**LAPORAN PRAKTIKUM SISTEM OPERASI**

**MODUL 1**

**“PENGENALAN SISTEM PENGEMBANGAN OS DENGAN**

**PC SIMULATOR BOCHS”**



**Oleh:  
Daffa Putra Alwansyah**

**L200190031**

**Fakultas Komunikasi dan Informatika**

**Universitas Muhammadiyah Surakarta**

1. Apa yang dimaksud dengan kode ‘ASCII’, buatlah tabel kode ASCII lengkap cukup kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary dan hexadesimal serta karakter dan symbol yang dikodekan.

**Jawaban:**

ASCII ialah singkatan bagi American Standard Code for Information Interchange. ASCII adalah kod – kod yang digunakan untuk memudahkan interaksi antara user dan komputer. Dengan kata lain, ASCII digunakan untuk pertukaran maklumat dan komunikasi data dengan cara menukarkan angka kepada karakter aksara. ASCII sebenarnya lebih dikenali sebagai kod angka 7-bit yang mewakili sebuah karakter, ASCII digunakan kerana komputer hanya memahami angka sahaja.

Asalnya kod ASCII mempunyai bilangan binary sebanyak 8-bit yang bermula dari 00000000 hingga 11111111. Jumlah kombinasi yang dihasilkan adalah sebanyak 255, bermula dari kod 0 hingga 255 dalam sistem Desimal. Kod ASCII 0 hingga 127 merupakan kod untuk manipulasi teks, manakala kod ASCII 128 hingga 255 untuk manipulasi grafik.

## Tabel :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ANSI ASCII**  **(Desimal)** | | **Unicode (Heksa Desimal)** | **Binner** | **Symbol** |
| 0 | | 00 | 00000000 | NUL |
| 1 | | 01 | 00000001 | SOH |
| 2 | | 02 | 00000010 | STX |
| 3 | | 03 | 00000011 | ETX |
| 4 | | 04 | 00000100 | EOT |
| 5 | | 05 | 00000101 | ENQ |
| 6 | | 06 | 00000110 | ACK |
| 7 | | 07 | 00000111 | BEL |
| 8 | | 08 | 00001000 | BS |
| 9 | | 09 | 00001001 | HT |
| 10 | | 0A | 00001010 | LF |
| 11 | | 0B | 00001011 | VT |
| 12 | | 0C | 00001100 | FF |
| 13 | | 0D | 00001101 | CR |
| 14 | | 0E | 00001110 | SO |
| 15 | | 0F | 00001111 | SI |
| 16 | | 10 | 00010000 | DLE |
| 17 | | 11 | 00010001 | DC1 |
| 18 | | 12 | 00010010 | DC2 |
| 19 | | 13 | 00010011 | DC3 |
| 20 | | 14 | 00010100 | DC4 |
| 21 | | 15 | 00010101 | NAK |
| 22 | | 16 | 00010110 | SYN |
| 23 | | 17 | 00010111 | ETB |
| 24 | | 18 | 00011000 | CAN |
| 25 | | 19 | 00011001 | EM |
| 26 | | 1A | 00011010 | SUB |
| 27 | | 1B | 00011011 | ESC |
| 28 | | 1C | 00011100 | FS |
| 29 | | 1D | 00011101 | GS |
| 30 | | 1E | 00011110 | RS |
| 31 | | 1F | 00011111 | US |
| 32 | | 20 | 00100000 | space |
| 33 | | 21 | 00100001 | ! |
| 34 | | 22 | 00100010 | " |
| 35 | | 23 | 00100011 | # |
| 36 | | 24 | 00100100 | $ |
| 37 | | 25 | 00100101 | % |
| 38 | | 26 | 00100110 | & |
| 39 | | 27 | 00100111 | ' |
| 40 | | 28 | 00101000 | ( |
| 41 | | 29 | 00101001 | ) |
| 42 | | 2A | 00101010 | \* |
| 43 | | 2B | 00101011 | + |
| 44 | | 2C | 00101100 | , |
| 45 | | 2D | 00101101 | - |
| 46 | | 2E | 00101110 | . |
| 47 | | 2F | 00101111 | / |
| 48 | | 30 | 00110000 | 0 |
| 49 | | 31 | 00110001 | 1 |
| 50 | | 32 | 00110010 | 2 |
| 51 | | 33 | 00110011 | 3 |
| 52 | | 34 | 00110100 | 4 |
| 53 | | 35 | 00110101 | 5 |
| 54 | | 36 | 00110110 | 6 |
| 55 | | 37 | 00110111 | 7 |
| 56 | | 38 | 00111000 | 8 |
| 57 | | 39 | 00111001 | 9 |
| 58 | | 3A | 00111010 | : |
| 59 | | 3B | 00111011 | ; |
| 60 | | 3C | 00111100 | < |
| 61 | | 3D | 00111101 | = |
| 62 | | 3E | 00111110 | > |
| 63 | | 3F | 00111111 | ? |
| 64 | | 40 | 01000000 | @ |
| 65 | | 41 | 01000001 | A |
| 66 | | 42 | 01000010 | B |
| 67 | | 43 | 01000011 | C |
| 68 | | 44 | 01000100 | D |
| 69 | | 45 | 01000101 | E |
| 70 | | 46 | 01000110 | F |
| 71 | | 47 | 01000111 | G |
| 72 | | 48 | 01001000 | H |
| 73 | | 49 | 01001001 | I |
| 74 | | 4A | 01001010 | J |
| 75 | | 4B | 01001011 | K |
| 76 | | 4C | 01001100 | L |
| 77 | | 4D | 01001101 | M |
| 78 | | 4E | 01001110 | N |
| 79 | | 4F | 01001111 | O |
| 80 | | 50 | 01010000 | P |
| 81 | | 51 | 01010001 | Q |
| 82 | | 52 | 01010010 | R |
| 83 | | 53 | 01010011 | S |
| 84 | | 54 | 01010100 | T |
| 85 | | 55 | 01010101 | U |
| 86 | | 56 | 01010110 | V |
| 87 | | 57 | 01010111 | W |
| 88 | | 58 | 01011000 | X |
| 89 | | 59 | 01011001 | Y |
| 90 | | 5A | 01011010 | Z |
| 91 | | 5B | 01011011 | [ |
| 92 | | 5C | 01011100 | \ |
| 93 | | 5D | 01011101 | ] |
| 94 | | 5E | 01011110 | ^ |
| 95 | | 5F | 01011111 | \_ |
| 96 | | 60 | 01100000 | ` |
| 97 | | 61 | 01100001 | a |
| 98 | | 62 | 01100010 | b |
| 99 | | 63 | 01100011 | c |
| 100 | | 64 | 01100100 | d |
| 101 | | 65 | 01100101 | e |
| 102 | | 66 | 01100110 | f |
| 103 | | 67 | 01100111 | g |
| 104 | | 68 | 01101000 | h |
| 105 | | 69 | 01101001 | i |
| 106 | | 6A | 01101010 | j |
| 107 | | 6B | 01101011 | k |
| 108 | | 6C | 01101100 | l |
| 109 | | 6D | 01101101 | m |
| 110 | | 6E | 01101110 | n |
| 111 | | 6F | 01101111 | o |
| 112 | | 70 | 01110000 | p |
| 113 | | 71 | 01110001 | q |
| 114 | | 72 | 01110010 | r |
| 115 | | 73 | 01110011 | s |
| 116 | | 74 | 01110100 | t |
| 117 | | 75 | 01110101 | u |
| 118 | | 76 | 01110110 | v |
| 119 | | 77 | 01110111 | w |
|  | 120 | 78 | 01111000 | x |
|  | 121 | 79 | 01111001 | y |
|  | 122 | 7A | 01111010 | z |
|  | 123 | 7B | 01111011 | { |
|  | 124 | 7C | 01111100 | | |
|  | 125 | 7D | 01111101 | } |
|  | 126 | 7E | 01111110 | ~ |
|  | 127 | 7F | 01111111 | DEL |

