User Interfaces

EECE6029

Yizong Cheng

2/19/2016

Keyboard Software

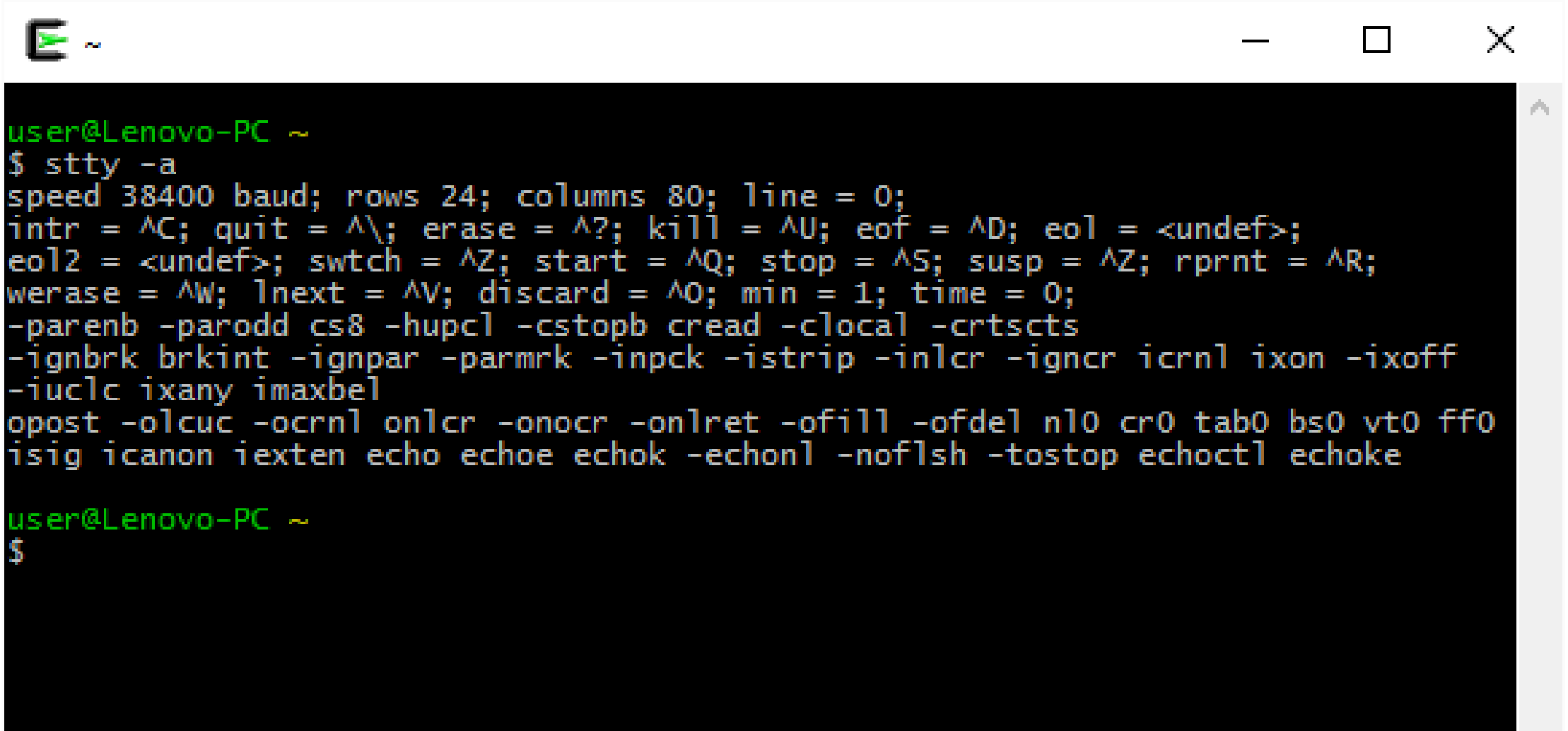
- Most keyboards have fewer than 128 keys and 7 bits encode one as the scan code.
- An interrupt is generated and when a key is pressed or released.
- The I/O register contains the scan code plus one bit for press/release.
- It’s up to the keyboard software/driver to convert these combinations into characters in ASCII or signals.
- raw mode or cooked mode
- canonical mode (line based) or noncanonical mode.

