KEYBOARD INPUT

Module Number 5. Section 5
COP4600 – Operating Systems
Richard Newman

INPUT/OUTPUT HANDLING

- Hardware – handling low-level I/O
- Software – layering
- Mass storage
- Clocks
- Keyboard 
- Mouse
- Monitor
- Thin Clients
- Power Management
- Minix3 I/O

KEYBOARDS

- Keyboards originally tied to monitors
- Teletype
- Then glass tty
- Now, keyboard separate from display
- Even soft keyboard!

Teletype image by Rama & Musée Bolo - Own work – creative commons
Virtual keyboard image by matt buchanan - originally posted to **Flickr** as Apple iPad Event
VT100 image by Jason Scott - Flickr: IMG_9976, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=29457452

TERMINAL TAXONOMY

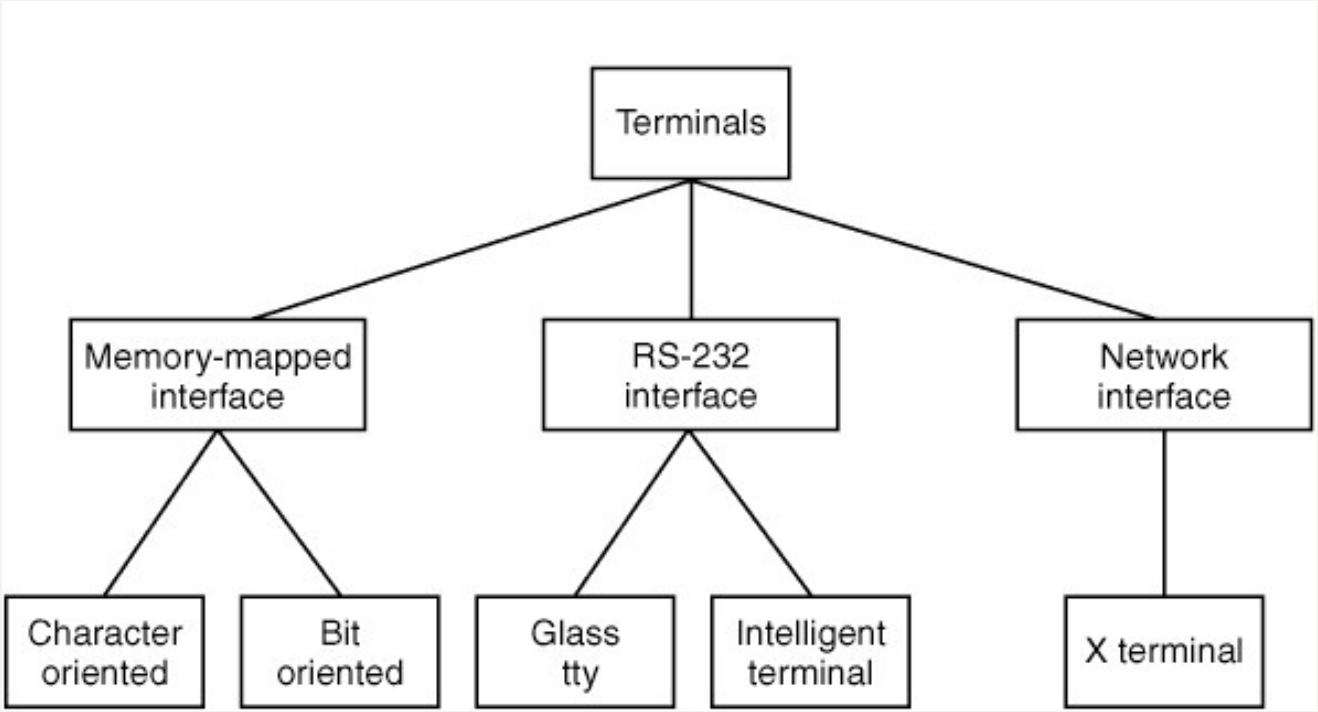


Figure 3-24. Terminal types.

