)
- 2 (Receive Data) – 3 (Transmit Data)
- 3 (Transmit Data) – 2 (Receive Data)
- 4 (Request To Send) – 5 (Clear To Send)
- 5 (Clear To Send) – 4 (Request To Send)
- 6 (Data Set Ready) – 20 (Data Terminal Ready)
- 7 (Signal Ground) – 7 (Signal Ground)
- 8 (Carrier Detect) – 20 (Data Terminal Ready)
- 20 (Data Terminal Ready) – 6 (Data Set Ready)
- 20 (Data Terminal Ready) – 8 (Carrier Detect)

For two DB9 connectors, the connections are:

- 1 (Carrier Detect) – 4 (Data Terminal Ready)
- 2 (Receive Data) – 3 (Transmit Data)
- 3 (Transmit Data) – 2 (Receive Data)
- 4 (Data Terminal Ready) – 6 (Data Set Ready)
- 4 (Data Terminal Ready) – 1 (Carrier Detect)
- 5 (Signal Ground) – 5 (Signal Ground)
- 6 (Data Set Ready) – 4 (Data Terminal Ready)
- 7 (Request To Send) – 8 (Clear To Send)
- 8 (Clear To Send) – 7 (Request To Send)
- 9 (Ring Indicator) not connected

For a DB9 (listed first) to a DB25 (second), the connections are:

- 1 (Carrier Detect) – 20 (Data Terminal Ready)
- 2 (Receive Data) – 2 (Transmit Data)
- 3 (Transmit Data) – 3 (Receive Data)
- 4 (Data Terminal Ready) – 6 (Data Set Ready)
- 4 (Data Terminal Ready) – 8 (Carrier Detect)
- 5 (Signal Ground) – 7 (Signal Ground)
- 6 (Data Set Ready) – 20(Data Terminal Ready)
- 7 (Request To Send) – 5 (Clear To Send)
- 8 (Clear To Send) – 4 (Request To Send)
- 9 (Ring Indicator) not connected

---

7. Frequently Asked Questions (FAQ)

7.1 Why does it say "space" after each character?

Your DECtalk Express has old firmware. Use the Emacspeak command `C−e d V' to find out your version. You should be running a version no older than 4.2bw from March 1995. If you have an earlier version, you can find an updated version at http://www.ultranet.com/~rongemma/tips_upd.htm, a WWW site maintained by Ron Jemma of the Dectalk Group at DEC. Alternatively, you can send email to Anne Nelson at DEC`nelson@dectlk.enet.dec.com`. The most recent version at this writing is 4.3 release AA X01 May 20 1996.

7.2 Why the high pitched tone from a DECtalk?

On occasion when reading the DECtalk will produce high pitch tones that last for several words or more, if this happens in a buffer it will often repeat within the same buffer.

The problem is due to remaining bugs in the Dectalk firmware. When emacspeak produces tones, especially when split caps is on, the dtk sometime goes into squealing mode.

If you notice this happening in particular text documents, just turn off split caps mode locally with `C−e d s'.

7.3 Why the "No library autorevert in search path" message?

What is the significance of the message "No library autorevert in search path" which is spoken (or at least queued for speech) when emacspeak 7.0 starts?

It's completely insignificant. autorevert is a new minor mode provided by emacs 20 and emacspeak looks for it and if found speech−enables it.

7.4 How do I change Emacspeak defaults?

What is the best way to change Emacspeak defaults for things such as speech rate, default punctuation mode, etc.?

Just put the needed changes in your .emacs file. ".emacs" is a file in your home directory that is loaded by emacs when it starts (shortly after reading emacspeak). It is the place to do personal modifications.

Here is an example which resets the speech rate and character scale, enables auditory icons, disables character echo and split caps mode, and redefines a function (courtesy of Greg Priest−Dorman priestdo@cs.vassar.edu and Matt Campbell mattcamp@feist.com):

```
;;
(dtk-set-rate 380 t)
(emacspeak-toggle-auditory-icons t)
(dtk-set-character-scale 1.05 t)
(emacspeak-toggle-character-echo t)
(dtk-toggle-split-caps t)
;;
(defun emacspeak-speak-time ()
  "Speak the time. "
  (interactive)
```

```
  (dtk-speak  (format-time-string "%A %B %e, %I %M %p"
      (current-time) )))
;;
```

7.5 What if an error message is not spoken?

An error message was displayed on the screen, but not reported by Emacspeak

In general, if emacs hits an error and you don't get to hear the message because it went by too quickly, here is what you should do:

1) Switch to the *Messages* buffer and see if you see the message there. 99% of the time it will be there.

2) If some portion of emacs or an emacs application throws an error without leaving an appropriate message in the *Messages* buffer, then turn on emacs' debugging facility by `M−x set−variable ret' `debug−on−error ret' `t ret' this sets debugon−on−error to true now repeat what caused the error you will be placed in a backtrace buffer.

Warning: do not read the entire contents of the backtrace buffer at one shot with emacspeak. Go line by line. (courtesy of T. V. Raman `raman@Adobe.COM`)

7.6 The terminal emulator is the most problematic portion of Emacspeak.

It's not an emacspeak problem−− it's a terminal problem. Getting a terminal emulator working correctly inside emacs is not a joke−− and eterm is an excellent piece of work, but depending on how some old app behaves you will (or should always expect to) see some weirdness.

After all, remember all of these UNIX terminal apps work from all kinds of terminals ranging from vt52's to new devices.

7.7 In a term buffer, the first half screen is not spoken

When starting a term buffer, no output is spoken in line mode by Emacspeak until the cursor has made it about half way down the screen

I've not seen this−− and I am running emacs 20.

When you start term it is in character mode. When you switch to line mode make sure the cursor is at the end of the buffer−− otherwise what you send will never go to the shell.

Also, in general use M−x shell (comint−mode) for running things you would run in line mode in eterm−− it's been around longer and far more robust. M−x term specializes in its character−mode of terminal emulation and may not always take care of everything in line mode. (courtesy of T. V. Raman `raman@Adobe.COM`)

7.8 Term mode gets flaky after a long session.

Sometimes, after long sessions with a term buffer, I will experience similar problems as do you with spech in the term becoming very flaky. Output may or may not be spoken (I have not determined the pattern).

if the term becomes flaky, enter character mode and issue a reset. the above is usually a result of running some curses based app that leaves the terminal in a screwy state.

Many curses based apps follow slightly different conventions on where they leave the cursor while you edit–– 60–70% of them leave the cursor just after the character you typed with a null character after point. Others leave a space character there which is what you hear.

Experiment between setting TERM to vt100 and eterm. Depending on whether the app was linked against curses or ncurses (former uses TERMCAP and latter uses TERMINFO) you will get different results.

8. Legalese

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `tjbynum@metalab.unc.edu` or `linux-howto@metalab.unc.edu`.