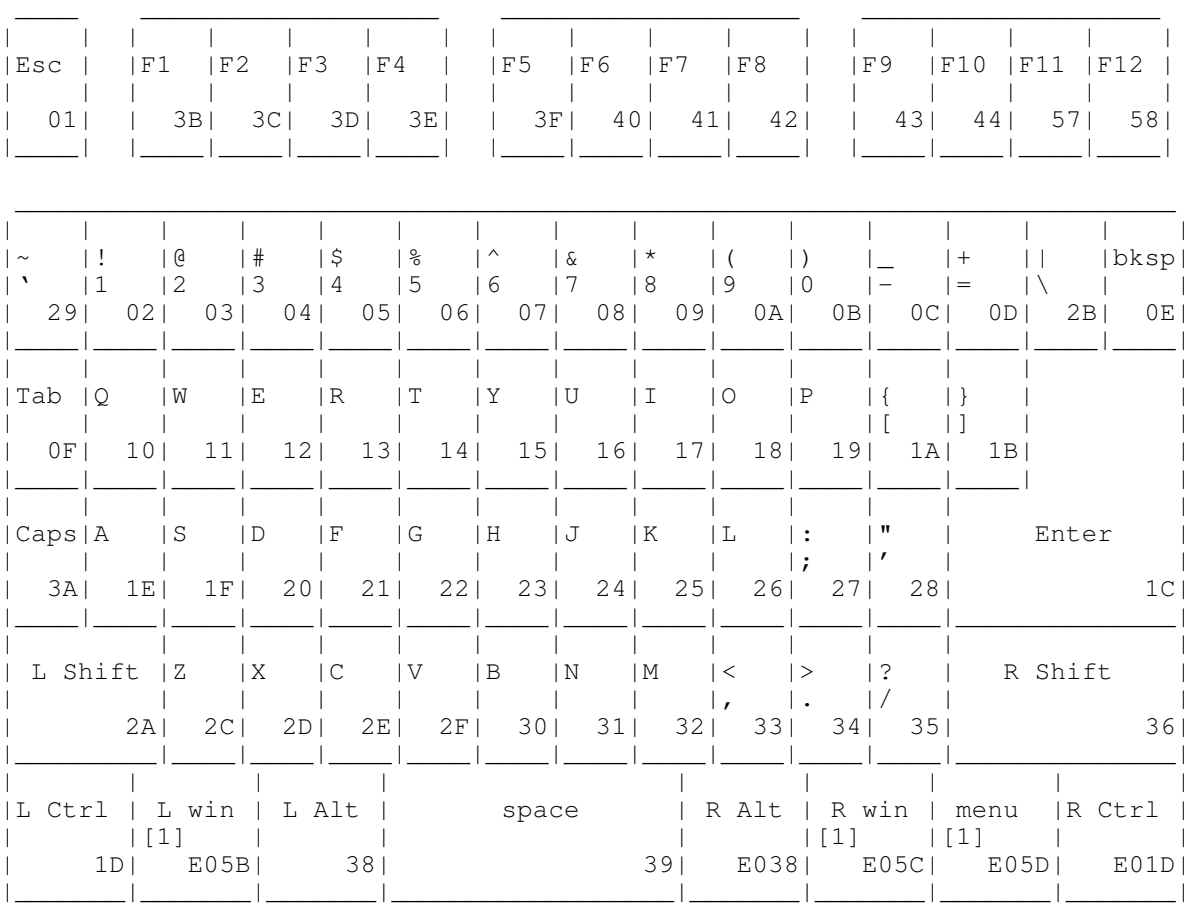
```
================================================================
IBM PC KEYBOARD INFORMATION FOR SOFTWARE DEVELOPERS
================================================================
Your host: Chris Giese
http://www.execpc.com/~geezer/os

Distribute freely. Last revised on Jan 3, 2002
Sources:
        PORTS.A of Ralf Brown's interrupt list collection
        repairfaq.org keyboard FAQ(doesn't appear to exsist)
        Linux source code
Test hardware:
        New Samsung KB3T001SAXAA 104-key keyboard
        Old Maxi 2186035-00-21 101-key keyboard

NO WARRANTY. NO GUARANTEE. I have tried to make this information
accurate. I don't know if I succeeded. Corrections or additional
information would be welcome.

This is a plain-text document. If you use a word-processor to view
it, use a fixed-pitch font (like Courier) so columnar data and
ASCII art lines up properly.

Lessons learned:
- Both the 8048 MCU in the keyboard and the 8042 controller
  on the motherboard accept command bytes.
- There is a bit (KCC) in the poorly-named "Command Byte" which
  seems to enable AT-to-XT scancode conversion (scancode set 2
  to scancode set 1). After booting DOS, my keyboard uses
  scancode set 2 with this conversion bit turned on. If I turn
  the bit off and switch to scancode set 1, operation remains
  the same.
- Scancode set 3 is probably the most elegant, in that it returns
  a one-byte make code for _every_ key. Unfortunately, not all
  keyboards support it.
- The scancodes of some keys depend on the internal num lock state
  of the keyboard.

================================================================
KEYBOARD I/O REGISTERS ON THE PC
================================================================
60h     data
64h     command (write)
64h     Status (read)

Bits in Status register (names from Linux source)
    b7  PERR    parity error in data received from keyboard
    b6  GTO     receive timeout
    b5          transmit timeout (or PS/2 mouse?)
    b4          keyboard is locked
    b3          0=60h was the port last accessed, 1=61h was last (?)
    b2          System Flag status: 0=power-up/reset, 1=selftest OK (?)
    b1  IBF     input buffer full (data from host to keyboard)
    b0  OBF     output buffer full (data from keyboard to host)

Bits in Output Port of 8042 chip (Table P0383 in PORTS.A)
The Output Port is written by controller command D1h,
and read by controller command D0h
    b7          keyboard data output
    b6          keyboard clock output
    b5          input buffer NOT full
    b4          output buffer NOT empty
    b3          (varies)
    b2          (varies)
    b1          A20 gate
    b0          system reset (THIS BIT SHOULD ALWAYS BE SET TO 1)

Bits in Input Port of 8042 chip
```

```
The Input Port is read by controller command C0h
    b7          keyboard NOT locked
    b6-b0       (varies)

Bits in "Command Byte" (confusing name; from Table P0404 in PORTS.A)
The "Command Byte" is written by controller command 60h
and read by controller command 20h
(names from Linux source)
    b7          (reserved)
