

संख्या पद्धति

(Number System)

1. संख्या पद्धति (Number System)

किसी भी संख्या को निरूपित करने के लिए एक विशेष संख्या पद्धति का प्रयोग किया जाता है। प्रत्येक संख्या को संख्या पद्धति में प्रयोग किए जाने वाले अंक या अंकों के समूह से दर्शाया जाता है। प्रत्येक संख्या पद्धति का एक निश्चित आधार (Base) होता है जो उस संख्या पद्धति में प्रयोग किए जाने वाले मूल अंकों (Basic Digits) की संख्या के बराबर होता है। किसी भी संख्या में अंकों (Digits) की स्थिति दायीं से बायीं ओर गिनी जाती है। किसी संख्या में प्रत्येक अंक का मान उसके संख्यात्मक मान (Face Value) तथा स्थानीय मान (Position Value) पर निर्भर करता है। किसी संख्या का कुल मान (Value) प्रत्येक अंक के मान का योगफल होता है। दशमलव संख्या पद्धति सर्वाधिक प्राचीन और सबसे प्रचलित संख्या पद्धति है।

आधार (Base) : किसी संख्या को निरूपित करने के लिए प्रयोग की जाने वाली मूल अंकों (Basic Digits) की कुल संख्या उस संख्या पद्धति का आधार कहलाती है। उदाहरण के लिए, दशमलव संख्या पद्धति में सभी संख्याओं को 10 मूल अंकों (0, 1, 2, 3, 4, 5, 6, 7, 8 तथा 9) से निरूपित किया जाता है। अतः इसका आधार 10 है। द्विआधारी संख्या पद्धति (Binary Number System) में 2 मूल अंकों (0 तथा 1) का प्रयोग किया जाता है। अतः इसका आधार 2 है। आक्टल संख्या पद्धति (Octal Number System) में आठ मूल अंकों (0, 1, 2, 3, 4, 5, 6, तथा 7) का प्रयोग होता है, अतः इसका आधार 8 है। हेक्साडेसिमल संख्या पद्धति (Hexadecimal Number System) का आधार 16 है क्योंकि इसमें सभी संख्याओं को 16 मूल अंकों (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D तथा E) से दर्शाया जाता है।

क्या आप जानते हैं?

संख्या पद्धति में किसी संख्या के आधार को दर्शाने के लिए उसका आधार अंक संख्या के बाद नीचे छोटे अक्षरों (Sub Script) में कोष्ठक में लिख दिया जाता है। इससे उस संख्या पद्धति का पता चलता है। जैसे— $589_{(10)}$; $275_{(8)}$; $1A5_{(16)}$; $101_{(2)}$ आदि।

संख्यात्मक मान (Face Value) : किसी संख्या में किसी अंक का संख्यात्मक मान उस संख्या की स्थिति पर निर्भर करता है। संख्या में अंकों की स्थिति को दायीं से बायीं ओर गिना जाता है। सबसे दायीं ओर अर्थात् इकाई के स्थान पर स्थित अंक का संख्यात्मक मान '0' होगा। दहाई के अंक का संख्यात्मक मान '1', सैकड़े के अंक का संख्यात्मक मान '2' जबकि हजार के अंक का संख्यात्मक मान '3' होता है।

स्थानीय मान (Position Value) : किसी संख्या में किसी अंक (Digit) का स्थानीय मान संख्या के आधार (Base) तथा उसके

संख्यात्मक मान (Face Value) पर निर्भर करता है। किसी संख्या का स्थानीय मान संख्या के आधार पर संख्यात्मक मान के घात के बराबर होता है।

$$\text{स्थानीय मान} = (\text{आधार})^{\text{संख्यात्मक मान}}$$

$$(\text{Position Value} = (\text{Base})^{\text{Face Value}})$$

किसी संख्या का मान प्रत्येक अंक के संख्यात्मक मान तथा स्थानीय मान के गुणनफल का योग होता है।

$$\text{उदाहरण : संख्या} = 4206_{(10)}$$

	चौथा अंक (हजार)	तीसरा अंक (सैकड़ा)	दूसरा अंक (दहाई)	पहला अंक (इकाई)
संख्या	4	2	0	6
संख्यात्मक मान (Face Value)	3	2	1	0
स्थानीय मान (Position Value)	$10^3=1000$	$10^2=100$	$10^1=10$	$10^0=1$
संख्या का मान= अंक×स्थानीय मान	$4 \times 1000=4000$	$2 \times 100=200$	$0 \times 10=0$	$6 \times 1=6$

$$\text{संख्या का कुल मान} = 4000+200+0+6=4206_{(10)}$$

2. कम्प्यूटर में प्रयुक्त होने वाली संख्या पद्धति



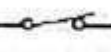
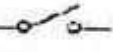