Special Characters in Canonical Mode

Character	POSIX name	Comment
CTRL-H	ERASE	Backspace one character
CTRL-U	KILL	Erase entire line being typed
CTRL-V	LNEXT	Interpret next character literally
CTRL-S	STOP	Stop output
CTRL-Q	START	Start output
DEL	INTR	Interrupt process (SIGINT)
CTRL-\	QUIT	Force core dump (SIGQUIT)
CTRL-D	EOF	End of file
CTRL-M	CR	Carriage return (unchangeable)
CTRL-J	NL	Line feed (unchangeable)

Figure 5-31. Characters that are handled specially in canonical mode.

The stty Command



stty -a

- speed 38400 baud; rows 24; columns 80; line = 0;
- intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
- eol2 = <undef>; swtch = ^Z; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
- werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
- -parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
- -ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
- -iuclc ixany imaxbel
- opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
- isig icanon iexten echo echoe echok -echonl -noflsh -tostop echoctl echoke

```
#include <termios.h>

int tcgetattr(int filedes, struct termios *termptr);

int tcsetattr(int filedes, int opt, const struct termios *termptr);
```

Both return: 0 if OK, –1 on error

```
struct termios {
  tcflag_t  c_iflag;     /* input flags */
  tcflag_t  c_oflag;     /* output flags */

  tcflag_t  c_cflag;     /* control flags */
  tcflag_t  c_lflag;     /* local flags */
  cc_t      c_cc[NCCS]; /* control characters */
};
```

c_iflag: Input Flags

Field	Flag	Description	POSIX.1	SVR4	4.3+BSD
				extension	
c_iflag	BRKINT	generate SIGINT on BREAK	•		
	ICRNL	map CR to NL on input	•		
	IGNBRK	ignore BREAK condition	•		
	IGNCR	ignore CR	•		
	IGNPAR	ignore characters with parity errors	•		
	IMAXBEL	ring bell on input queue full		•	•
	INLCR	map NL to CR on input	•		
	INPCK	enable input parity checking	•		
	ISTRIP	strip eighth bit off input characters	•		
	IUCLC	map uppercase to lowercase on input		•	
	IXANY	enable any characters to restart output		•	•
	IXOFF	enable start/stop input flow control	•		
	IXON	enable start/stop output flow control	•		
	PARMRK	mark parity errors	•		

c_oflag: Output Flags

	PARMRK	mark parity errors	•		
c_oflag	BSDLY	backspace delay mask		•	
	CRDLY	CR delay mask		•	
	FFDLY	form feed delay mask		•	
	NLDLY	NL delay mask		•	
	OCRNL	map CR to NL on output		•	
	OFDEL	fill is DEL, else NUL		•	
	OFILL	use fill character for delay		•	
	OLCUC	map lowercase to uppercase on output		•	
	ONLCR	map NL to CR-NL (ala CRMOD)		•	•
	ONLRET	NL performs CR function		•	
	ONOCR	no CR output at column 0		•	
	ONOEOT	discard EOTs (^D) on output			•
	OPOST	perform output processing	•		
	OXTABS	expand tabs to spaces			•
	TABDLY	horizontal tab delay mask		•	
	VTDLY	vertical tab delay mask		•	
					•

c_cflag: Control Flags

	VTDLY	vertical tab delay mask		
c_cflag	CCTS_OFLOW	CTS flow control of output		•
	CIGNORE	ignore control flags		•
	CLOCAL	ignore modem status lines	•	
	CREAD	enable receiver	•	
	CRTS_IFLOW	RTS flow control of input		•
	CSIZE	character size mask	•	
	CSTOPB	send two stop bits, else one	•	
	HUPCL	hangup on last close	•	
	MDMBUF	flow control output via Carrier		•
	PARENB	parity enable	•	
	PARODD	odd parity, else even	•	

