

NAME

stty — inspect and alter teletype and window attributes

SYNOPSIS

```
stty [-F teletype]
stty [-F teletype] -a
stty [-F teletype] -g
stty [-F teletype] saved-settings
stty [-F teletype] size|speed|attribute...
```

DESCRIPTION

Writes (some) attributes of *teletype* (the standard input stream by default) or sets them.

By default, a difference from **stty sane** is written as such:

```
speed 38400 baud; rows 48; cols 128; line = 1;
iutf8 ...
```

or as such:

```
ispeed 38400 baud; ospeed 19200 baud; rows 54; cols 226; line = 0;
kill = ^Y;
-brkint -imaxbel ...
```

if the speeds differ.

The first line lists the teletype speed in bits per second (which likely means very little if it's a pty(4)), the kernel's idea of the window size, and the current line discipline.

The next line lists the **Special Control Characters** in a visual format — "<undef>" if disabled, "M-" if top bit set, then literal if printable or *^b*.

Subsequent lines list the individual *attributes* for **Input Attributes**, **Output Attributes**, **Control Attributes**, and **Local Attributes**.

OPTIONS

- F** *teletype* Consult/modify *teletype*, opened O_RDONLY | O_NONBLOCK, instead of the standard input stream.
- a**, --**all** Instead of the control characters and attributes that differ between the current settings and the current settings with **sane** applied, list *all* known characters and attributes of the current settings, negated if applicable.
- g**, --**save** Write current settings in a concise reproducible format suitable for use as *saved-settings*, cf. **Saved Settings**, below.

ATTRIBUTES

Specifying any attribute suppresses the default output, except for [-]**drain**.

Queries

Write information about *teletype* to the standard output stream.

Query	Information	Format
size	Window size	Rows, then columns, separated by a space
speed	Baud-rate	Single number if the same, otherwise input speed, space, output speed

Special Control Characters

Must be followed by either the empty string, meaning **0x00**, a single byte, used verbatim, "^-" or "undef", disabling the character, "^?" for DEL (**0x7F**, bottom 7 bits set), "*^b*" where *b* is any byte, of which the bottom **5** bits are used (this most likely corresponds to holding the Control key on your keyboard, then pressing the desired character, and yields distinct favourable results for @, the alphabet, brackets, \, ^, and _), or an integer.

cc	Index	Purpose	When
intr	VINTR	Send SIGINT to foreground process group	isig
quit	VQUIT	Send SIGQUIT to foreground process group	isig
erase	VERASE	Erase last character	icanon
kill	VKILL	Kill whole line	icanon
eof	VEOF	Discard the eof and terminate line immediately	icanon
eol	VEOL	Alternative keying for the newline	icanon
eol2	VEOL2	Likewise	icanon
swtch	VSWTCH	cf. HISTORY, UNIX System V Release 2.0	SVr2 & sh1(1)
start	VSTART	Resume output suspended with stop	ixon, ixoff
stop	VSTOP	Temporarily halt output to <i>teletype</i>	ixon, ixoff
susp	VSUSP	Send SIGTSTP to foreground process group	isig
rprnt	VREPRINT	Re-echo current input line	ixexten & echo
discard	VDISCARD	Toggle flusho	ixexten
werase	VWERASE	Erase last word	icanon
lnext	VLNEXT	Next byte is literal — all processing is turned off	ixexten
dsusp	VDSUSP	Like susp , but sent when the byte would be read, not instantly	isig
status	VSTATUS	Send SIGINFO to foreground process group	icanon

These are also part of the **Special Control Characters**, but aren't actually characters, and control non-canonical input; as such, they take an integer only:

cc	Index	Purpose	When
min	VMIN	Minimum bytes to return from a read()	-icanon
time	VTIME	Tenths of a second to allow a read() to block	-icanon

If both are nonzero, wait at most **time** after the last received byte that didn't satisfy **min** to return. If neither — **read()** returns immediately with whatever's available.

Input Attributes

Except where specified otherwise, these all correspond to flags of the same name, but in upper-case. A BREAK means being on the receiving end of a `tcsendbreak(3)`.

Attrib.	Inverse	Purpose	When
ignbrk	-ignbrk	Discard BREAKs	
brkint	-brkint	Send SIGINT to foreground process group under BREAK	-ignbrk
ignpar	-ignpar	Discard framing/parity errors	
parmrk	-parmrk	(see below)	
inpck	-inpck	Check received parity for correctness	
istrip	-istrip	Strip input bytes to 7 bits	
inlcr	-inlcr	Convert input newlines to carriage returns	
igncr	-igncr	Discard input carriage returns	
icrnl	-icrnl	Convert input carriage returns to newlines	
iucle	-iucle	Convert upper-case input bytes to lower-case	ixexten, after istrip
ixon	-ixon	Handle start and stop and as-described	
ixany	-ixany	Allow any input byte to resume stopped output	ixon
ixoff	-ixoff	Send start and stop to the input to prevent input queue overflow	
imaxbel	-imaxbel	Ring bell when input queue full	
iutf8	-iutf8	Process input line as UTF-8 instead of byte-wise	icanon

If **-ignpar parmrk**, framing/parity errors yield bytes **0xFF**, **0**, then the erroneous one; to disambiguate, normal **0xFF** bytes are doubled unless **istrip** in this case. If **-ignpar -parmrk** — a single **0** byte.

If **-ignbrk -parmrk**, a break condition yields a single **0** byte, if **-ignbrk parmrk** — **0xFF**, **0**, **0**.

Output Attributes

These allow correct output to quirky or moving teletypes.

Attrib. Inverse Purpose

opost	-opost	Enable all other Output Attributes
olcuc	-olcuc	Convert lower-case bytes to upper-case for output
onlcr	-onlcr	Output newline as carriage return, newline
ocrnl	-ocrnl	Output carriage return as newline
onocr	-onocr	Don't output carriage return at the first column
onlret	-onlret	Assume newline returns carriage to first column; use cr? delays instead
ofill	-ofill	Transmit 0 bytes as filler instead of using time-based delays
ofdel	-ofdel	Use DEL (0x7F)s instead of 0s

n10	No newline delay
n11	Approximately 0.10s or two filler bytes
cr0	No carriage return delay
cr1	Dependent on current column or two filler bytes
cr2	Approximately 0.10s or four filler bytes
cr3	Approximately 0.15s
tab0	No tab delay
tab1	Dependent on current column or two filler bytes
tab2	Approximately 0.10s or two filler bytes
tab3	Expand tabs to spaces
bs0	No backspace delay
bs1	Approximately 0.05s or a filler byte
ff0	No form-feed delay
ff1	Approximately 2s
vt0	No vertical tab delay
vt1	Approximately 2s

Control Attributes

Attrib.	Inverse	Purpose	Inverse Purpose
		cs5 <i>teletype</i> bytes are 5 bits	
		cs6 <i>teletype</i> bytes are 6 bits	
		cs7 <i>teletype</i> bytes are 7 bits	
		cs8 <i>teletype</i> bytes are 8 bits	

cstopb	-cstopb	Send two stop bits	Send one
cread	-cread	Enable receiver	Don't receive (read) anything
parenb	-parenb	Generate a parity bit for each byte	Don't
parodd	-parodd	Odd parity	Even parity
hupcl	-hupcl	Hang up when the last file descriptor for <i>teletype</i> is closed	
clocal	-clocal	Local line — ignore link status	
cmspar	-cmspar	(see below)	Normal parity
crtsets	-crtsets	Flow control	No flow control

If **cmspar -parodd**: space (always-**0**) parity; if **cmspar parodd**: mark (always-**1**) parity.

Local Attributes

Attrib.	Inverse	Purpose	Inverse Purpose	When
isig	-isig	Send signals when intr , quit , susp , or dsusp		
icanon	-icanon	Canonical editable line-buffered input	Raw input, minimally -buffered	
xcase	-xcase	(see below)		icanon
echo	-echo	Echo input back	Input is hidden	
echoe	-echoe	(see below)		icanon & echo

echok	-echok	Represent kill (per [-]echoke)	Echo kill per echoctl	icanon & echo
echoke	-echoke	kill by erasing the whole line	kill by echoing a newline	↑ & echok
echonl	-echonl	Echo newlines regardless of echo		icanon
noflsh	-noflsh	Don't discard buffers when sending signals due to isig		isig
tostop	-tostop	Send SIGTTOU to background process group attempting output		
echoctl	-echoctl	Echo control bytes (sans start , stop) as ^b instead of verbatim		echo
echoprt	-echoprt	(see below)		icanon & echo
flusho	-flusho	Output discarded	Output flowing	
iexten	-iexten	Enable implementation-defined flags		
extproc	-extproc	External processing for pty(4) s, cf. ioctl(TIOCPKT)		
altwerase	-altwerase	Use alternative word-breaking algorithm for werase		

Blocking **SIGTTOU** has the same effect as setting **-tostop** for the process would.

On systems where **xcase** does anything, a backslash followed by a lowercase byte is converted to an up-percase byte, and a backslash followed by any of **' ! ^ () ** becomes **` | ~ { } **, respectively. Output is written with the mapping applied in reverse.

When erasing (**icanon**) (via **erase**, **werase**, **kill** with **echok** **echoke**, &c.), if **echo**:

if **echoprt** echo a backslash, each erased column in erasure order; if another byte on the same line is echoed, the first is preceded with a slash; this is friendly to *teletypes* that display to write-once media like paper,

if **echoe** and erased via **erase**
echo **erase**,

otherwise, echo backspace, space, backspace; this works best for re-writable media since it hides the erased bytes.

On systems where **flusho** does anything, it's enabled by **discard** and disabled by any input.

Line Speed

Attrib.	Purpose
<i>speed</i>	Equivalent to ispeed <i>speed</i> ospeed <i>speed</i>
ispeed <i>speed</i>	Sets input baud-rate to <i>speed</i> , or same as ospeed if 0
ospeed <i>speed</i>	Sets output baud-rate to <i>speed</i> , or hangs up if 0

Where *speed* is any of:

50	200	2400	57600	576000	2000000
75	300	4800	115200	921600	2500000
110	600	9600	230400	1000000	3000000
134.5	1200	19200	460800	1152000	3500000
150	1800	38400	500000	1500000	4000000

all of which correspond to directly to the gross-bits-per-second speed of the link, or one of the following:

0	(see above)
134	134.5
exta	19200
extb	38400

Under Linux, **38400** potentially means **57600**, **115200**, **230400**, **460800**, or some division of the clock, depending on additional link-specific configuration (**ioctl(TIOCSSERIAL)**, **setserial(8)** **spd_hi/spd_vhi/spd_shi/spd_warp/spd_cust divisor**),

Windowing

These take a single integer; the "window" is the system's concept of the window dimensions — these won't resize an emulator, but can help get full-size windows and user-space wrapping on a remote system.

Dimension	Dimension	Unit
rows	Vertical	Lines

cols Horizontal Columns

Line Discipline

Takes a single integer as "**line discipline**", which determines how bytes are packaged for delivery to/from the peer; in many ways this could be considered the "protocol" — "raw serial", X.25, GSM, PPP, &c. There should be no reason to set this manually.

Flushing

Argument	Effect
drain	Block until all output is processed (TCSADRAIN), this is the default
-drain	Affect the changes immediately (TCSANOW)

Combos

Argument	Effect
ek	Set erase and kill to their default values
raw	Clear <i>all</i> Input Attributes , min 1 time 0 -opost -isig -icanon -xcase
cooked	Set eof and eol to their default values, brkint ignpar istrip icrnl ixon opost isig icanon
sane	Set all Special Control Characters to their default values, -ignbrk brkint -inlcr -igncr icrnl -iucL -ixany -ixoff imaxbel -iutf8 opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 ff0 vt0 cread isig icanon -xcase echo echoe echok echoke -echonl -noflsh -tostop echoctl -echoprt -flusho iexten -extproc

Aliases

Argument	Inverse	Is	Inverse Is
reprint		rprnt	
decctlq	-decctlq	-ixany	ixany
tandem	-tandem	ixoff	-ixoff
tabs	-tabs	tab0	tab3
hup	-hup	hupcl	-hupcl
cbreak	-cbreak	-icanon	icanon
crterase	-crterase	echoe	-echoe
crtkill	-crtkill	echoke	-echoke
ctlecho	-ctlecho	echoctl	-echoctl
prterase	-prterase	echoprt	-echoprt
columns		cols	
	-cooked		raw
	-raw		cooked
lcase	-lcase	LCASE	-LCASE
parity	-parity	evenp	-evenp
	-oddp		-evenp

Gang Aliases

Argument	Is
dec	intr ^C erase ^? kill ^U -ixany echoe echoke echoctl
litout	-istrip -opost cs8 -parenb
-litout	istrip opost cs7 parenb
pass8	-istrip cs8 -parenb
-pass8	istrip cs7 parenb
nl	-icrnl -onlcr
-nl	-inlcr -igncr icrnl onlcr -ocrnl -onlret
LCASE	iucL olcuc xcase
-LCASE	-iucL -olcuc -xcase
oddp	cs7 parenb parodd
evenp	cs7 parenb -parodd

```

-evenp      cs8 -parenb
 crt        echoe echoke echoctl

```

Saved Attributes

Input Attributes, Output Attributes, Control Attributes, Local Attributes, then all **Special Control Characters**, hexadecimal, colon-separated.

Naturally, this doesn't preserve/restore **Windowing** information or the **Line discipline**.

EXIT STATUS

1 if *teletype* couldn't be opened, new attributes couldn't be applied, or they weren't fully updated (the state after applying wasn't the state being applied — most likely this means altering a read-only attribute for the *teletype*).

EXAMPLES

Undo any weird modes, kill line with `^K`:

```
$ stty sane kill ^K
```

Fully-portably save/restore *teletype* state around querying the teletype size (provided the line speed is at least **110**, which all ANSI-X3.64-compliant video terminals support):

```
#!/bin/sh
saved=$(stty -g; printf "rows %u cols %u" $(stty size))

stty -echo -icanon min 0 time 1
printf '\033[%s' s '99999;99999;H' 6n u
IFS='[;R' read -r _ r c
echo rows $r cols $c

stty $saved
```

SEE ALSO

`tcgetattr(3)`, `tcgetwinsize(3)`, `termios(4)`, `tty(4)`, `ascii(7)`

`/proc/tty/lldiscs`, under Linux, listing *discipline* mappings

High-level teletype semantics that result from these settings (**raw** and **cooked** in particular):

K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v1/UNIX_ProgrammersManual_Nov71.pdf, pp. 162-164, November 3, 1971, `/dev/tty0 . . . tty5 (IV)`.

STANDARDS

Conforms to the IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 base spec, and, except where specified, its XSI parts — **-ag** are standard, also cf. **Standard Attributes**, below. No format is specified: **-a** must include: the **ATTRIBUTES**, separated by spaces, but may (and other implementations do) include additional information; "speed" or "ispeed"/"ospeed" if they're different, in the format listed in the **DESCRIPTION**, up to the semicolon, of which only "baud" is localisable; and **Special Control Characters** in the format listed in the **DESCRIPTION**, the value being either "<undef>", if undefined, the literal character if printable, or a "visual representation" thereof — this implementation's `M-` prefix for the top bit and `^@` for **0** matches the GNU system.

The default output is to be a "subset" of **-a** — this in general means delta from some default settings (and maybe additional data considered relevant, like **min** and **time** if **-icanon**).

The **-g** output is required to not require quoting (but may field-split, so it shouldn't be quoted for portability) and is required to save all standard **ATTRIBUTES** except **Windowing** information, which is optional (this naturally also excludes the **Line Discipline**). This implementation's format matches the GNU system's, which is largely equivalent to AT&T System V UNIX.

-F is an extension, compatible with the GNU system (4.4BSD spells it **-f**, but that clashes with AT&T System V UNIX **-flusho**); `O_NONBLOCK` means that this can be used on devices that aren't usable yet, and would block if spelled `< teletype` (cf. `open(2)`).

Standard Attributes

Queries

size, as described; exactly one space.

Special Control Characters

intr, **quit**, **erase**, **kill**, **eof**, **eol**, **start**, **stop**, **susp**, with arguments as-described (except for empty-string-as-**0**, an extension), but `^b` is defined explicitly for the alphabet, brackets, `\`, `^`, and `_`.

min, **time**, with just an integer, as-described.

`VMIN` is allowed to be `VEOF`, and `VTIME` `VEOL`, which is why **raw** and **cooked** set them, respectively.

Input Attributes

All as-described.

Attrib.	Inverse	Attrib.	Inverse	Attrib.	Inverse	Attrib.	Inverse
ignbrk	-ignbrk	parmrk	-parmrk	inlcr	-inlcr	ixon	-ixon
brkint	-brkint	inpck	-inpck	igncr	-igncr	ixany	-ixany
ignpar	-ignpar	istrip	-istrip	icrnl	-icrnl	ixoff	-ixoff

Output Attributes

opost and **-opost** are part of the base spec, and as-described. All others are shaded XSI:

Attrib.	Inverse	Attrib.	Inverse	Attrib.	Inverse
onlcr	-onlcr	onocr	-onocr	ofill	-ofill
ocrnl	-ocrnl	onlret	-onlret	ofdel	-ofdel
nl0	nl1				
cr0	cr1	cr2	cr3		
tab0	tab1	tab2	tab3		
bs0	bs1				
ff0	ff1				
vt0	vt1				

Control Attributes

All as-described.

Attrib.	Inverse	Attrib.	Inverse	Attrib.	Inverse
cs5	cs6	cs7	cs8		
cstopb	-cstopb	parenb	-parenb	hupcl	-hupcl
cread	-cread	parodd	-parodd	clocal	-clocal

Local Attributes

All a superset of described.

Attrib.	Inverse	Note
isig	-isig	
icanon	-icanon	
echo	-echo	
echoe	-echoe	Defined generically as erasing the last character from the display, else echo applies [-] echoke selects between the two allowed behaviours
echok	-echok	
echonl	-echonl	
noflsh	-noflsh	
tostop	-tostop	

ixten **-ixten** If **-ixten**, **icanon**, **isig**, **ixon**, and **ixoff** are guaranteed to behave conformantly

Line Speed

Either "*number*" for "terminal baud rate" or "**ispeed** *number*"/ "**ospeed** *number*" for "terminal input baud rate"/"terminal output baud rate", with **0** semantics as described. The speeds supported are at *least*:

Hang up	B0
50 baud	B50
75 baud	B75
110 baud	B110
134.5 baud	B134
150 baud	B150
200 baud	B200
300 baud	B300
600 baud	B600
1200 baud	B1200
1800 baud	B1800
2400 baud	B2400
4800 baud	B4800
9600 baud	B9600
19200 baud	B19200
38400 baud	B38400

It's unclear what the minimum required mapping to the *numbers* for these is.

Windowing

rows and **cols**, as described.

See also the standard **LINES** and **COLUMNS** environment variables in `environ(7)`.

Line Discipline

Not part of the standard.

Flushing

The standard doesn't require **stty** to use `tcsetattr(3)`, and doesn't specify any flushing characteristics.

Combos

ek	As described
sane	"Reset all modes to some reasonable, unspecified, values."
oddp	As described
evenp	As described
-evenp	As described

Aliases

All as-described.

Argument	Inverse
tabs	-tabs
hup	-hup
	-raw
parity	-parity
	-oddp

Gang Aliases

Argument	Inverse	Is	Inverse Is
raw	cooked	(see below)	

nl **-nl** **-icrnl** **-inlcr -igncr icrnl**

raw (and **-raw**, **cooked**) is shaded XSI, and supposed to enable (or disable for the inverse) "raw input and output". This raw mode is supposed to be equivalent to "**cs8 erase ^- kill ^- intr ^- quit ^- eof ^- eol ^- -post -inpck**"; what **cooked** is supposed to mean is unclear. This implementation's aliases match the GNU system's.

The additional **-onlcr** for **nl** and **onlcr -ocrnl -onlret** for **-nl** match the GNU system, and, since the additional attributes are shaded XSI while **[-]nl** isn't, may be considered a safe extension.

Obsolescent Aliases

Version 3 of the Single UNIX Specification ("SUSv3") removes these, previously marked LEGACY and shaded EX (equivalent to present-day XSI):

iucl	-iucl	Defined in terms of characters and only in the POSIX locale; a problem,
olcu	-olcu	given that locales are user-space, unlike the line and teletype drivers
xcas	-xcas	Performs "canonical lower and canonical upper presentation", unexplained
lcas	-lcas	As described
LCAS	-LCAS	As described

(Nominally, IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 doesn't contain **-xcas**, but this appears to be a <https://www.mail-archive.com/austin-group-1@opengroup.org/msg10123.html>: **mistake** in the **CHANGE HISTORY**.)

HISTORY

UNIX Programmer's Manual

`gtty(II)` and `stty(II)` syscalls appear in the first edition of the UNIX Programmer's Manual, operating on the DC11 asynchronous communications interfaces attached (`/dev/tty0 ... tty5 (IV)`; cf. *UNIX Programmer's Manual* biblio. in **SEE ALSO**), but not the console (`/dev/tty (IV)`, *ibid.*, p. 161) Teletype Model 33 ASR typewriter which is an integral part of the PDP-11.

`stty(II)` (*ibid.*, p. 135) takes a file descriptor and a pointer to three 16-bit integers, the first of which (`dcrsr`) "is placed into the typewriter's reader control and status register", and the second (`dcpsr`) "is placed in the printer control and status register" — it then defers to the DC11 manual for their format and "until the typewriter is quiescent" (**drain**, **TCSADRAIN**); the **BUGS** section notes:

This call should be used with care. It is all too easy to turn off your typewriter.

Indeed, the DC11's two registers are the Receiver Status Register (**RCSR**), laid out as such (in short; read-only bits and bit **8** omitted), compared to the present-day equivalents:

0	DATA TERMINAL READY	Controls receiving data from modem — if 0 , no input is processed. All too easy indeed!
1	BREAK	Outputs a BREAK if set <code>tcsendbreak(3)</code>
[3, 4]	RECEIVER SPEED SELECT	(see below) <code>cfsetispeed(3)</code>
6	INTERRUPT ENABLE	The DC11 will send an interrupt if there's data. Clearing this would likely also "turn off" the teletype.
[9, 10]	CHARACTER LENGTH	8 — the value, i.e. down to 5 cs[5678]

and the Transmitter Status Register (**TSCR**) (in short; read-only bits and bit **2** omitted):

0	REQUEST TO SEND	Controls sending data to modem.
[3, 4]	TRANSMITTER SPEED SELECT	(see below) <code>cfsetospeed(3)</code>
6	INTERRUPT ENABLE	Send an interrupt if ready to send byte. Setting this inadvertently would most likely have undesirable effects.
8	STOP CODE	2 stop bits — the value [-]cstopb

The **SPEED SELECT** fields select between the four speeds available on the specific DC11 model — according to `/etc/init (VII)` (*ibid.*, p. 202) and `logging in and logging out(VII)` (*ibid.*, p. 208) the default teletype is a **150**-baud Model 37 Teletype with a **BREAK** at the "login:" prompt selecting the **300**-baud GE TermiNet 300; `type(I)` (*ibid.*, p. 101) supports printing to **134.5**-baud IBM 2741s — the remaining speed required to disambiguate the model used is **1200** baud, given in `/dev/tty0 ... tty5 (IV)`, points squarely at the DC11-AG.

These are described in grueling detail in *DC11 asynchronous line interface manual*, DEC-11-HDCD-D, 5th Printing, http://bitsavers.org/pdf/dec/unibus/DEC-11-HDCD-D_DC11_Asynchronous_Line_Interface_Sep73.pdf, Chapter 3 (Operational Programming), Digital Equipment Corporation, Maynard, Massachusetts, September 1973.

The third integer is the *mode*, whose contents "determine the system's treatment of the typewriter" (octal):

```
200 even (M37 tty) parity allowed
100 odd (non-M37 tty) allowed
040 raw mode: wake up on all characters
020 map CR into LF; echo LF or CR as CR-LF
010 don't echo (half duplex)
004 map upper case to lower case on input (M33 TTY)
```

Received bytes with invalid parity are discarded. The closest present-day analogue to 300 is **parodd** to 100 and **parodd** to 200, but allowing both is valid and common in this case.

Raw mode is the moral equivalent of present-day, well, **raw**, but spelled as the inverse of the default **icanon** (with **min 1 time 0**) implied: "every character is passed back immediately to the program. No erase or kill processing is done; the end-of-file character (EOT), the interrupt character (DELETE) and the quit character (FS) are not treated specially."; today, **erase**, **kill**, and **eof** are predicated on **icanon** directly, and **intr** and **quit** on **isig**.

020 is equivalent to **icrnl onlcr**, "used for GE TermiNet 300's".

010 can be recognised as **-echo** and 004 — **iucl**.

The delays are hard-coded for the Model 37. It's unclear if they're the same at **300** baud or if they're altered for the TermiNet, except that it works.

The **Special Control Characters** are unchangeable:

```
intr ASCII DELETE (^?, 0x7F), "sometimes called »rub-out«"; (see below)
quit ASCII FS (^ \, 0x1C), (see below)
eof ASCII EOT (End Of Transmission, ^D, 4), as present-day
erase # (with \# typed for literal #), as present-day
kill @ (with \@ typed for literal @), as present-day
```

It's important to note that no additional output processing is done for erasing, à la **echoe -echok**.

intr and **quit** represent a primitive form of signals (even before signals got signals, job control is *king*. it is known): "for practical purposes at random" a process whose *control typewriter* — the first `/dev/tty*` file opened by the process, inherited on `fork(II)` (cf. today's similar but much more complex `credentials(7)`); this means all processes in a login session share the same one — matches the one on which the **intr** (**quit**) was received and which hasn't ignored the "signal" by passing **0** to `intr(II)` (*UNIX Programmer's Manual*, p. 122) (`quit(II)` (*ibid.*, p. 127)), is "forced to exit" (and dump core). (If a function is passed to `intr(II)` (`quit(II)`) it is executed, but this causes a cascade of errors that make it not worthwhile, and, as the **BUGS** proclaim: "I don't know how to make it work.")

Hanging up the call makes all future reads return empty (the Manual calls this simulating end-of-file); nowadays, a `SIGHUP` is also delivered to the foreground process group.

`gtty(II)` (*ibid.*, p. 120), expectedly, copies those three integers into the provided buffer.

`/dev/tty (IV)` (the "hard-wired" "console typewriter") is nominally out-of-scope since `[gs]tty(II)` don't apply to it, but it explicitly cannot be placed in raw mode (it's unclear how that would happen), **quit** is generated with the "alt mode" key, it's always in **iucl** (004) mode, and "a carriage return is echoed when a line-feed is typed" — this doesn't appear to be a documented `stty(II)`able mode, and doesn't appear to have survived as a modern one either.

`type(I)`, or

```
NAME type -- type on 2741
```

"produces output on an IBM 2741 terminal with a Correspondence type ball." on `/dev/tty5` for reasons too murky to be understood outside of its geographical context and usage procedure from hell. Indeed, the **BUGS** say: "The best thing would be to support 2741's as a standard terminal."

Of interest is also the `hup(I)` (ibid., p. 63) program, which "hangs up the phone on the typewriter which uses it", much like `stty 0` would, but, per **BUGS**, "should not be used; sometimes causes the typewriter channel to be lost".

UNIX Programmer's Manual, 2nd Edition

The argument names are aligned with the DC11 manual in K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Second Edition, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v2/v2man.pdf, pp. 166-167, Bell Telephone Laboratories, Inc., June 12, 1972, `stty(II)`. `dcrsr` becomes "the typewriter's receiver control and status register" and `dcpsr` — "transmitter control and status register".

The `[-]parodd` and `iucle` equivalents are clarified:

```
200 even parity allowed on input (e. g. for m37s)
100 odd parity allowed on input
004 map upper case to lower on input (e. g. M33)
```

`echo` is inverted to match present-day:

```
010 echo (full duplex)
```

`icrnl onlcr` is reversed:

```
020 map CR into LF; echo LF or CR as LF-CR
```

and new modes appear in the bottom bits:

```
002 echo and print tabs as spaces
001 inhibit all function delays (e. g. CRTs)
```

where 001 is ideologically equivalent to `n10 cr0 tab0 ff0` and 002 to `tab3` if set and `tab[12]/tab0` if clear/clear but 001.

The reasoning for 020 (`icrnl onlcr`) is expanded past the TerminoNets, to "other terminals without the new-line function" — i.e. ones for which a newline doesn't return the carriage.

The top byte of the `mode` is "used to indicate that the terminal is an IBM 2741 and to specify 2741 modes.":

```
400 terminal is an IBM 2741
1000 the 2741 has the transmit interrupt feature
      (currently ignored)
2000 use correspondence code conversion on output
4000 use correspondence code conversion on input
      (currently ignored)
```

400 also ignores modes 331 and selects proper delays.