मनुष्य गणना के लिए दशमलव आधारित संख्या पद्धति (Decimal number system) का प्रयोग करता है जिसमें 0 से 9 तक (कुल 10) अंकों का प्रयोग किया जाता है। अन्य सभी अंक इन्हीं अंकों से मिलकर बनते हैं। परन्तु कम्प्यूटर दशमलव आधारित संख्या पद्धति का प्रयोग नहीं करता है। कम्प्यूटर में प्रयोग होने वाली संख्या पद्धतियाँ हैं—

- द्विआधारी संख्या पद्धति (Binary number system)
- आक्टल संख्या पद्धति (Octal number system)
- हेक्साडेसिमल संख्या पद्धति (Hexadecimal number system)

संख्या पद्धति	आधार (Base)	कुल अंक	महत्तम अंक (Highest Digit)
द्वि-आधारी	2	0, 1	1
आक्टल	8	0, 1, 2, 3, 4, 5, 6, 7	7
दशमलव	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	9
हेक्सा डेसिमल	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	F(15)

3. द्विआधारी संख्या पद्धति (Binary number System)

कम्प्यूटर एक इलेक्ट्रॉनिक मशीन है जो विद्युत धारा पर कार्य करता है। यह केवल दो ही परिस्थितियों को जान सकता है। पहला, जब परिपथ में धारा प्रवाहित हो रही है अर्थात् परिपथ का स्विच ऑन (ON) है तो इसे संकेत '1' कहा जाता है। दूसरी स्थिति में परिपथ में धारा प्रवाहित नहीं हो रही है, अर्थात् परिपथ का स्विच ऑफ (OFF) है तो इसे संकेत '0' कहा जाता है। इससे हम कह सकते हैं कि कम्प्यूटर केवल द्विआधारी संख्या पद्धति की पहचान कर सकता है। द्विआधारी (Binary) संख्या पद्धति में सभी संख्याएं दो अंक (Digits), 0 तथा 1 का प्रयोग कर लिखी जाती है। इसी कारण कम्प्यूटर को डाटा या निर्देश देने से पहले उसे 0 या 1 (ऑफ या ऑन) में बदलना पड़ता है।

द्विआधारी स्थिति	ऑन (1)	ऑफ (0)
बल्ब की स्थिति		
स्विच		
धारा		

चित्र संख्या : 7.1

स्विच	ऑन (On)	ऑफ (Off)
बल्ब	जलता है	बुझा है
धारा	प्रवाहित	नहीं प्रवाहित
बाइनरी स्थिति	1	0

बाइनरी संख्या पद्धति में इन दो अंकों 0 और 1 को बाइनरी डिजिट (Binary Digit) या संक्षेप में बिट (Bit) कहते हैं। संख्या पद्धति में किसी भी संख्या का मान उसके स्थानीय मान पर निर्भर करता है।

दशमलव पद्धति में आधार 10 होता है तथा इकाई के अंक का स्थानीय मान $10^0 = 1$ होता है, दहाई के अंक का स्थानीय मान $10^1 = 10$ तथा सैकड़ा के अंक का स्थानीय मान $10^2 = 100$ होता है। किसी अंक का कुल मान उस अंक तथा उसके स्थानीय मान के गुणनफल के बराबर होता है।

इसी प्रकार द्विआधारी संख्या पद्धति में आधार 2 होता है। इकाई के अंक का स्थानीय मान $2^0 = 1$ होता है, दहाई के अंक का स्थानीय मान $2^1 = 2$ तथा सैकड़ा के अंक का स्थानीय मान $2^2 = 4$ होता है।

"संख्याओं का स्थानीय मान"

स्थानीय मान	हजार	सैकड़ा	दहाई	इकाई
अंक पद्धति				
दशमलव (10)	$10^3=1000$	$10^2=100$	$10^1=10$	$10^0=1$
द्विआधारी (2)	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
आक्टल (8)	$8^3=512$	$8^2=64$	$8^1=8$	$8^0=1$
हेक्सा डेसिमल (16)	$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$