    b6  KCC     convert set 2 scancodes to set 1 ("IBM PC compatibility mode")
    b5  DMS     disables PS/2 mouse when set
    b4          disables keyboard when set
    b3          ignore keyboard lock switch when set
    b2  SYS     System Flag (same as b2 in Status register, it seems)
    b1          enables IRQ12 from PS/2 mouse when set
    b0  EKI     enables IRQ1 on keyboard output buffer full

Result Byte for interface self-tests (Table P0406 in PORTS.A)
Returned by controller commands A9h or ABh
    0           no error
    1           clock line stuck low
    2           clock line stuck high
    3           data line stuck low
    4           data line stuck high

================================================================
CONTROLLER COMMANDS (from Table P0401 of PORTS.A)
================================================================
Before writing each byte of these commands to port 64h,
poll the status register (port 60h) until bit b1=0.

20h-2Fh         reads byte with address=lower 5 bits of command
                  The byte at address 0 is the "Command Byte".
60h-7Fh nn      writes byte nn to address=lower 5 bits of command
                  The byte at address 0 is the "Command Byte".
A7h             disables PS/2 mouse port (MCA only?)
A8h             enables PS/2 mouse port (MCA only?)
A9h             self-test mouse interface, returns Result Byte (see above)
AAh             self-test controller; returns 55h if success, FCh if failure
ABh             self-test keyboard interface, returns Result Byte (see above)
ADh             disables keyboard (sets b4 of "Command Byte")
AEh             enables keyboard (clears b4 of "Command Byte")
C0h             reads Input Port
D0h             reads Output Port
D1h nn          writes Output Port
                  Important: bit 0 (system reset) should always be set here,
                  as the system may hang constantly. To reset the PC, pulse
                  b0 of the Output Port with command FEh instead.
DDh             disable A20 (Not all systems support this byte)
DFh             enable A20 (Not all systems support this byte)
E0h             read test inputs. return value=
                    b1      kbd data
                    b0      kbd clock
EDh nn          write LEDs. nn=
                    b2      Caps Lock
                    b1      Num Lock
                    b0      Scroll Lock
F0h-FFh         pulse bit(s) of Output Port low for 6 microseconds.
                  If b0-b3 of the command is low, the corresponding bit
                  in the Output Port will be pulsed low. b0=system reset,
                  and should ALWAYS be PULSED low, never set low constantly.

================================================================
KEYBOARD COMMANDS (from Table P0386 of PORTS.A)
================================================================
Before writing each byte of these commands to port 60h,
poll the status register (port 60h) until bit b1=0.
Unless otherwise noted: each command responds with FAh (ACKnowledge)
or FEh (Resend) after receiving each byte of the command.
```