MEMORY-MAPPED TERMINALS

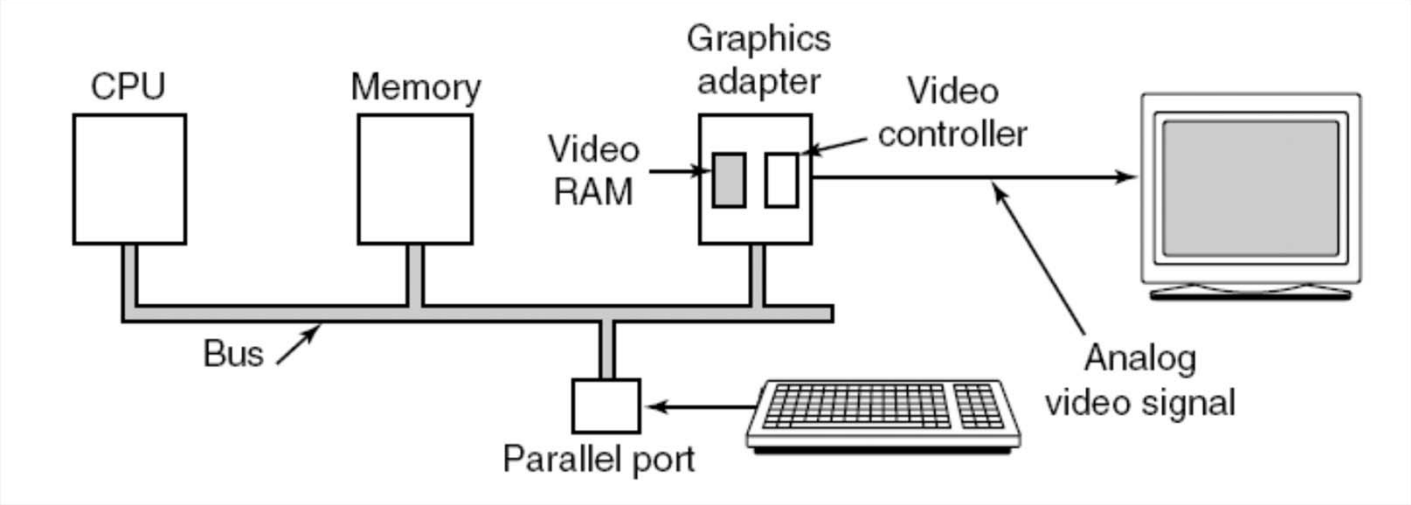


Figure 3-25. Memory-mapped terminals write directly into video RAM.