दशमलव के बाद की संख्याओं का स्थानीय मान

दशमलव (.)	-1	-2	-3
दशमलव बिंदु	$10^{-1}=1/10$	$10^{-2}=1/100$	$10^{-3}=1/1000$
बाइनरी बिंदु	$2^{-1}=1/2$	$2^{-2}=1/4$	$2^{-3}=1/8$
आक्टल बिंदु	$8^{-1}=1/8$	$8^{-2}=1/64$	$8^{-3}=1/512$
हेक्सा डेसिमल बिंदु	$16^{-1}=1/16$	$16^{-2}=1/256$	$16^{-3}=1/4096$

इसी स्थानीय मान के आधार पर एक संख्या पद्धति के अंकों का परिवर्तन दूसरी संख्या पद्धति में किया जाता है।

3. बाइनरी अंकगणित (Binary Arithmetic)

जब बाइनरी अंकों पर साधारण अंकगणितीय गणनाएं जैसे— जोड़, घटाव, गुणा व भाग (Addition, Subtraction, Multiplication and Division) आदि संपादित की जाती है, तो इसे बाइनरी अंकगणित कहा जाता है।

बाइनरी अंकगणित का उपयोग कम्प्यूटर की गणना के लिए सर्किट तैयार करने तथा उनके सरलीकरण में किया जाता है।

3.1. दशमलव का द्विआधारी में परिवर्तन (Conversion from Decimal to Binary) : दशमलव को द्विआधारी में बदलने के लिए दशमलव संख्या को 2 से भाग देते हैं। भागफल को नीचे लिखकर शेष को अलग लिखते हैं जो द्विआधारी संख्या का

क्या आप जानते हैं ?

दशमलव पद्धति को किसी अन्य संख्या पद्धति में बदलने के लिए 'भाग शेष विधि' (Division Remainder Method) का प्रयोग किया जाता है। इसमें—

- दशमलव संख्या को नये आधार वाली संख्या से विभाजित करते हैं।
- भागफल को नीचे लिखकर शेष को अलग लिखते हैं।
- भागफल को पुनः विभाजित करते हैं जब तक कि भागफल शून्य न रह जाए।
- अंतिम शेष (बायां अंक-MSD) से शुरू कर प्रथम शेष (दायां अंक-LSD) तक संख्या लिखते हैं जो नये आधार में संख्या को निरूपित करता है।

सबसे दायां अंक (LSD-Least Significant Digit) होता है। पुनः भागफल में 2 से भाग देते हैं और यह प्रक्रिया तब तक दुहरायी जाती है जब तक भागफल 0 न रह जाय। अंतिम शेष द्विआधारी संख्या का सबसे बायां अंक (MSD-Most Significant Digit) होता है। प्रत्येक स्थिति में लिखी गई शेष संख्या नीचे (MSD) से ऊपर (LSD) की ओर द्विआधारी संख्या को निरूपित करता है।

उदाहरण : $25_{(10)}$ को द्विआधारी में बदलें।

हल :

2	25	शेष	
2	12	1	
2	6	0	
2	3	0	
2	1	1	
	0	1	

दायां अंक (LSD)
↑
बायां अंक (MSD)

$$25_{(10)} = 11001_{(2)}$$

दशमलव अंकों के बाइनरी तुल्यांक

दशमलव	बाइनरी
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

3.2 द्विआधारी का दशमलव में परिवर्तन (Conversion from Binary to Decimal) : द्विआधारी अंकों को दशमलव में परिवर्तित करने के लिए उसके अंकों के मान को स्थानीय मान से गुणा कर उन्हें जोड़ दिया जाता है।

उदाहरण : $10101_{(2)}$ को दशमलव में बदलें।

हल : संख्या 1 0 1 0 1

स्थानीय मान $2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

$$\begin{aligned} 10101_{(2)} &= (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) \\ &\quad + (0 \times 2^1) + (1 \times 2^0) \\ &= (1 \times 16) + (0 \times 8) + (1 \times 4) \\ &\quad + (0 \times 2) + (1 \times 1) \end{aligned}$$

$$= 16 + 0 + 4 + 0 + 1$$

$$= 21_{(10)}$$

3.3 दशमलव भिन्न का बाइनरी में परिवर्तन (Conversion of Decimal Fraction to Binary Fraction) : दशमलव भिन्न को 2 से गुणा करते हैं। गुणनफल में पूर्ण संख्या को अलग लिखते हैं जो बाइनरी भिन्न का बायां अंक (MSD) होता है। भिन्न को पुनः 2 से गुणा करते हैं और यह तब तक दुहराते हैं जब तक भिन्न शून्य न हो जाय या बाइनरी भिन्न के आवश्यक अंक पूरे न हो जाए।