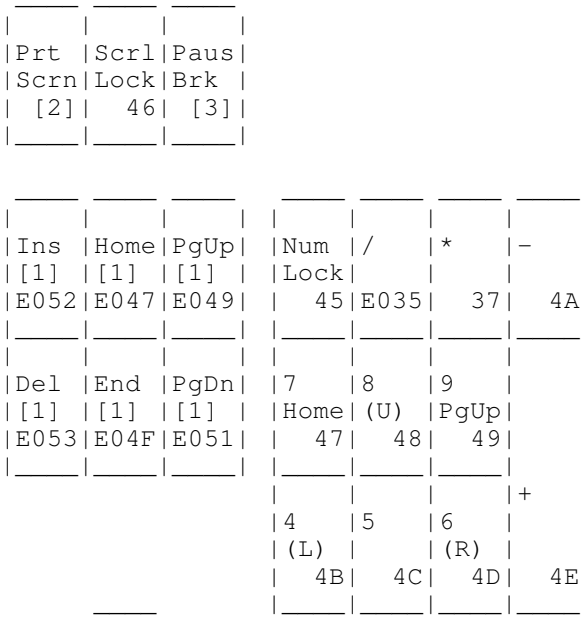
```
EDh nn          write LEDs, as above
EEh             echo, keyboard responds with EEh
EFh             no-operation (reserved)
F0h nn          selects scancode set nn=1-3 or 0 to return current set
F2h             read ID. Keyboard responds with ACK (FAh) and two optional
                ID bytes:
                    (none)      AT keyboard
                    83h ABh     (?)
                    ABh 41h     MF2, translation mode
                    ABh 83h     MF2, pass-through mode
F3h nn          set typematic (auto-repeat) rate/delay. nn=
                    b7      unused
                    b6..5   Repeat delay (00=250 msec ... 11=1000msec)
                    b4..0   Repeat rate (00000=30 Hz ... 11111=2 Hz).
F4h             clears output buffer, enables keyboard
F5h             disables keyboard, resets to defaults
F6h             sets keyboard defaults

F7h             make all keys typematic (auto-repeat)   [*]
F8h             make all keys make-break                [*]
F9h             make all keys make-only                 [*]
FAh             make all keys typematic and make-break  [*]
FBh nn          make one key typematic                  [*]
FCh nn          make one key make-break                 [*]
FDh nn          make one key make-only                  [*]
                        [*] these commands may work only for
                            scancode set 3; I'm not sure.
FEh             resend previous scan code
FFh             reset keyboard CPU, do power-on self-test, return
                self-test result byte

non-key status bytes
--------------------
00h    Key detection error or buffer full.
AAh    Power-on/reset diagnostics successful.
E0h    (scancode sets 1 and 2) Prefix byte for "gray" keys
        (keys not on original 83-/84-key keyboard)
EEh    Sent in response to ECHO command.
F0h    (scancode sets 2 and 3) Prefix byte for break codes.
FAh    ACKknowledge; response to most commands.
FCh    Diagnostics failed (MF keyboard).
FDh    Diagnostics failed (AT keyboard).
        The keyboard stops scanning and waits for next command
        after returning code FCh or FDh
FEh    Last command was invalid or had parity error; resend it.
FFh    Key detection error or buffer full.

================================================================
SCANCODES FOR SCANCODE SET 1 (XT)
================================================================
US 104-key keyboard, set 1 scancodes


"Make" code is generated when key is pressed.
"Break" code is generated when key is released.
Hex value of make code for each key is shown.

Most keys:
        one-byte make code      = nn
        one-byte repeat code    = nn
        one-byte break code     = 80h + nn

"Gray" keys (not on original 84-key keyboard):
        two-byte make code      = E0nn
        two-byte repeat code    = E0nn
        two-byte break code     = E0 followed by 80h + nn

"Gray" keys noted by [1] are NumLock-sensitive.
```