RS-232 TERMINALS

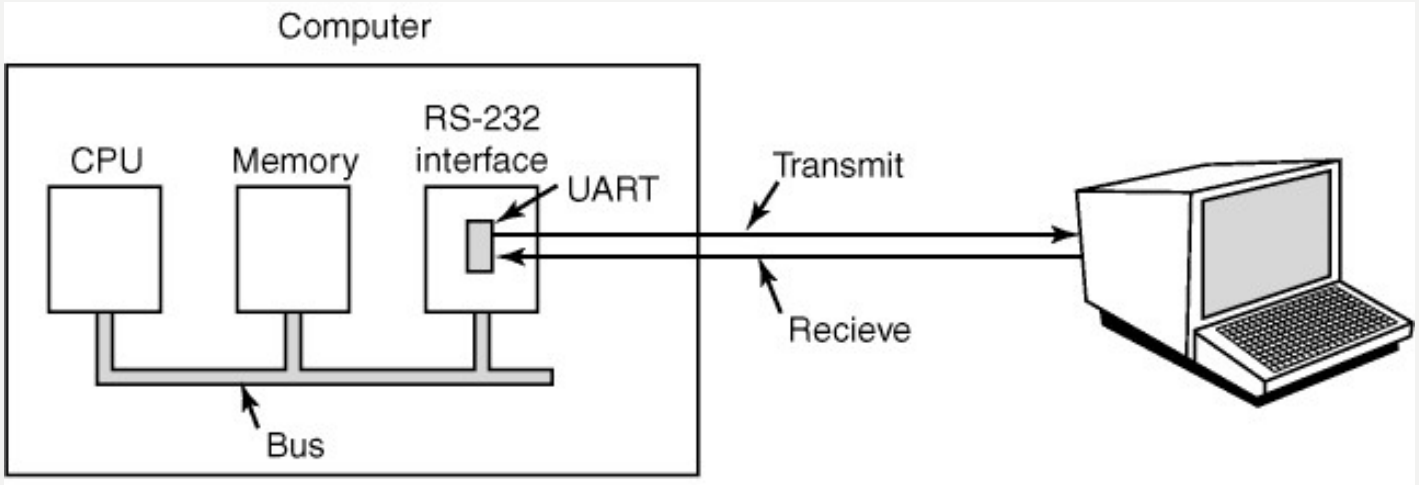


Figure 3-27. An RS-232 terminal communicates with a computer over a communication line, one bit at a time. The computer and the terminal are completely independent.

KEYBOARD DATA FLOW

User presses or releases key

Scan code sent to driver

Keyboard Driver

Driver converts to ASCII code(s) Stores in buffer. Cooks if in canonical mode

Driver Buffers

Process reads from keyboard

User process

Characters copied, depending on mode

Process Space

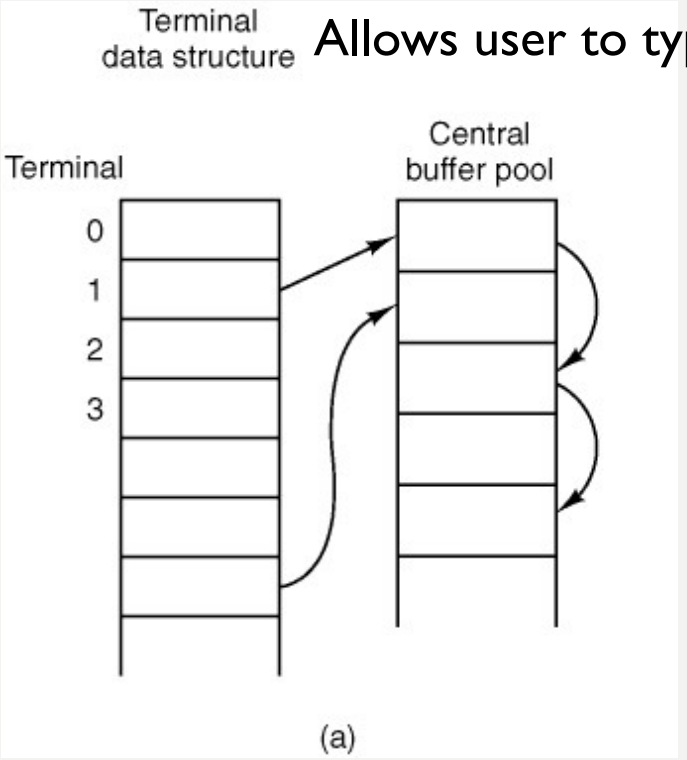
Echo sent to display (if on)

Display Driver

BUFFERING TERMINAL INPUT

Allows user to type ahead

Buffers in pool are typically small (8-16 bytes)



Buffers are big enough for all typing (100's of bytes)

Figure 3-28. (a) Central buffer pool.
(b) Dedicated buffer for each terminal.

KEYBOARD DRIVER (1)

42	35	163	170	18	146	38	166	38	166	24	152	57	185
L+	h+	h-	L-	e+	e-	l+	l-	l+	l-	o+	o-	SP+	SP-

54	17	145	182	24	152	19	147	38	166	32	160	28	156
R+	w+	w-	R-	o+	o-	r+	r-	l+	l-	d+	d-	CR+	CR-

Figure 3-41. Scan codes in the input buffer, with corresponding key actions below, for a line of text entered at the keyboard. L and R represent the left and right Shift keys. + and - indicate a key press and a key release. The code for a release is 128 more than the code for a press of the same key.