अंतिम पूर्णांक बाइनरी भिन्न का दायां अंक (LSD) होता है। दशमलव के बाद बाइनरी भिन्न को ऊपर से नीचे की ओर लिखा जाता है।

उदाहरण : $0.8125_{(10)}$ को बाइनरी में बदलें

हल : $0.8125 \times 2 = 1.625 \rightarrow 1$ (MSD)

$$0.625 \times 2 = 1.250 \rightarrow 1$$

$$0.250 \times 2 = 0.500 \rightarrow 0$$

$$0.500 \times 2 = 1.000 \rightarrow 1$$
 (LSD)

$$\text{अतः } 0.8125_{(10)} = 0.1101_{(2)}$$

3.4 बाइनरी भिन्न का दशमलव भिन्न में परिवर्तन (Conversion of Binary Fraction to Decimal Fraction) : बाइनरी बिंदु के बाद के अंकों को उनके स्थानीय मान से गुणा कर जोड़ देते हैं।

उदाहरण : $0.101_{(2)}$ को दशमलव में बदलें

$$\text{हल : } .101_{(2)} = (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$$

$$= (1 \times 1/2) + (0 \times 1/4) + (1 \times 1/8)$$

$$= (1 \times 0.5) + (0 \times 0.25) + (1 \times 0.125)$$

$$= .5 + 0 + .125$$

$$= .625_{(10)}$$

$$\text{अतः } 0.101_{(2)} = 0.625_{(10)}$$

क्या आप जानते हैं ?

आक्टल और हेक्साडेसिमल संख्या पद्धति का विकास बाइनरी अंकों का सरलता से कोडिंग करने तथा बड़ी संख्याओं को आसानी से लिखने के लिए किया गया है।

4. आक्टल संख्या पद्धति (Octal number system)

इसमें आधार 8 होता है तथा 0, 1, 2, 3, 4, 5, 6 और 7 (कुल 8) संख्याओं का प्रयोग किया जाता है। आक्टल संख्या पद्धति में सभी संख्याएं इन्हीं आठ अंकों (digits) का प्रयोग कर लिखी जाती हैं। जिस प्रकार, दशमलव संख्या पद्धति में 9 के बाद का अंक लिखने के लिए सबसे छोटी संख्या 1 के दायां ओर 0 लिखा जाता है, उसी प्रकार, आक्टल संख्या पद्धति में 7 के बाद का अंक लिखने के लिए सबसे छोटी संख्या 1 के दायां ओर 0 लिखा जाता है। अतः आक्टल संख्या पद्धति में 10 का मूल्य दशमलव संख्या पद्धति में 8 के बराबर होता है।

4.1 दशमलव का आक्टल में परिवर्तन (Conversion from Decimal to Octal) : इसमें भाग शेष विधि का प्रयोग किया जाता है।

उदाहरण : $828_{(10)}$ को आक्टल में बदले

हल :	8	828	शेष
	8	103	4
	8	12	7
	8	1	4
		0	1

दायाँ अंक (LSD) ↑
बायाँ अंक (MSD)

अतः $828_{(10)} = 1474_{(8)}$

4.2 आक्टल का दशमलव में परिवर्तन (Conversion from Octal to Decimal) : अंकों को उनके स्थानीय मान से गुणा कर जोड़ देते हैं।

उदाहरण : $1672_{(8)}$ को दशमलव में बदलें

$$\begin{aligned} \text{हल : } 1672_{(8)} &= (1 \times 8^3) + (6 \times 8^2) + (7 \times 8^1) + (2 \times 8^0) \\ &= (1 \times 512) + (6 \times 64) + (7 \times 8) + (2 \times 1) \\ &= 512 + 384 + 56 + 2 \\ &= 954_{(10)} \end{aligned}$$

अतः $1672_{(8)} = 954_{(10)}$

4.3 आक्टल का बाइनरी में परिवर्तन (Conversion from Octal to Binary)

उदाहरण : $704_{(8)}$ को बाइनरी में बदलें।

हल : प्रथम विधि : आक्टल संख्या को दशमलव में बदलते हैं और फिर दशमलव को बाइनरी में बदलते हैं।

$$\begin{aligned} 704_{(8)} &= (7 \times 8^2) + (0 \times 8^1) + (4 \times 8^0) \\ &= 7 \times 64 + 0 + 4 \times 1 \\ &= 448 + 4 = 452_{(10)} \end{aligned}$$