The counterpart to the "correspondence code" is the default "EBCDIC (e. g. [sic] 963 ball and corresponding keyboard)" — since the 2741s are IBM Selectric typewriters, these refer to the two common type balls: the "correspondence" ball (and keyboard) are those of the standard retail Selectric for office correspondence. It's unclear what EBCDIC actually means here, since the counterpart to correspondence is PTT/[E]BCD, neither of which even remotely correspond to a character set encoding¹.

With this, `type(I)` (ibid., pp. 211-214) becomes

```
NAME      type -- type on single sheet paper
copying the input files in chunks of 66 lines, then waiting for a newline between them to allow manual feeding of loose paper.
```

`ttY0(IV)` (ibid., pp. 211-214), now with eight devices, now explicitly notes the input buffer of **150** characters and echoes a # for each one lost. The newline, tab, and form-feed delays are appropriate for the Model 37, but the carriage return delay is appropriate for the GE TerminoNet 300. The speeds now listed are also **110** and **600**, which implies heterogeneity in the devices used since any given DC11 is capable of being on-line programmable for one set of four speeds, but this is excluded by the `SPEED SELECT` fields being hard-coded for autodetection.

ASCII EOT, which causes the Model 37 to hang up, is stripped from the output (previously it was sent with wrong parity). No mention is made of formerly-similar special processing for NAK, which would lock the keyboard.

Arrangements for the 2741 (and other supported teletypes) are made automatically with the new `getty(VII)` (*ibid.*, pp. 251-252) as **cs7 -cstopb -echo 134.5**. **echo** is off because 2741 is half-duplex (transmission-wise; all characters typed are always hardware-echoed, much like on a typewriter, which it is), and incoming data excludes the keyboard being unlocked. Indeed, the keyboard is only unlocked when a read is being performed, **intr** is done by catching a **BREAK** from the **ATTN** key, **quit** and **eof** are impossible, the encoding conversion renders swathes of important punctuation missing and even more of the commonly-used ASCII plane with entirely unrelated characters. It's astonishing this doesn't simply render the 2741 unfit for purpose, but letter-quality prints may well have made it just about worth it — Joe Morris and Dennis Ritchie, "IBM Selectric parallel/serial interface", *alt.folklore.computers*, <https://groups.google.com/g/alt.folklore.computers/c/M7H9LZrK894/m/-ncoZ9gVzPAJ> <https://groups.google.com/g/alt.folklore.computers/c/M7H9LZrK894/m/zqHmiGrbqh4J>, 1998-09-03/04.

Login processing and rudimentary **150-baud/300-baud** detection in `/etc/init (VII)` is delegated to the new `getty(VII)`, executed after the call is established: the speed selection is generalised slightly, in that the "login:" message at **150** baud is prefixed with `'\033:\a'` to put the Model 37 in full-duplex mode; at **300** baud — with `'\033;'` to do the same for the GE TermiNet 300; this "is acceptable for other 300 baud terminals also."; at **134.5** — with nothing, see above.

The username is read in raw mode (with standard **erase** and **kill** processing emulated) with either parity allowed, which allows the next speed to be picked and the message re-printed when a zero byte is read (as part of a **BREAK**). At **150** baud, the 2741's (**134.5**-baud) **ATTN** and enter "looks like the ascii »~« (174₈)" (actually 176₈; 174₈ is also recognised but it's "I") and, if any of these is read, the **134.5**-baud speed will be picked immediately (it is noted, however (`login`, `logout(VII)` (*ibid.*, pp. 257-258)), that: "If the greeting doesn't appear after a few seconds, hang up and try again; something has gone wrong."). The "speed" here is loose, and includes model-specific flags (parity, 2741ness).

If the username is terminated with a carriage return instead of a newline, the post-**getty** mode will also have `020 (icrnl onlcr)` set. If it didn't have any lowercase characters, it's mapped to lowercase and `004 (iucl)` is set as well.

Following this, it executes the also-new `login(I)` (*ibid.*, p. 86), or

```
NAME          login -- sign onto UNIX
```

with the acquired username, and logging in proceeds as-expected (the passphrase is prompted for **-echo** (if possible, i.e. not on the 2741), the **UID** and working directory changed, the **MOTD** read and **mailbox** existence noted, the shell executed).

hup disappears, but `stty(I)` (*ibid.*, p. 117) appears, and bears recalling in its entirety:

```
NAME          stty -- set teletype options
```

```
SYNOPSIS      stty option1 ...
```

```
DESCRIPTION  Stty will set certain I/O options on the current
              output teletype. The option strings are selected
              from the following set:
```

```

even          allow even parity.
-even         disallow even parity.
odd           allow odd parity
-odd          disallow odd parity
raw           raw input (no erase/kill)
-canon        "
-raw          negate raw mode (erase/kill)
canon         "
cr            allow (and echo) cr for lf.
-nl           "
nl            negate cr mode.
-cr           "
echo          echo back every character typed.
full         "
```

```

-half          "
half         do not echo characters as typed.
-echo         "
-full         "
lcase        map upper case to lower case
-ucase        "
ucase        do not map case
-lcase        "
space       map tabs into spaces
-tab         "
tab         do not map tabs
-space       "
delay       calculate cr and tab delays.
-delay       no cr/tab delays
ebcdic      ebcdic ball conversion (2741 only)
-corres      "
corres      correspondence ball conversion (2741 only)
-ebcdic      "

```

FILES standard output.

SEE ALSO stty(II)

DIAGNOSTICS "Bad options"

BUGS --

OWNER jfo

These map trivially to the `stty(II) mode` argument. Of these, `[-]echo`, `[-]raw`, `[-]lcase`, and `[-]nl` survive semantically equivalent, `[-]canon` as `[-]icanon`, `[-]tab` similarly to `[-]tabs`, and `[-]even` `[-]odd` similarly to `[-]evenp` `[-]oddp`.

UNIX Programmer's Manual, 2nd Edition, add'l references

1:

Three keyboards/line codes/type balls for the 2741 are the "Standard Selectric Typewriter" (correspondence), "PTTC/BCD", and "PTTC/EBCD" — *IBM 2741 Communications Terminal*, TP-09 A24-3415-2, Third Edition, http://bitsavers.org/pdf/ibm/2741/A24-3415-2_2741_Communication_Terminal.pdf, figs. 5-10, pp. 15-20, International Business Machines Corporation, Data Processing Division, 112 East Post Road, White Plains, N.Y. 10601.

The line is stateful for the ball hemisphere selected (lower case/upper case headings in even-numbered fig. *ibid.*) et al., with the sent/received codes corresponding to the rotation and tilt of the ball: BA1 and C42 are: **g/G** and **6/¢** for the correspondence models, **a/A** and **6/{'** for the standard charset, **:** for the A and H ones, and **¢** for the typewriter(?) one} for the PTTC/BCD ones, and **a/A** and **6/'** for the PTTC/EBCD ones. Trivially, since EBCDIC (or anything contemporary) isn't stateful, none of these are any given encoding.

UNIX Programmer's Manual, 3rd Edition

The three substantive changes herein are the correction of the 200 mode description to "M37s" and removal of the "It is all too easy to turn off your typewriter." warning from **BUGS** in K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Third Edition, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v3/v3man.tar.gz, ./man2/stty.2, Bell Telephone Laboratories, Inc., February, 1973, `stty(II)`.

As well as a drastic culling of modes allowed in `stty(I)` (*ibid.*, ./man1/stty.1), now:

```

even      allow even parity.
-even     disallow even parity.
odd       allow odd parity
-odd      disallow odd parity
raw       raw mode input
          (no erase/kill/interrupt/quit/EOT)
-raw      negate raw mode
-nl       allow cr for lf (and echo lf cr)
nl        allow nl only
echo      echo back every character typed.
-echo     do not echo characters.
lcase     map upper case to lower case
-lcase    do not map case
-tabs     replace tabs by spaces
tabs      preserve tabs
delay     calculate cr and tab delays.
-delay    no cr/tab delays
ebcdic    ebcidic ball conversion (2741 only)
corres    correspondence ball conversion (2741 only)

```

i.e. `[-]tab` becoming the present-day-spelling of `[-]tabs` and the redundant `[-]canon`, `[-]cr`, `[-]full`, `[-]half`, `[-]ucase`, `[-]space`, `-ebcdic`, `-corres` spellings being removed.

UNIX Programmer's Manual, 4th Edition

Sees a rewrite of the kernel in C, of the manual in `troff(1)`, and is infested by gnomes — K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Fourth Edition, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v4/v4man.tar.gz, `./man0/basinf`, Bell Telephone Laboratories, Inc., November, 1973, How to Get Started, *Logging in*.

`getty(VII)` (ibid., `./man7/getty.7`) cycles between just **150-** and **300-**baud and kernel support for the 2731 is removed entirely, and `type(I)` (ibid., `./man1/type.1`) once again becomes

```
type - type on 2741
```

this time copying the input to `/dev/ttyc` "converting to 2741 EBCDIC output code."; the usage procedure specifies that while it still reads the teletype, the `ATTN` key is to be pressed (since the 2741 can't be **raw** and is always hard-**echo**, this is the only way to get indicate anything without messing up the letter, assuming the keyboard is even unlocked), and an inscrutable **BUGS** section follows; it appears that the 2741s were only worth as much hassle as was required to print letters, and no more — Tom Van Vleck, "Home Terminals", *Multicians*, <https://multicians.org/terminals.html>, para. 10, 1997-02-17.

The input buffer is bumped to **256** bytes (`dc(IV)` (ibid., `./man4/dc.4`)), and overrun characters are "thrown out without notice". ASCII EOT is only stripped from output if not **raw**.

The "upper-case mode" (004) is additionally described as present-day **xcase** (with the mapped characters in the same order, even! This (and the already-extant lowercase) directly corresponds to the characters untypeable on a Teletype Model 33 (and 35), present on all DEC PDP-11s as standard²), but not predicated on **icanon**.

Recognisable (but unreliable, so not quite present-day) signals appear as `signal(II)` (ibid., `./man2/signal.2`) and hanging up sends signal **1** (what we'd call `SIGHUP`), default action terminate without core, to *all* processes whose *control typewriter* matches the hanged-up one, then "returns with an end-of-file indication" as previous. This is as present-day.

This courtesy is extended to DEL's "interrupt signal" (**2**, `SIGINT`) and FS' "quit signal" (**3**, `SIGQUIT`). It's implied that but unclear how the shell protects itself from either signal.

`stty(II)` (ibid., `./man2/stty.2`) sees a major rework with a **SYNOPSIS** of

```

(stty = 31.)
(file descriptor in r0)
sys stty; arg
...

```

arg: speed; 0; mode

stty(fildes, arg)

int arg[3];

The "speed and general handling of the input side of the typewriter" (**ispeed**, unclear what "general handling" means) resides in the "low" byte of *speed*, and "the speed of the output side" (**ospeed**) — the "high" byte. It may therefore make more sense to present *arg* as a pointer to a `struct { u8 ispeed, ospeed, pad[2]; u16 mode; };`.

The speeds listed are the ones "supported by the DH-11 interface", lifted directly from *DH11 asynchronous 16-line multiplexer user's manual*, EK-ODH11-OP-002, 1st Edition, [http://bitsavers.org/pdf/dec/unibus/EK-](http://bitsavers.org/pdf/dec/unibus/EK-ODH11-OP-002_DH11_Asynchronous_16-line_Multiplexer_Users_Manual_Sep76.pdf)

[ODH11-OP-002_DH11_Asynchronous_16-line_Multiplexer_Users_Manual_Sep76.pdf](http://bitsavers.org/pdf/dec/unibus/EK-ODH11-OP-002_DH11_Asynchronous_16-line_Multiplexer_Users_Manual_Sep76.pdf), Chapter 3 (Programming), 3.2.3 (Line Parameter Register), p. 3-5, Digital Equipment Corporation, Maynard, Massachusetts, September 1976. Which are:

0 Zero Baud	4 134.5 Baud	8 600 Baud	12 4800 Baud
1 50 Baud	5 150 Baud	9 1200 Baud	13 9600 Baud
2 75 Baud	6 200 Baud	10 1800 Baud	14 External Input A
3 110 Baud	7 300 Baud	11 2400 Baud	15 External Input B

and compare 3.3.5 Zero Baud (*ibid.*, 3-9), which says: "A speed selection of 0 Baud is provided so that the program may turn off any line."

With two important differences: speed **0** is defined as "(turn off device)", and speeds **14** and **15** are defined as "External A" and "External B". This has made a lot of people very angry and been widely regarded as a bad move.

The attentive reader may notice that the speeds are listed for an entirely different device — a DH11 incessantly called a "DH-11" — than supposedly has a manual page and supposedly contains the `/dev/tty?` device family — `dc(IV)`: DC11 (likewise, "DC-11"). Indeed, the reference implementation of the UNIX system has just "DC-11 interfaces actually present", but the PREFACE states (*Fourth Edition*, `./man0/intro`):

The number of UNIX installations is now above 20, and many more are expected. None of these has exactly the same complement of hardware or software.

The manual lists asterisks after the speeds supported by its DC11s (unwaveringly **134.5**, **150**, **300**, **1200**), warning that "if a non-starred speed is selected, it will be ignored and the present speed left unchanged." Even further:

In the current configuration, only 150 and 300 baud are really supported, in that the code conversion and line control required for 2741's (134.5 baud) must be implemented by the user's program, and the half-duplex line discipline required for the 202 dataset (1200 baud) is not supplied.

One could say that this is a primordial version of inter-UNIX portability, like the outsourcing of mounting of `/usr` (and `/sys`), changing the mode of the DEctape nodes and spawning `dpd(VII)` (*ibid.*, `./man7/dpd.7`) from the `init(VII)` process itself to the new-in-*Fourth Edition* `/etc/rc`. Or, given that `getty(VII)` still uses a fixed table of remote teletype types and `init(VII)` — of local teletype lines to wait on, one could be reaching.

Beyond removal of the 2741 400, 1000, 2000 modes, 020 is re-described to echo "as CR-LF" despite a subsequent paragraph describing it and `dc(IV)` remaining unchanged as "LF-CR", and a new mode:

10000 no delays after tabs (e.g. TN 300)

which seems best-matched to **tab0**; it's unclear if 001 implies 10000, and likely that 002 (**tab3**) overrides it by removing tabs from the output entirely.

The console teletype `kl(IV)` (*ibid.*, `./man4/kl.4`)'s "line disciplines" "are identical to those for `tty?`", and it accepts `gtty(II)/stty(II)`, with the only "salient differences" being the different key for SIGQUIT and that "the speed-select word is ignored". One could say that the latter isn't a difference, since that's the behaviour for most speeds on `dc(IV)`, but.

stty(I) (ibid., ./man1/stty.1)'s **raw** parenthetical gains "parity bit passed back"(?), **-nl** also turns around as "CR-LF" to match the syscall, **[-]delay** also mentions form-feed (no functional change, presumably? given that form feed is mentioned in the *Second Edition* as well), and ebcdic corres are replaced with

tdelay calculate tab delays
-tdelay no tab delays

The **BUGS** section becomes

There should be 'package' options such as **execuport**, **33**, or **terminet**.

UNIX Programmer's Manual, 4th Edition, add'l references

2:

KL11 teletype control manual, DEC-11-HR4C-D, 3rd Printing, http://bitsavers.org/pdf/dec/unibus/KL11_TeletypeControlManual.pdf, Chapter 2 (General Description), 2.1 Introduction, para. 2, Digital Equipment Corporation, Maynard, Massachusetts, September 1970.

H720 power supply and mounting box manual, DEC-11-HR5B-D, 5th Printing, http://bitsavers.org/pdf/dec/pdp11/pwrSupply/H720_PowerSupplyMan.pdf, Chapter 4 (Installation), LT33 Teletype, LT35 Teletype, pp. 4-7, 4-8, Digital Equipment Corporation, Maynard, Massachusetts, September 1970. As well as ibid., Table 4-1 (PDP-11 System Configurations), p. 4-1.

Technical Manual, Model 32 and 33, Bulletin 273B, http://bitsavers.org/communications/teletype/33/273B_Mod32_33_TechMan_Jan63.pdf, Section 7 (Principles of Operation), 7-2. Code, Fig. 7-1.a. (Character Arrangement), Teletype Corporation, 555 Touhy Avenue, Skokie, Illinois, 1962, 1963.

The Model 35 has a Shift key, so it could be used to type lower-case letters, but it inverts bit 5, so it can't be used to type special characters from **xcase**: *35 Keyboard and Base for Keyboard Send-Receive (KSR) and Receive-Only (RO) Teletypewriter Sets, Description and Operation*, Bulletin 221B, Section 574-221-100, Volume 1, http://bitsavers.org/communications/teletype/35/281B_Mod35_TechVol1_Apr73.pdf, Section 2 (Principles of Operation), 2.20 (Shift Key Mechanism), p. 12, Teletype Corporation, 555 Touhy Avenue, Skokie, Illinois, 1962, 1964, 1965.

It can't print lower-case characters: *35 Typing Unit, General Description and Principles of Operation*, Bulletin 221B, Section 574-220-100, Volume 1, http://bitsavers.org/communications/teletype/35/281B_Mod35_TechVol1_Apr73.pdf, Section 2 (Principles of Operation), 2.04 (Signaling Code), Fig. 5, p. 9, Teletype Corporation, 555 Touhy Avenue, Skokie, Illinois, 1963.

UNIX Programmer's Manual, 5th Edition

300-baud teletypes are now the default — K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Fifth Edition, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v5/v5man.pdf, p. iv, Bell Telephone Laboratories, Incorporated, June, 1974, How to Get Started, *Logging in..* So the **getty** (VIII) (ibid., p. 267) speed ordering is reversed to try it first, then fall back to **150**-baud on **BREAK**. The **300**-baud "login:" prefix additionally returns "the GSI terminal to non-graphic mode".

init (VIII) (ibid., p. 268) now reads `/etc/ttys` (**ttys** (V) (ibid., 211)) as a list of lines to listen on and programs to spawn when called (but "the only program currently specifiable is `/etc/getty` (q.v.)", since the program is an offset into an **init**-internal table) and accepts "the *hangup* signal (signal #1)" (**SIGHUP**) to reload and reprocess it "as in a reboot" — remnants of this remain in all present-day **inits**.

tty (IV) (ibid., p. 194) is re-instated as a description of all "low-speed asynchronous communications ports", with the hardware-specific peculiarities relegated to their respective **dc** (IV) (ibid., 180), **dh** (IV) (ibid., 181), and **k1** (IV) (ibid., 184) pages, all of which unchanged and equally uninteresting, except to see that the DC11 `/dev/tty[01234567abcd]` is described as "not currently connected", and appears replaced on the reference implementation by the DH11 at `/dev/tty[f-u]`. The only significant change is that the default speed is also **300** baud and individual settable speeds are not listed.

The "parity bit passed back" bit is clarified in the **raw** paragraph as

The input parity bit is passed back to the reader, but parity is still generated for output characters. clarification is achieved, but thankfully half of the new **BUGS** also make the meaning clear:

On raw-mode output, parity should be transmitted as specified in the characters written.

"If you find it hard to type" FS, "try control-\ or control-shift-L."; the other half of the new **BUGS** is that "Half-duplex terminals are not supported." — this is supported by the nonexistence of **type**.

`stty` (II) (ibid., p. 131) loses asterisks after "supported speeds" and generalises the unknown-speed \Rightarrow ignored behaviour to all the aforementioned devices for the aforementioned reasons. Besides that, it appears unchanged — the `dc` (IV) speeds are still listed as if globally true and mode 10000 (**tab0**) is crossed out by hand entirely, with the description of mode 001 (**nl0 cr0 tab0 ff0**) crossed out and "hangup on last close" penned in — this is **[-]hupcl**, with that exact phrasing of the flag surviving to IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 .

It's unclear what this means, especially given that `stty` (I) (ibid., p. 79) is unchanged entirely.

UNIX Programmer's Manual, 6th Edition

Sees the introduction of `/dev/tty`, corresponding to the process's *control typewriter* (the console is exclusively `/dev/tty8`, a synonym in the *Fifth edition*), as present-day, "a variety of delays after function characters", rather than model-class enumeration for each "function" (NL/CR/tab/FF), speed changes explicitly predicated on "if the hardware permits" — K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Sixth Edition, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v6/v6doc.tar.gz, `./man/man4/tty.4`, May, 1975, `ttY` (IV). `#` and `@` are "By default", and EOT is finally parenthesised as "control-D".

`init` (VIII) (ibid., `./man/man8/init.8`) always runs **getty**, and the third `ttys` (V) (ibid., `./man/man5/ttys.5`) column is instead passed as an argument thereto:

For normal lines, the character is `'0'`; other characters can be used, for example, with hard-wired terminals where speed recognition is unnecessary or which have special characteristics. (Getty will have to be fixed in such cases.)

`getty` (VIII) (ibid., `./man/man8/getty.8`) mentions no such modes, simply predicating the description of the former **300-/150**-baud operation on being passed `'0'`. However, the **getty** of the reference system is available at https://www.tuhs.org/Archive/Distributions/Research/Dennis_v6/v6src.tar.gz, `./s1/getty.c`. And reveals: a `'-` mode for the "Console TTY" whose **110 icrnl onlcr tab3 iuclc olcuc xcasc cr1** point to it continuing to be the same Model 33, with an entry in the default table after the **150**-baud mode.

There's also a `'1'` table, which contains only the **150**-baud entry, and a `'2'` one, described only as "9600" — **9600 icrnl onlcr tab3 ff1**.

`stty` (II) (*Sixth Edition*, `./man/man2/stty.2`) finally sees a new **SYNOPSIS**:

```
(stty = 31.)
(file descriptor in r0)
sys stty; arg
...
arg: .byte ispeed, ospeed; .byte erase, kill; mode
```

```
stty(fildes, arg)
struct {
    char ispeed, ospeed;
    char erase, kill;
    int mode;
} *arg;
```

The *erase*, *kill* members are self-explanatory. The *mode* table is vastly expanded:

```
100000 Select one of two algorithms for backspace delays
040000 Select one of two algorithms for form-feed and vertical-tab delays
```

030000 Select one of four algorithms for carriage-return delays
 006000 Select one of four algorithms for tab delays
 001400 Select one of four algorithms for new-line delays
 000200 even parity allowed on input (e. g. for M37s)
 000100 odd parity allowed on input
 000040 raw mode: wake up on all characters
 000020 map CR into LF; echo LF or CR as CR-LF
 000010 echo (full duplex)
 000004 map upper case to lower on input (e. g. M33)
 000002 echo and print tabs as spaces
 000001 hang up (remove 'data terminal ready,' lead CD) after last close

Which matches the scribble in *Fifth Edition*.

The delays are very similar to present-day, but described in terms of narrow-bit integers, rather than bit-fields, and form-feed is overloaded with vertical tab, and a zeroed delay value corresponds to no delay for that function byte; respectively, they correspond to: BSDLY, FFDLY=VTDLY, CRDLY, TABDLY, NLDLY, and are expounded as follows.

Value	Field	Equiv.	Description
100000	BSDLY =1	bs1	"currently ignored but will be used for Terminet 300's."
040000	FFDLY =1	ff1/vt1	"about 2 seconds."
030000	CRDLY =3		"is unimplemented and is 0."
20000	CRDLY =2	cr3	"lasts about .16 seconds and is suitable for the VT05 and the TI 700."
10000	CRDLY =1	cr2	"lasts about .08 seconds and is suitable for the Terminet 300."
006000	TABDLY=3		unimplemented, 0
4000	TABDLY=2		unimplemented, 0
2000	TABDLY=1	tab1	"dependent on the amount of movement and is tuned to the Teletype model 37."
001400	NLDLY =3		"is unimplemented and is 0."
1000	NLDLY =2	nl1	"useful for the VT05 and is about .10 seconds."
400	NLDLY =1		"dependent on the current column and is tuned for Teletype model 37's."

Mode 000001 corresponds directly to **[−]hupcl**, and "is useful when a port is to be used for some special purpose; for example, if it is associated with ACU used to place outgoing calls.", with "ACU" presumably standing in for an Automatic Calling Unit.

`stty` (II) is "also used with certain special files other than typewriters" — it's the only non-I/O entry in the device driver interfaces, cf. Dennis M. Ritchie, *The Unix I/O System*, https://www.tuhs.org/Archive/Distributions/Research/Dennis_v6/v6doc.tar.gz, **Character device drivers**, Bell Telephone Laboratories, `./iosys/iosys`. And compare `v6src.tar.gz` `./usr/sys/conf.h` `bdevsw/cdevsw` — "but since none of them are part of the standard system the specifications will not be given."; one could consider this a primordial `ioctl(2)`.

`stty` (I) (*Sixth Edition*, `./man/man1/stty.1`) allows no arguments, in which case it "reports the current settings of the options", much like present-day, The **[−]delay** and **[−]tdelay** *options* are removed in favour of:

cooked	same as '−raw'
and	
ek	reset erase and kill characters back to normal # and @.
erase <i>c</i>	set erase character to <i>c</i> .
kill <i>c</i>	set kill character to <i>c</i> .
cr0 cr1 cr2 cr3	select style of delay for carriage return (see below)
nl0 nl1 nl2 nl3	select style of delay for linefeed (see below)
tab0 tab1 tab2 tab3	select style of delay for tab (see below)
ff0 ff1	select style of delay for form feed (see below)
tty33	set all modes suitable for Teletype model 33
tty37	set all modes suitable for Teletype model 37
vt05	set all modes suitable for DEC VT05 terminal

tn300	set all modes suitable for GE Terminet 300
ti700	set all modes suitable for Texas Instruments 700 terminal
tek	set all modes suitable for Tektronix 4014 terminal
hup	hang up dataphone on last close.
-hup	do not hang up dataphone on last close.
0	hang up phone line immediately
50 75 110 134 150 200 300 600 1200 1800 2400 4800 9600 exta extb	Set typewriter baud rate to the number given, if possible. (These are the speeds supported by the DH-11 interface).

It should come as no surprise that "(see below)" points to nothing, except to clarify that "various delay algorithms are tuned to various kinds of terminals", and reiterating that the **0**-terminated options mean no delay.

Of these, **cooked**, **[-]hup**, and the speeds are retained verbatim (but see **STANDARDS, Line Speed**), **ek** using a system default (**CERASE**, **CKILL**), and **erase kill** are equivalent for the *cc*-is-single-byte case.

v6src.tar.gz ./s2/stty.c reveals that there are **[-]LCASE**, aliases for **[-]lcase**, as present-day, a **134** alias for **134.5**, **33 37 05 tn ti** aliases for **tty33 tty37 vt05 tn300 ti700**, respectively, which replace delay flags with **cr1, ff1 cr2 tab1 nl1, nl2, cr1, cr2, ff1**, respectively. The **bs[01]** delays, conspicuously missing from the manual, are there, but just conspicuously missing from the manual.

The no-argument output is ideologically much like present-day **sane**, deviations except much less formalised (and written to the standard error stream): the first lines are either

```
input speed  ispeed baud
output speed  ospeed baud
```

or

```
speed speed baud
```

where the *speed* is an integer (i.e. **134.5** is truncated), and **0** for **exta extb**. The observant reader will notice that there's no way to set differing speeds with **stty**. It's unclear if they were used at all, at least on interactive lines. A line of

```
erase = 'cc'; kill = 'cc'
```

follows (these are unescaped and written verbatim), then a listing of the modes set as:

```
0200  even
0100  odd
040   raw
020   -nl
010   echo
04    lcase
02    -tabs
01    hup
```

followed by the non-**0** delays (**nl, tab, cr, ff, bs**), separated by spaces, as the final line.

UNIX Programmer's Manual, 7th Edition

Many more and different models are listed, as expected from four years of increasingly affordable cash cows being bred; the TTY *37 terminal* section is generalised as *1200- and 150-baud terminals* to cover causing BREAKs until the correct speed is achieved, and a new *Hard-wired terminals* section appears, saying that they "usually begin at the right speed, up to 9600 baud", and defers to the normal instructions otherwise — *UNIX™ Time-Sharing System: UNIX Programmer's Manual*, Seventh Edition, Volume 1, https://www.tuhs.org/Archive/Distributions/Research/Henry_Spencer_v7/v7.tar.gz, *usr/man/man0/intro*, Bell Telephone Laboratories, Incorporated, January, 1979, Introduction to Volume 1, How to Get Started, *Logging in.*

The "'break' or 'interrupt' key" — the one that generates the BREAK used to cycle between speeds — sends SIGINT, with the usual DEL method seemingly being second-class (*ibid.*, *How to communicate through your terminal.*, para. 4).