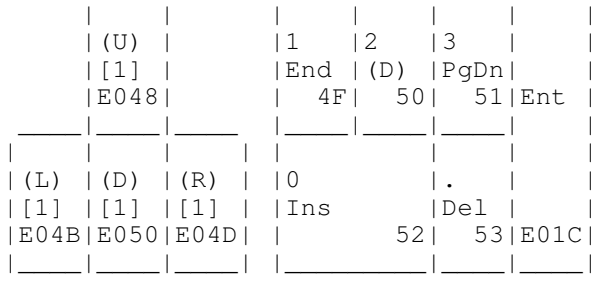
When the keyboard's internal NumLock is active:
four-byte make code = E02AE0nn
two-byte repeat code = E0nn
four-byte break code = E0 followed by 80h + nn followed by E0AA



[2] For PrintScreen/SysRq key: make code = E02AE037,
repeat code = E037, break code = E0B7E0AA

[3] The Pause/Break key does not repeat, and it does not
generate a break code. Its make code is E11D45E19DC5





code	key	code	key	code	key	code	key
01	Esc	0F	Tab	1D	L Ctrl	2B	|
02	1!	10	Q	1E	A	2C	Z
03	2"	11	W	1F	S	2D	X
04	3#	12	E	20	D	2E	C
05	4$	13	R	21	F	2F	V
06	5%	14	T	22	G	30	B
07	6^	15	Y	23	H	31	N
08	7&	16	U	24	J	32	M
09	8*	17	I	25	K	33	,<
0A	9(	18	O	26	L	34	.>
0B	0)	19	P	27	;:	35	/?
0C	-_	1A	[{	28	'"	36	R Shift
0D	=+	1B	]}	29	`~	37	*
0E	BackSpace	1C	Enter	2A	L Shift	38	L Alt

code	key	code	key	code	key	code	key
39	Space	41	F7	49	PageUp 9	51	PageDown 3
3A	CapsLock	42	F8	4A	-	52	Insert 0
3B	F1	43	F9	4B	(left) 4	53	Del .
3C	F2	44	F10	4C	5		
3D	F3	45	NumLock	4D	(right) 6	57	F11
3E	F4	46	ScrollLock	4E	+	58	F12
3F	F5	47	Home 7	4F	End 1		
40	F6	48	(up) 8	50	(down) 2		

code	key
E01C	Enter (on numeric keypad)
E01D	R Ctrl
E02A	make code prefix for keyboard internal numlock
E02AE037	PrintScreen make code
E035	/
E037	PrintScreen repeat code
E038	R Alt
E047	Home
E048	(up)
E049	PageUp
E04B	(left)
E04D	(right)
E04F	End
E050	(down)
E051	PageDown
E052	Insert
E053	Del
E05B	L Win
E05C	R Win
E05D	Menu
E0AA	break code suffix for keyboard internal numlock
E0B7E0AA	PrintScreen break code
E11D45E19DC5	Pause