पुनः,

2	452	शेष
2	226	0
2	113	0
2	56	1
2	28	0
2	14	0
2	7	0
2	3	1
2	1	1
2	0	1

(LSD) ↑
(MSD)

$704_{(8)} = 452_{(10)} = 111000100_{(2)}$

दूसरी विधि (संक्षिप्त विधि) : आक्टल संख्याओं को उनके तीन अक्षरों के बाइनरी अंक समूह तुल्यांक से प्रतिस्थापित कर देते हैं।

$704_{(8)} = 111, 000, 100_{(2)}$

आक्टल संख्या	तीन अंकों का बाइनरी तुल्यांक
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

4.4. बाइनरी का आक्टल में परिवर्तन (Conversion from Binary to Octal)

उदाहरण : $10110_{(2)}$ को आक्टल में बदलें।

हल : प्रथम विधि : बाइनरी को दशमलव में बदलें और फिर दशमलव को भाग शेष विधि द्वारा आक्टल में बदलें।

$$\begin{aligned} 10110_{(2)} &= (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) \\ &= 16 + 0 + 4 + 2 + 0 \\ &= 22_{(10)} \end{aligned}$$

पुनः,

8	22	शेष (LSD)
8	2	6
	0	2

↑ $22_{(10)} = 26_{(8)}$
(MSD)

अतः $10110_{(2)} = 22_{(10)} = 26_{(8)}$

दूसरी विधि (संक्षिप्त विधि) : दायाँ से शुरू कर तीन बाइनरी अंकों का समूह बना लेते हैं तथा उनका आक्टल तुल्यांक प्रतिस्थापित कर देते हैं। सबसे बायीं ओर के समूह को तीन अंकों का बनाने के लिए बायीं ओर शून्य जोड़ देते हैं।

$$\begin{aligned} 10 \ 110_{(2)} &= 010, 110_{(2)} \\ &= 2 \quad 6 = 26_{(8)} \end{aligned}$$

5. हेक्साडेसिमल संख्या पद्धति (Hexadecimal Number System)

इसमें 16 मूल अंकों का प्रयोग किया जाता है, जिसमें 0 से 9 तक अंक तथा A, B, C, D, E और F अक्षर होते हैं। इसमें बाइनरी अंकों का परिवर्तन चार बाइनरी समूहों में किया जाता है। इससे डाटा स्थानांतरण की गति तेज होती है।

हेक्सा डेसिमल (16)	डेसिमल (10)	बाइनरी (2)	आक्टल (8)
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7
8	8	1000	10
9	9	1001	11
A	10	1010	12
B	11	1011	13
C	12	1100	14
D	13	1101	15
E	14	1110	16
F	15	1111	17
10	16	00010000	20

5.1 हेक्साडेसिमल का दशमलव में परिवर्तन (Conversion from Hexadecimal to Decimal) : हेक्साडेसिमल अंकों को उनके स्थानीय मान से गुणा कर जोड़ देते हैं।

उदाहरण : $124_{(16)}$ को दशमलव में बदलें।

$$124_{(16)} = (1 \times 16^2) + (2 \times 16^1) + (4 \times 16^0) \\ = 256 + 32 + 4 = 292_{(10)}$$

अतः $124_{(16)} = 292_{(10)}$

उदाहरण : $1AC_{(16)}$ को दशमलव में बदलें।

$$1AC_{(16)} = (1 \times 16^2) + (A \times 16^1) + (C \times 16^0) \\ = (1 \times 256) + (10 \times 16) + (12 \times 1) \\ = 256 + 160 + 12 \\ = 428_{(10)}$$

5.2. दशमलव का हेक्साडेसिमल में परिवर्तन (Conversion from Decimal to Hexadecimal)

भाग शेष विधि द्वारा

उदाहरण : $431_{(10)}$ को हेक्साडेसिमल में बदलें।

16	431	शेष
16	26	15 = F (LSD)
16	1	10 = A
	0	1 (MSD)

$$431_{(10)} = 1AF_{(16)}$$

5.3 हेक्साडेसिमल का बाइनरी में परिवर्तन (Conversion from Hexadecimal to Binary)

उदाहरण : $12D_{(16)}$ को बाइनरी में बदलें।

हल : प्रथम विधि : पहले हेक्साडेसिमल को डेसिमल में बदले और फिर डेसिमल को बाइनरी में बदलें।

$$12D_{(16)} = (1 \times 16^2) + (2 \times 16^1) + (D \times 16^0) \\ = 1 \times 256 + 2 \times 16 + 13 \times 1 \\ = 256 + 32 + 13 \\ = 301_{(10)}$$