As for *Stty and gty*, "These system calls have been extensively altered, see *ioctl(2)* and *tty(4)*." (*ibid.*, *Converting From the 6th Edition*).

Indeed, they're now described in the new `ioctl(2)` (`ibid.`, `usr/man/man2/ioctl.2`) as shims for `ioctl(TIO[GS]ETP)`, and are provided as such by Standard C Library (`libc`, `-lc`); the formerly-`stty` (II) information is merged into `tty(4)` (`ibid.`, `usr/man/man4/tty.4`), though it's now about "terminals", rather than "typewriters". The individual-device pages are shed, as are references thereto.

"The set of processes that thus share a control terminal is called a *process group*" — this has no bearing on the actual semantics of the system, and process groups are acquired (and shed) differently in present-day, but this is the first notion of the process group — `kill(2)` (`ibid.`, `usr/man/man2/kill.2`), when signalling pid 0, signals the members of the caller's process group: *this* is as present-day.

The argument format is the same, but now defined in `<sgtty.h>`, designated `struct sgttyb`, with members prefixed with `sg_`, and `mode` — `sg_flags`. The speeds themselves are defined with their present-day `Bintspeed`, `EXTA`, `EXTB` macros.

The `TIOCSETN ioctl()` is like `TIOCSETP`, but doesn't wait for output empty then dump input buffer. Mode 000001 (HUPCL) is moved to `ioctl(TIOCHPCL)` (description retained), and becomes "TANDEM".

The delay mask macros appear (humorously, with "VTDELAY" governing FF[01]), and "XTABS" becomes the third tab delay mode (present-day TAB3). Backspace delays are downgraded to "might".

Besides those, the low-order byte bit layout is the same:

EVENP	0000200	Even parity allowed on input (most terminals)
ODDP	0000100	Odd parity allowed on input
RAW	0000040	Raw mode: wake up on all characters, 8-bit interface
CRMOD	0000020	Map CR into LF; echo LF or CR as CR-LF
ECHO	0000010	Echo (full duplex)
LCASE	0000004	Map upper case to lower on input
CBREAK	0000002	Return each character as soon as typed
TANDEM	0000001	Automatic flow control

Of these, ECHO survives as a macro today and TANDEM as `ixoff` with an alias.

Raw mode is extended with

There are no delays and no echoing, and no replacement of one character for another; characters are a full 8 bits for both input and output (parity is up to the program).

It's unclear what this means.

The new "CBREAK" mode is described as

CBREAK is a sort of half-cooked (rare?) mode. Programs can read each character as soon as typed, instead of waiting for a full line, but quit and interrupt work, and output delays, case-translation, CRMOD, XTABS, ECHO, and parity work normally. On the other hand there is no erase or kill, and no special treatment of \ or EOT.

It's unclear what this means. Well, it's clear what it means, but unclear why — a reasonable guess would be, upon reflection on the modern `[-]cbreak` alias, a lack of sufficient fragmentation — the contemporary driver had all of eight bits without delays, modern ones barely make do with `I20`.

The `TIOC[GS]ETC` (misdocumented as `TIOCSETC` and `TIOCSETP` again) `ioctl(s)`, allows changing the "special characters", with defaults outlined below, via the following `<sgtty.>>` type:

```
struct tchars {
    char    t_intrc;        /* interrupt */
    char    t_quite;       /* quit */
    char    t_startc;     /* start output */
    char    t_stopc;      /* stop output */
    char    t_eofc;       /* end-of-file */
    char    t_brkc;       /* input delimiter (like nl) */
};
```

Character	ASCII Default	Printable Default
<code>intr</code>	DEL	^?

quit	FS	^ \
start	DC1	^ Q
stop	DC3	^ S
eof	EOT	^ D
eol		^ _

These all match present-day defaults.

Indeed — this is the first system with truly settable **Special Control Characters** and it includes a disabling functionality, though spelled as **-1** (rather than today's `_POSIX_VDISABLE`).

start and **stop** are fully formed (in **ixany** mode).

eol is fully formed:

The `t_brkc` character, by default `-1`, acts like a new-line in that it terminates a 'line,' is echoed, and is passed to the program. The 'stop' and 'start' characters may be the same, to produce a toggle effect. It is probably counterproductive to make other special characters (including erase and kill) identical.

The new "TANDEM" mode sends `t_stopcd/t_startc` in response to the input buffer water level: "It is useful for flow control when the 'terminal' is actually another machine that obeys the conventions.". The name hints that this was intended for talking with Tandem Computers, Inc. systems, presumably; it's not used in the userland of the distribution tape at all.

The **BUGS** lose the "parity should be transmitted" sentence and gain an even more indicting paragraph:

The terminal handler has clearly entered the race for ever-greater complexity and generality. It's still not complex and general enough for TENEX fans.

`init(8)` (*ibid.*, `usr/man/man8/init.8`), rather than checking the console switches, just starts `/bin/sh </dev/console` (the latter replaced `tty8`), and boots multi-user when the shell exits; as expected, `SIGHUP` restarts the **init** sequence at single-user mode.

`getty(8)` (*ibid.*, `usr/man/man8/getty.8`) is generalised to say that the "login:" message "can contain character sequences to put various kinds of terminals in useful states." and enumerates the following `ttys(5)` second-column values:

- 0** Cycles through 300-1200-150-110 baud. Useful as a default for dialup lines accessed by a variety of terminals.
- Intended for an on-line Teletype model 33, for example an operator's console.
- 1** Optimized for a 150-baud Teletype model 37.
- 2** Intended for an on-line 9600-baud terminal, for example the Teletype 4104.
- 3** Starts at 1200 baud, cycles to 300 and back. Useful with 212 datasets where most terminals run at 1200 speed.
- 5** Same as '3' but starts at 300.
- 4** Useful for on-line console DECwriter (LA36).

Which do not exhaust the actual tables available.

Special Control Characters are reset to their defaults together with setting the initial username-asking mode.

`stty(1)` (*ibid.*, `usr/man/man1/stty.1`) lists **bs[01]** in the manual, replacing the ominous "(see below)"s on all delay modes and "various delay algorithms" note with a "(see `ioctl(2)`)" on the first (**cr**), adds

- cbreak** make each character available to `read(2)` as received; no erase and kill
- cbreak** make characters available to `read` only when newline is received

and allows **erase** and **kill** to "be of the form '^X' which is interpreted as a 'control X'". A cursory examination of the code (*ibid.*, `usr/src/cmd/stty.c`) reveals that this is achieved by masking it to the bottom **5** bits, as present-day, and as-expected for ASCII. It's salient to note that the other new **Special Control Characters** are indeed not changeable.

`[-]hup` is replaced with just **hup**, calling `ioctl1(TIOCHPCL)`. It's unclear how you'd unset this. Removal of **-hup** is undocumented.

An undocumented **gspeed** mode, equivalent to **ispeed 300 ospeed 9600** (but keep in mind neither primitive exists yet) appears, and **tab3** is, naturally, removed (but not from the manual),

The no-argument output formats nonprintable (control) **erase** and **kill** in the expected way; the flag line is unchanged as much as possible, with the explicit singling-out of XTABS as **-tabs** (and no **tab[012]**).

Third Berkeley Software Distribution

The section covering **@#** notes that "Most users prefer to use a backspace rather than '#', and many prefer control X instead of '@'", pointing to *tset(1)* (cf. **Not Considered (TODO: name)**) and *stty(1)* to alter this — *UNIX Programmer's Manual, Seventh Edition, Virtual VAX-11 Version*, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, *3bsd*, *usr/man/man0/intro*, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, December, 1979, Introduction to Volume 1, How to Get Started, *How to communicate through your terminal.*, para. 3.

getty, *vm(8)* (*ibid.*, *usr/man/man8/getty.vm.8*) loses the table enumeration in the manual (and moves some tables around, but there are 13 with 16 total modes, one of which being an "LSI Chess Terminal" (?) whose login message is "*"; these are out of scope) and special escapes before the login messages (except for table '1'), sends a line consisting of "Virtual *myname*", where the latter is a primordial form of hostname defined in *<ident.h>*, before the "login:" one, explicitly ignores (i.e. re-does the identification and prompt at the same speed) login names without letters, and sleeps for a second after prompting and before trying to read the name, then **stty**(s) again to dump the buffer:

```
/*
 * Wait a while, then flush input to get rid
 * of noise from open lines
 */
```

There is a *stty(2)* (*ibid.*, *usr/man/man2/stty.2*) but it's an odd mishmash of *Sixth Edition* values (esp. HUPCL) with *Seventh Edition* *tty(4)* format. *tty(4)* (*Seventh Edition, Virtual VAX-11 Version, December, 1979, usr/man/man4/tty.4*) is expectedly as in *Seventh Edition*.

stty(1) (*Seventh Edition, Virtual VAX-11 Version, December, 1979, usr/man/man1/stty.1*) removes the '^X' description, lists default values for **erase kill** and finally allows setting the **Special Control Characters**:

```
intr c    set interrupt character to c (default DEL.)
quit c    set quit character to c (default control \.)
start c   set start character to c (default control Q.)
stop c    set stop character to c (default control S.)
eof c     set end of file character to c (default control D.)
brk c     set break character to c (default undefined.) This character is an extra wakeup causing
character.
```

These all — sans **brk** — match their present-day spellings.

By peeping the source (*ibid.*, *usr/src/cmd/stty.c*), we find that despite the '^X' removal, it's both still valid, and ^? is accepted as DEL (**0177, 0x7F**), as well as that **Special Control Characters** not followed by a character is ignored instead of segfaulting.

The no-argument output is suitably updated as well: after the speed lines, the **Special Control Characters** are listed as such:

```
erase = cc; kill = cc; intr = cc; quit = cc
start = cc; stop = cc; eof = cc; brk = cc
```

Where *cc* is also as present-day: "<undef>" if **0377 (-1)**, else "M-" if the top bit set then "^?" / "^c" / "c".

The settings for non-delay modes (incl. XTABS (**[-]tabs**)) are always written, format otherwise unchanged.

Fourth Berkeley Software Distribution

Refuses input and rings the bell on input overflow "if your prompt was a %" — that is, with `cs`(1), that is, with the "new" line discipline — and declares that "many prefer control-U instead of '@'" now; additionally, "Many users change this interrupt character to be ^C (control-C) using `stty`(1)."; finally,

It is also possible to suspend output temporarily using ^S (control-s) and later resume output with ^Q. In a newer terminal driver, it is possible to cause output to be thrown away without interrupting the program by typing ^O; see `newtty`(4).

and

If you use `cs`(1) the key ^Z (control-z) will cause jobs to "stop". If this happens before you learn about it, you can simply continue by saying "fg" (for foreground) to bring the job back.

UNIX Programmer's Manual, Seventh Edition, Virtual VAX-11 Version, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.0, `usr/man/man0/intro`, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, November, 1980, Introduction to Volume 1, How to Get Started, *How to communicate through your terminal.*, paras. 2-4.

Indeed, as the **PREFACE** (*ibid.*, `usr/man/man0/intro`) states: "This release finds the system in transition: a number of facilities have been added in experimental versions (job control, resource limits) and" "Applications which use facilities that are in transition should be aware that some of the system calls and library routines will change in the near future." — the extent to which the latter is not the case is remarkable: these facilities remain enshrined, essentially unchanged, to this day.

`ttty`(4) (*ibid.*, `usr/man/man4/ttty.4`) more than doubles in size; this is reasonable, given that it describes double the amount of "terminal drivers used for conversational computing" (nominally three, but the third is the "net" NETLDISC one, used for Berknet, described in full in `bk`(4) (*ibid.*, `usr/man/man4/bk.4`), and entirely out of scope here) — line disciplines, configurable with `ioctl`(TIOC[GS]ETD), with the "old (standard) terminal driver", "used when using the standard shell `sh`(1) and for compatibility with other standard version 7 UNIX systems" defined as OTTYDISC (0 by convention), and the "newer terminal drivers, with features for job control", which (must be used when using `cs`(1)), as NTTYDISC. These aren't strictly line disciplines in the modern sense (that would be ideologically closer to `new` vs `net` split), but this does seem to be the first system with line disciplines that are used and behave as they do from userspace and can be usefully applied to any teletype. (*Seventh Edition* comes close with its `pk`(4), but that's more akin to *Seventh Edition, Virtual VAX-11 Version, November, 1980* `bk`(4).)

This is also the system that originates modern-day process groups — complete with `[gs]etpgrp`(0), `ioctl`(TIOC[GS]PGRP), &c., cf. `jobs`(3J) (*ibid.*, `usr/man/man3/jobs.3j`) — which are fully formed, to ϵ , with the "new" discipline, and as such so truly *remarkably* unremarkable that one may as well consult modern `credentials`(7). As such, only outstanding differences will be noted.

The three major modes — cooked, CBREAK, RAW — are described completely and separately as "all driver functions" available, no input processing, and no processing (sans LFLUSHO), respectively. RAW returns BREAKs as NULs; this is the first time this is documented outside `getty`(8).

For input, NTTYDISC introduces "a literal-next character ^V" (**lnext**, as present-day) "which can be typed in both cooked and CBREAK mode preceding **any** character to prevent its special meaning.", which is preferred to escaping **erase kill** with a `\`, retained only for "historical reasons" (but even that sees an update with only the escaped character being echoed). as well as "The word-erase character, normally ^W", which "erases the preceding word", "defined as a sequence of non-blank characters, with tabs counted as blanks" (**werase**, as present-day, to precision of **altwerase**). and "Finally, the reprint character, normally ^R, retypes the pending input beginning on a new line.", which happens "automagically in cooked mode if characters which would normally be erased from the screen are fouled by program output."; it's unclear how this is mediated.

In OTTYDISC **erase** is just echoed and **kill** is additionally followed by a newline (as-if **echok -echoke**), "(even if the character is not killing the line, because it was preceded by a '\!.)" (notably, this isn't documented in any manuals heretofore; it's unclear when exactly this stopped being **-echok**) — whereas "new terminal driver has several modes for handling the echoing of terminal input, controlled by bits in a local

mode word." (notably, this is the first occurrence of the "local flags", so spelled and so separated). These flags are largely as present-day behaviourally, though decidedly not as present-day spelling-wise, and are outlined below.

For output, EOT is not transmitted in cooked mode (as opposed to only if RAW), so "programs using raw or cbreak mode should be careful"; NTTYDISC adds "a output flush character, normally ^O" (**discard**, as present-day), "which sets the LFLUSHO bit in the local mode word, causing subsequent output to be flushed until it is cleared by a program or more input is typed." (**flusho**, as present-day) — which happens in cooked and CBREAK modes and input is **rprinted** after resumption.

If LCASE, under NTTYDISC, upper-case bytes are preceded by `^`, which is finally as present-day. A provision for "Hazeltime terminals, which do not understand that `~` has been made into an ASCII character" is made via LTILDE, which outputs `~` (**0x7E**)s as ``` (**0x60**)s.

When the line is hung up, the "distinguished" (foreground) process group is signalled, as present-day; LNOHANG disables this.

The interface is, expectedly, insane: "there are 4 different structures which contain various portions of the driver data."; `struct sgttyb sg_flags` returns to being 16-bit by becoming `short`, with no other layout changes, but CR3 "is suitable for the concept-100 and pads lines to be at least 9 characters at 9600 baud." The former refers to Human Designed Systems Concept-series video terminals; the latter is unclear. This corresponds to no modern mode.

`struct tchars` is unchanged.

The new "Local mode", a word, applies (except for LNOHANG), to NTTYDISC only, settable with TIOCL[GS]ET, with bit-manipulation `ioctl`(s) TIOCLBI[CS] having the expected result.

Macro	Equiv.	Description
LCRTBS	(see below)	Backspace on erase rather than echoing erase
LPRTERA	echopt	Printing terminal erase mode
LCRTERA	(see below)	Erase character echoes as backspace-space-backspace
LTILDE		Convert <code>~</code> to <code>`</code> on output (for Hazeltime terminals)
LMDBUF		Stop/start output when carrier drops
LLITOUT	(see below)	Suppress output translations
LTOSTOP	tostop	Send SIGTTOU for background output
LFLUSHO	flusho	Output is being flushed
LNOHANG		Don't send hangup when carrier drops
LETXACK		Diablo style buffer hacking (unimplemented)
LCRTKIL	(see below)	BS-space-BS erase entire line on line kill
LINTRUP		Generate interrupt SIGTINT when input ready to read
LCTLECH	echoctl	Echo input control chars as ^X, delete as ^?
LPENDIN		Retype pending input at next read or input character

The erasure mode tree is slightly different; assuming **echo**:

```

if LCRTBS | LCRTERA  backspace, space, backspace (present-day -echopt -echoe);
if LCRTBS            backspace;
if LPRTERA           backslash, columns, then slash on next output (present-day echopt);
otherwise            erase (present-day -echopt echoe).
```

kills are backspace, space, backspace for the whole line if LCRTERA or backspace to the beginning of the line otherwise (closest to **echok echoke**) if LCRTKIL and unfouled, otherwise echo **kill**, newline (**-echok -echoke**).

LMDBUF functions like TANDEM, but rather than reacting to water-marks, it does to carrier events, and only does something under the "new" DM-BB driver, which bears a "NB: WE HAVEN'T TESTED dhdm CODE ON VAX.", disclaimer. LETXACK is unused and otherwise undocumented. LPENDIN is a flag set internally by the driver to facilitate reprinting input after TIOCSET[PN] from CBREAK to cooked input (settable by userspace, but not used by the distribution userland).

LINTRUP is unused in the distribution userland except as **stty** [-]intrup. A mode rather similar to LNOHANG may be found in **cllocal**.

Likewise for the new 42000u334180u "Local special chars", TIOC[GS]LTC:

```
struct ltchars {
    char    t_suspc;        /* stop process signal */
    char    t_dstopc;      /* delayed stop process signal */
    char    t_rprntc;      /* reprint line */
    char    t_flushc;      /* flush output (toggles) */
    char    t_werasc;      /* word erase */
    char    t_lnextc;      /* literal next character */
};
```

With expected present-day defaults of ^Z, ^Y, ^R, ^O, ^W, ^V, and expectedly disabled if -1.

The **BUGS** concedes to TENEX fans and that paragraph disappears.

This is undocumented, but the "Local special chars" are reset to kernel defaults when a teletype device is opened and doesn't have any open descriptors. This explains why **getty** (ibid., usr/src/cmd/getty.c) just resets the discipline to OTTYDISC.

stty(1) (ibid., 4.0/usr/man/man1/stty.1), rather than the "current settings", reports:

the speed of the terminal and the settings of the options which are different from their defaults. With the argument "all", all normally used option settings are reported. With the argument "everything", everything stty knows about is printed.

Character arguments may start with "u" to mean undefined (-1) — the manual describes this as "»u« or »undef«".

raw cbreak are updated to reflect their behaviour in tty(4), and **hup -hup** are removed (but only from the manual, **hup** remains unchanged in usr/src/cmd/stty.c). The only new options in the main block are, at last:

tandem enable flow control, so that the system sends out the stop character when its internal queue is in danger of overflowing on input, and sends the start character when it is ready to accept further input
-tandem disable flow control

A separate chunk of options is added, caveated on applying only to the new driver (with the new **csch**):

new Use new driver (switching flushes typeahead).
crt Set options for a CRT (crtbs, ctlecho and, if >= 1200 baud, crterase and crtkill.)
crtbs Echo backspaces on erase characters.
prterase For printing terminal echo erased characters backwards within "\" and "/".
crterase Wipe out erased characters with "backspace-space-backspace."
-crterase Leave erased characters visible; just backspace.
crtkill Wipe out input on like kill ala **crterase**.
-crtkill Just echo line kill character and a newline on line kill.
ctlecho Echo control characters as "^x" (and delete as "^?") Print two backspaces following the EOT character (control D).
-ctlecho Control characters echo as themselves; in cooked mode EOT (control-D) is not echoed.
tostop Background jobs stop if they attempt terminal output.
-tostop Output from background jobs to the terminal is allowed.
tilde Convert "~" to "^" on output (for Hazeltine terminals).
-tilde Leave poor "~" alone.
flusho Output is being discarded usually because user hit control O (internal state bit).
-flusho Output is not being discarded.
pendin Input is pending after a switch from cbreak to cooked and will be re-input when a read becomes pending or more input arrives (internal state bit).
-pendin Input is not pending.
intrup Send a signal (SIGTINT) to the terminal control process group whenever an input record (line in cooked mode, character in cbreak or raw mode) is available for reading.

-intrap	Don't send input available interrupts.
mdmbuf	Start/stop output on carrier transitions (not implemented).
-mdmbuf	Return error if write attempted after carrier drops.
litout	Send output characters without any processing.
-litout	Do normal output processing, inserting delays, etc.
nohang	Don't send hangup signal if carrier drops.
-nohang	Send hangup signal to control process group when carrier drops.
etxack	Diablo style etx/ack handshaking (not implemented).

All of these correspond directly to the "Local mode" bits, except for "**new**", which, as expected, sets NTTYDISC, "**crt**" that TODOs, the positive "**crt***" options that unset LPRTERA, and "**prterase**" that unsets all "**crt***" flags. The conspicuously-missing **-crtbsd** **-prterase** are present but undocumented; the less-conspicuously missing "**old**" sets OTTYDISC. There is a "**newcrt**" which is equivalent to "**new crt**", and is described only in `tty(4)`.

And a similarly-caveatted (and "not normally changed") list of characters:

susp <i>c</i>	set suspend process character to <i>c</i> (default control Z.)
dsusp <i>c</i>	set delayed suspend process character to <i>c</i> (default control Y.)
rprnt <i>c</i>	set reprint line character to <i>c</i> (default control R.)
flush <i>c</i>	set flush output character to <i>c</i> (default control O.)
werase <i>c</i>	set word erase character to <i>c</i> (default control W.)
lnext <i>c</i>	set literal next character to <i>c</i> (default control V.)

These are all entirely as present-day.

There is an (undocumented, of course) query: **speed**, which writes the output speed of `/dev/tty` to the standard output stream and exits **0** (or writes "unknown" and exits **1** if the speed is unknown).

"**crt**", additionally to described, clears LPRTERA; the rationale is, naturally, found in `tty(4)`: "At speeds less than 1200 baud, the LCRTERA and LCRTKILL processing is painfully slow" — and can be approximated at a tenth of a second per column erased at **300** baud.

The output format is globally compressed; the first line starts with "net discipline, " if `bk(4)` is in use, "new tty, " if NTTYDISC, or "old tty, " if "**everything**". The speeds are printed on the same line (by just dropping the newline from the end of the speed printing function; this, of course, means that if the input and output speeds are different, like in "**gspeed**" mode, they're written as "input speed 300output speed 9600"; this once more supports the point that split-speed teletypes just weren't in use). Thus the line ends if "**everything**" or a semicolon separates otherwise.

The mode output is identical (sans "**[-]tandem**", after "**[-]lcase**") if "**everything**", otherwise the "defaults" are EVENP | ODDP, !RAW, CRMOD, ECHO, !LCASE, !TANDEM, (XTABS always reported), !CBREAK, and no delays. If "**all**" or "**everything**" the line is broken.

The default "Local mode"s are all unset; in NTTYDISC, they follow in (mostly) the order listed. The only quirk is that "**crt**" is coalesced and listed as-if a first-class option if the "**crt***" and "**prterase**" all match, except that it's considered to start to include LCRTERA | LCRTKIL at **600** baud, and if "**everything**" is listed as "crt: (crtbs crterase crtkill ctlecho)"; the line is broken here if any flags were listed.

In no-argument output, unless under `bk(4)`, non-default **Special Control Characters** (and, under NTTYDISC "Local special chars") are written in the same format as previous, but only if they differ from the defaults (as described, except for **erase** which is `^H`); otherwise, if "**all**" or "**everything**", they're listed in a two-line tabulated format:

```
erase kill intr quit stop eof
```

for OTTYDISC, with undefined ones not written, and in the `stop` column either **stop** or **start** listed if just one is set (or they're the same) or both but separated by a `/`. The same holds for **eof** and **eol** in `eof`.

For NTTYDISC, the columns are:

```
erase kill werase rprnt flush lnext susp intr quit stop eof
```

additionally, the `susp` column can list **susp** and/or **dsusp** in the same manner.

This is, perhaps, best illustrated with an example; for a VT05 in a reasonable `stty 1200 vt05 new crt kill ^X rprnt u susp u nohang` mode, `stty all` yields:

```
new tty, speed 1200 baud; nl2
crt nohang
erase kill werase rprnt flush lnext susp intr quit stop eof
^H ^X ^W ^O ^V ^Y ^? ^\ ^S/^Q ^D
```

and for a Model 37 in a reasonable `stty 150 new tty37 nl prterase` mode `stty` yields:

```
speed 150 baud; nl nl1 tab1 cr1 ffl prterase
erase #
```

4.1 Berkeley distribution

Adds

```
LDECCTQ /* only ^Q starts after ^S */
LNOFLSH /* dont flush output on signals */
```

Which are equivalent to `[-]ixany` and `[-]noflsh`.

Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, 4.1BSD, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.1/4.0.upgrade, sys/h/ioctl.h, sys/dev/ttynew.c.

4.2 Berkeley Software Distribution

`getty`, taking the table name, reaches its full potential by iterating through a (fully-formed) termcap at `/etc/gettytab`, which includes a default with the (now actual) hostname — *UNIX Programmer's Manual, 4.2 Berkeley Software Distribution, Virtual VAX-11 Version, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.2, usr/man/man8/getty.8, usr/man/man5/gettytab.5, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, August, 1983, getty(8), gettytab(5).* The latter's **BUGS** note that:

Some ignorant peasants insist on changing the default special characters, so it is wise to always specify (at least) the erase, kill, and interrupt characters in the **default** table. In **all** cases, '#' or '^H' typed in a login name will be treated as an erase character, and '@' will be treated as a kill character.

The **he** capability is stupid.

Termcap format is horrid, something more rational should have been chosen.

A description of the **he** capability has been omitted here for the sanity of the reader; it is, by a long shot, the dumbest part of the source materials for this manual. Truly, this is the pinnacle of software.

The relevant revolutionary new part of `getty(8)` is that can hang up if the login name is not obtained in time (**to** property).

All mentions of `stty()/gTTY()` are removed from `ioctl(2)` (*ibid.*, `usr/man/man2/ioctl.2`), which just directs the reader to the section-4 page they're interested in; `stty(3C)` (*ibid.*, `usr/man/man3/stty.3c`), where C is for Compatibility, is described as defunct and obsoleted by `ioctl(2)`.

LINTRUP is "replaced by a call to `fcntl` and the SIGIO signal." — Samuel J. Leffler, *Bug fixes and changes in 4.2BSD, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.2, usr/doc/uchanges/3.t, Berkeley, California 94720, July 28, 1983, Section 2..* The equivalent call is `F_SETFL` with `FNDELAY` | `FASYNC`.

`tty(4)` (*4.2 Berkeley Software Distribution, usr/man/man4/tty.4*) expands on `LDECCTQ` as "Only ^Q restarts output after ^S, like DEC systems"; `LNOFLSH` remains entirely undocumented.

`struct ltchars t_dstopc` is renamed to `t_dsuspc` (it's unclear why it hasn't been that from the beginning), and `t_werasesc` to `t_werasc` (to fit in the 8-byte limit of older compilers, presumably).

`stty(1)` (ibid., `usr/man/man1/stty.1`) sees the addition of

- dec** set all modes suitable for Digital Equipment Corp. operating systems users; (erase, kill, and interrupt characters to `^?`, `^U`, and `^C`, `decctlq` and "newcrt".)
- decctlq** After output is suspended (normally by `^S`), only a start character (normally `^Q`) will restart it. This is compatible with DEC's vendor supplied systems.
- decctlq** After output is suspended, any character typed will restart it; the start character will restart output without providing any input. (This is the default.)

These all survive today as-described.

The DEC operating systems are pretty much any of them, to an extent: TENEX doesn't count, since that's originally BBN, but **Third Berkeley Software Distribution** says that "many prefer control X instead of '@'" — TENEX uses `^X`: Theodore H. Myer, Jogn R Barnaby, and William W. Plummer, *TENEX Executive Manual, TENEX Executive Language, Manual for Users, Bolt Beranek and Newman Inc.*, http://bitsavers.org/pdf/bbn/tenex/TenexExecRef_Apr73.pdf, p. 18, 50 Moulton Street, Cambridge, Massachusetts 02138, January, 1971; revised April, 1973, II. Executive Language Structure, C. Editing and Errors in TENEX commands, 4. (CTRL)-X, Rubout. And single-character erasure echoes a \ followed by the erased column every time. `^R` reprints the line after CR.