KEYBOARD DRIVER (2)

Key	Scan code	"ASCII"	Escape sequence
Home	71	0x101	ESC [H
Up Arrow	72	0x103	ESC [A
Pg Up	73	0x107	ESC [V
–	74	0x10A	ESC [S
Left Arrow	75	0x105	ESC [D
5	76	0x109	ESC [G
Right Arrow	77	0x106	ESC [C
+	78	0x10B	ESC [T
End	79	0x102	ESC [Y
Down Arrow	80	0x104	ESC [B
Pg Dn	81	0x108	ESC [U
Ins	82	0x10C	ESC [@

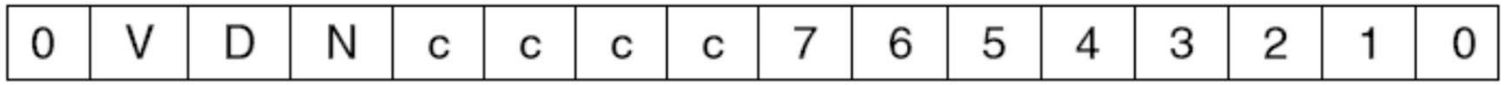
Figure 3-42. Escape codes generated by the numeric keypad. When scan codes for ordinary keys are translated into ASCII codes the special keys are assigned "pseudo ASCII" codes with values greater than 0xFF.

LOADABLE KEYMAPS

Scan code	Character	Regular	SHIFT	ALT1	ALT2	ALT+SHIFT	CTRL
00	none	0	0	0	0	0	0
01	ESC	C('[')	C('[')	CA('[')	CA('[')	CA('[')	C('[')
02	'1'	'1'	'!'	A('1')	A('1')	A('!')	C('A')
13	'='	'='	'+'	A('=')	A('=')	A('+')	C('@')
16	'q'	L('q')	'Q'	A('q')	A('q')	A('Q')	C('Q')
28	CR/LF	C('M')	C('M')	CA('M')	CA('M')	CA('M')	C('J')
29	CTRL	CTRL	CTRL	CTRL	CTRL	CTRL	CTRL
59	F1	F1	SF1	AF1	AF1	ASF1	CF1
127	???	0	0	0	0	0	0

Figure 3-37. A few entries from a keymap source file.

TERMINAL DRIVER SUPPORT CODE



V: IN_ESC, escaped by LNEXT (CTRL-V)
D: IN_EOF, end of file (CTRL-D)
N: IN_EOT, line break (NL and others)
cccc: count of characters echoed
7: Bit 7, may be zeroed if ISTRIP is set
6-0: Bits 0-6, ASCII code

Figure 3-40. The fields in a character code as it is placed into the input queue.

TERMINAL INPUT CODES

- Canonical (cooked) mode
  - Line at a time input
  - Must still echo characters when typed
  - But allow user to correct input before processing
- Raw mode
  - Characters can be passed to the process as they are typed
  - Used for games, visual editors, etc.
  - Program interprets meaning of each character
- cbreak Mode
  - Similar to raw
  - But INTR, QUIT, START, STOP retain effects

KEYBOARD SOFTWARE

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	Linefeed (unchangeable)

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

Characters handled specially by cbreak mode.

WHEN TO RETURN IN RAW MODES

	TIME = 0	TIME > 0
MIN = 0	Return immediately with whatever is available, 0 to N bytes	Timer starts immediately. Return with first byte entered or with 0 bytes after timeout
MIN > 0	Return with at least MIN and up to N bytes. Possible indefinite block.	Interbyte timer starts after first byte. Return N bytes if received by timeout, or at least 1 byte at timeout. Possible indefinite block

Figure 3-31. MIN and TIME determine when a call to read returns in noncanonical mode. N is the number of bytes requested.

INPUT/OUTPUT HANDLING

- Hardware – handling low-level I/O
- Software – layering
- Mass storage
- Clocks
- Keyboard
- Mouse Next
- Monitor
- Thin Clients
- Power Management
- Minix3 I/O