पुनः

2	301	शेष
2	150	1
2	75	0
2	37	1
2	18	1
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

LSD
↑
MSD

$$301_{(10)} = 1,0010, 1101_{(2)}$$

दूसरी विधि (संक्षिप्त विधि) : हेक्साडेसिमल अंकों को चार अंकीय बाइनरी तुल्यांक से प्रतिस्थापित कर दिया जाता है।

$$= 1 \quad 2 \quad D$$

$$12D_{(16)} = 0001, 0010, 1101_{(2)}$$

5.4. बाइनरी का हेक्साडेसिमल में परिवर्तन (Conversion from Binary to Hexadecimal)

उदाहरण : $1011011_{(2)}$ को हेक्साडेसिमल में बदलें।

हल : प्रथम विधि : बाइनरी को डेसिमल में बदले और फिर डेसिमल को हेक्साडेसिमल में बदलें।

$$1011011_{(2)} = (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) \\ + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) \\ + (1 \times 2^0) \\ = 64 + 0 + 16 + 8 + 0 + 2 + 1 \\ = 91_{(10)}$$

पुनः

16	91	शेष
16	5	11 = B (LSD)
16	0	5 (MSD)

अतः $91_{(10)} = 5B_{(16)}$

दूसरी विधि (संक्षिप्त विधि) : दायें से प्रारंभ कर बाइनरी अंकों को चार के समूह में विभाजित करते हैं। अंतिम समूह को चार का बनाने के लिए आवश्यकता पड़ने पर बायीं ओर '0' लिख देते हैं। अब बाइनरी के 4 अंकों के तुल्यांक हेक्साडेसिमल अंक लिख देते हैं।

$1011011^{(2)} = 0101, 1011$

$= 5 \quad B = 5B_{(16)}$

6. बाइनरी अंकगणित (Binary Arithmetic)

6.1. बाइनरी जोड़ (Binary Addition) : बाइनरी जोड़ सामान्य जोड़ की तरह होता है पर इसमें निम्नलिखित जोड़ नियमों को ध्यान में रखा जाता है।

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$ तथा हासिल (Carry) 1

बाइनरी जोड़ का उदाहरण—

बाइनरी संख्या 110 (6_{10}) तथा 101 (5_{10}) को जोड़ें

बाइनरी	दशमलव
$\begin{array}{r} 110 \\ + 101 \\ \hline 1011 \end{array}$	$\begin{array}{r} 6 \\ + 5 \\ \hline 11 \end{array}$
1 Carry ←	Carry 1 ←

6.2. बाइनरी घटाव (Binary Subtraction) : यह साधारण घटाव की तरह होता है जिसमें निम्नलिखित बिन्दुओं को ध्यान में रखा जाता है।

$1 - 1 = 0$

$0 - 0 = 0$

$1 - 0 = 1$

$0 - 1 = 1$ (0 द्वारा बायीं ओर से उधार लेने पर वह 10 बन जाता है तथा $10 - 1 = 1$ क्योंकि $1 + 1 = 10$ होता है)

बाइनरी घटाव का उदाहरण—

बाइनरी संख्या 101 (5_{10}) में से 10 (2_{10}) को घटाये

बाइनरी	दशमलव
$\begin{array}{r} 101 \\ - 10 \\ \hline 011 \end{array}$	$\begin{array}{r} 5 \\ - 2 \\ \hline 3 \end{array}$
Borrow 10	

रोचक तथ्य

अधिकांश कम्प्यूटर गुणा या भाग करने के लिए बार बार जोड़ने या बार बार घटाने का सहारा लेते हैं। इसे गुणन या भाग की योगात्मक विधि (Additive Method of Multiplication or Division) कहा जाता है। इससे हार्डवेयर की बचत होती है।

6.3. बाइनरी गुणा (Binary Multiplication) : यह साधारण गुणा की तरह है। लेकिन जोड़ करते समय बाइनरी नियमों का पालन किया जाता है।

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 1$

बाइनरी गुणा का उदाहरण—

बाइनरी संख्या 110 (6_{10}) को 10 (2_{10}) से गुणा करें।

बाइनरी	दशमलव
$\begin{array}{r} 110 \\ \times 10 \\ \hline 000 \\ + 110 \\ \hline 1100 \end{array}$	$\begin{array}{r} 6 \\ \times 2 \\ \hline 12 \end{array}$

6.4. बाइनरी भाग (Binary Division) : यह साधारण भाग की तरह है लेकिन घटाव करते समय बाइनरी नियमों का पालन किया जाता है।

जैसे—

$0 \div 1 = 0$

$1 \div 1 = 1$

क्या आप जानते हैं ?