TOPS-20 is almost there: DEL is **erase**, `^U` is **kill** — *TOPS-20 System Manager's Guide*, Version 6.0 Interim Release Draft, http://bitsavers.org/pdf/dec/pdp10/TOPS20/TOPS-20_V6_System_Managers_Guide_Interim_Dec84.pdf, p. 2-9, Digital Equipment Corporation, December 1984 (first printing October 1976), Creating the TOPS-20 File System, Step 13, DELETE Key.

VMS sees the full range here: `^O`, `^S`, `^Q`, `^R`, `^U`, all as-described, and `^X` which is equivalent to `^U` in the UNIX context — *VAX/VMS I/O User's Guide*, AA-D028A-TE, http://bitsavers.org/pdf/dec/vax/vms/1.0/AA-D028A-TE_VAX_VMS_IO_Users_Guide_Aug78.pdf, pp. 2-5, 2-6, 2-7, Digital Equipment Corporation, Maynard, Massachusetts, August 1978, Chapter 2, Terminal Driver, 2.2.6 Control Characters and Special Keys, Table 2-1, Terminal Control Characters.

Per the juices (*4.2 Berkeley Software Distribution*, `usr/src/bin/stty.c`), **19200 38400** speeds are added as aliases for **exta extb** and are written as such, rather than "0", in the current-option listing, "[**- extack**" is `#ifdefed` away, and [**-noflsh** added. Default characters are now part of the global `<sys/ttychars.h>`, as present-day, and with present-day names; the surprising difference is that **erase kill intr** got DECified into `^?`, `^U`, `^C` (from `#`, `@`, `^?` in the kernel and `^H`, `@`, `^?` in userspace (`stty.reset` — *Seventh Edition, Virtual VAX-11 Version, November, 1980*, `usr/src/cmd/reset.c`)).

4.3 Berkeley Software Distribution

No longer "must" one "call UNIX from an appropriate terminal"; instead, a phone number is necessary only "If you will be accessing UNIX remotely". A direct connection "generally runs at 9600 or 19200 baud", and modem speed is "typically 300, 1200, or 2400 baud" — *UNIX User's Reference Manual (URM)*, 4.3 *Berkeley Software Distribution, Virtual VAX-11 Version*, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.3, `usr/man/man0/intro.ms`, Computers Systems Research Group, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, April, 1986, Introduction to User's Reference Manual, How to get Started, *Logging in.* For logging out, the **logout** `csh(1)` primitive is recommended above EOFing the shell (ibid., *Logging out.*). The *How to communicate through your terminal.* section is adamant that DEL and `^H` have replaced `@#` by default — there is no evidence of this: they remain `^H` and `^U`. And provides an ominous note:

If the terminal does not appear to be echoing anything that you type, it may be stuck in "no-echo" or "raw" mode. Try typing "(control-J)reset(control-J)" to recover.

Text processing. replaces the eternal (since *Third Edition*, which introduced the How to Get Started section) **pr** recommendation ("paginates the text, supplies headings, and has a facility for multi-column

output") with a **more** one ("is useful for preventing the output of a command from scrolling off the top of your screen.")

`ttys(5)`, rather than being a trivial columnated list, is now fully comment-capable and parameterised with regards to what to run with full `cmdline`, where, what type of terminal is on it, and arbitrary parameters to the programs run; this includes running **xterms**; this is as present-day in most Berkeley derivatives — *UNIX Programmer's Reference Manual (PRM), 4.3 Berkeley Software Distribution, Virtual VAX-11 Version*, https://archive.org/details/The_CSRG_Archives_CD-ROM_1_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 1, Berkeley Systems 1978-1986, 4.3, `usr/man/man5/ttys.5`, Computers Systems Research Group, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, April, 1986. `tty(4)` (*ibid.*, 4.3/`usr/man/man4/tty.4`) finally stops mentioning `bk(4)` (*ibid.*, 4.3/`usr/man/man4/bk.4`) (though it's not fully removed, only marked "(obsolete)" with a **BUGS** section that lays out why it's not fit for purpose), and the listing is reworded to "there are two line disciplines available for use with terminals" which makes much more sense, given that `bk(4)` is (unsuitable) for uploading data from microprocessors; ominously, "Other disciplines may exist for special purposes, such as use of communications lines for network connections."

`TIOCNOTTY`, detaching the caller from its controlling terminal, available since *4.2 Berkeley Software Distribution*, is finally documented. **eof** is finally documented as one of the inputs that end lines in cooked mode in-line. The DECified defaults are propagated to the description. `TIOCHPCL` is now "usually done on outgoing lines and dialups.", and the userland agrees with this.

Rather than always outputting even parity, odd parity may be generated by setting `ODDP !EVENP`.

`TIOCSTOP` and `TIOCSTART` **ioctl**(s) appear to emulate pressing **stop start**. It's unclear if `TIOCSTI`ing them would suffice.

One new "Local mode" bit appears:

`LPASS8` Pass all 8 bits through on input, in any mode
`LNOFLSH` Inhibit flushing of pending I/O when an interrupt character is typed.

`LPASS8` (in place of `LINTRUP`) is additionally described as overriding normal parity validation in cooked and `CBREAK` modes (and supposedly what happens in `RAW` mode). It's unclear what this actually means. `[-]pass8` survives as an alias; it's unclear if it retains its meaning.

Window size provisions appear fully-formed as present-day, "to accommodate terminals and workstations with variable-sized windows": `SIGWINCH`, `TIOC[GS]WINSZ`, and

```
struct winsize {
    unsigned short ws_row;    /* rows, in characters */
    unsigned short ws_col;    /* columns, in characters */
    unsigned short ws_xpixel; /* horizontal size, pixels */
    unsigned short ws_ypixel; /* vertical size, pixels */
};
```

`stty(1)` (*PRM, 4.3 Berkeley Software Distribution, usr/man/man1/stty.1*) is prefixed with a list of **Queries**:

speed The terminal speed alone is printed on the standard output.
size The terminal (window) sizes are printed on the standard output, first rows and then columns. of which the latter is new, and as present-day; before job control flags are seen its corresponding setters:
rows *n* The terminal size is recorded as having *n* rows.
columns *n* The terminal size is recorded as having *n* columns.
cols *n* is an alias for *columns*.
equally as present-day.

The new "Local mode"s are prescribed as such

pass8 Passes all 8 bits through on input, in any mode.
-pass8 Strips the 0200 bit on input except in raw mode.

It's even more unclear how this relates to parity validation.

size is implemented (ibid., `usr/src/bin/stty.c`), like "**speed**", by re-opening `/dev/tty` and exiting **0**; the purpose of this is unclear.

The window size is written as "`, ws_row rows, ws_col columns`" after the speed, if "**all**" or "**everything**".

UNIX Programmer's Manual, CB Version

Sees a truly unique **Logging in.** section, alluding to well-developed port-specific **getty** tables:

Usually, the same telephone number serves terminals operating at speeds of 110, 150, and 300 baud.

A different number may be used for 1200-baud service.

and caveats *Higher-speed terminals* on their performance:

Terminals designed to run at higher data rates than 300 baud (i.e., 1200 baud) can be utilized in full-duplex mode provided *input* remains character-by-character, typing speed.

(it's unclear what the latter means, but this would point at having two models of DC11s installed, since that particular combination of speeds is unavailable and DH11s can do all of these, and more, at once; but this is not the case, see below). Aside from the "*TELETYPE® Model 37*", used as an example of <300-baud teletypes, no devices, models, or brands are mentioned (conversely, *Seventh Edition* mentions no fewer than 7 models from 5 brands). It is also uniquely noted that: "The system may prompt you for a dialup password which is established by the system administrator (you need to know it to use any dial port into the system)." — this is mediated by **login** after the normal password prompt. And a special, shockingly content-free for the length, *Non-dial Terminals* section. J. D. Doan, *CB-UNIX Programmer's Manual, Department 59473, Bell Telephone Laboratories, Incorporated, Edition 2.3, Volume 1*, https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix_cbunix_intro.pdf, pp. viii-ix; `cbunix_man1_04.pdf`, p. 4, Columbus, OH 43213, May 1981, How To Get Started; `login(1)`.

A new **Logging out**, method of sitting around for a while is noted, based on (lack of) teletype activity. As compared to *Sixth Edition* (from which a "large part of the present manual's contents is descended" (*CB-UNIX Programmer's Manual*, `cbunix_intro.pdf`, ACKNOWLEDGEMENTS, p. iii)), `^S` can "temporarily stop output", which "is useful with CRT terminals to prevent output from disappearing before it can be read.", and is only resumed with a `^Q` — these are as-present-day **stop start** defaults in **-ixany** mode. Additionally,

On CB-UNIX only, output may also be stopped by typing the break or escape keys. In this case, typing another escape (or any other characters, for that matter) will cause output to be resumed.

which corresponds to **ixany** for this alternate keying.

2.3 is, of course, a late version of the system, but, in the words of an unnamed "retired CB/UNIX developer";

Other things that immediately come to mind that we added first in Columbus Unix were power-fail restart (myself and Jim McGuire did the initial work) and line-disciplines and terminal types (Bill Snider did the initial work).

and

we started using the C version in early 1975 as I recall.

as per https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/readme.txt, final para.. This mostly lines up with the citing of *Sixth Edition*: the manual includes parts of *Seventh Edition, Virtual VAX-11 Version, December, 1979* and *Seventh Edition, Virtual VAX-11 Version, November, 1980* — in some portions with the branding in tact. Nevertheless, given that

I was the supervisor of the group in Columbus for a number of years that was responsible for the development of CB-UNIX.

and

The SCCS version of UNIX had a number of unique features for the times: semaphores and line disciplines (in 1974!) for example. Hal Pearson was responsible for semaphores, and Bill Snider for line disciplines.

mostly lines up with the aforementioned — Dale DeJager, "UNIX History", *net.unix*, https://groups.google.com/g/net.unix/c/-H9x36DMOBQ/m/P_G_s9SJBrgJ, 1984-01-17. It's not outlandish to say that the extensive discipline system, outlined below, *does* land somewhere in the fore of the tail half of the 1970s more so than its aft, perhaps to the precision of its extent. `pr(1)`s of the kernel are also scanned, and dated; helpfully, it's "Sat Feb 9 09:01:00 1980" and "Jan 26 17:20" so

using them as anything more than a disambiguation is useless — CB-UNIX Source Code, https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/SourceCode,cbunix3.pdf. Most (sans `ioctl(2)`) of the teletype-related lines are dated "November 1979" and branded CB-UNIX 2.1; the wording of `tty (4)` (*CB-UNIX Programmer's Manual*, [cbunix_man4.pdf](https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/SourceCode,cbunix_man4.pdf), p. 27)'s controllable-teletype-file and it mentioning `dh (4)` and `kl (4)` points the most recent common ancestor being *Sixth Edition*.

The core difference there is that that file is `/dev/ln` — this follows the console being `ln00` and the dial-in lines being `ln*`. `dh (4)` (*CB-UNIX Programmer's Manual*, [cbunix_man4.pdf](https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/SourceCode,cbunix_man4.pdf), p. 2) is extended to also cover DZ11s (also incessantly called "DZ-11"s): "(For DZ-11 lines, output speed is always the same as input speed. The 200 speed and the two externally clocked speeds (*exta*, *extb*) are missing on the DZ-11.)". The DZ11 supports a different set of speeds:

0	50 Baud	4	150 Baud	8	1800 Baud	12	4800 Baud
1	75 Baud	5	300 Baud	9	2000 Baud	13	7200 Baud
2	110 Baud	6	600 Baud	10	2400 Baud	14	9600 Baud
3	134.5 Baud	7	1200 Baud	11	3600 Baud	15	Not used

none of which correspond directly to the corresponding speeds on the DH11, and "The receiver clock is turned on and off by setting and clearing bit 12" — *DZ11 asynchronous multiplexer technical manual*, EK-DZ110-TM-002, 3rd Printing (Rev), http://bitsavers.org/pdf/dec/unibus/EK-DZ110-TM-002_DZ11_Asynchronous_Multiplexer_Technical_Manual_Oct78.pdf, p. 3-10, Digital Equipment Corporation, Maynard, Massachusetts, October 1978, Chapter 3 (Programming), 3.3 Programming Features, 3.3.1 Baud Rate, Table 3-4 Baud Rate Selection Chart.

And the "Transmitter clock is always on." (*ibid.*, Table 3-3 LPR Bit Functions, p. 3-8). How this is negotiated is not noted, but *CB-UNIX Source Code*, [cbunix7.pdf](https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/SourceCode,cbunix7.pdf), pp. 64-72 (`dz.c`), bears:

```
/* this array maps the standard unix transmit and receive speeds
 * into the speeds used in the dz11 hardware. If a speed is selected
 * that is not available on the dz11 it will be ignored.
 */
```

```
char dzspeed[]
{
    0000,          /* not available */
    0000,          /* 50 baud */
    0001,          /* 75 baud */
    0002,          /* 110 baud */
    0003,          /* 134.5 baud */
    0004,          /* 150 baud */
    0000,          /* not available */
    0005,          /* 300 baud */
    0006,          /* 600 baud */
    0007,          /* 1200 baud */
    0010,          /* 1800 baud */
    0012,          /* 2400 baud */
    0014,          /* 4800 baud */
    0016,          /* 9600 baud */
    0000,          /* not available */
    0000,          /* not available */
};
```

and, in `dzparam()`:

```
lpr = ((dzspeed[tp->t_speeds.lobyte]<<8) | ((dev.d_minor&07) |
      RECV_ON));
```

which is very obviously wrong: unmappable DH11 speeds all become **50** baud.

The `kl (4)` (*CB-UNIX Programmer's Manual*, [cbunix_man4.pdf](https://www.tuhs.org/Archive/Distributions/Distributions/USDL/CB_Unix/SourceCode,cbunix_man4.pdf), p. 6) `ln00` name is "for historical reasons only": the canonical keying for "the real system console" is `sys tty`, and the current console if going single-user — *Init (1M)* and *reboot (1M)* talk to this device" — is `syscon`, which `telinit (1M)` links to the invoking teletype when passed runlevel **S**.

This nomenclature is matched with `line(1)` (ibid., `cbunix_man1_03.pdf`, p. 100) — "get line identification" — which "gives the name of the user's line in the form **lnnn** in the range 00 to 99" (or "**lnxx** if the standard input file is not a typewriter"); the familiar `tty(1)` (ibid., `cbunix_man1_09.pdf`, p. 21) interface is also available, writing the full path and accepting `-s`, as present-day.

Opening `tty(4)` "causes a wait to take place at the first read or write.", rather than waiting "until a connection is established.". The description for *raw* mode is extended; rather being "awakened on each character":

Control is returned to the reading program only when the *read(2)* character count has been satisfied (as well as if an *alarm(2)* signal occurs, or if the line hangs up).

And output parity, rather than being "generated" is noted as "all 8-bits are sent if parity is set to *even* and *odd* and the number of data bits is set to 8.". Expectedly, in non-*raw* mode, output parity is no longer listed as being always-even, too.

The **start stop**-and-friends spiel is reproduced here entirely, but "A BREAK character is treated like and ESC character when not in raw mode." only; this makes sense.

`getty(1M)` (ibid., `cbunix_man1_03.pdf`, p. 40) — "the second process in the series, *init-getty-login-shell*, that ultimately connects a terminal user with UNIX" — copies `issue(5)` (ibid., `cbunix_man5.pdf`, p. 32) to the calling teletype before the prompt, as present-day, and sees a primary **SYNOPSIS** of:

```
/etc/getty [ -h ] line [ speed [ type [ linedisc ] ] ]
```

(Though "Additional arguments may be typed after the login name. These are passed to *login*, which will place them in the environment." — the latter is part of the **login** calling convention, the former is presumably mediated by just adding whitespace after the username.)

`-h` suppresses the default behaviour of forcing "a hangup on the line by setting the speed to zero before"hand. It's unclear how this is supposed to work with DZ11s, given the above (it is: it doesn't, see above and note that `RECV_ON` is **010000** (bit 12)). *line* is the filename under `/dev`, *speed* is the `gettydefs(5)` (ibid., p. 26) entry, itself nothing more than the extraction of the built-in tables, and so tabulating the label, flags when reading the username (with **hupcl -echo raw** implied), final flags, speeds, the message, and the label to go on BREAK. *type* describes "what type of terminal is connected to the line in question":

none	default	tektonix, tek	Tektronix
tec	TEC scope	ds40-1	Teletype DS40-1
vt61	DEC vt61	hp45	Hewlett-Packard HP45
vt100	DEC vt100	ds40-2b	Teletype DS40-2b

where "The default terminal is »**none**«; i.e., any crt or normal terminal unknown to the system."; this is more akin to present-day `$TERM` (coupled with `terminfo(5)`), but as a first-class kernel layer, complete with emulating missing functions. And *linedisc* describes "which line discipline to use in communicating with the terminal":

full_duplex, full	default
transparent, trans	transparent (see <code>ioctl(2)</code>)
half_duplex, half	half duplex
votrax	votrax

"The default is »**full_duplex**«, i.e., line discipline zero."

The interesting differences in the username adaptation mode are:

In addition to the standard UNIX erase and kill characters, `'#'` and `'@'`, *getty* also understands `'\b'`.

If the user uses a `'\b'` as a rubout, *getty* sets the standard erase character to backspace and the standard kill character to `'@'` instead of `'#'` and `'@'`.

Which is, notably, the worst possible way to spell this, and **How To Get Started** continues to say "The ASCII »delete« (a.k.a. »rubout«) character", overloading it even further.

And

Getty also undertands the "standard" ESS protocols for erasing, killing and aborting a line, and terminating a line. If *getty* sees the ESS erase character, `'_'`, or kill character, `'$'`, or abort character, `'&'`, or the ESS line terminator, `'/'` or `'!'`, it attempts to set up the terminal into **STDTTY** mode (see `ioctl(2)`), which has those characters as the erase, kill, and line terminator characters. If it doesn't

succeed, the standard erase and kill characters will be used. Here, the ESS is the Electronic Switching System — AT&T's line of computerised telephone switches (cf. *UNIX History*: "The system was derived from the UNIX operating system that was used in the SCCS (Switching Control Center System), which incidentally was the first application of UNIX outside of research."). This spelling (so far as `_`, `!` (7:25) and `$` (9:27)) is attested on the mostly-contemporaneous 3ESS — Æstrid Smith and Sarah Autumn, *Connections Museum Seattle*, Booting up the 3ESS, <https://youtu.be/k865-VjWUk8>, 7000 East Marginal Way South, Seattle, WA 98108, Mar 23, 2022.

The **BUGS** mention that none of the characters mentioned above (or `"^D"`) is quotable.

`gtty(1)` (*CB-UNIX Programmer's Manual*, `cbunix_man1_03.pdf`, p. 58) appears, and bears recalling in its entirety:

NAME

`gtty` – get terminal line options

SYNOPSIS

`gtty` [`line ...`]

DESCRIPTION

`Gtty` will get certain I/O options on the lines specified. If no line is given the standard input is used (file descriptor 0). See `stty(1)` for an explanation of the options reported by `gtty`.

SEE ALSO

`gtty(2)`, `stty(1)`

BUGS

On devices which do not allow variable speeds, the input and output speed is meaningless. it follows, then, that `stty(1)` (*ibid.*, `cbunix_man1_08.pdf`, p. 38) sees a pre-*Sixth Edition* calling convention. In addition to the precise set of options in *Fourth Edition*, a list of new options follows.

Option	Equiv.	Description
hdplx		half duplex (disable reception during transmission)
-hdplx		full duplex
tstread		test read (don't wait for carriage return to terminate read)
-tstread		normal read termination
hangup	hupcl	dial-up line (hangup after last close)
-hangup	-hupcl	non dial-up
lfdelay	nl1(?)	line-feed delay
-lfdelay	nl0	no line-feed delay
sccmod		enable sccmode (special editing, break and del characters)
-sccmod		disable sccmode
xclude		exclude future opens of this teletype
-xclude		Allow future opens of this teletype
tandemi	ixon	respond to xoff and xon when transmitted by the terminal by stopping and starting output.
-tandemi	-ixon	treat xoff and xon as normal input characters

[**-**]**tandemi** are highlit in pink — the meaning of this is unclear; perhaps this is to indicate that they should be bolded and aren't?

And these options, each taking a numeric argument:

speed allows the user to change the speed of a line
stop specify the number of stop bits to be transmitted
data data bits
vrow set the variable row on crts above which the screen will be frozen

The format for **"speed"** is unspecified, and could be either the actual baud-rate just as well as **[0, 15]**.

"the terminal type" may also be specified, but "If this is done it must be the first option."; depending on the system configuration, the following may be meaningful (this is implied to not be an exhaustive list):

"none", **"tec"**, **"delta"**, **"dta"**, **"tektronix"**, **"tex"**, **"vt61"**, **"ds40-1"**, **"ds40-2b"**, **"hp45"**

note that here Tektronix is spelled correctly, and its short form is **"tex"**, not **"tek"**. The VT100 is missing, and an unexplained **"delta"** appears.

"When a terminal type other than »none« is in use, several more flags become valid:"

anl	automatic new-line when character printed in column 80
-anl	no automatic new-line
snl	special new-line function (only implemented for "ds40-1")
-snl	no special new-line function
lcf	special action on last column of last line
-lcf	no special action on last column of display
default	set terminal flags to default for this terminal type

`stty:o(2)` (ibid., `cbunix_man2_04.pdf`, pp. 23-28) — where the `:o` denotes being for compatibility with "release 1 of CB-UNIX" (`intro(2)` (ibid., `cbunix_man2_01.pdf`, **OLD C COMPILER**, p. 4)) — is noted as superseded by `ioctl(2)` (ibid., `cbunix_man2_02.pdf`, pp. 1-7) but nevertheless contains a more cohesive overview of the system's general teletype system (though, notably, the mode layout and nomenclature outside of the core XTABS-EVENP spectrum are completely different between the two). Line disciplines **1**, **2** are "reserved for project specific line disciplines", and [**5**, **9**] "for future common line disciplines". Of those, **0** (STD_LTYPE) is the "normal" full-duplex discipline; **4** (HF_LTYPE) is the half-duplex line discipline — at last! — both of which "require no additional information". It's unclear how HF_LTYPE differs from setting HDPLX ("For those communications controllers with the capability, disable reception during transmission.").

The interesting discipline is **3** (TRANS_LTYPE):

a line discipline that allows the user full eight bit transparency ion input and output with or without parity, For this line discipline, a *write* will perform no mapping. A *read* will return upon the occurrence of the first of the three conditions as specified by the user:

- 1) The requested number of characters have arrived.
- 2) The number of seconds, *ts_quanta*, has elapsed.
- 3) A break character has arrived.

If *ts_quanta* is zero, timing is disabled; otherwise, *ts_quanta* is the maximum wait time in seconds. If *ts_brk0* and *ts_brk1* are both zero, no break characters will awaken the process. If *ts_brk1* is 0377 then *ts_bbrk0* is taken as a single break character. Otherwise, both break characters are assumed valid. NCDELAY, XTABS, LCASE, ECHO, CRMOD, RAW, NTDELAY, NLDELAY, and STDTTY have no meaning for this line discipline.

Which is, to the precision of the temporal resolution and the explicit wake-up characters, *exactly* the present-day **raw** mode.

Backronymising, it may be that "[**-tstread**]" may actually be TranSparenT READ, otherwise it's unclear what it *could* be, aside maybe from a murky description of the equally-murky NOSLEEP:

Return a zero if a read is performed and no characters are present. Don't wait to flush output on *close* or *stty*. Don't wait for carrier in the first *read* or *write* after an *open* if carrier is not up. Normally a process will block when waiting for carrier to come up after an *open*. This roadblock will take place in the first *read* or *write* not the *open*.

`stty:o(2)` has an explicit NLDELAY mode, `ioctl(2)` has a range of CR[0123] NL[0123] TAB[01] FF[01] BS[01] with the would-be TAB2 bit defined as NOAL ("no allocation", presumably), since XTABS is unmoved from **2**. "[**-xclude**]" corresponds directly to XCLUDE, described identically with addition of "Cleared upon last close."

TANDEMI ("response to xon/xoff", [**-ixon**]) is contrasted with TANDEMO (but only in `ioctl(2)`) — "transmission of xon/xoff": "turns off the keyboard when there are too many characters in the terminal hardware queue" — which itself is [**-ixoff**].

erase kill are changeable, layout-compatible with *Sixth Edition*. No other character is, hence STDTTY, which is exactly as-described (**erase _ kill \$ eol / eol2 !**) with the addition of **icrnl** and a proverbial "**kill2 &**" (in addition to the primary DEL) (though, curiously, the flag itself does *not* (appear to, *CB-UNIX Source Code*, `cbunix10.pdf`, pp. 36-58 (`tty.c`), esp. `ttyinput()` (p. 45); the manual itself would suggest it does) imply the **erase kill** keyings, so those need to be set separately). This is likely [**-]scmod** as Switching Control Center MODE.

ioctl(DIOCKETT) "is used to specify the type of CRT connected to a line":

```
/*
 * Terminal types
 */
#define TERM_NONE      0      /* tty */
#define TERM_TEC       1      /* TEC Scope */
#define TERM_V61       2      /* DEC VT61 */
#define TERM_V10       3      /* DEC VT100 */
#define TERM_TEX       4      /* Tektronix 4023 */
#define TERM_D40       5      /* TTY Mod 40/1 */
#define TERM_H45       6      /* Hewlett-Packard 45 */
#define TERM_D42       7      /* TTY Mod 40/2B */
```

"If a type other than **TERM_NONE** is specified, input and output mapping will occur for the CRT language defined in the header file `<sys/crtctl.h>`. In this case, the ESC character takes on special meaning, escaping the subsequent characters on input and output." Helpfully, `crtctl.h` doesn't appear to have been scanned. Nevertheless, this points to a bizarre world where the motion/underlining/esc. interface is *to the system*, which then provides a Cohesive Baseline Video Terminal, shimmed as required. Mayhap it was less bizarre before the monopolisation of ECMA-48 ("Control functions for coded character sets") et al. — the only terminal listed verifiably supporting these is the VT100.

The video-terminal-specific modes are as such:

TM_SNL	Handle new line specially, if the terminal driver is so equipped.
TM_ANL	Provide a carriage return and new line when writing beyond column 80.
TM_LCF	Immediately before placing a character in the last column and now, delete the top line, print the character in the last column of the now second-to-last-row, and then move the cursor to column one of the new last line. This function is required for terminals that move the cursor to "bad" places when printing in the last position.
TM_CECHO	Echo the control sequences, such as "cursor up", when received.
TM_CINVIS	Do not pass the cursor control characters to the user program on input.

And agree with `"[-]anl"`, `"[-]snl"`, and `"[-]lcf"`.

"vrow" is additionally noted as "for split screen operation". The actual semantics of this are unnoted.

ioctl(VIOCG[GS]ETD) is described as "Get/set versatec parameters.". Votrax International, Inc. manufactured voice synthesizers, originally the VS4 "Votrax", and is attested as discipline **5** (`VOT_LTYPE`) (*CB-UNIX Source Code*, `cbunix3.pdf`, p. 95 (`votrax.h`); *ibid.*, `cbunix10.pdf`, pp. 65-69 (`vs.c`)).

No use of `TM_SNL` is attested, especially in *ibid.*, `cbunix7.pdf`, pp. 59-63 (`ds40.c`), which is supposed to host it.

The lost-media `<sys/crtctl.h>` is also shed light upon via the emulation therein:

```
char ds40map[] {
    LCA,    NONE,    /* Load Cursor Address (simulated) */
    CUP,    SDESC|'7', /* Move Cursor Up */
    CDN,    SDESC|'B', /* Move Cursor Down */
    CRI,    SDESC|'C', /* Move Cursor Right */
    CLE,    010,     /* Move Cursor Left */
    NL,     NONE,    /* Special DS40 newline */
    CRTN,   SDESC|'G', /* Return Cursor to start of line */
    HOME,   SDESC|'H', /* Home Cursor */
    CS,     SDESC|'R', /* Clear Screen */
    EEOP,   SDESC|'J', /* Erase to end of page */
    EEOL,   NONE,    /* Erase to end of line */
    IL,     SDESC|'L', /* Insert Line */
    DL,     SDESC|'M', /* Delete Line */
    IC,     SDESC|'^', /* Insert Char */
    DC,     SDESC|'P', /* Delete Char */
    STB,    SDESC|'3', /* Start Blink */
```

```

        SPB,      SDESC|'4',      /* Stop Blink */
        STAB,     SDESC|'0',      /* Set Single Tab */
        ATAB,     SDESC|'1',      /* Set Column of Tabs */
        CTAB,     SDESC|'2',      /* Clear Tabs */
        USCRL,    SDESC|'S',      /* Scroll Up */
        DSCRL,    SDESC|'Y',      /* Scroll Down */
        ASEG,     SDESC|'U',      /* Advance Segment */
/*      KBL,      NONE,          /* Keyboard Lock */
/*      KBU,      NONE,          /* Keyboard Unlock */
        UVSCN,    NONE,
        DVSCN,    NONE,
        0,
    };

```

Where NL is handled as `\33G\33B`, regardless of any mode. Mayhap it was optional in a previous version of the system?