==========

SCANCODES FOR SCANCODE SET 2 (AT)
==========

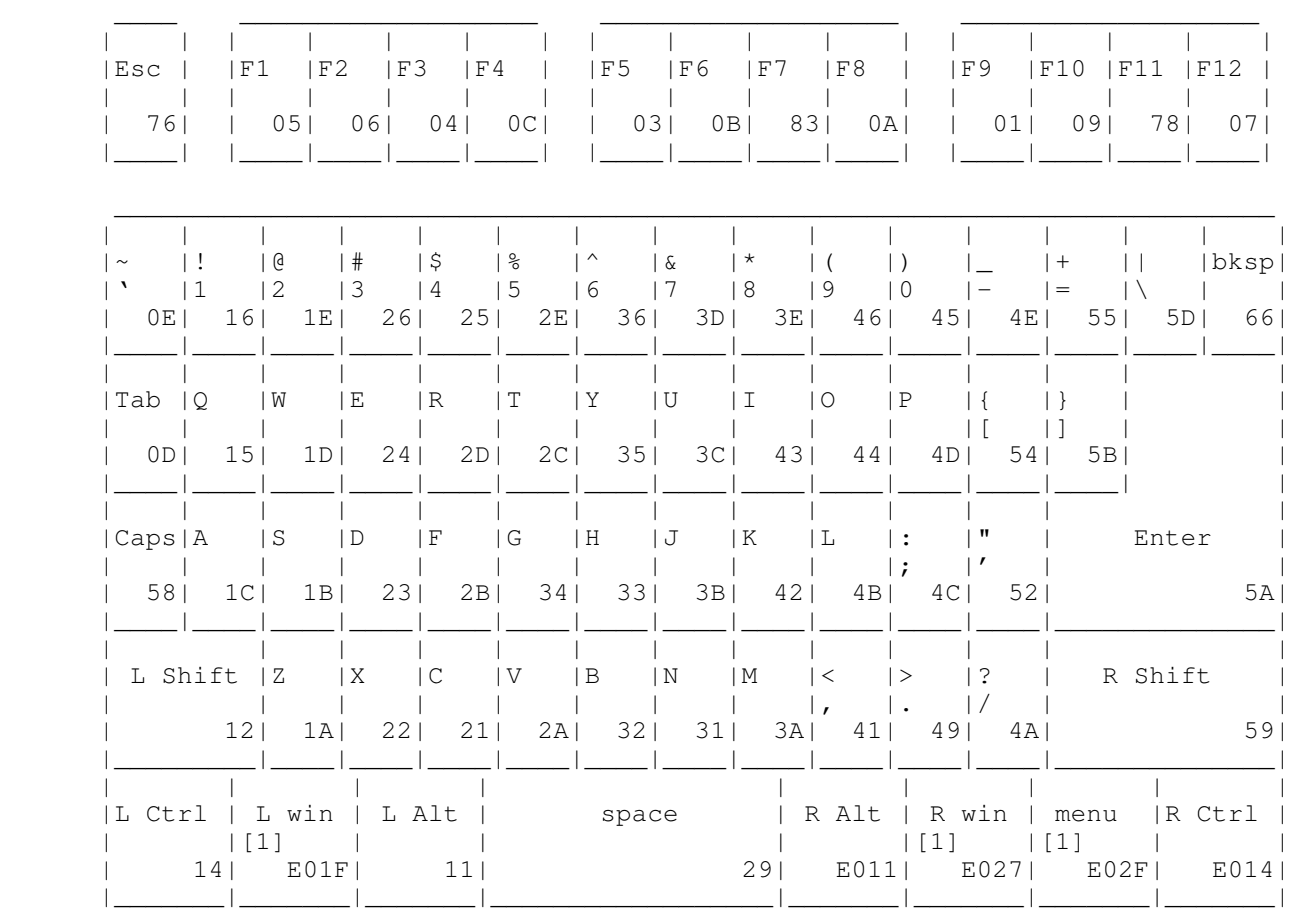
US 104-key keyboard, set 2 scancodes,
8042 AT-to-XT scancode translation OFF

"Make" code is generated when key is pressed.
"Break" code is generated when key is released.
Hex value of make code for each key is shown.