एक बाइट में आठ बिट होते हैं। किसी बिट का मान 0 या 1 हो सकता है। इस तरह एक बाइट से $2^8 = 256$ कैरेक्टर निरूपित किये जा सकते हैं।

7. बाइनरी मेमोरी (Binary Memory)

बिट (Bit) : यह कम्प्यूटर मेमोरी का सबसे छोटा भाग है। यह बाइनरी डिजिट (Binary digit) का संक्षिप्त रूप है। इसे 0 या 1 (आफ या ऑन) में व्यक्त किया जाता है। कम्प्यूटर में प्रत्येक डाटा, अनुदेशों तथा परिणामों को बाइनरी डिजिट या बिट में निरूपित और स्टोर किया जाता है। अतः, बिट मेमोरी को प्रदर्शित करने वाली सबसे छोटी इकाई है।

क्या आप जानते हैं ?

किसी संख्या का 0 से विभाजन स्वीकार्य नहीं है। ऐसा करने पर कम्प्यूटर शून्य से विभाजन (Division by Zero) की त्रुटि (Error) देता है।

निबल (Nibble) : चार बिट के समूह को निबल कहा जाता है। यह आधे बाइट के बराबर होता है।

बाइट (Byte) : आठ बिट या दो निबल के समूह को एक बाइट कहा जाता है। कम्प्यूटर मेमोरी में किसी अक्षर या कैरेक्टर को दर्शाने के लिए कम से कम आठ बिट अर्थात एक बाइट की जरूरत पड़ती है। एक खाली स्थान (Space) भी 1 बाइट जगह घेरता है। कम्प्यूटर मेमोरी को बाइट में ही मापा जाता है। बाइट मेमोरी की वह सबसे छोटी इकाई है जिसके द्वारा किसी अंक, अक्षर या चिह्न को निरूपित किया जा सकता है। एक बाइट (8 बिट) द्वारा कुल 256 ($2^8=256$) अलग-अलग कैरेक्टर निरूपित किए जा सकते हैं।

शब्द की लम्बाई (Word Length) : कम्प्यूटर शब्द की लंबाई एक कम्प्यूटर से दूसरे कम्प्यूटर में भिन्न हो सकती है जबकि किसी एक कम्प्यूटर के लिए यह निश्चित होती है। कम्प्यूटर शब्द की लम्बाई कम्प्यूटर के हार्डवेयर पर निर्भर करता है। विभिन्न प्रकार के कम्प्यूटर में शब्द की लंबाई 1 बिट से 64 बिट तक हो सकती है। सुपर कम्प्यूटर में शब्द लंबाई का परास 64 बिट होता है।

7.1 कम्प्यूटर मेमोरी की माप

- 1 निबल = 4 बिट
- 1 बाइट = 8 बिट = 2 निबल
- 1 किलोबाइट (KB) = 2^{10} बाइट
= 1024 बाइट $\cong 10^3$ बाइट
- 1 मेगाबाइट (MB) = 2^{10} किलोबाइट
= 1024 किलोबाइट
= 2^{20} बाइट $\cong 10^6$ बाइट
= 1024×1024 बाइट
= 10,48,576 बाइट
= 10 लाख बाइट
= 1 मिलियन (Million) बाइट
- 1 गीगा बाइट (GB) = 2^{10} मेगाबाइट
= 1024 मेगाबाइट
= 1024×1024 किलोबाइट
= 2^{20} किलोबाइट $\cong 10^9$ बाइट
= 10,48,576 किलोबाइट
= 2^{30} बाइट
= 1,07,37,41,424 बाइट
- 1 टेराबाइट (TB) = 1024 गीगा बाइट
= 2^{40} बाइट $\cong 10^{12}$ बाइट
- 1 पेटाबाइट (PB) = 1024 टेराबाइट
= 2^{50} बाइट $\cong 10^{15}$ बाइट
- 1 एक्साबाइट (EB) = 1024 पेटाबाइट
= 2^{60} बाइट $\cong 10^{18}$ बाइट
- 1 जेटाबाइट (ZB) = 1024 एक्सा बाइट
= 2^{70} बाइट $\cong 10^{21}$ बाइट