c_lflag: Local Flags

	PARODD	odd parity, else even	•	
c_lflag	ALTWERASE	use alternate WERASE algorithm		•
	ECHO	enable echo	•	
	ECHOCTL	echo control chars as ^(Char)		• •
	ECHOE	visually erase chars	•	
	ECHOK	echo kill	•	
	ECHOKE	visual erase for kill		• •
	ECHONL	echo NL	•	
	ECHOPRT	visual erase mode for hardcopy		• •
	FLUSHO	output being flushed		• •
	ICANON	canonical input	•	
	IEXTEN	enable extended input char processing	•	
	ISIG	enable terminal-generated signals	•	
	NOFLSH	disable flush after interrupt or quit	•	
	NOKERNINFO	no kernel output from STATUS		•
	PENDIN	retype pending input		• •
	TOSTOP	send SIGTTOU for background output	•	
	XCASE	canonical upper/lower presentation		•

c_cc: Control Characters

Character	Description	c_cc subscript	Enabled by field	Enabled by flag	Typical value	POSIX.1	SVR4	4.3+BSD
							extension	
CR	carriage return	(can't change)	c_lflag	ICANON	\r	•		
DISCARD	discard output	VDISCARD	c_lflag	IEXTEN	^O		•	•
DSUSP	delayed suspend (SIGTSTP)	VDSUSP	c_lflag	ISIG	^Y		•	•
EOF	end-of-file	VEOF	c_lflag	ICANON	^D	•		
EOL	end-of-line	VEOL	c_lflag	ICANON		•		
EOL2	alternate end-of-line	VEOL2	c_lflag	ICANON			•	•
ERASE	backspace one character	VERASE	c_lflag	ICANON	^H	•		
INTR	interrupt signal (SIGINT)	VINTR	c_lflag	ISIG	^?, ^C	•		
KILL	erase line	VKILL	c_lflag	ICANON	^U	•		
LNEXT	literal next	VLNEXT	c_lflag	IEXTEN	^V		•	•
NL	linefeed	(can't change)	c_lflag	ICANON	\n	•		
QUIT	quit signal (SIGQUIT)	VQUIT	c_lflag	ISIG	^\	•		
REPRINT	reprint all input	VREPRINT	c_lflag	ICANON	^R		•	•
START	resume output	VSTART	c_iflag	IXON/IXOFF	^S	•		
STATUS	status request	VSTATUS	c_lflag	ICANON	^T			•
STOP	stop output	VSTOP	c_iflag	IXON/IXOFF	^Q	•		
SUSP	suspend signal (SIGTSTP)	VSUSP	c_lflag	ISIG	^Z	•		
WERASE	backspace one word	VWERASE	c_lflag	ICANON	^W		•	•

Power Consumption

Device	Li et al. (1994)	Lorch and Smith (1998)
Display	68%	39%
CPU	12%	18%
Hard disk	20%	12%
Modem		6%
Sound		2%
Memory	0.5%	1%
Other		22%

Figure 5-40. Power consumption of various parts of a notebook computer.