Most keys:
- one-byte make code = nn
- one-byte repeat code = nn
- two-byte break code = F0nn

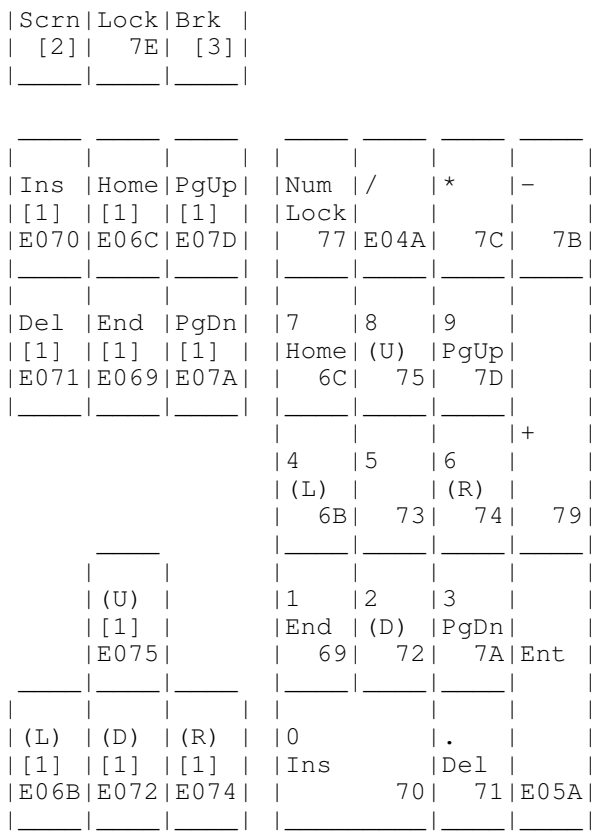
"Gray" keys (not on original 84-key keyboard):
- two-byte make code = E0nn
- two-byte repeat code = E0nn
- three-byte break code = E0F0nn

"Gray" keys noted by [1] are NumLock-sensitive.
When the keyboard's internal NumLock is active:
- four-byte make code = E012E0nn
- two-byte repeat code = E0nn
- six-byte break code = E0F0nnE0F012



[2] For PrintScreen/SysRq key: make code = E012E07C,
repeat code = E07C, break code = E0F07CE0F012

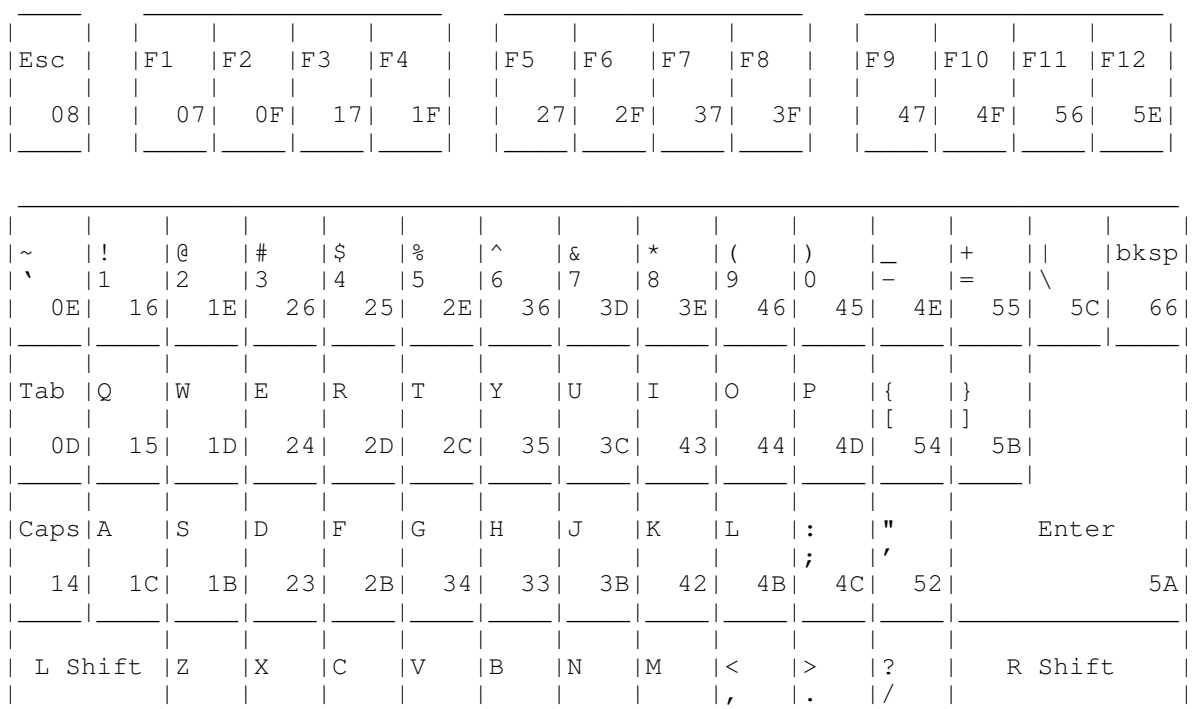
[3] The Pause/Break key does not repeat, and it does not
generate a break code. Its make code is E11477E1F014F077