For comparison, for the VT100 (*ibid.*, `cbunix10.pdf`, pp. 70-74 (`vt100.c`)), the tables alone are at least twice that.

A unique feature of CB-UNIX is `ioctl(DIOCSETS)`:

Set spy mode. All output directed to the terminal specified by *fildevs* will be copied to the terminal of the process performing the *ioctl*. Only one spy operation may be active in the entire system at any time. The spy continues until explicitly turned off. Currently, spy is only effective on lines using the `STD_LTYPE` line discipline and is restricted to the super-user.

It's unclear how the explicit turn-off is accomplished.

UNIX User's Manual, Release 3.0

Rattles off **110**, **150**, **300** and **1200** baud as "Common terminal speeds", with **2400**, **4800**, and **9600** "occasionally" available. The phone number spiel is generalised to

On some UNIX systems, there are separate telephone numbers for each available terminal speed, while on other systems several speeds may be served by a single telephone number.

and the itemised list of teletype speeds and how to handle them, depending on what the default is, to

In the latter case, there is a "preferred" speed; if you dial in from a terminal set to a different speed, you will be greeted by a string of meaningless characters (the **login:** message at the wrong speed.

Keep hitting the "break" or "attention" key until the **login:** message appears.

and likewise for the return/new-line post-login mapping tactics (rather, text lowered from the former *300-baud Terminals:* section is used verbatim; the note of preference for the new-line key where supported, previously part of the *TELETYPE® Model 37* (and other terminals less than 300 baud): section, is lost) — T. A. Dolotta, S. B. Olsson, and A. G. Petrucci, *UNIX User's Manual, Laboratory 364, Bell Telephone Laboratories, Incorporated, Release 3.0*, http://bitsavers.org/pdf/att/unix/System_III/UNIX_Users_Manual_Release_3_Jun80.pdf, p. 7, Murray Hill, NJ 07974, June 1980, How To Get Started, **Logging in.**

A record low of "two ways to log out" is recorded, by losing direct-**login** (*ibid.*, **Logging out.**).

The "On CB-UNIX only" note rings true — `^S/^Q` are the only flow control method, but now in **ixany** mode (*ibid.*, **How to communicate through your terminal.**, para. 4).

The individual drivers are rich: `dj(4)` (*ibid.*, p. 560) for DJ11s (all together now: incessantly called "DJ-11"s) — "Line speeds and other characteristics are not programmable but are set by switches in the hardware in groups of 4 lines. Only parameters such as character delays and mapping can be altered."† — `dz(4)` (*ibid.*, p. 566) for DZ11s (**dz**), DZ11/KMC11s (**dzk**), and DH11s (**dh**) (yes.) — the DZ11 note is exactly the same as on CB-UNIX; "The behavior of *dzk* lines is indistinguishable from that of *dz* lines, except that on the *dzk* backspace delays are implemented using fill characters (rubouts) instead of timed delays.", which corresponds to hard `[-]ofill` based on the driver‡; "Note that the DH-11 is considered obsolete and is not supported on the VAX-11/780." — same old `k1(4)` (*ibid.*, p. 571), though marked (**PDP-11 only**) in the header — on `/dev/console`, "Since they run at a constant speed, attempts to change the speed are ignored."

`init(8)` (ibid., p. 681) distinguishes entries by arbitrary $\{state, id\}$, with no relation to line state; if `no /etc/inittab` (`inittab(5)`, ibid., p. 618) is present or `init` is the only process, it runs a shell on `/dev/console` — this is described as more of a rescue state than a distinct single-user mode.

`getty(8)` (ibid., p. 674) sees a **SYNOPSIS** of

`/etc/getty` name type delay

since it now opens the line itself — "*Name* should be the name of a terminal in `/dev/` (e.g., `tty03`)." — and "*delay* is relevant for dial-up ports only. It specifies the time in seconds that should elapse before the port is disconnected if the user does not respond to the **login** request."

The text itself is a generalisation of the *Sixth Edition* into a mimicry of the *Seventh Edition* format. Rather than just the "Terminet 300 terminal" being put in full-duplex by the "login:" messages, they may additionally "take the DASI terminals out of the plot mode, or put a TELETYPE® Model 37 into full-duplex.". Returning "the GSI terminal to non-graphic mode" is lost.

The `type` table is compiled-in, but this is, at least, listed in the **BUGS**. A special (undisclaimed with the "(may vary locally)" for the others) type of `! type` "tells `getty` to update `/etc/utmp` and exit". What this update does is unspecified. The `types` recognised in the distribution *Release 3.0* are:

- B110; for 100 baud console TTY.
- 0** B300–B150–B110–B1200; normal dial-up sequence starting at B300.
- 1** B150; no sequence.
- 2** B2400; no sequence.
- 3** B1200–B300–B150–B110; normal dial-up sequence starting at B1200.
- 4** B300; for console DECwriter.
- 5** B9600; no sequence.
- 6** B4800–B9600; for Tektronix 4014.

The source itself is derived from *Sixth Edition* (provided to me by my good friend Juliano: UNIX System III Source Code, <https://vetsuware.com/download/UNIX%20System%20III%20Source%20Code%20SYSIII/?id=11576,src/cmd/getty.c>. Don't be a cop if you're affiliated with whoever bought the rights from SCO Group) and modes `'-'` and `'1'` agree completely, the `'0'` loop is extended with the **1200**-baud mode, and the new `'3'` just starts there, the `'2'` **9600**-baud mode is replaced, all up from there is new.

The `utmp(5)` (*Release 3.0*, p. 631) processing is done as the first thing and consists of setting the entry for `name` to $\{\text{current-time}, \emptyset\}$ in `/etc/utmp` and appending the same to `/usr/adm/wtmp`.

`name` is ignored if the standard input stream is a teletype, and the name of the attached teletype used instead; this is, presumably, to be used in conjunction with `!`, explained as

```
/* If the term type arg is special, exit only,
   this is so you can update utmp without spawning
   an open line
*/
```

which does just exit. Timing out on `delay` (if specified and non-zero) hangs up the line bimodally by setting it to `B0|HUPCL` and exiting. Some **Special Control Characters** are set to their defaults — `intr 0177 quit ^\ ("/* FS, cntl | *") erase # kill @ eof ^D eol 00` — from `<termio.h>` (ibid., `src/head/termio.h`), under the present-day `V`-prefixed indices and `C`-prefixed default macros no less!

`tty(4)` (*Release 3.0*, pp. 585-592) sees, fore-mostly, a fleshed-out controlling teletype and process group system: `/dev/tty` is "a synonym for the control terminal associated with the process group of that process, if any", and "The very first terminal file opened by the process group leader of a terminal file not already associated with a process group becomes the *control terminal* of that process group.", "A process can break this association by changing its process group using `setpgrp(2)`." — this corresponds *directly* to final present-day semantics and this is where high-ubiquitous techniques like double-forked daemonisation take their root.

"A line is delimited by a new-line (ASCII LF) character, an end-of-file (ASCII EOT) character, or an end-of-line character." — this fore-shadows an almost-normal interface, but one would be remiss to not note that EOT is End Of Transmission, hence why it wouldn't be outlandish for a teletype to hang up on its re-

ceipt.

'@' "optionally outputs a new-line character." and the table of "Certain characters" is re-tagged in terms of the logical actions, rather than the raw (now-default) values, and includes all characters with special meaning:

INTR	(rubout or ASCII DEL) [same text as <i>Seventh Edition</i> , incl. its all-processes-with-this-controlling-terminal target])
QUIT	(Control-I or ASCII FS) [likewise, though the core-dump filename is expounded]
ERASE	(#) erases the preceding character. It will not erase beyond the start of a line, as delimited by a NL, EOF, or EOL character.
KILL	(@) deletes the entire line, as delimited by a NL, EOF, or EOL character.
EOF	(Control-D or ASCII EOT) [likewise]
EOL	(ASCII NUL) is an additional line delimiter, like NL. It is not normally used.
STOP	(Control-S or ASCII DC3) can be used to temporarily suspend output. It is useful with CRT terminals to prevent output from disappearing before it can be read. While output is suspended, STOP characters are ignored and not read.
START	(Control-Q or ASCII DC1) is used to resume output which has been suspended by a STOP character. While output is not suspended, START characters are ignored and not read. The start/stop characters can not be changed or escaped.

The observant reader will note that (sans NL), these correspond directly to V-prefixed indices. Additionally, "The character values for INTR, QUIT, ERASE, KILL, EOF, and EOL may be changed to suit individual tastes. The ERASE, KILL, and EOF characters may be escaped by a preceding \ character, in which case no special function is done." This EOF handling is appears to be unique. No trimming of EOT on output is done.

<sgtty.h> as the container for the structure used by the "primary calls" <termio.h>:

```
#define NCC 8
struct termio {
    unsigned short c_iflag; /* input modes */
    unsigned short c_oflag; /* output modes */
    unsigned short c_cflag; /* control modes */
    unsigned short c_lflag; /* local modes */
    char c_line; /* line discipline */
    unsigned char c_cc[NCC]; /* control chars */
};
```

"The special control characters are defined by the array *c_cc*. The relative positions and initial values for each function are as follows:"

0	INTR	DEL
1	QUIT	FS
2	ERASE	#
3	KILL	@
4	EOF	EOT
5	EOL	NUL
6	reserved	
7	reserved	

This is the origin of the phrase "**Special Control Characters**", and, more importantly, *struct termio* is pretty much exactly as present-day, and so are the indices (mostly, see below).

"The *c_iflag* field describes the basic terminal input control", and is outlined below.

<i>c_iflag</i>	Equiv.	Description
IGNBRK	ignbrk	Ignore break condition.
BRKINT	brkint	Signal interrupt on break.
IGNPAR	ignpar	Ignore characters with parity errors.
PARMRK	parmrk	Mark parity errors.
INPCK	inpck	Enable input parity check.
ISTRIP	istrip	Strip character.

INLCR	inlcr	Map NL to CR on input.
IGNCR	igncr	Ignore CR cr.
ICRNL	icrnl	Map CR to NL on input.
IUCLC	iucle	Map upper-case ot lower-case on input.
IXON	ixon	Enable start/stop output control.
IXANY	ixany	Enable any character to restart output.
IXOFF	ixoff	Enable start/stop input control.

The attentive reader may note that these are *all* as present-day, sans the lack of foreground/background process groups (signals are targeted as for SIGINT) and START/STOP always being stripped from the input (nowadays they're passed through if not IXON). The less-obvious flags are described below the original table; they are mostly found with mildly-evolved wording in IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1, where IUCLC is lost. "The initial input control value is all bits clear."

"The *c_oflag* field specifies the system treatment of output", and is outlined below.

<i>c_oflag</i>	Equiv.	Description
OPOST	opost	Postprocess output.
OLCUC	olcuc	Map lower case to upper on output.
ONLCR	onlcr	Map NL to CR-NL on output.
OCRNL	ocrnl	Map CR to NL on output.
ONOCR	onocr	No CR output at column 0.
ONLRET	onlret	NL performs CR function.
OFILL	ofill	Use fill characters for delay.
OFDEL	ofdel	Fill is DEL, else NUL.
NLDLY	(mask)	Select new-line delays:
NL0	nl0	
NL1	nl1	
CRDLY	(mask)	Select carriage-return delays:
CR0	cr0	
CR1	cr1	
CR2	cr2	
CR3	cr3	
TABDLY	(mask)	Select horizontal-tab delays:
TAB0	tab0	
TAB1	tab1	
TAB2	tab2	
TAB3	tab3	Expand tabs to spaces.
BSDLY	(mask)	Select backspace delays:
BS0	bs0	
BS1	bs1	
VTDLY	(mask)	Select vertical-tab delays:
VT0	vt0	
VT1	vt1	
FFDLY	(mask)	Select form-feed delays:
FF0	ff0	
FF1	ff1	

They just don't miss! In IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1, OPOST is the only truly portable flag, all others are shaded XSI (X/Open Systems Interfaces — extension), and, symmetrically, OLCUC is lost (*Release 3.0* even points out that "This function is often used in conjunction wiht IUCLC."). The *entire* description for the now-XSI parts lives *verbatim* in the standard. "The initial output control value is all bits clear."

"The *c_cflag* field specifies the hardware control of the terminal", and is outlined below.

<i>c_cflag</i>	Equiv.	Description
CBAUD	(mask)	Baud rate:
B0	0	Hang up
B50	50	50 baud
B75	75	75 baud
B110	110	110 baud
B134	134	134.4 baud
B150	150	150 baud
B200	200	200 baud
B300	300	300 baud
B600	600	600 baud
B1200	1200	1200 baud
B1800	1800	1800 baud
B2400	2400	2400 baud
B4800	4800	4800 baud
B9600	9600	9600 baud
EXTA	exta	External A
EXTB	extb	External B
<hr/>		
CSIZE	(mask)	Character size:
CS5	cs5	5 bits
CS6	cs6	5 bits
CS7	cs7	6 bits
CS8	cs8	7 bits
<hr/>		
CSTOPB	cstopb	Send two stop bits, else one.
CREAD	cread	Enable receiver.
PARENB	parenb	Parity enable.
PARODD	parodd	Odd parity, else even.
HUPCL	hupcl	Hang up on last close.
CLOCAL	clocal	Local line, else dial-up.

"For any particular hardware, impossible speed changes are ignored.", and note that this is a global (RX and TX) speed — this further supports the hypothesis that no-one has ever actually used uneven speeds.

The same deal as for *c_iflag* applies; the speeds are a separate table with specialised setters (see below), but are otherwise identical, except EXTA, EXTB are spelled as B19200, B38400 with the expected descriptions; IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 has an additional explicit side-effect for CLOCAL as functioning as-if O_NONBLOCK: this is attested in the drivers and makes logical sense, but not laid forth. "The initial hardware control value after open is B300, CS0, CREAD, HUPCL.""

"The *c_lflag* field of the argument structure is used by the line discipline to control terminal functions. The basic line discipline (0) provides" flags outlined below.

<i>c_lflag</i>	Equiv.	Description
ISIG	isig	Enable signals.
ICANON	icanon	Canonical input (erase and kill processing).
XCASE	xcase	Canonical upper/lower presentation.
ECHO	echo	Enable echo.
ECHOE	echoe	Echo erase character as BS-SP-BS.
ECHOK	echok -echoke	Echo NL after kill character.
ECHONL	echonl	Echo NL.
NOFLSH	noflsh	Disable flush after interrupt or quit.

Same deal here, but this is the first field that's missing flags found in IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 (IEXTEN and TOSTOP). ISIG is, naturally, not yet SUSP-capable. The standard defines ECHOE as its function — erasing the last character — rather than the output sequence. ECHOK is allowed to do either [-] **echoke** mode — this system reasonably does **-echoke** since it doesn't have the

flag. XCASE is lost.

The only weirdness here is that in the !ECHO ECHOE case, in *Release 3.0* "the erase character is echoed as ASCII SP BS". It also says "Note that an escape character preceding the erase or kill character removes any special function." (and likewise for EOT), for which no provisions are made in the standard, and this is not reproducible on modern systems. ECHONL is noted to be "useful for terminals set to local echo (so-called half duplex)". "The initial line-discipline control value is all bits clear."

The "lost" flags all last appear in Version 2 of the Single UNIX Specification ("SUSv2") and are only defined for the POSIX locale. This is a bit of an issue when the kernel has to process them (alternatively, "mangle my preferred non-seven-bit-ASCII encoding").

The ISIG "special input functions may be disabled individually by changing the value of the control character to an unlikely or impossible value (e.g. 0377)". Admittedly, the common spelling is `_POSIX_VDISABLE` nowadays, and that value is usually handled explicitly, but yeah. There's also nothing preventing this working from other remappable values in ICANON, this is likely just suboptimal phrasing location.

"The MIN and TIME values are stored in the position for the EOF and EOL characters respectively.", and to this day IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 requires that indices for all **Special Control Characters** are unique, "except that the VMIN and VTIME subscripts may have the same values as the VEOF and VEOL subscripts, respectively."

Quite funnily, there is a marked *loss* of function against *Seventh Edition*, since START and STOP can't be remapped. There is also a provision for this (but a lesser one, in that they may just be ignored when setting, but must be otherwise provided).

Admittedly, there is also a loss of `ioctl`(s): since all parameters are integrated into one complete package, the `struct tchars TIOC[GS]ETC` requests are lost entirely, the new spellings (well, the old ones are still supported as `<sys/ttold.h>` compat shims, but)

TIOCGETP→TCGETA	Get the parameters associated with the terminal and store it in the <i>termio</i> structure referenced by arg .
TIOCSETN→TCGETA	Set the parameters associated with the terminal from the structure referenced by arg . The change is immediate.
TCSETAW	Wait for the output to drain before setting the new parameters. This form should be used when changing parameters what will affect output.
TIOCSETP→TCGETAF	Wait for the output to drain, then flush the input queue and set the new parameters.

TIOCHPCL is removed, since it's once again provided by HUPCL.

The system ships with only one line discipline, **0**, imaginatively branded

```
/*
 * Line discipline 0
 */
```

IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 says in its **stty RATIONALE**:

The original *stty* description was taken directly from System V and reflected the System V terminal driver **termio**. It has been modified to correspond to the terminal driver **termios**.

But this doesn't appear to be supported by either the source or the manual itself — "termio" is just the "terminal `ioctl`()" argument and its declaration and arguments live in an eponymous header. This is like calling the pre-*Release 3.0* `tty.c` "the UNIX **sgtty** driver". Parameterising it differently (hell, not even that much against *Seventh Edition*, if anything the driver is simpler) and putting the discipline-specific routines into `tt0.c` didn't really make it *different*. "the terminal driver **termios**" is also *literally* just *Release 3.0* `tty(4)` plus insubstantial wording fiddling to make it standard. To a *staggering* degree.

A "synchronous terminal interface" is included as `st(4)` (*ibid.*, p. 582), described as "a pseudo-device driver that enables a UNIX system to communicate with a TELETYPE® Model 40/0 ASCII synchronous terminal." utilising `vpm(4)` ("The Virtual Protocol Machine"; *ibid.*, 594), itself a KMC11 (this time spelled correctly) driver and compiler toolchain that "translates a high-level language description protocol into an intermediate language that is interpreted by an interpreter running in the KMC11." — this machinery is entirely unrelated to the teletype driver and line disciplines, and driven entirely from

user-space (though a default is provided the, cf. `st(1M)` (*ibid.*, p. 343) and *UNIX System III Source Code src/cmd/st/*, in particular `mod40.r` therein; the corresponding `st(4)` driver lives in `src/uts/pdp11/pwb/st.c` and `src/uts/vax/pwb/st.c`, the former branded "Synchronous terminal interface for the PDP 11." and the port "TTY 40/4 driver for VAX 11/780").

`ioctl(2)` (*Release 3.0*, p. 446) is a stub even more so than the *Seventh Edition* one, giving only the prototype and errors.

`login(1)` (*Release 3.0*, p. 240) — "sign on" — is delightfully uneventful; passphrases may now expire, delightfully termed "password aging", and the ">dialup« password" becomes "a second »external« password", with a prompt of "External security:", and "If you do not complete the login successfully within a certain period of time (e.g., one minute), you are likely to be silently disconnected.", which goes hand-in-hand with `getty`'s `delay`.

`stty(1)` (*ibid.*, pp. 348-350) sees a **SYNOPSIS** of

```
stty [ -a ] [ -g ] [ options ]
```

and switches to using "the device that is the current standard input" (as opposed to output, and more a saner description). Without `-ag`, "settings of certain options" are reported; with `-a`, "all of the option settings", and with `-g`, "current settings in a form that can be used as an argument to another `stty` command." — the options are fully formed here, and the output is directed to the standard output stream.

It's odd that this is the first system noting that "many combinations of the options make no sense, but no sanity checking is performed". The settable options follow. ← indicates the argument is identical to present-day (← with notes); → indicates the description is the same as in *Seventh Edition* (to the precision of whitespace, punctuation, and link to `tty(4)`).

Control Modes	Description
<code>parenb (-parenb)</code>	← enable (disable) parity generation and detection.
<code>parodd (-parodd)</code>	← select odd (even) parity.
<code>cs5 cs6 cs7 cs8</code>	← select character size (see <code>tty(4)</code>).
<code>0</code>	↔ hang up phone line immediately.
<code>50 75 110 134 150 200 300 600 1200 1800 2400 4800 9600 exta extb</code>	↔ Set terminal baud rate to the number given, if possible (these are the speeds supported by the DH-11 interface).
<code>hupcl (-hupcl)</code>	← hang up (do not hang up) DATA-PHONE® connection on last close.
<code>hup (-hup)</code>	← same as <code>hupcl (-hupcl)</code> .
<code>cstopb (-cstopb)</code>	← use two (one) stop bits per character.
<code>cread (-cread)</code>	← enable (disable) the receiver.
<code>local (-local)</code>	← assume a line without (with) modem control.

Input Modes	Description
<code>ignbrk (-ignbrk)</code>	← ignore (do not ignore) break on input.
<code>brkint (-brkint)</code>	← signal (do not signal) INTR on break.
<code>ignpar (-ignpar)</code>	← ignore (do not ignore) parity errors.
<code>parmrk (-parmrk)</code>	← mark (do not mark) parity errors (see <code>tty(4)</code>).
<code>inpck (-inpck)</code>	← enable (disable) input parity checking.
<code>istrip (-istrip)</code>	← strip (do not strip) input characters to seven bits.
<code>inlcr (-inlcr)</code>	← map (do not map) NL to CR on input.
<code>igncr (-igncr)</code>	← ignore (do not ignore) CR on input.
<code>icrnl (-icrnl)</code>	← map (do not map) CR to NL on input.
<code>iucL (-iucL)</code>	← map (do not map) upper-case alphabetic to lower case on input.
<code>ixon (-ixon)</code>	← enable (disable) START/STOP output control. Output is stopped by sending an ASCII DC3 and started by sending an ASCII DC1 .
<code>ixany (-ixany)</code>	← allow any character (only DC1) to restart output.
<code>ixoff (-ixoff)</code>	← request that the system send (not send) START/STOP characters when the input queue is nearly empty/full.

Output Modes	Description
opost (-opost)	← post-process output (do not post-process output; ignore all other output modes).
olcuc (-olcuc)	← map (do not map) lower-case alphabetic to upper case on output.
onlcr (-onlcr)	← map (do not map) NL to CR-NL on output.
ocrnl (-ocrnl)	← map (do not map) CR to NL on output.
onocr (-onocr)	← do not (do) output CRs at column zero.
onlret (-onlret)	← on the terminal NL performs (does not perform) the CR function.
ofill (-ofill)	← use fill characters (use timing) for delays.
ofdel (-ofdel)	← fill characters are DELs (NULs).
cr0 cr1 cr2 cr3	↔ select style of delay for carriage returns (see <i>tty(4)</i>).
nl0 nl1	← select style of delay for line-feeds (see <i>tty(4)</i>).
tab0 tab1 tab2 tab3	← select style of delay for horizontal tabs (see <i>tty(4)</i>).
bs0 bs1	↔ select style of delay for backspaces (see <i>tty(4)</i>).
ff0 ff1	↔ select style of delay for form-feeds (see <i>tty(4)</i>).
vt0 vt1	← select style of delay for vertical tabs (see <i>tty(4)</i>).
Local Modes	Description
isig (-isig)	← enable (disable) the checking of characters against the special control characters INTR and QUIT.
icanon (-icanon)	← enable (disable) canonical input (ERASE and KILL processing).
xcase (-xcase)	← canonical (unprocessed) upper/lower-case presentation.
echo (-echo)	← echo back (do not echo back) every character typed.
echoe (-echoe)	← echo (-echo) (do not echo) ERASE character as a backspace-space-backspace string. Note: this mode will erase the ERASEed character on many CRT terminals; however, it does <i>not</i> keep track of column position and, as a result, may be confusing on escaped characters, tabs, and backspaces.
echok (-echok)	← echo (do not echo) NL after KILL character.
lfkc (-lfkc)	the same as echok (-echok); obsolete.
echonl (-echonl)	← echo (do not echo) NL.
noflsh (-noflsh)	← disable (enable) flush after INTR or QUIT.
Combination Modes	Description
evenp or parity	← enable parenb and cs7 .
oddp	← enable parenb , cs7 , and parodd .
-parity , -evenp , or -oddp	← disable parenb , and set cs8 .
raw (-raw or cooked)	enable (disable) raw input and output (no ERASE, KILL, INTR, QUIT, EOT, or output post processing).
nl (-nl)	← unset (set) icrnl , onlcr . In addition -nl unsets inlcr , igncr , ocrnl , and onlret .
lcase (-lcase)	← set (unset) xcase , iucle , and olcuc .
LCASE (-LCASE)	← same as lcase (-lcase).
tabs (-tabs or tab3)	← preserve (expand to spaces) tabs when printing.
ek	← set ERASE and KILL characters back to normal # and @.
sane	resets all modes to some reasonable values.
term	set all modes suitable for the terminal type <i>term</i> , where <i>term</i> is one of tty33 , tty37 , vt05 , tn300 , ti700 , or tek .

The "134.5" spelling of "134" is retained, but undocumented.

"DATA-PHONE@" is a line of Bell modems. The only spelling change from "[**-**]hup" being trademarkisation of "dataphone" is, well.

[**-**]iucle is predicated on **ixten** since its loss in Version 3 of the Single UNIX Specification ("SUSv3").

"[**-**]lfkc" is the only psycho mode here. There's never been any mention of it anywhere before.

"[**-**]isig" is the same to the precision of *Release 3.0* not having job control. Modern [**-**]echoe implementations are usually smarter than described and *do* process tabs normally, and \-escaping is thankfully lost.

Embedded backspaces in `-echoctl` mode are usually sufficiently confusing, though.

"`echok`" corresponds to `echok -echoke` mode.

"`evenp`" and "`parity`" are documented as `parenb cs7` but are actually also `-parodd`, so as present-day.

"`ek`" uses `CERASE` and `CKILL` macros, so it is the same, but the system defaults are different.

"`sane`" is `-ignbrk brkint ignpar -parmrk -inlcr -igncr icrnl -iucle ixon -ixoff opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 ff0 vt0 cs7 cread parenb -parodd -clocal isig icanon -xcase echo -echoe echok -echonl -noflsh` and all **Special Control Characters** default. This corresponds to, going from *Release 3.0* to present-day, a loss of `ignpar -parmrk -inlcr -igncr icrnl -iucle ixon -ixoff opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 ff0 vt0 cs7 cread parenb -parodd -clocal isig icanon -xcase echo -echoe echok -echonl -noflsh` and gain of `-ixany imaxbel -iutf8 echoe echoke -tostop echoctl -echoprt -flusho iexten -extproc`.

The `terms` are the same as in *Seventh edition*, but the short forms are lost.

Due to the `VMIN/VTIME VEOF/VEOL` overlap, "`raw`" (all **Input Attributes** clear, `-opost cs8 -parenb -isig -icanon -xcase`) sets `min 1 time 1` (or, since no parsing provisions are made for `min time`, "`min ^A time ^A`") "`-raw`"/"`cooked`" (`brkint ignpar istrip icrnl ixon opost cs7 parenb isig icanon`) set `eof eol` to default. The latter two (sans `cs7`) is as present-day.

The **Control Assignments** table reads, for "`control-character c`":

set `control-character` to `c`, where `control-character` is `erase`, `kill`, `intr`, `quit`, `eof`, `eol`, `min`, or `time` (`min` and `time` are used with `-icanon`; see `tty(4)`). If `c` is preceded by an (escaped from the shell) caret (^), then the value used is the corresponding CTRL character (e.g., "`^d`" is a **CTRL-d**); "`^?`" is interpreted as DEL and "`^-`" is interpreted as undefined.

and for "`line i`" — "set line discipline to `i` ($0 < i < 127$)". The argument is a decimal integer, and the range enforced by the kernel.