1. Carilah daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap (dari buku referensi atau internet). Daftar perintah ini dapat digunakan sebagai pedoman untuk memahami program ‘boot.asm’ dan ‘kernel.asm’

|  |  |
| --- | --- |
| **Instructions** | **information** |
| DIV | Divide |
| DJNZ | Decrement and Jump if Not Zero |
| INC | Increment |
| JB | Jump if Bit Set |
| JBC | Jump if Bit Set and Clear Bit |
| JC | Jump if Carry Set |
| JMP | Jump to Address |
| JNB | Jump if Not Bit Set |
| JNC | Jump if Carry Not Set |
| JNZ | Jump if Accumulator Not Zero |
| JZ | Jump if Accumulator Zero |
| LCALL | Long Call |
| LJMP | Long Jump |
| MOV | Move from Memory |
| MOVC | Move from Code Memory |
| MOVX | Move from Extended Memory |
| MUL | Multiply |
| NOP | No Operation |
| ORL | OR Logic |
| POP | Pop Value From Stack |
| PUSH | Push Value Onto Stack |
| RET | Return From Subroutine |
| RETI | Return From Interrupt |
| RL | Rotate Left |
| RLC | Rotate Left through Carry |
| RR | Rotate Right |
| RRC | Rotate Right through Carry |
| SETB | Set Bit |
| SJMP | Short Jump |
| SUBB | Subtract With Borrow |
| SWAP | Swap Nibbles |
| XCH | Exchange Bytes |
| XCHD | Exchange Digits |
| XRL | Exclusive OR Logic |
| ACALL | Absolute Call |
| ADD | Add |
| ADDC | Add with Carry |
| AJMP | Absolute Jump |
| ANL | AND Logic |
| CJNE | Compare and Jump if Not Equal |
| CLR | Clear |
| CPL | Complement |
| DA | Decimal Adjust |
| DEC | Decrement |