Zoning Illumination

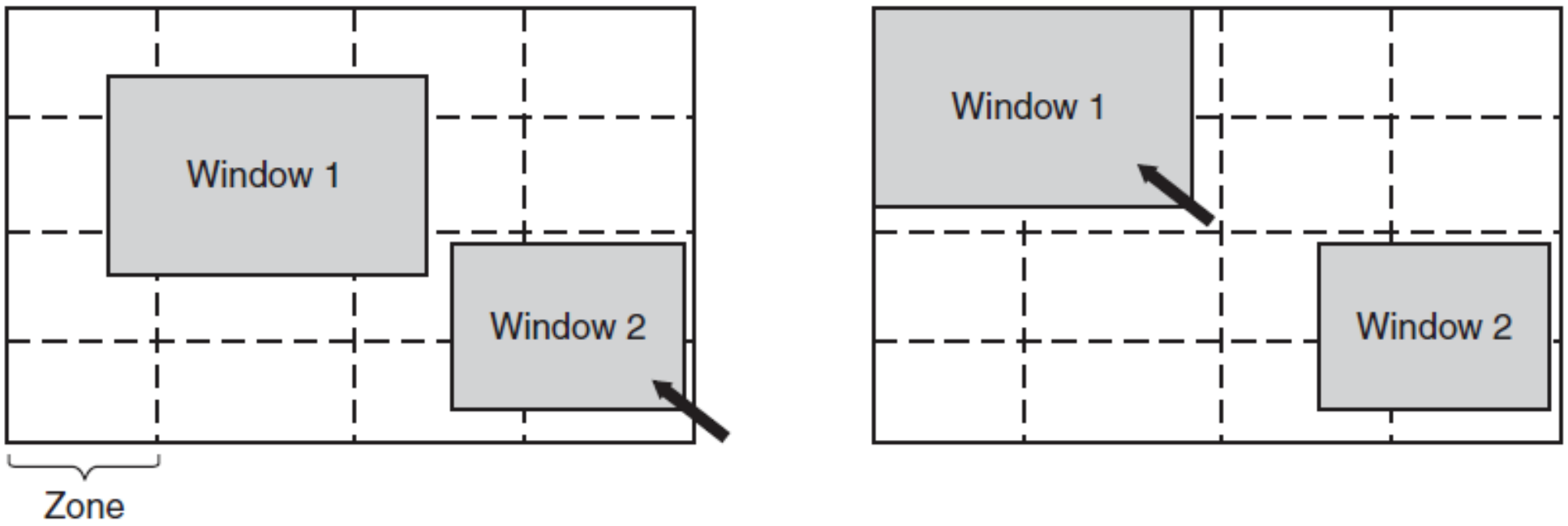


Figure 5-41. The use of zones for backlighting the display. (a) When window 2 is selected, it is not moved. (b) When window 1 is selected, it moves to reduce the number of zones illuminated.

CPU Power Consumption

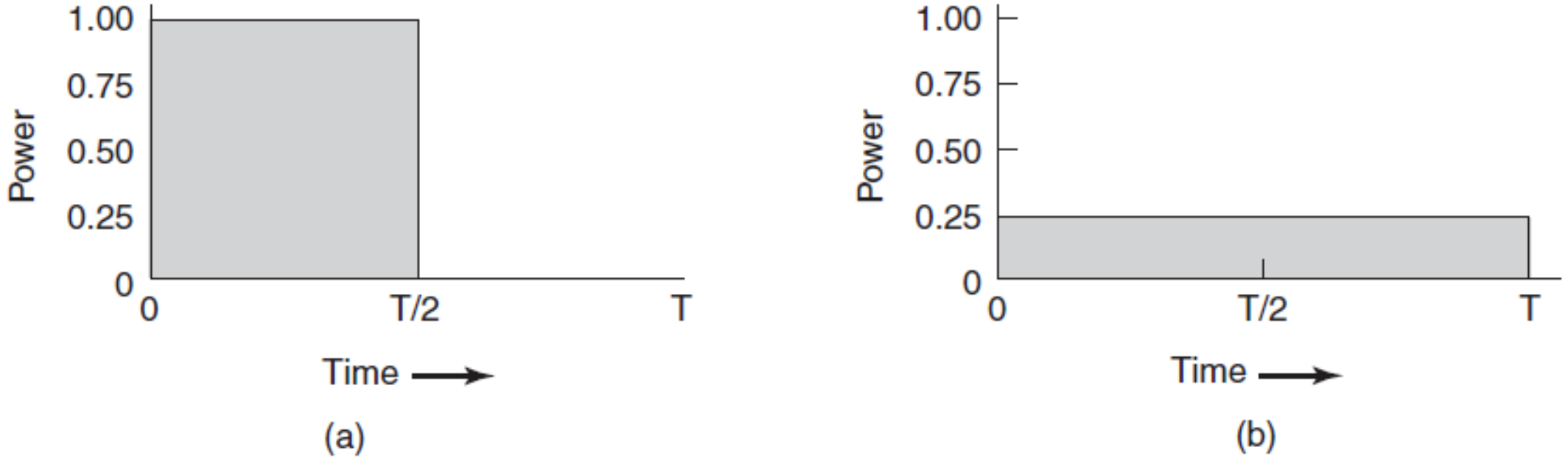


Figure 5-42. (a) Running at full clock speed. (b) Cutting voltage by two cuts clock speed by two and power consumption by four.