It's interesting to note here that "`^-`" — **0377** (*UNIX System III Source Code*, `src/cmd/stty.c`) — is explicitly described as "undefined", rather than just "unlikely". No special provisions are made for `0xFF` in the kernel, it *is* just very unlikely — enough to be used as an undefined value. The gain of "`^?`" matches that of *Seventh Edition, Virtual VAX-II Version, December, 1979*. It's unclear what "escaped from the shell" means — ^ doesn't appear to be special to `sh(1)` (*Release 3.0*, pp. 237-333).

The `-g` format is the same as this implementation's (**Saved Attributes**), and read back the same way.

The first line of the no-argument format starts with

```
speed 300 baud; evenp
(or oddp as the case may be), or
speed 300 baud; -parity
```

if `-parenb`. This continues with the character size if `parenb /cs7` or `-parenb /cs8`, and as usual with defaults of `-cstopb -hupcl cread -clocal` (though, if `-cread`, "`cread`" is written; this can only be a bug).

The second line starts with "`line = c_line;`" (if not `0`) and continues with all **Special Control-Characters** separated with semicolons, as, given `intr ^- quit ^? erase ^E kill K eof ^@ eol 0xD8` (which is uninputtable, but):

```
intr <undef>; quit = DEL; erase = ^e; kill = K; eof = ^`; eol = -X;
```

The third line is the **Input Modes**, with defaults of `-ignbrk -brkint inlcr -ignpar -parmrk istrip -inlcr -igncr -icrnl -iucle ixon ixany -ixoff` (`ignbrk` hides the value of `[-]brkint`; `-inlcr` — of `[-]ignpar`; `-xon` — of `[-]ixany`).

The fourth line is the **Output Modes**: either `-opost`, or flags in manual order, default off (except for `ofill ofdel` rendered as "del-fill" and `ofill -ofdel` rendered as "nul-fill") and `0` delays hidden. These (and two others, all listed below) are the only modes accepted as input but not noted in the manual.

The fifth line is the **Local Modes**, with defaults of `isig icanon -xcase`, `[-]echo [-]echoe [-]echok` always written, then of `-echonl -noflsh`,

The `-a` format is similar, but arranged differently: the first line lists the speed, line discipline, and **Special Control Characters**; the second the **Control Modes**, third — **Input Modes** (broken into fourth before `[-]ixon`), fifth — **Local Modes**, sixth — **Output Modes**, all in manual order with no aggregation.

Secret mode	Is
fill	ofill –ofdel
–fill	–ofill –ofdel
nul-fill	ofill –ofdel
del-fill	ofill ofdel

UNIX User's Manual, Release 3.0, add'1 references

†:

DJ11 asynchronous 16-line multiplexer maintenance manual, EK-DJ11-MM-003, 4th Printing, http://bitsavers.org/pdf/dec/unibus/EK-DJ11-MM-003_DJ11_Maint_Man_Aug76.pdf, p. 1-5, Digital Equipment Corporation, Maynard, Massachusetts, August 1976, General Description, 1.6 Specifications, **Performance, Receivers and Transmitters, Data Rate.**

‡:

This configuration is called out explicitly in the manual:

A typical application of the KMC11 is the control of several peripheral devices attached to the UNIBUS. These devices, for example the DZ11 8-line asynchronous multiplexer, typically operate by programmed I/O, interrupting the PDP-11 processor for each character input or output. However, with the addition of a KMC11, this processor overhead can be substantially reduced and I/O throughput increased.

ofilling delays doesn't appear to be a fundamental side-effect of this. *KMC11 Programmer's Manual*, AA-5244B-TC, Second Printing, http://bitsavers.org/pdf/dec/unibus/KMC11_ProgMan.pdf, p. 1-2, Digital Equipment Corporation, Maynard, Massachusetts 01754, December 1977, Introduction, 1.2 KMC11 General Description, 1.2.1 Controlling Peripherals over the UNIBUS.

UNIX System User's Manual, System V; UNIX Administrator's Manual, Release 5.0

Yes, there's two of them, and yes, they're branded differently; this coincides with the Bell System break-up, and the User's Manual title page has a beautifully-crafted "we're kinda mad about this but don't wanna be openly super mad about it" blurb — *UNIX System User's Manual, System V, Western Electric Company, Incorporated.*, 301-905 Issue 1, http://bitsavers.org/pdf/att/unix/System_V_Release_1/301-905_UNIX_System_V_Release_1_Users_Manual_Jan83.pdf, p. 1, January 1983.

The manual sections are helpfully rearranged: i.a., devices ("Special Files") now live in section 7 — *UNIX System Administrator's Manual, Release 5.0, Bell Laboratories, Inc.*, 301-926 ISS. 1, http://bitsavers.org/pdf/att/unix/System_V_Release_1/301-926_UNIX_System_V_Release_1_Administrators_Manual_Jun82.pdf, p. 3, June 1982, Introduction.

The haphazard assembly of random uploadable code is productised as `stermio(7)` (ibid., p. 234) — "general synchronous terminal interface" — which roughly simulates the normal teletype interface, alongside `stgetty(1M)` (ibid., p. 160) and `stlogin(1)` (*UNIX System User's Manual, System V*, p. 351). We are, thankfully, saved from a hypothetical `ststty(1)`, but not `scat(1)` (ibid., p. 305).

`login(1)` (ibid., pp. 231-232)'s "»external« password" reverts to its CB-UNIX spelling (or, more likely, the ~~in-breeding~~cross-pollination goes the other way, or, likelier still, since CB-UNIX is noted as a source system for *Release 3.0* and post-dates it, both).

The same holds for `getty(1M)` (*UNIX System Administrator's Manual, Release 5.0*, pp. 116-117), they're both dated March 13, 1981 (with header for "CB-UNIX 2.3") in the *CB-UNIX Programmer's Manual*, so it's best-described in terms of the differences from *there*. "[`-t` timeout]" "specifies that *getty* should exit if the open the line succeeds and no one types anything in the specified number of seconds." — this is essentially a port of *Release 3.0 delay*. The *type* list is (unexpectedly?) *shorter*:

none	default
vt61	DEC vt61
vt100	DEC vt100
hp45	Hewlett-Packard HP45
c100	Concept 100

And caveatted:

Also, for terminal type to have any meaning, the virtual terminal handlers must be compiled into the operating systems. They are available, but not compiled in the default condition. The more robust line discipline zero is infinitely better than the gang of CB-UNIX ones, so so is *linedisc*:

Again the hooks for line disciplines are available in the operating system but there is only one presently available, the default line discipline, **LDISC0**.

`^U` is also understood when reading the username, and **kill** adjusted. If the "»standard« ESS2" (note the odd updated branding) line editing characters are seen, "it arranges for this set of characters to be used for these functions.", rather than setting an exotic mode, since they all do map here, as **intr & erase _ kill \$ eol / eol2 !** — this checks out in the source, likewise provided to me by my good friend Juliano: UNIX SYSVr1 Source Code, <https://vetusware.com/download/UNIX%20SYSVr1%20Source%20Code%20r1/?id=11570>, `sysv-pdp11_usr-src/cmd/getty.c`.

`dz(7)` (*UNIX System Administrator's Manual, Release 5.0*, p. 199) is marked "(DEC only)"; the "(3B20S only)" `tn4(7)` (*ibid.*, p. 246) and `tn74(7)` (*ibid.*, p. 257) appear, identical except the former is eight-line and the latter – two-line: "The `c_flag` items of **B200**, **EXTA**, and **EXTB** are not available.", which sounds an awful lot *just* like the DZ11.

`tty(4)` is split into `tty(7)` (*ibid.*, p. 252) — "controlling terminal interface" — with the `/dev/tty` description paragraph and `termio(7)` (*ibid.*, pp. 238-244) contains the rest, unchanged.

Of course, the header itself *does* change, if only to accomodate the `STDTTY` replacement — now `<sys/termio.h>`, `VEOL2` is allocated as `c_cc[6]` (*UNIX SYSVr1 Source Code, sysv-pdp11_usr-src/uts/sys/termio.h*). Video-terminal-specific flags also appear — the list of terminals is taken *verbatim*, down to "Hewlett", from CB-UNIX, as are the flags, and argument format (this is also copied *verbatim*, save for the new `ioctl()` names):

```
/*
 * structure of ioctl arg for LDGETT and LDSETT
 */
struct termcb {
    char    st_flg;          /* term flags */
    char    st_termt;       /* term type */
    char    st_crow;        /* gtty only - current row */
    char    st_ccol;        /* gtty only - current col */
    char    st_vrow;        /* variable row */
    char    st_lrow;        /* last row */
};
```

`stty(1)` (*UNIX System User's Manual, System V*, pp. 355-357) is unchanged, save for the addition of some `sttermio(7)`-specific modes. These are outlined below, but they aren't "actually" local, since that interface is much simpler (if insane).

Local Modes	Description
stwrap (<code>-stwrap</code>)	disable (enable) truncation of lines longer than 79 characters on a synchronous line.
stflush (<code>-stflush</code>)	enable(disable) flush on a synchronous line after every <code>write(2)</code> .
stappl (<code>-stappl</code>)	use application mode (use line mode) on a synchronous line.

More importantly, a copy-paste error that meant `-lcase` included `-xcase` but `-LCASE` didn't is fixed — (*UNIX SYSVr1 Source Code, sysv-pdp11_usr-src/cmd/stty.c*).

This does leave the system in an odd state where `VEOL2` is supported (*ibid.*, `sysv-pdp11_usr-src/uts/pdp11/io/tt?.c`), but only settable by **getty**, and not even observable with **stty**. This must be a fun mode to try to get out of if you trigger it by accident.

UNIX System V Release 2.0

The only way you'd know that these are AT&T System V Release 2 UNIX is by process of elimination, since that's the largest release number that appears on exactly one page out of the entire 530-page volume — Steven V. Earhart, *UNIX Programmer's manual, AT&T, CBS College Publishing's UNIX System*

Library, Select Code 230-031 ISBN 0-03-009317-1, Volume 1: Commands and utilities, http://bitsavers.org/pdf/att/unix/System_V_Release_2/UNIX_Programmers_Manual_Vol_1_Commands_and_Utilities_1986.pdf, pp. 77-78, 1986, `convert(1)`.

Rather than saying that @ "»kills«" all prior characters on the line, it now "cancels" them (*ibid.*, Introduction, How To Get Started, How to communicate through your terminal, para. 3, p. vii).

`stermio(7)` and friends appear to be gone, and the individual driver pages are lost — Steven V. Earhart, *UNIX Programmer's manual, AT&T, CBS College Publishing's UNIX System Library*, Select Code 230-033 ISBN 0-03-009313-9, Volume 3: System administration facilities, http://bitsavers.org/pdf/att/unix/System_V_Release_2/UNIX_Programmers_Manual_Vol_3_System_Administration_Facilities_1986.pdf, pp. vii-viii, 1986, Table of Contents.

`termio(7)` (*ibid.*, pp. 119-129) now has "SWTCH" allocated as `c_cc[7]` with no default listed, but the summary describes it as

(Control-z or ASCII SUB) is used by the job control facility, *shl*, to change the current layer to the control layer.

Picking SUB(stitute) is probably the only sane part of this model of "job control".

ISIG and NOFLSH also govern **switch** p.a. to **intr quit**.

A `c_cflag` LOBLK ("Block layer output.") is added, described as

If LOBLK is set, the output of a job control layer will be blocked when it is not the current layer.

Otherwise the output generated by that layer will be multiplexed onto the current layer.

Which is kinda like **tostop** if **tostop** just blocked instead of SIGTTOUing.

The **SEE ALSO** defers to `shl(1)` (*UNIX Programmer's manual, Volume 1, 1986*, pp. 397-398) — "shell layer manager", no-argument **SYNOPSIS**,

Shl allows a user to interact with more than one shell from a single terminal. The user controls these shells, known as *layers*, using the commands described below.

The I/O is as-expected — reading from a background "layer" blocks, setting "[**-**]loblk" "within the layer" (since all of these sessions are just on different teletypes) is as-described.

The *stty* character **switch** (set to ^Z if NUL) is used to switch control to *shl* from a layer. *Shl* has its own prompt, >>>, to help distinguish it from a layer.

Notably, this is the *only* instance of special treatment of a character, and there *is* an established C-prefix convention which is used for everything else *and* this — CSWTCH is defined correctly — but the kernel doesn't use it, and initialises it to 0. At least the special treatment is only in the userspace, and **shl** sets **switch** to CSWTCH if it's... CNSWTCH, which is defined to 0 just below its non-no variant. Note also that this *overlaps* with the default settings for **eol eol2** — and takes precedent over them, so "EOL", whose description starts with "(ASCII NUL)" is invalidated by default if **isig**.

A *layer* is a shell which has been bound to a virtual tty device (`/dev/sxt???`). The virtual device can be manipulated like a real tty device using *stty* (1) and *ioctl* (2). Each layer has its own process group id.

There's a prompt, so this is naturally conversational. A *summary* of meaningful commands (for the full inglorious, see the original) is included below, with their job control equivalents. For the purposes of this analysis, one could consider a ^Z here and ^Z, **bg** on a modern ("BSD-style") system the same fundamental operation; things diverge from there.

Command	Equivalent	Add'l summary
create [<i>name</i>]	<code>PS1="name " sh</code>	The default <i>name</i> is a digit — the last character of the layer's <code>sxt(7)</code> device — in parentheses. Explicit "(<i>digit</i>)" <i>names</i> are therefore illegal.
delete name [<i>name ...</i>]	<code>kill -HUP %name...</code>	Deletes the specified layers; the SIGHUPping is a result of that.

layers [-l] [<i>name ...</i>]	jobs [<i>%name...</i>]	Roughly. No- l lists just the names and "process group" (which is more akin to jobs -p). "The -l option produces a <i>ps</i> (1)-like listing.", which is closer to present-day no-flag jobs .
resume [<i>name</i>]	fg or fg %name	
toggle	fg -	"the layer that was current before the last current layer"
quit	^D	All layers get SIGHUPped.
<i>name</i>	fg %name	

Of course, modern shells will decidedly object to being job-controlled, but in the context of *Seventh Edition, Virtual VAX-11 Version, November, 1980* under **cs**h (with a job-control-agnostic **sh**), this does hold (as verified on a 4.2BSD system). Not that in general you'd actually want to job-control entire shell sessions, but.

Hidden in the **create** description is the blurb that "A maximum of seven layers can be created." This is by far the worst "job control" scheme ever devised; its only "redeeming" quality is that *sxt*(7) (*UNIX Programmer's manual, Volume 3, 1986*, pp. 116-118) provides almost-general virtual teletypes.

getty(1M) (*ibid.*, pp. 64-66) appears to lose both the ESS and '**b**'/'**^U**' processing, but doesn't, it just stops being documented. Also undocumented is that it starts setting **xc**ase for lines it deems upper-case-only — this is apparent in the source, similarly provided to me by my good friend Juliano: *UNIX SYSVr2 Source Code*, <https://vetusware.com/download/UNIX%20SYSVr2%20Source%20Code%20r2/?id=11571>, `src/cmd/getty.c`.

It's important to note that the manuals post-date the system by approximately two years — the kernel branding is for 1984 (*ibid.*, `src/uts/vax/os/main.c`) and other, themselves unsourced, secondary sources agree.

login(1) (*UNIX Programmer's manual, Volume 1, 1986*, pp. 232-234) gains a fun facility where if the shell *passwd*(5) field is *, it chroots to the home directory and reexecs `/etc/login`,

The aforementioned **swtch/eol[2]** precedence is *UNIX SYSVr2 Source Code*, `src/uts/vax/io/tt0.c`, `ttin()`, `if (flg) while (nchar--)`.

termio(7) Extends "or the timeout value TIME has expired" to "has expired between characters"; *ibid.* and `tt[y1].c` corroborate that this just a wording change.

Ibid., `src/uts/sys/termio.h` concretifies speeds:

```
#define B19200 0000016
#define B38400 0000017
/* EXTA and EXTB should be deleted when */
/* cleaned out of the source */
#define EXTA 0000016
#define EXTB 0000017
/* they are aliases for b19200 and B38400 */
```

This clean-out doesn't appear to have been done, and there are zero users, but.

stty(1) (*UNIX Programmer's manual, Volume 1, 1986*, pp. 420-423) loses the *st*ermio(7) references (but *not* the "**st***" modes), gains the LOBLK proxy, below, adds SWTCH notes to "**isig (-isig)**", "**noflsh (-noflsh)**", and "**raw (-raw or cooked)**" and a **swtch** one to **Control Assignments**.

In "**hupcl (-hupcl)**" "a DATA-PHONE@ data set connection" is reduced to "DATA-PHONE@ connection".

The description for "**llocal (-llocal)**" starts with an 'n', for "n assume...". Unclear why.

Control Modes	Description
----------------------	--------------------

loblk (-loblk)	block (do not block) output from a non-current layer.
-----------------------	---

UNIX SYSVr2 Source Code, `src/cmd/stty.c` lines up with this, except all mentions of LOBLK and VSWTCH are predicated on not being built for the PDP-11. Presumably due to memory constraints? But enforcing this *here* is odd.

The default for no-argument output is "**-loblk**".

UNIX® System V User's Reference Manual; AT&T System V.3 Administrator's Reference Manual

In this case the only way to figure out that this 350-page volume is for AT&T System V Release 3 UNIX is in the **SEE ALSO** section in *UNIX® User's Reference manual, Prentice-Hall C and UNIX® Systems Library*, ISBN 0-13-940487-2 025, http://bitsavers.org/pdf/att/unix/System_V_Release_3/UNIX_System_V_Users_Reference_Manual_1987.pdf, pp. 116-117, AT&T, 1987, 1986, `getopts(1)`.

The "valid user name" requirement is changed to "valid login id" — the purpose of this is unclear, unless the editor was smoking on that psychoanalysis mid — and the "common"/"occasionally available" dichotomy is hegelled by commonalising all of them and removing **110**, **150**, and **300** baud, all of which between ten- and four-fold slower than the slowest listed, now **1200**, **2400**, **4800**, and **9600** baud (*ibid.*, Introduction, How To Get Started, Logging In, paras. 1-2, p. 4).

"you can simply hang up the phone" is for the first time predicated — "If you've dialed in," (*ibid.*, Logging Out, p. 5).

The gnome is fired, replaced by impersonal language; automation gets to us all in the end (*ibid.*, How to Communicate Through Your Terminal, para. 1, p. 5).

"(@ is called the line kill character.)" — which wasn't actually clear before (*ibid.*, para. 3). One wonders what the purpose of changing the description in the first place was.

The first occurrence of the "**CONTROL-**" notation explains how to type the characters described ("pressing the control key and alphabetic s simultaneously"). The rest of the **stop/start** paragraph is rewritten to be almost cute (if you forget that this is AT&T product):

Thus, if you had typed **cat yourfile** and the contents of **yourfile** were bassing by on the screen more rapidly than you could read it, you would type **CONTROL-S** to freeze the output for a moment.

Typing **CONTROL-Q** would allow the output to resume its rapid pace.

The "(or any other character, for that matter)" is removed, even though **ixany** is still the default (*ibid.* para. 4, pp. 5-6).

And the only extant archived document that should have `getty(1M)` instead defers to D-NIX-specific documentation — *AT&T System V.3 Administrator's Reference Manual*, 089-9717-00, Version A, http://bitsavers.org/pdf/diab/DS90/AT_T-System-V.3-administrators-reference-manual.pdf, p. 264, Diab Data AB, S-183 02, Täby, Sweden, 89-11-01, Manual references, G.

`termio(7)` (*ibid.*, pp. 229-236) finally describes what it means by "flushing" a buffer by expounding

When the input limit is reached, the buffer is flushed and all the saved characters are thrown away without notice.

and notes `EOL2` in the summary as "is another additional line delimiter."; the allocation table remains unchanged. New `c_cflags` are listed below.

<code>c_cflag</code>	Pos'd before	Description
B19200	EXTA	19200 baud
B38400	EXTB	38400 baud

RCV1EN
XMT1EN LOBLK

The pre-LOBLK insertions also move the values: LOBLK was **0010000**, now that's RCV1EN, and they continue to be shifted left by one as-if normal.

These new flags are not described (in the manual or the comment) or indeed used anywhere on the system, so the only side-effect of this is that **stty**, the only user-space LOBLK user, needs to be rebuilt — thanks as ever to my good friend Juliano for lending me the dumps: UNIX SYSVr3 Source Code, <https://vetusware.com/download/UNIX%20SYSVr3%20Source%20Code%20r3/?id=11572,32/usr/src/uts/3b2/sys/termio.h>.

The system-default speed is set to **1200** baud on the U3B16 (*ibid.*).

In AT&T System V Release 3.2 UNIX specifically (cf. `31/usr/src/uts/3b2/sys/termio.h`), `TIOC[GS]WINSZ` (*ibid.*) and the corresponding `struct winsize` appear, all as present-day (*ibid.*, `32/usr/src/uts/3b2/sys/ptem.h`; dated 1984, nominally predating **4.3 Berkeley Software Distribution** by two years).

`stty(1)` (*UNIX® User's Reference manual, 1987, 1986*, pp. 257-259) starts with explaining the character format in the output (and failing to escape the accents into proper carets and backticks, so, especially in double quotes, they're absolute hash; this is fixed below), most notably:

For example, normally `stty -a` will report that the value of `swtch` is `"^"`; however, if `shl (1)` or `layers (1)` has been invoked, `stty -a` will have the value `"^z"`.

(Where `layers(1)` (*ibid.*, pp. 148-149) is "layer multiplexor for windowing terminals", out of scope, impossible to characterise without the `xt(7)` (which may have "channel groups", likely some sort of generalisation of `sxt(7)`, backing `shl(1)`) manual; layers are "manipulated in a terminal-dependent manner" and "Each layer is in most ways functionally identical to a separate terminal.", so you could think of it like the back of an `xterm(1)`, whose front (render, display, windowing, &c.) is provided by the terminal hardware.)

Control Modes `exta extb` are replaced with **19200 38400**.

"DATA-PHONE@" in "**hupcl (-hupcl)**" description is replaced with "Dataphone".

The descriptions for the three "**st***" **Local Modes** all gain "(Does not apply to the 3B2.)".

Indeed, not even compatibility aliases remain, but secret "**19.2**" and "**38.4**" aliases appear (*UNIX SYSVr3 Source Code*, `301/usr/src/cmd/stty/stty.c`). The output speed table is updated appropriately to not show the newly-numeric speeds as **0**.

Additionally, `sane` doesn't reset anymore `swtch`, accompanied by a `/* SWTCH purposely not set */` comment. Maybe it would've been a better idea to just make the default... sane? And not have "`stty sane`" effectively turn off job control? We may never know.

X/OPEN Portability Guide (July 1985)

`termio(7)`, as found in the *System V Interface Definition, Issue 1 (Spring 1985)* is optional, *X/OPEN Portability Guide, The X/OPEN Group Members*, ISBN: 0 444 87839 4, Part II The X/OPEN System V Specification, http://bitsavers.org/pdf/xOpen/X_Open_Portability_Guide_1985, Chapter 1 Interface Definition, 1.2 Status of Interfaces, 1.2.2 Optional, The Netherlands, July 1985, `xpg_2_xopen_system_v_specification_1.pdf`.

Because "Some X/OPEN systems may not support any asynchronous lines, or may only support them over networks." (*ibid.*, 1.10 Caveats, 1.10.2 Termio(7)). Provisions are made for awful V.24/RS232C implementations with no modem control — the carrier is to be assumed always-present.

Ibid., `xpg_2_xopen_system_v_specification_3.pdf`, Chapter 7 Special Files, `console(7)` (p. `CONSOLE(7).1`) — "System console interface" — and `tty(7)` (p. `TTY(7)`) are included verbatim, "except that the SVID states that the system console works through the `termio` interface. This is not necessarily true of X/OPEN systems."

`termio(7)` (*ibid.*, pp. `TERMIO(7).1-TERMIO(7).11`) — "general terminal interface (**OPTIONAL**)" — is "Identical to the SVID entry" "except for minor changes to the first paragraph", but since that's lost to time, the delta below is presented from just AT&T System V Release 2 UNIX, and likely applies to both documents. Expectedly, the specific process names (like being "opened by `getty`") are generalised to "the system".

Rather than a hard **256** bytes, the current limit becomes "`{MAX_CHAR}` characters", defined in `limits(5)` (*ibid.*, Chapter 5 Header Files, pp. `LIMITS(5).1-LIMITS(5).2`) — "Implementation Specific Constants" — in `<limits.h>` as "max size of character input buffer" with no minimum value.

If the process being STOPped hasn't "made other arrangements", additionally "the abnormal termination routines will be executed", rather than explicitly a `core` core dump.

`SWTCH` is missing from the summary (presumably because it just defers to a specific program?) and the subsequent list of **Special Control Characters** changeable by the user, but the `c_cc` allocation table is unchanged. This has the side-effect of continuing to omit `VEOL2` and the out-of-order allocation and lack of `VSWTCH` default being unexplained in (now) a standards document.

NCC is generalised to

```
/* size of the array
 * c_cc for special control characters */
```

And the unlabelled delay macros gain descriptions in the form of (Carriage-return delay type *N*) (except it's "Backspace-delay" for some reason); NL[01] are instead described as "New-Line character type *N*" with no corresponding description change – the meaning of this is unclear.

EXT[AB] are removed entirely and replaced with B19200, B38400 (notably more extreme than AT&T System V Release 3 UNIX, but at that point no device in common use has supported the "external" speeds for years).

TC[GS]ETA spellings are replaced with TC[GS]ETS, with links to **APPLICATION USAGE**, noting:

TCGETA and *ioctl(2)* commands that are reserved to maintain source code compatibility. Their use is even more system dependent than the *termio* interface and source code that uses these commands may not work correctly on all systems.

This may ring roughly true on some weird derivatives, but doesn't appear close to reasonable for the range of systems described herein so far. Of course, TCSETA[WF] remain unchanged.

A **MIN/TIME Interaction** section is added, expounding the minutiae of how **min time** govern **-icanon** input buffering. It's all as-described and the wording largely survives with editorial changes to IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1 — the prose re-states the in-line "This allows fast bursts of input to be read efficiently while still allowing single character input." "The time value represents tenths of seconds." prescription of purpose as "TIME is a timer of 0.1 second granularity that is used to timeout bursty and short term data transmissions."

A notable mention is also found in *ibid.*, Part VI Source Code Transfer, Chapter 5 Other Techniques (*xpg_6_source_code_transfer.pdf*, p. 5.1) with the following requirement from the "far from easy to use" **uucp**:

The TTY drivers should support XON/XOFF (ASCII DC1/DC3) handshaking and an 8-bit transparent mode.

X/Open Portability Guide (December 1988)

X/Open Portability Guide Issue 2 ("XPG2") doesn't appear to have been archived at all, and of *Issue 3* only *Volume 1* seems to've been, so tracking changes to the teletype interface is, unfortunately, impossible. *Issue 2* is the first one to include utilities, and "Little material in this volume has been updated since Issue 2." — *X/Open Portability Guide, XSI Commands and Utilities, X/Open Company, Ltd.*, Issue 3 ISBN 0-13-685835-X, Volume 1, <https://archive.org/details/xopen-portability-xsi-commands-utilities/page/n5/mode/2up>, p. 1, Abbots House, Abbey Street, Reading, Berkshire, RG1 3BD, United Kingdom, December 1988, Chapter 1 Introduction, 1.1 OVERVIEW, 1.1.2 Important Note.

Notably, however, i.a., "the current descriptions of the utilities are incomplete and lack precision" and "There are many inconsistencies in the use of terminology in this volume", so

The **IEEE P1003.2 Shell and Application Utilities Interface for Computer Operating System Environments** Working Group is currently redefining most of the utilities in this volume. This volume will be aligned with **IEEE Std. 1003.2-19xx** after **IEEE Std. 1003.2-19xx** attains full use status. During this alignment process the problems noted above will be addressed.

Which is a bit of a cop-out from the back-cover "X/Open is not a standards-setting body."

Ibid., *Utilities*, **stty** (pp. 252-256) is largely just the AT&T System V Release 2 UNIX one — with the first *termios(7)* reference appropriately updated to *Volume 2* and the rest removed — and shaded UN ("Possibly unsupported feature."). No changes from *Issue 2* are noted.

"0" is generalised to

hang up line immediately. This applies to all terminal lines, not just modem lines. A SIGHUP signal is sent to all processes attached to the line.

The list of speeds becomes a "number" mode, described as

set terminal baud rate to the *number* given, if possible. (Not all speeds are supported by all hardware interfaces.)

The **"hupcl (-hupcl)"** "DATA-PHONE@" is debranded to "modem".
The stray "n" in **"clocal (-clocal)"** is pruned.