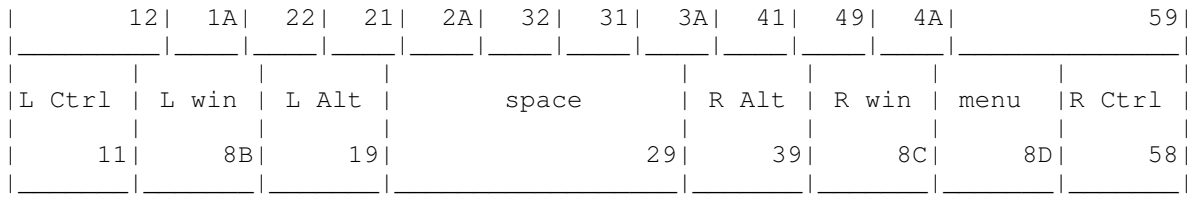


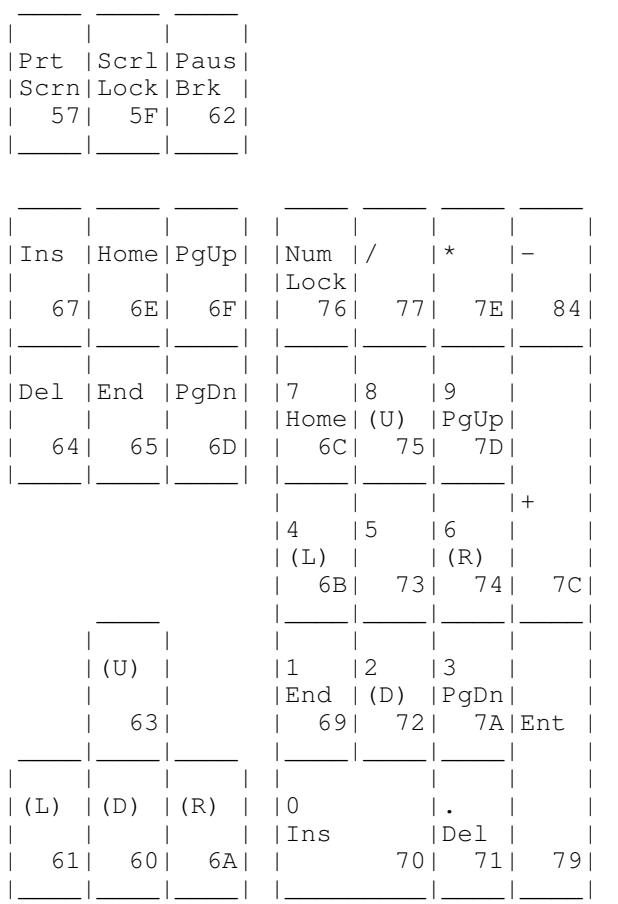
code	key	code	key	code	key	code	key
01	F9					66	BackSpace
		21	C	41	,<		
03	F5	22	X	42	K	69	End 1
04	F3	23	D	43	I		
05	F1	24	E	44	O	6B	(left) 4
06	F2	25	4$	45	0)	6C	Home 7
07	F12	26	3#	46	9(		
						70	Ins 0
09	F10	29	Space	49	.>	71	Del .
0A	F8	2A	V	4A	/?	72	(down) 2
0B	F6	2B	F	4B	L	73	5
0C	F4	2C	T	4C	;:	74	(right) 6
0D	Tab	2D	R	4D	P	75	(up) 8
0E	`~	2E	5%	4E	-_	76	Esc
						77	NumLock
11	L Alt	31	N	52	'"	78	F11
12	L Shift	32	B			79	+
		33	H	54	[{	7A	PageDown 3
14	L Ctrl	34	G	55	=+	7B	-
15	Q	35	Y			7C	*
16	1!	36	6^	58	CapsLock	7D	PageUp 9
				59	R Shift	7E	ScrollLock
1A	Z	3A	M	5A	Enter		
1B	S	3B	J	5B	]}	83	F7
1C	A	3C	U				
1D	W	3D	7&	5D	|		
1E	2@	3E	8*				

code	key
E011	R Alt
E012E07C	PrintScreen make code
E014	R Ctrl
E01F	L Win
E027	R Win

```
E02F            Menu
E04A            /
E05A            Enter (on numeric keypad)
E069            End
E06B            Left
E06C            Home
E070            Ins
E071            Del
E072            (down)
E074            (right)
E075            (up)
E07A            PageDown
E07C            PrintScreen repeat code
E07D            PageUp
E0F07CE0F012    PrintScreen break code
E11477E1F014F077 Pause

=====================================================================
SCANCODES FOR SCANCODE SET 3
=====================================================================
US 104-key keyboard, set 3 scancodes


"Make" code is generated when key is pressed.
"Break" code is generated when key is released.
Hex value of make code for each key is shown.

All keys:
        one-byte make code      = nn
        one-byte repeat code    = nn
        two-byte break code     = F0nn

When operating in scancode set 3, the keyboard
does not maintain an internal NumLock state.
```







code	key	code	key	code	key	code	key
07	F1	2A	V	4A	/?	6B	(left) 4
08	Esc	2B	F	4B	L	6C	Home 7
		2C	T	4C	;:	6D	PageDown
0D	Tab	2D	R	4D	P	6E	Home
0E	`~	2E	5%	4E	-_	6F	PageUp
0F	F2	2F	F6	4F	F10	70	Ins 0
						71	Del .
11	L Ctrl	31	N	52	'"	72	(down) 2
12	L Shift	32	B			73	5
		33	H	54	[{	74	(right) 6
14	CapsLock	34	G	55	=+	75	(up) 8
15	Q	35	Y	56	F11	76	NumLock
16	1!	36	6^	57	PrintScr	77	/
17	F3	37	F7	58	R Ctrl		
				59	R Shift	79	Enter (on numeric keypad)
19	L Alt	39	R Alt	5A	Enter	7A	PageDown 3
1A	Z	3A	M	5B	]}		

1B	S	3B	J	5C	|	7C	+
1C	A	3C	U			7D	PageUp 9
1D	W	3D	7&	5E	F12	7E	*
1E	2@	3E	8*	5F	ScrollLock		
1F	F4	3F	F8	60	(down)	84	-
				61	(left)		
21	C	41	,<	62	Pause	8B	L Win
22	X	42	K	63	(up)	8C	R Win
23	D	43	I	64	Del	8D	Menu
24	E	44	O	65	End		
25	4$	45	0)	66	BackSpace		
26	3#	46	9(	67	Ins		
27	F5	47	F9				
				69	End 1		
29	Space	49	.>	6A	(right)		