"loblk (-loblk)" is caveatted on

The *loblk* option is supported on systems that support *shl*. This option may not be supported by all hardware interfaces.

The odd **"brkint (-brkint)"** "INTR" signalling subject becomes a reasonable "SIGINT".

"ixon (-ixon)" loses the description of how its "START/STOP output control" actually happens, and **"ix-any (-ixany)"** loses the definition for the counterfactual.

All "NL"s and "CR"s are expanded to "newline"s and "carriage-return"s; for **"onlret (-onlret)"** "on the terminal NL performs" becomes "the terminal newline key performs"; "carriage returns" in **"cr0 cr1 cr2 cr3"** are matched to this new punctuation.

The *VSWTCH* erasure continues in its elision from **"isig (-isig)"**, **"noflsh (-noflsh)"**, and **"raw (-raw or cooked)"**.

The "obsolete" **"lfkc (-lfkc)"** is removed and the **"st*"** modes naturally do not fit into the standard.

"swtch" in the **Control Assignments** is shaded OP ("Dependent on optional service in XSI."), and the line discipline range is rephrased as "[0, 127]".

"raw (-raw or cooked)" additionally says it means "and no parity". It's unclear what this could mean.

The **"tabs (-tabs or tab3)"** becomes **"tabs (-tabs or tab8)"** it's wildly unclear what that could *possibly* mean, except maybe that the editor needs new glasses?

"ek" resets "back to the system defaults" rather than any explicit values.

The **"term"** combo group is gone.

The **APPLICATION USAGE** section notes that "Typical implementations of this utility require a communications line configured to use the *termios* interface" (and, if none exist, you may not ship **stty**) — this (or *Issue 2*, naturally) is the first occurrence of "termios", at least nominally distinct from any existing interface.

IEEE Standard Portable Operating System Interface for Computer Environments, Approved August 22, 1988

This isn't strictly spelled out in *Volume 1*, but "XSI Issue 3 is fully compliant with IEEE Std 1003.1-1988 (POSIX.1)." — *XPG3-XPG4 Base Migration Guide, Version 2, X/Open Company Limited*, ISBN: 1-85912-156-X X/Open Document Number: G501, <https://pubs.opengroup.org/onlinepubs/9294999899/toc.pdf>, p. 6, Apex Plaza, Forbury Road, Reading, Berkshire, RG1 1AX, United Kingdom, December 1995, Part 1: General Information, Chapter 1 Introduction, 1.3 Conformance to Standards.

The deltas listed for the interfaces agree (*ibid.*, Part 3: System Interfaces and Headers Migration, Chapter 7 System Interfaces, all functions **cf*()** and **tc*()**, pp. 127-128, 204-207) (*ibid.*, *Chapter*, Chapter 8 Headers, 8.3 Header Migration Information, **<termios.h>**, p. 235), so this is a reasonable proxy (\pm X/OPEN bugs) to the interface of *Issue 3* (and *Issue 2*, presumably, it's hinted it used a draft ("IEEE P1003.1 Trial Use Standard (April 1986)" — *Issue 3 Volume 1, Referenced Documents*, p. xii)).

"{_POSIX_VDISABLE}" is defined as

Terminal special characters defined in **<termios.h>** §7.1.2 can be disabled using this character value, if it is defined. See *tcgetattr()* and *tcsetattr()* §7.2.1.

which is as present-day — *IEEE Standard Portable Operating System Interface for Computer Environments 1003.1 POSIX*, SH12211 IEEE Std 1003.1-1988 FIPS PUB 151-1 ISBN 1-55937-003-3, <https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub151-1.pdf>, p. 48, The Institute of Electrical and Electronics Engineers, Inc, 345 East 47th Street, New York, NY 10017, USA, August 22, 1988, **Table 2-9**. Execution-Time Symbolic Constants.

And "{_POSIX_JOB_CONTROL}" as "If this symbol is defined, it indicates that the implementation supports job control." *ibid.*, **Table 2-8**. Compile-Time Symbolic Constants — this is echoed through **setpgid()** (*ibid.*, 4. Process Environment, 4.3 Process Groups., 4.3.3 Set Process Group ID for Job Control.), **tcgetpgrp()** (*ibid.*, 7. Device- and Class-Specific Functions, **7.2 General Terminal Interface Control Functions.**, 7.2.3 Get Foreground Process Group ID., p. 139), and **tcsetpgrp()** (*ibid.*, 7.2.4 Set Foreground Process Group ID.), all of which are allowed to ENOSYS out if job control is not im-

plemented.

Job control is defined succinctly (*ibid.*, 2, **2.3 General Terms., job control.**, p. 31) as

A facility that allows users to selectively stop (suspend) the execution of *processes* and continue (resume) their execution at a later point. The user typically employs this facility via the interactive interface jointly supplied by the terminal I/O driver and a command interpreter. *Conforming implementations* may optionally support *job control* facilities; the presence of this option is indicated to the application at compile time or run time by the definition of the `{_POSIX_JOB_CONTROL}` symbol; see **Symbolic Constants** §2.10).

This agrees with *Seventh Edition, Virtual VAX-11 Version, November, 1980* more so than AT&T System V Release 2 UNIX, and this is confirmed in *IEEE Std 1003.1-1988*, Appendix B. Rationale and Notes, B.2 Definitions and General Requirements., **B.2.3 General Terms., job control.** (pp. 197-201; yes, four and a half pages), Implementing Job Control Systems, para. 1 (p. 200):

The intent in adding 4.2BSD-style *job control* functionality was to adopt the necessary 4.2BSD programmatic interface with only minimal changes to resolve syntactic or semantic conflicts with System V or to close recognized security holes.

Indeed, it's much more useful to look at the interface through the lens of the rationale — *ibid.*, **B.7 Device- and Class-Specific functions.** (pp. 258-265) Appendix B. Rationale and Notes, **B.7 Device- and Class-Specific functions.** — "After considerable debate, a structure similar to, but more flexible than, the AT&T System III *termio* was agreed upon. The format of that structure, referred to as the *termios* structure, has formed the basis for the current section.", and brings with it in particular new `tc*`-group functions which universally replace the traditional `ioctl()` agglomeration: `tcsetattr()` for `TCSETA` (and the rest, but unnoted; see below) and `tcgetattr()` for `TIOCGSPEED` (*ibid.*, **B.7.2 General Terminal Interface Control Functions., B.7.2.3 Get Foreground Process Group ID. and B.7.2.4 Set Foreground Process Group ID.**, p. 265; "except for the additional security restriction that the referenced terminal must be the controlling terminal for the calling process" for both and "the specified new process group must be currently in use in the caller's session" for the latter), since they allow for full (any) type-checking, and for "networked implementations" it's impossible to pass arbitrary `ioctl()`s through, since "only the device driver knows for sure" in which direction data flows (*ibid.*, B.7, paras. 9-15, p. 259).

The standard specifies that if an attempt to set the input baud rate to zero is made by `cfsetispeed()`, the input baud rate will be instead set to the output baud rate by `cfsetospeed()`. This allows implementations to provide support for split baud rates or not.

which is a bit of a cop-out (and, frankly, frivolous, since no-one has ever used them) in conjunction with

In historical implementations, the baud rate information is traditionally kept in `c_cflag`. Applications should be written to presume that this might be the case (and thus not blindly copy `c_cflag`) but not to rely on it, in case it is in some other field of the structure. Setting the `c_cflag` field absolutely after setting a baud rate is a bad idea because of this.

(*ibid.*, **B.7.1 General Terminal Interface., B.7.1.2 Settable Parameters., B.7.1.2.7 Baud Rate Functions.**, p. 264); note also that "traditionally" starts at AT&T System III UNIX, which'd successfully shed the "just put device registers" day-one legacy, which is *re*-introduced, if anything.

Ibid., **B.7.1.2.6 Special Control Characters.** (!, p. 263) concedes that "Permitting VMIN and VTIME to overlap with VEOF and VEOL was a compromise", and that "Only when backwards compatibility of object code is a serious concern to an implementor should an implementation continue this practice.". Indeed, why AT&T System III UNIX did it, besides saving two bytes, is a mystery.

Similarly,

The alternate behavior possible when ECHOK or ECHOE are specified with ICANON is permitted as a compromise depending on what the actual hardware can do. Erasing characters and lines is preferred, but is not always possible.

(*ibid.*, **B.7.1.2.5 Local Modes.**), and

Although a more optimal structure for implementations may be possible, the degree of change to applications would be significantly larger.

(*ibid.*, **B.7.1.2.1 termios Structure.**, p. 262).

The removal of all *c_oflag* values sans `OPOST` (ibid., 7, **7.1 General Terminal Interface.**, 7.1.2 Settable Parameters., **7.1.2.3 Output Modes.**, p. 129) is rationalised as "There is nothing that a portable application should do to its output for a terminal because that would require knowledge of the operation of the terminal. It is the responsibility of the operating system to provide postprocessing appropriate to the output device", and "Extensions to the standard to control the type of postprocessing already exist, and are expected to continue into the future. The control of these features is primarily to adjust the interface between the system and the terminal device so the output appears on the display correctly. This should be set up before use by any application." (ibid., Appendix B, B.7, B.7.1, B.7.1.2, **B.7.1.2.3 Output Modes.**, p. 263).

`VERASE` is explicitly intended to erase by character, with proper handling of multibyte characters; at time of writing (appx. 34 years after this publication) a survey of common implementations reveals that this holds for Linux (if `utf8` and input is UTF-8) and *almost* the illumos gate (which nevertheless fails). Analogously, *t_werasc*, spelled "4.3BSD"'s "WERASE character",

has not been specified in the standard because it is difficult to define in the international environment. It is only useful for languages where words are delimited by blanks. In some ideographic languages, such as Japanese and Chinese, words are not delimited at all. The WERASE character should presumably take one back to the beginning of a sentence in those cases: practically, this means it would not get much use for those languages.

(ibid., B.7.1.1 Interface Characteristics., **B.7.1.1.6 Canonical Mode Input Processing.**, p. 262).

Minutiae realign 4.2BSD job control edge cases (orphan processes `EIO`ing instead of being killed) with present-day semantics (ibid., **B.7.1.1.4 Terminal Access Control.**, p. 261, cf. ibid., 6. Input and Output Primitives, 6.4 Input and Output., 6.4.2 Write to a File., **6.4.2.4 Errors.**, p. 115).

While "normal" controlling teletype allocation (`getty`) is outside the scope of the standard, `open()` isn't, so in another compromise;

The Working Group did not wish to require the traditional behavior, because it is not very straightforward or flexible for either implementations or applications. However, because of its prevalence, it was not practical to disallow this behavior either.

So `O_NOCTTY` is created "to allow applications to avoid unintentionally acquiring a controlling terminal as a side-effect of opening a terminal file.", as present-day — ibid., Appendix B, **B.5 Files and Directories.**, **B.5.3 General File Creation.**, **B.5.3.1 Open a File.**, para. 7, p. 244; ibid., B.7, B.7.1, B.7.1.1, **B.7.1.1.3 The Controlling Terminal.**, pp. 260-261.

"The issue of modem control was excluded from POSIX" because Europeans didn't care (ibid., B.7, paras. 16-19, p. 260). It's unclear what this refers to.

It definitely shows why "This section has probably undergone more debate and revision than any other in the standard." (ibid., B.7, p. 258)!

With this in mind, it is now prudent to look at the re-defined teletype interface, unsullied by hard compatibility. Chapter 7 — ibid., pp. 123-140 — is indeed most similar to *Release 3.0 tty(4)*, in that it also describes the controlling teletype and doesn't contain `switch`. Of course, the new `O_NONBLOCK` flag is explicitly noted as equivalent to `CLOCAL` for `open()` (*IEEE Std 1003.1-1988*, 7, 7.1, 7.1.1. Interface Characteristics., **7.1.1.1 Opening a Terminal Device File.** para. 2, p. 123), "If the implementation supports job control (if `{_POSIX_JOB_CONTROL}` is defined", "A terminal may have a foreground process group associated with it" which is what plays the "special role" for signal-generating characters instead of the "control terminal", and `tc[gs]etpgrp()` and `setsid()`, rather than `[gs]etpgrp()`, are meaningful in controlling it (ibid., **7.1.1.2 Process Groups.**) (and `O_NOCTTY` prevents it allocation; ibid., **7.1.1.3 The Controlling Terminal.**, p. 124).

Conversely, ibid., **7.1.1.4 Terminal Access Control.** inserts the 4.2 *Berkeley Software Distribution* phonotactics governing `SIGTTIN/SIGTTOU` and `TOSTOP` for background process groups doing I/O or teletype mode control.

"`{MAX_CHAR}` in early drafts" of the standard is re-spelled as "`{MAX_INPUT}`" for obvious reasons (ibid., Appendix B, B.2, **B.2.9 Numerical Limits.**, B.2.9.5 Pathname Variable Values), and

The system may impose a limit, `{MAX_INPUT}`, on the number of bytes that may be stored in the input queue. The behavior of the system when the limit is exceeded is implementation-defined. Similarly, for all of the `c_[iocl]flag` and `c_cc` fields, the default values are implementation-defined (ibid., 7, 7.1, 7.1.1, **7.1.1.5 Input Processing and Reading Data.**, p. 125).

"If `{MAX_CANON}` is defined for this terminal device, it is a limit on the number of bytes in a line." with implementation-defined results if overrun (ibid., **7.1.1.6 Canonical Mode Input Processing.**, para. 2). The only new thing about these limits is nomenclature — the Minimum Value (prefixed with `_POSIX_`) is **255** for both (notably, **1** less than heretofore documented; ibid., **Table 2-5. Pathname Variable Values**, p. 46; ibid., **Table 2-2. Minimum Values**, p. 45).

For `-icanon`, when setting `min > {MAX_INPUT}`, "the response to the request" is explicitly implementation-defined (even though it falls out of the usual definition for this limit; ibid., Appendix B, 7, 7.1, 7.1.1, **7.1.1.7 Non-Canonical Mode Input Processing.**, p. 126), and the `min time` interactions are editorialised to twice the volume, but remain at no functional change (ibid., **7.1.1.7.1-7.1.1.7.4**, pp. 126-127).

Ibid., **7.1.1.9 Special Characters.** (pp. 127-128) are, naturally, editorialised, with the targets for `intr quit` now being the same as for `susp` (below), and see the addition of:

SUSP If job control is supported (see **Special Control Characters** §7.1.2.6), the SUSP special character is recognized on input. If the ISIG flag is enabled, receipt of the SUSP character causes a SIGTSTP signal to be sent to all processes in the foreground process group for which the terminal is the controlling terminal, and the SUSP character is discarded when processed.

CR Special character on input and is recognized if the ICANON flag is set; it is the `'\r'`. When ICANON and ICRNL are set and IGNCR is not set, this character is translated into a NL, and has the same effect as a NL character.

"It is implementation-defined whether the START and STOP characters can be changed." — this covers both \geq *Release 3.0* and \geq *Seventh Edition*; but "The values for INTR, QUIT, ERASE, KILL, EOF, EOL, and SUSP (job control only), shall be changeable to suit individual tastes." "If `{_POSIX_VDISABLE}` is in effect for the terminal file" (`[f]pathconf(_PC_VDISABLE)` isn't `-1`: *IEEE Std 1003.1-1988*, **Table 5-2. Configurable Pathname Variables**, p. 105), "special character functions associated with changeable special control characters can be disabled individually", as present-day; if any two overlap, it's "undefined" what happens. Provisions are made for context-based disambiguation, like: "an implementation may define multibyte sequences that have a meaning different from the meaning of the bytes when considered individually." if `ixten`; the example is a red herring — this provision is actually for `\`-escaping `erase kill eof`, allowing for preservation of historical behaviour.

Ibid., 7, 7.1, 7.1.1, **7.1.1.10 Modem Disconnect.** (p. 128) concretifies hangup \wedge `-local` \Rightarrow `SIGHUP`, retains subsequent `read()`s returning empty, and defines subsequent `write()`s to `EIO` — this appears to be new behaviour.

HUPCL is defined to take place after committing all output and flushing input, rather than instantly: this is the same behaviour as *Release 3.0*, but undocumented (*IEEE Std 1003.1-1988*, 7, 7.1, 7.1.1, **7.1.1.11 Closing a Terminal Device File.**).

Ibid., 7.1.2, 7.1.2.1-7.1.2.6, incl. **Table 7-1-Table 7-5** (pp. 129-134) correspond to the bulk of the remainder of `tty(4)`, and are reproduced faithfully below.

Table 7-1. *termios* Structure

Member Type	Array Size	Member Name	Description
<code>icflag_t</code>		<code>c_iflag</code>	Input modes
<code>icflag_t</code>		<code>c_oflag</code>	Output modes
<code>icflag_t</code>		<code>c_cflag</code>	Control modes
<code>icflag_t</code>		<code>c_lflag</code>	Local modes
<code>cc_t</code>	NCCS	<code>c_cc</code>	Control characters

Where both types are "unsigned integral types" (as present-day, with the missing requirement of being smaller than a `long`), and the structure not being limited to these members.

Table 7-2. *termios* *c_iflag* Field

Mask Name	Description
BRKINT	Signal interrupt on break.
ICRNL	Map CR to NL on input.
IGNBRK	Ignore break condition.
IGNCR	Ignore CR.
IGNPAR	Ignore characters with parity errors.
INLCR	Map NL to CR on input.
INPCK	Enable input parity check.
ISTRIP	Strip character.
IXOFF	Enable start/stop input control.
IXON	Enable start/stop output control.
PARMRK	Mark parity errors.

Sans the removal of `IUCLC` (since characters are possibly multi-byte) and `IXANY` (unnoted, likely just doesn't really matter) the table is unchanged.

Long break conditions are explicitly amalgamated (cf. *ibid.*, Appendix B, B.7, B.7.1, B.7.1.2, **B.7.1.2.2 Input Modes.**, p. 263), and, if `-ignbrk brkint`, clear I/O queues and `SIGINT` the foreground process group (the former is unnoted in *Release 3.0*, nor is the target of the latter).

`VSTART` and `VSTOP` are only allowed to be discarded from the input if `ixon` (rather than always). The wording doesn't change, other than by omission, so `ixon` is allowed to function in either `[-]ixany` mode.

`IXOFF` is described more verbosely and explicitly mentions `{MAX_INPUT}`, but "The precise conditions under which `STOP` and `START` characters are transmitted are implementation-defined."

7.1.2.3. Output Modes.

Mask Name	Description
OPOST	Perform output processing.

"If `OPOST` is set, output data is processed in an implementation-defined fashion so that lines of text are modified to appear appropriately on the terminal device, otherwise characters are transmitted without change.", see above.

Table 7-3. *termios* *c_cflag* Field

Mask Name	Description
CLOCAL	Ignore modem status lines.
CREAD	Enable receiver.
CSIZE	Number of bits per byte*:
CS5	5 bits
CS6	6 bits
CS7	7 bits
CS8	8 bits
CSTOPB	Send two stop bits, else one.
HUPCL	Hang up on last close.
PARENB	Parity enable.
PARODD	Odd parity, else even.

* `CSIZE` has historically described "character" size.

See above for the reasoning for moving speeds, or below for the new list.

Oddly, `CREAD` is not rephrased in terms of bytes.

If `hupcl`, "The modem connection shall be broken." when the last file descriptor is closed.

Despite the age of this publication, provisions are made:

If the object for which the control modes are set is not an asynchronous serial connection, some of the modes may be ignored; for example, if an attempt is made to set the baud rate on a network connection to a terminal on another host, the baud rate may or may not be set on the connection between that terminal and the machine it is directly connected to.

Table 7-4. *termios c_lflag* Field

Mask Name	Description
ECHO	Enable echo.
ECHOE	Echo ERASE as an error-correcting backspace.
ECHOK	Echo KILL.
ECHONL	Echo '\n'.
ICANON	Canonical input (erase and kill processing).
IEXTEN	Enable extended (implementation-defined) functions.
ISIG	Enable signals.
NOFLSH	Disable flush after interrupt, quit, or suspend.
TOSTOP	Send SIGTTOU for background output.

XCASE is replaced with IEXTEN; ISIG and NOFLSH expectedly mention "SUSP (job control only)". The TOSTOP description matches the *Seventh Edition, Virtual VAX-11 Version, November, 1980* LTOSTOP one verbatim.

IEXTEN is fully defined as

If IEXTEN is set, implementation-defined functions shall be recognized from the input data. It is implementation-defined how IEXTEN being set interacts with ICANON, ISIG, IXON, or IXOFF. If IEXTEN is not set, then implementation-defined functions shall not be recognized and the corresponding input characters shall be processed for ICANON, ISIG, IXON, and IXOFF.

The description for TOSTOP derives heavily from the *Seventh Edition, Virtual VAX-11 Version, November, 1980* one as well (though, naturally, process groups aren't "distinguished" any-more).

ECHO is not rephrased in terms of bytes.

ECHOE

shall cause the terminal to erase the last character in the current line from the display, if possible. If there is no character to erase, an implementation may echo an indication that this was the case or do nothing.

if **icanon** — this allows for the *Release 3.0* behaviour of **icanon echo echok** yielding "\b \b" and **icanon -echo echok** " \b".

If **icanon**, ECHOK is allowed to either "erase the line from the display" — like *Seventh Edition, Virtual VAX-11 Version, November, 1980* LCRTKIL — or "echo the '\n' character after the KILL character" — like *Release 3.0*.

Table 7-6. *termios c_cc* Special Control Characters

Subscript Usage		Description
Canonical Mode	Non-Canonical Mode	
VEOF		EOF character
VEOL		EOL character
VERASE		ERASE character
VINTR	VINTR	INTR character
VKILL		KILL character
	VMIN	MIN value
VQUIT	VQUIT	QUIT character
VSUSP	VSUSP	SUSP character
	VTIME	TIME value
VSTART	VSTART	START character

VSTOP VSTOP STOP character

"Implementations that do not support job control may ignore the SUSP character value in the `c_cc` array indexed by the VSUSP subscript.", and similarly for **start stop**, but for them they "shall return the value in use when `tcgetattr` is called."

Even if disabling is supported and enabled, "If ICANON is not set, the value of `{_POSIX_VDISABLE}` has no special meaning for the VMIN and VTIME entries of the `c_cc` array."

Speed control is moved to the appropriately-named *IEEE Std 1003.1-1988*, 7, 7.1, 7.1.2, **7.1.2.7 Baud Rate Functions**. (pp. 134-135), with `cf[gs]et[io]speed()` — these either take or return `speed_t` (defined the same way as `tcflag_t`). The only surprising part is that `cfsetispeed(B0)` sets the input speed to be the same as the output speed, and only `cfsetospeed(B0)` actually hangs up. The pre-defined speeds follow.

Table 7-6. *termios* Baud Rate Values

Name	Description	Name	Description
B0	Hang up	B600	600 baud
B50	50 baud	B1200	1200 baud
B75	75 baud	B1800	1800 baud
B110	110 baud	B2400	2400 baud
B134	134.5 baud	B4800	4800 baud
B150	150 baud	B9600	9600 baud
B200	200 baud	B19200	19 200 baud
B300	300 baud	B38400	38 400 baud

Ibid., 7.2, **7.2.1 Get and Set State**. (pp. 136-137) provides `tcgetattr()` as an equivalent for `ioctl(TCGETA)` and

`tcsetattr(TCSANOW)` for `ioctl(TCSETA)`,
`tcsetattr(TCSADRAIN)` for `ioctl(TCSETAW)`, and
`tcsetattr(TCSAFLUSH)` for `ioctl(TCSETAF)`.

Descriptions for the former decidedly mirror those for the latter.

UNIX® System V Release 4

"login id" becomes "login ID"; "common terminal speeds" are now "300, 1200, 2400, 4800, 9600, 19200, and 38400 baud" — *UNIX® System V Release 4 User's Reference Manual*, ISBN 0-13-947037-9, http://bitsavers.org/pdf/att/unix/System_V_Release_4/

0-13-947037-9_Unix_System_V_Rel4_Users_Reference_Manual_1990.pdf, para. 1, p. 4, AT&T: UNIX Software Operation, 1990, Introduction, How to Get Started, Logging In.

The special-character spelling is updated to the more reasonable "CTRL-s" (ibid., How to Communicate Through Your Terminal, para. 4, p. 6).

Unfortunately, "The shell layer manager continues to be supported in Release 4.0.", but "Release 4.0 offers an optional shell (`jsch`) that features a new implementation of job control that conforms to the POSIX standard." (of course, `csch` and the KornShell are also available, but the former is far cry from normal-`sh`-but-with-job-control) — *UNIX® System V Release 4 Migration Guide*, ISBN 0-13-933821-7, http://bitsavers.org/pdf/att/unix/System_V_Release_4/

0-13-933821-7_Unix_System_V_Rel4_Migration_Guide_1990.pdf, p. 3-3, AT&T: UNIX Software Operation, 1990, 3. Migrating from Release 2, Release 2.0 Features, Shell Enhancements (Job Control).

Worryingly, however,

In Release 4.0, the entire terminal (tty) subsystem in the kernel has been rewritten to use the STREAMS mechanism. The move to a STREAMS implementation increases the modularity of the tty subsystem and provides a more flexible framework for future enhancements. Pseudo-ttys (ptys) have been implemented under STREAMS as well.

And the same happens to pipes. (Ibid., 5. UNIX System V Release 4, Input/Output, STREAMS, p. 5-21)

The WERASE character (the character control-W) erases the last "word" typed in the current output line (but not any preceding spaces or tabs). A "word" is defined as a sequence of non-blank characters, with tabs counted as blanks.

This is the *exact* wording in *ibid.*, §B.7.1.1.6 (p. 262). Similarly,

The REPRINT character (the character control-R) prints a newline followed by all characters that have not been read. Reprinting also occurs automatically if characters that would normally be erased from the screen are fouled by program output. The characters are reprinted as if they were being echoed; consequencely, if ECHO is not set, they are not printed.

Being "fouled by" brands the first two sentences as good but nevertheless obvious adaptation of the *Seventh Edition, Virtual VAX-11 Version, November, 1980* description. Being predicated on **echo** is undocumented there, but matches the actual behaviour (*ibid.*, `usr/src/sys/dev/ttynew.c`, `ntyretype()`, `ntyecho()`).

In the **Special Control Characters** summary, **intr quit** spell it "a SIGQUIT signal"/"a SIGINT signal" instead of "a *quit* signal"/"an *interrupt* signal", but target "all frequent processes associated with the controlling terminal" (i.e., only "frequent" is added, and it's unclear how that makes it be job-controlly). **erase kill** finally correctly list **eol2** as a character they won't erase past.

The unchanged-since-*Release 3.0* **eof** trimming paragraph is moved up and extended with **echoctl** as

The EOF character is not echoed unless it is escaped or ECHOCTL is set. Because EOT is the default EOF character, this prevents terminals that respond to EOT from hanging up.

The "The start/stop characters can not be changed or escaped." sentence is lost. **swtch** is the only remnant of layer-based "job control"; its new description and the new synopses follow.

- WERASE (CTRL-W or ASCII ETX) erases the preceding "word". It does not erase beyond the start of a line, as delimited by a NL, EOF, EOL, or EOL2 character.
- REPRINT (CTRL-R or ASCII DC2) reprints all characters, preceded by a newline, that have not been read.
- SWTCH (CTRL-Z or ASCII EM) is used only when `sh1` layers is invoked.
- SUSP (CTRL-Z or ASCII SUB) generates a SIGTSTP signal. SIGTSTP stops all processes in the foreground process group for that terminal.
- DSUSP (CTRL-Y or ASCII EM) generates a SIGTSTP signal as SUSP does, but the signal is sent when a process in the foreground process group attempts to read the DSUSP character, rather than when it is typed.
- DISCARD (CTRL-O or ASCII SI) causes subsequent output to be discarded. Output is discarded until another DISCARD character is typed, more input arrives, or the condition is cleared by a program.
- LNEXT (CTRL-V or ASCII SYN) causes the special meaning of the next character to be ignored. This works for all the special characters mentioned above. It allows characters to be input that would otherwise be interpreted by the system (for example, KILL, QUIT).

The most obvious idiocy here is that `^Z` is SUB (that's even internally inconsistent!), and the CNSWTCH/CSWTCH dichotomy is still there and it's *still* initialised to CNSWTCH (*UNIX SYSVr4 Source Code*, `ATT-SYSVr4/uts/i386/sys/termios.h`, `ATT-SYSVr4/uts/i386/io/ldterm.c`).

These new **Special Control Characters** are all the same functionally and phonotactically as in 4.2 *Berkeley Software Distribution*, sans the "DISCARD" `t_flushc` spelling.

`_POSIX_VDISABLE` is **0**, so **eol eol2 swtch** are all disabled by default (this also means that the default **eol** description continues to be wrong, but for a different reason this time, but oh well).

`\`-escaping **erase kill eof** is mentioned again here and doesn't appear to be predicated on IEXTEN; this is a very clear violation of the standard.

`struct termios` consists only of the POSIX-prescribed members, with `tcflag_t` and `speed_t` being *unsigned longs* and `cc_t` an *unsigned char*; the `-icanon c_cc` listing is the same as for `struct termio`, but for **icanon**, expectedly, it grows to:

0	VINTR	DEL
1	VQUIT	FS

2	VERASE	#
3	VKILL	@
4	VEOF	EOT
5	VEOL	NUL
6	VEOL2	NUL
7	VSWTCH	NUL
8	VSTRT	DC1
9	VSTOP	DC3
10	VSUSP	SUB
11	VDSUSP	EM
12	VREPRINT	DC2
13	VDISCRD	SI
14	VWERASE	ETB
15	VLNEXT	SYN
16-19	reserved	

The "VSTRT" is wacky: it's a typo; it's correctly `VSTART` in `<sys/termios.h>`. "VDISCRD" is equally wacky and equally just the editor's sticky keyboard. Conversely, the `VSWTCH` default is magically correct here.

Input Modes see the subsumption of all descriptions with their *IEEE Std 1003.1-1988* equivalents, if there is one (i.e. not for `IUCLC`, `IXANY` and additionally not for `IX*`), though sometimes very poorly copied (as in "If `INPCK` is not set"), and with "bytes" reverted to "characters", plus the addition of

`IMAXBEL` Echo BEL on input line too long.
described as

If `IMAXBEL` is set, the ASCII BEL character is echoed if the input stream overflows. Further input is not stored, but any input already present in the input stream is not disturbed. If `IMAXBEL` is not set, no BEL character is echoed, and all input present in the input queue is discarded if the input stream overflows.

Which is spelled so obnoxiously as to sound a like standardese, but isn't; `-imaxbel` corresponds to the classic behaviour, and `imaxbel` – to that of *Seventh Edition, Virtual VAX-11 Version, November, 1980's* `NTTYDISC`.

"The initial input control value is `BRKINT`, `ICRNL`, `IXON`, `ISTRIP`."

Output Modes see the addition of *Seventh Edition*-style `XTABS` `TAB3` alias.

"The initial output control value is `OPOST`, `ONLCR`, `TAB3`."

The mysterious `RCV1EN` `XMT1EN` and the all-too-well-described `LOBLK` **Control Modes** are removed in favour of

`CIBAUD` Input baud rate, if different from output rate
`PAREXT` Extended parity for mark and space parity

The former is exquisitely-adapted for the specific *IEEE Std 1003.1-1988* split-speed interface:

If the `CIBAUD` bits are not zero, they specify the input baud rate, with the `CBAUD` bits specifying the output baud rate; otherwise, the output and input baud rates are both specified by the `CBAUD` bits. The values for the `CIBAUD` bits are the same as the values for the `CBAUD` bits, shifted left `IBSHIFT` bits.

The latter is undescribed and doesn't appear to be used at all outside the userland.

"The initial hardware control value after open is `B300`, `CS8`, `CREAD`, `HUPCL`."

Local Modes grow

`TOSTOP` Send `SIGTTOU` for background output.
`ECHOCTL` Echo control characters as `^char`, delete as `^?`.
`ECHOPRT` Echo erase character as character erased.
`ECHOKE` BS-SP-BS erase entire line on line kill.
`FLUSHO` Output is being flushed.
`PENDIN` Retype pending input at next read or input character.

IEXTEN Enable extended (implementation-defined) functions.

Of these, **FLUSHO** is the **discard** companion, equally as present-day as **LFLUSHO** was in *Seventh Edition, Virtual VAX-11 Version, November, 1980*. Likewise for **[L]TOSTOP**, which bears the *IEEE Std 1003.1-1988* description.

ECHOCTL, **ECHOPRT**, and **ECHOKE** act as present-day re-spellings of **LCTLECH**, **LPRTERA**, and **LCRTKIL**. Similarly for **[L]PENDIN**, which has no present-day equivalent, and which bears in its only kernel-side usage, in `ldterm.c` (i.e. `ldterm(7)` — "standard STREAMS terminal line discipline module" — *UNIX® System V Release 4 System Files and Devices Reference Manual for Motorola Processors*, pp. 159-161):

```
/*
 * Yuk. The C shell file completion code actually
 * uses this "feature", so we have to support it.
 */
```

ISIG and **ICANON**'s domains grow by the new **Special Control Characters** and **eol2**.

If **IEXTEN** is set, the following implementation-defined functions are enabled: special characters (**WERASE**, **REPRINT**, **DISCARD**, and **LNEXT**) and local flags (**TOSTOP**, **ECHOCTL**, **ECHOPRT**, **ECHOKE**, **FLUSHO**, and **PENDIN**).

a more rigorous analysis may nickel-and-dime the system on the extension gating, or bring up this from `ldterm.c`:

```
/*
 * EOL2/XCASE should be conditioned with IEXTEN to be truly POSIX conformant.
 * This is going to cause problems for pre-SVR4.0 programs that don't
 * know about IEXTEN. Hence EOL2/IEXTEN is not conditioned with IEXTEN.
 */
```

"The initial line-discipline control value is **ISIG**, **ICANON**, **ECHO**, **ECHOK**,"

`struct winsize` is explicitly noted now in the **Terminal Size** section, alongside its `TIOC[GS]WINSZ` `ioctl(s)`s. The `ioctl(s)` taking `struct termios`, to which `termios(2)` decays, all correspond to the `struct termio` ones but with the final **A** substituted for **S**, just like *X/OPEN Portability Guide (July 1985)* fore-told.

A consequence of making the teletype interface layered, `ttcompat(7)` (*ibid.*, pp. 413-418), described as "V7, 4BSD and XENIX STREAMS compatibility module" isn't *as* insane as it sounds; of course, it *is* insane in just how relatively well it manages to map the entirely different and completely unrelated format of *Seventh Edition, Virtual VAX-11 Version, November, 1980* `struct sgttyb/struct ltchars` with sometimes-unmappable modes to something that makes relative sense and provides a sufficient compatibility layer to Just work.

`stty(1)` (*UNIX® System V Release 4 User's Reference Manual*, pp. 432-437) escapes accents properly and removes the `swtch/shl(1)/layers(1)` paragraph. There are two new sets of modes for controlling the serial/modem hardware (**Hardware Flow Control Modes** and **Clock Modes**), which correspond to `termiox(7)` (*UNIX® System V Release 4 System Files and Devices Reference Manual for Motorola Processors*, pp. 396-400) but are otherwise outside the scope of this document, because most of them biblio down to single pins in EIA-232-D.

"**0**" hangs up "line", rather than "phone line"; "**hupcl (-hupcl)**" loses "Dataphone" and just hangs up "connection".

Input mode	Description
parext (-parext)	enable (disable) extended parity generation and detection for mark and space parity.
ispeed 0 110 300 600 1200 1800 2400 4800 9600 19200 38400	← Set terminal input baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the input baud rate is set to zero, the input baud rate will be specified by the value of the output baud rate.

ospeed 0 110 300 600 1200 1800 2400 4800 9600 19200 38400

← Set terminal output baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the output baud rate is set to zero, the line will be hung up immediately.

Input Modes-wise, "**ixon (-ixon)**" sends **start stop** rather than hard DC1/DC3. This courtesy is not extended to "**ixany (-ixany)**" (in description only, **-ixany** does actually accept the current **start**).

Local mode	Description
tostop (-tostop)	← send (do not send) SIGTTOU when background processes write to the terminal.
echoctl (-echoctl)	← echo (do not echo) control characters as <i>^char</i> , delete as <i>^?</i>
echoprt (-echoprt)	← echo (do not echo) erase character as character is "erased".
echoke (-echoke)	← BS-SP-BS erase (do not BS-SP-BS erase) entire line on line kill.
flusho (-flusho)	← output is (is not) being flushed
pendin (-pendin)	← retype (do not retype) pending input at next read or input character.
iexten (-iexten)	← enable (disable) extended (implementation-defined) functions for input data.

"*control character c*" grows all the new characters, except **min time** which are split off, as below. Despite the long index macro, **VREPRINT** is **rprnt**.

Control assignment	Description
min, time number	Set the value of min or time to <i>number</i> . MIN and TIME are used in Non-Canonical mode input processing (-icanon)

Combination mode	Description
spacep	enable parenb , cs7 , and parext .
markp	enable parenb , cs7 , parodd , and parext .
-parity , or -evenp	disable parenb , and set cs8 .
-oddp	disable parenb and parodd , and set cs8 .
-spacep	disable parenb and parext , and set cs8
-markp	disable parenb , parodd , and parext , and set cs8
async	set normal asynchronous communications where clock settings are xcibrg , rcibrg , tsetcoeff , rsetcoeff .

"**-parity**, or **-evenp**" retain the old (present-day) semantics, "**-oddp**" is changed to violate *Issue 3*; this (and spelling **tab3** correctly, and POSIX-conformant **0/hang-up** semantics (cf. **X/Open Portability Guide Issue 4 TODO DOODODODO**)) is the only incompatibility.

There's also a new **Window Size** section, corresponding directly to *struct winsize* members, outlined below. "**rows**" is as present-day; "**columns**" similarly, but survives as **cols** in IEEE Std 1003.1-202x ("POSIX.1"), Draft 2.1.

Window size	Description
rows n	← set window size to <i>n</i> rows.
columns n	← set window size to <i>n</i> columns.
ypixels n	set vertical window size to <i>n</i> pixels.
xpixels n	set horizontal window size to <i>n</i> pixels.

UNIX SYSVr4 Source Code, ATT-SYSVr4/ATT-SYSVr4/cmd/ttymon/sttytable.c reveals that "**spacep**" is additionally **-parodd**, and that "**loblk (-loblk)**" is excluded from the PDP-11 implicitly (by the system not even remotely targeting it), rather than explicitly.

As hinted at, **min time** are parsed as decimal integers if they start with a digit (and like normal **Special Control Characters** otherwise; *ibid.*, ATT-SYSVr4/ATT-SYSVr4/cmd/ttymon/sttyparse.c). The **Window Sizes** are always parsed as decimal integers.

Accepting "**line i**" is predicated on not being attached to a `termio(7)`-only STREAM; conversely, accepting the new modes is predicated on being attached to a `termios(7)`-capable one.

On `termios(7)`-capable STREAMS, if a split speed is set and if the speeds are actually different, they're printed as "`ispeed is baud; ospeed os baud;`" again (*ibid.*, ATT-SYSVr4/ATT-SYSVr4/cmd/ttymon/stty.c).

The default-output **cread** sign mismatch bug is fixed; "**parenb parext parodd**" agglomerates to "**markp**" and "**parenb parext -parodd**" - "**spacep**".

On `TIOC[GS]WINSZ`-capable STREAMS, the window size is written as

```
rows = rows; columns = cols; ypixels = ypixels; xpixels = xpixels;
```

If **-icanon**, **min time** are always written as

```
min = min; time = time;
```

eol2 is listed after **eol** and on `termios(7)`-capable STREAMS, the new characters are written in index order.

All new modes are default-off for purposes of no-argument output. The **-g** mode is, naturally, longer, since `NCCS > NCC`, but all inputs that don't have the exact field count for the `termios(7)`-generated output are trimmed to `termio(7)` length.

A 4.3BSD-like **stty** is also found in `/usr/ucb`. It's been heavily tinkered with, but it retains most, if not all, of its idiosyncracies.

4.4BSD

The biggest area of change is a new terminal driver. The terminal driver is similar to the System V terminal driver with the addition of the necessary extensions to get the functionality previously available in the 4.3BSD terminal driver. 4.4BSD also adds the IEEE Std1003.1 job control interface, which is similar to the 4.3BSD job control interface, but adds a security model that was missing in the 4.3BSD job control implementation. A new system call, *setsid*, creates a job-control session consisting of a single process group with one member, the caller, that becomes a session leader. Only a session leader may acquire a controlling terminal. This is done explicitly via a `TIOCSCTTY ioctl` call, not implicitly by an *open* call. The call fails if the terminal is in use.

With emulation for "both the old *ioctl* calls and old options to *stty*" — *4.4BSD User's Reference Manual (URM)*, ISBN 1-56592-075-9, https://archive.org/details/The_CSRG_Archives_CD-ROM_3_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 3, Final Berkeley Releases, 4.4BSD-Lite2, `usr/src/share/man/man0/title.urm`, Computer Systems Research Group, University of California at Berkeley, April, 1994, Preface, 2. Changes in the Kernel, 2.4. POSIX terminal driver changes.

The typical speeds get bumped to **1200**, **2400**, and **9600** baud (*ibid.*, Introduction, How to Get Started, *Logging in*, para. 2).

"The `^U` (control-U) character", rather than `DEL`, is the default **kill**, and "the delete character (`DEL`) or sometimes the backspace character (control-H)", rather than just the latter, is the default **erase**.

lnext replaces `^V` as the escape for **erase kill**, but the wording stays the same (*ibid.*, *How to communicate through your terminal.*, para. 2).

If, in `ttys(5)`, `console` isn't designated "**secure**", `init(8)` itself will prompt for the root passphrase — *4.4BSD Programmer's Reference Manual (PRM)*, ISBN 1-56592-078-3, https://archive.org/details/The_CSRG_Archives_CD-ROM_3_August_1998_Marshall_Kirk_McKusick Marshall Kirk McKusick, The CSRG Archives, CD-ROM 3, Final Berkeley Releases, 4.4BSD-Lite2, `usr/src/sbin/init/init.8`, Computer Systems Research Group, University of California at Berkeley, April, 1994. `tty(4)` (*ibid.*, `usr/src/share/man/man4/tty.4`) becomes a general overview of the teletype subsystem and `ioctl(1)` listing; the substantive change is that "The available line disciplines are listed in `<sys/termios.h>` and currently are:"

```
TTYDISC    Termios interactive line discipline.
TABLDISC   Tablet line discipline.
```

SLIPDISC Serial IP line discipline.
 which marks the removal of OTTYDISC and relabeling, with the aforementioned updates, of NTTYDISC, and also I hope they recover from the injury.

Of which TABLDISC is described as

```
/*
 * Line discipline for RS232 tablets;
 * supplies binary coordinate data.
 */
```

(*ibid.*, `usr/src/sys/kern/tty_tb.c`, available since **4.3 Berkeley Software Distribution**), where "tablet" is used to mean a digitiser/graphics tablet.

"When the `termios` line discipline is in effect, the terminal file behaves and is operated according to the rules described in `termios(4)`." (*ibid.*, `usr/src/share/man/man4/termios.4`) — "general terminal line discipline" — "describes a general terminal line discipline that is supported on tty asynchronous communication ports." Job control, its edge cases, controlling teletypes, `O_NONBLOCK`, `[-icanon]` processing, base **Special Control Characters**, and "**Modem Disconnect**" are *IEEE Std 1003.1-1988* verbatim, with choices for implementation-defined behaviour noted, if any. The value of `_POSIX_VDISABLE` is not listed, but the former `-1` recommendation is reused (but spelled as `0xFF` since `cc_t` is unsigned; *4.BSD Programmer's Reference Manual (PRM)*, `usr/src/sys/sys/termios.h`).

{`MAX_CANON`} is a limit on the number of bytes in a line. The behavior of the system when this limit is exceeded is the same as when the input queue limit {`MAX_INPUT`}, is exceeded. This appears to match **UNIX® System V Release 4**, but is undocumented there.

"The following special characters are extensions defined by this system and are not a part of 1003.1 `termios`." `eol`'s verbatim inclusion in the base listing essentially re-spells `t_brkc`.

EOL2	Secondary EOL character. Same function as EOL.
WERASE	Special character on input and is recognized if the <code>ICANON</code> flag is set. Erases the last word in the current line according to one of two algorithms. If the <code>ALTWERASE</code> flag is not set, first any preceding whitespace is erased, and then the maximal sequence of non-whitespace characters. If <code>ALTWERASE</code> is set, first any preceding whitespace is erased, and then the maximal sequence of alphabetic/underscores or non alphabetic/underscores. As a special case in this second algorithm, the first previous non-whitespace character is skipped in determining whether the preceding word is a sequence of alphabetic/underscores. This sounds confusing but turns out to be quite practical.
REPRINT	Special character on input and is recognized if the <code>ICANON</code> flag is set. Causes the current input edit line to be retyped.
DSUSP	Has similar actions to the <code>SUSP</code> character, except that the <code>SIGTSTP</code> signal is delivered when one of the processes in the foreground process group issues a <code>read()</code> to the controlling terminal.
LNEXT	Special character on input and is recognized if the <code>IEXTEN</code> flag is set. Receipt of this character causes the next character to be taken literally.
DISCARD	Special character on input and is recognized if the <code>IEXTEN</code> flag is set. Receipt of this character toggles the flushing of terminal output.
STATUS	Special character on input and is recognized if the <code>ICANON</code> flag is set. Receipt of this character causes a <code>SIGINFO</code> signal to be sent to the foreground process group of the terminal. Also, if the <code>NOKERNINFO</code> flag is not set, it causes the kernel to write a status message to the terminal that displays the current load average, the name of the command in the foreground, its process ID, the symbolic wait channel, the number of user and system seconds used, the percentage of cpu the process is getting, and the resident set size of the process.

Of these, `eol2` is new (presumably for compatibility with **UNIX® System V Release 4**), and so is `status` (and the rest of the infrastructure it describes). No note is made about automatic `REPRINT` on fouling.

"Values of the `c_iflag` field describe the basic terminal input control, and are composed of following masks:"

Input mode	Description
IGNBRK	/* ignore BREAK condition */
BRKINT	/* map BREAK to SIGINTR */
IGNPAR	/* ignore (discard) parity errors */
PARMRK	/* mark parity and framing errors */
INPCK	/* enable checking of parity errors */
ISTRIP	/* strip 8th bit off chars */
INLCR	/* map NL into CR */
IGNCR	/* ignore CR */
ICRNL	/* map CR to NL (ala CRMOD) */
IXON	/* enable output flow control */
IXOFF	/* enable input flow control */
IXANY	/* any char will restart after stop */
IMAXBEL	/* ring bell on input queue full */

It's interesting to note that *all* of these descriptions are custom, but the order matches **UNIX® System V Release 4**; besides that, this is the standard set of *IEEE Std 1003.1-1988* modes plus `IXANY` `IMAXBEL`, which are described compatibly with **UNIX® System V Release 4** but worse:

If `IXANY` is also set, then any character may restart output.

If `IMAXBEL` is set and the input queue is full, subsequent input shall cause an ASCII `BEL` character to be transmitted to the the output queue.

(it's also described earlier to actually be the same).

`IXOFF` corresponds to the *Seventh Edition, Virtual VAX-11 Version, November, 1980* `TANDEM` mode, `ICRNL` to the input half of `CRMOD`, and `ISTRIP` is the inverse of `LPASS8`; `IXANY` is the inverse of **4.1 Berkeley distribution** `LDECCTQ`.

The other descriptions are verbatim the POSIX ones, sans changing the tense from "shall" to "does". This extends to the default value and `ixoff` conditions, which are also listed as "implementation defined" (note the loss of punctuation).

"Values of the `c_oflag` field describe the basic terminal output control, and are composed of the following masks:"

Output mode	Description
OPOST	← /* enable following output processing */
ONLCR	← /* map NL to CR-NL (ala CRMOD) */
OXTABS	← /* expand tabs to spaces */
ONOEOT	/* discard EOT's '^D' on output) */

`OPOST` is the inverse of the former `LLITOUT`, `ONLCR` helpfully points out it's the output half of `CRMOD`. and `OXTABS` – to `XTABS`, though on modern-day 4.4BSD derivatives this, if at all, survives as `TAB3` (even if it's the only non-`TAB0` delay mode) instead. `ONOEOT` was the default in cooked mode.

No note is made of the default value.

"Values of the `c_cflag` field describe the basic terminal hardware control, and are composed of the following masks."

Control mode	Description
CSIZE	/* character size mask */
CS5	/* 5 bits (pseudo) */
CS6	/* 6 bits */
CS7	/* 7 bits */
CS8	/* 8 bits */
CSTOPB	/* send 2 stop bits */
CREAD	/* enable receiver */
PARENB	/* parity enable */
PARODD	/* odd parity, else even */

```

HUPCL      /* hang up on last close */
CLOCAL     /* ignore modem status lines */
CCTS_OFLOW /* CTS flow control of output */
CRTSCTS    /* same as CCTS_OFLOW */
CRTS_IFLOW /* RTS flow control of input */
MDMBUF     /* flow control output via Carrier */

```

Same applies to the labeling, order, and descriptions (though the lack of indentation for CS* yields another sentence on them being masked by CSIZE), and CREAD is opined on with the appendage of

Not all hardware supports this bit. In fact, this flag is pretty silly and if it were not part of the **termios** specification it would be omitted.

It's unclear why CS5 is marked "pseudo"; it appears to be supported on the devices that do and not on those that don't.

CRTS_IFLOW is not additionally described, but Clear to send and Request to send are part of RS232 hardware flow control.

The CCTS_OFLOW (CRTSCTS) flag is currently unused.

But does actually appear to be used on some platforms (like *4.4BSD Programmer's Reference Manual (PRM)*, `usr/src/sys/sparc/dev/zs.c`).

MDMBUF ("output flow control is controlled by the state of Carrier Detect.") corresponds to the LMDMBUF ("Stop/start output when carrier drops") *Seventh Edition, Virtual VAX-11 Version, November, 1980* mode.

No note is made of the default value.

"Values of the `c_lflag` field describe the control of various functions, and are composed of the following masks."

Local mode	Description
ECHOK	/* visual erase for line kill */
ECHOE	← /* visually erase chars */
ECHO	← /* enable echoing */
ECHONL	← /* echo NL even if ECHO is off */
ECHOPRT	← /* visual erase mode for hardcopy */
ECHOCTL	← /* echo control chars as ^ (Char) */
ISIG	← /* enable signals INTR, QUIT, [D]SUSP */
ICANON	← /* canonicalize input lines */
ALTWERASE	← /* use alternate WERASE algorithm */
IEXTEN	← /* enable DISCARD and LNEXT */
EXTPROC	← /* external processing */
TOSTOP	← /* stop background jobs from output */
FLUSHO	← /* output being flushed (state) */
NOKERNINFO	/* no kernel output from VSTATUS */
PENDIN	/* XXX retype pending input (state) */
NOFLSH	← /* don't flush after interrupt */

Same applies to the labeling (which is extra psycho here) and descriptions (down to "an implementation may echo an indication that" no erasing was needed for **icanon echoe**, and NOFLSH flushing "are not be done", which is "shall not be done" in the standard). The observant reader may notice that ECHOK is missing, but this just seems to be a documentation error, and it's in `termios.h`, commented "/* echo NL after line kill */", also as present-day.

ECHOK does indeed seem to replace/override ECHOK, rather than being a mode flag for it like in **UNIX® System V Release 4** and explicitly doesn't work with ECHOPRT (in which case it behaves like ECHOK; *ibid.*, `usr/src/sys/kern/tty.c`, `ttyinput()`). This is odd, and in contrast with LPRTERA (and **UNIX® System V Release 4**).

ECHOE, ECHOCTL, FLUSHO, PENDIN, and NOFLSH correspond to LCRTERA, LCTLECH, LFLUSHO, LPENDIN, and LNOFLSH, and match **UNIX® System V Release 4**. It's unclear why PENDIN is "XXX"ed, it appears fully-functional (ibid., `ttypend()`, `ttread()`).

ALTWERASE, NOKERNINFO, and EXTPROC are new and defer to previous sections, except for the latter, which is undescribed. It appears to short-circuit appx. all line editing, echoing, prevent OXTABS (ibid., `ttyinput()`, `tyrub()`, `tyecho()`, `tyoutput()`) and forward `ioctl()`s in `pty(4)` (ibid., `usr/src/share/man/man4/pty.4`) packet mode to the remote (ibid., `usr/src/sys/kern/tty_pty.c`, `ptyioctl()`).

The macro is actually as old as *Seventh Edition, Virtual VAX-11 Version, December, 1979*, described as `/* external processor (kmc) */` (`usr/include/sys/tty.h`), but unused until now.

No note is made of the default value.

Initial **Special Control Characters** values are "set according to the values in the header `<sys/ttydefaults.h>`." (of course, one is also directed to the nonexistent `<ttydefaults.h>` instead a few lines away). The entirely unsurprising summary from the manual follows; it's in declaration order, and doesn't match any other layout, not that it really should.

Index Name	Special Character	Default Value
VEOF	EOF	^D
VEOL	EOL	_POSIX_VDISABLE
VEOL2	EOL2	_POSIX_VDISABLE
VERASE	ERASE	^? \177
VWERASE	WERASE	^W
VKILL	KILL	^U
VREPRINT	REPRINT	^R
VINTR	INTR	^C
VQUIT	QUIT	^\\ \34
VSUSP	SUSP	^Z
VDSUSP	DSUSP	^Y
VSTART	START	^Q
VSTOP	STOP	^S
VLNEXT	LNEXT	^V
VDISCARD	DISCARD	^O
VMIN	---	1
VTIME	---	0
VSTATUS	STATUS	^T

Notably: all (actual) delay modes and LCASE are lost, and thank god. This distribution supports close to a dozen platforms and post-dates the relative ubiquity of display terminals by close to two decades.

LTILDE is lost, presumably because the last Hazeltine-branded product that used `~`s as escape introducers and didn't allow using ESCs was the 1500, released in 1977 — *Hazeltine 1500 Series Video Display Terminals Maintenance Manual*, HI-1053A, http://bitsavers.org/pdf/hazeltine/H1500/Hazeltine_1500_Series_Maintenance_Manual_Dec77.pdf, pp. 1-1, 1-8, Hazeltine Corporation Computer Terminal Equipment, Greenlawn, N.Y. 11740, December 1977, 1. Introduction and Description, 1.2 General Description, 1.2.1 Model 1500 and 1.2.2 Model 1510.

So is LETXACK but that's because it was never actually implemented.

LCRTBS is superseded by faster video terminals not needing it (so also by ECHOE).

"RAW" mode survives as one of the **Combos**, and maps very well. "CBREAK" mode is much iffier (`-icanon` excludes `lnext`), and is unrepresentable per se.

Rather than being encoded in some flag, `struct termios` has `c_[io]speed speed_t` members (although `speed_t` is a `long` so it's non-conformant), and the standard `cf[gs]et[io]speed()` functions set/return that (*4.BSD Programmer's Reference Manual (PRM)*, `usr/src/lib/libc/gen/tcsetattr.3`). A `cfsetspeed()` extension is also provided, expect-

edly just writing to both, and a `cfmakeraw()` that "sets the flags stored in the termios structure to a state disabling all input and output processing, giving a »raw I/O path.«" (`-ignbrk -brkint -parmrk -istrip -inlcr -igncr -icrnl -ixon -opost cs8 -parenb -isig -icanon -echo -echonl -iexten` but `"/ * XXX set MIN/TIME */"`; *ibid.*, `./4.4BSD-Lite2/usr/src/lib/libc/gen/termios.c`). The remainder of the interface is uneventful; the speed effects defer to `termios(4)` and the rest copies *IEEE Std 1003.1-1988* §7.2.1.

However, this is the first system the breaks away from the enshrined DH11 Line Parameter Register enumeration value, and "The value of the integer corresponds directly to the baud rate being represented" (with 134.5 baud truncated to **134**); compatibility macros are defined (the usual `EXTA EXTB` for **19200 38400**), but a speed finally is Just an integer, and many more pre-defined macros are provided (**7200, 14400, 28800, 57600, 76800, 115200, 230400** baud), presumably for means-testing.

then `stty,1`

remember to spec URM!p

`find 4.3 4.4BSD-Lite2 -name stty.1 -or -name tcsetattr.3 -or -path *bin*stty.c`

Not Considered (TODO: name)

UNIX/32V: it's unclear what the distribution tape looks like; the one available on TUHS reads like it's a snapshot of halfway between v6 and v7

`tabs 1`

2BSD: not a cohesive system, `(tset)` doesn't target anything in particular (defines for it) `>=3BSD tset: (a)` we'd have 20 more pages, `(b)` this is more part of curses because `(c)` it sets up environment and visuals more than the line itself (also `ncurses` ships it still), and `(d)` the calling convention and the domain is insane (see `(a)`)

Sub-/partial distributions of 4.1(abc12)

4.3reno: it's entirely different from both 4.3tahoe and 4.4lite2; uses `termios` but `tty.4` is still 4-ioctl