- 1 योद्वाबाइट (YB) = 1024 जेटा बाइट
= 2^{80} बाइट $\cong 10^{24}$ बाइट
- 1024 बाइट = 1 किलो बाइट (KB-Kilo Byte)
- 1024 किलोबाइट = 1 मेगा बाइट (MB-Mega Byte)
- 1024 मेगाबाइट = 1 गीगा बाइट (GB-Giga Byte)
- 1024 गीगाबाइट = 1 टेरा बाइट (TB-Tera Byte)
- 1024 टेराबाइट = 1 पेटा बाइट (PB-Peta Byte)
- 1024 पेटाबाइट = 1 एक्सा बाइट (EB-Exa Byte)
- 1024 एक्साबाइट = 1 जेटा बाइट (ZB-Zetta Byte)
- 1024 जेटाबाइट = 1 योद्वा बाइट (YB-Yotta Byte)

रोचक तथ्य

- एक पेटाबाइट मेमोरी में स्टोर किए गए MP₃ फॉर्मेट संगीत को सुनने में 2000 साल लगेंगे।
- 2013 में World Wide Web पर उपलब्ध सभी दस्तावेजों का कुल आकार 4 जेटाबाइट था।

8. कम्प्यूटर कोड (Computer Codes)

कम्प्यूटर में डाटा अक्षरों (Alphabets), विशेष चिह्नों (Special Characters) तथा अंकों (Numeric) में हो सकता है। अतः इन्हें अल्फान्यूमेरिक डाटा (Alphanumeric Data) कहा जाता है। डाटा में प्रत्येक अक्षर, चिह्न या अंक को एक विशेष कोड द्वारा व्यक्त किया जाता है।

8.1. बाइनरी कोडेड डेसिमल (BCD-Binary Coded Decimal) : इसमें संपूर्ण डेसिमल संख्या को बाइनरी में बदलने की बजाय डेसिमल संख्या के प्रत्येक अंक को उसके चार अंकीय बाइनरी तुल्यांक से प्रतिस्थापित कर दिया जाता है। इसे 4-बिट बीसीडी कोड (4 Bit BCD Code) कहा जाता है।

क्या आप जानते हैं ?

बुलियन अलजेबरा का आविष्कार ब्रिटेन के गणितज्ञ जार्ज बूले (George Boole) ने किया। इन्हीं के नाम पर इसे बुलियन अलजेबरा कहा गया।

8.2. आस्की (ASCII-American Standard Code for Information Interchange) : आस्की (ASCII) एक लोकप्रिय कोडिंग सिस्टम है जिसका प्रारंभ आन्सी (ANSI-American National Standards Institute) द्वारा 1963 में किया गया। इसमें एक कैरेक्टर के लिए 8 बिट और तीव्र निरूपण के लिए हेक्साडेसिमल संख्या पद्धति का प्रयोग किया गया। कम्प्यूटर के की-बोर्ड में प्रयुक्त प्रत्येक कैरेक्टर के लिए एक विशेष आस्की कोड निर्धारित किया गया है। इसमें एक कैरेक्टर के लिए 8 बिट का प्रयोग किया जाता है।

8.3. यूनिकोड (Unicode-Universal Code)

कम्प्यूटर के बढ़ते व्यवहार तथा अलग-अलग भाषाओं में कम्प्यूटर के उपयोग ने एक सार्वत्रिक कोड की आवश्यकता को जन्म दिया जिसमें संसार के प्रत्येक कैरेक्टर के लिए एक अलग कोड निर्धारित हो ताकि

प्रत्येक भाषा, प्रत्येक प्रोग्राम तथा प्रत्येक साफ्टवेयर में उसका प्रयोग किया जा सके। इसके लिए यूनिकोड की व्यवस्था की गई जिसमें एक लाख कैरेक्टर के निरूपण की क्षमता है। यूनिकोड विश्व की सभी भाषाओं में प्रयुक्त अक्षरों, अंकों तथा चिह्नों के लिए एक विशेष कोड निर्धारित करता है। यूनिकोड में प्रयुक्त पहले 256 कैरेक्टर का निरूपण आस्की कोड के समान ही है। इसमें प्रत्येक कैरेक्टर को 32 बिट में निरूपित किया जाता है। यूनिकोड में तीन प्रकार की व्यवस्था प्रयोग में लायी जाती है—

- यूटीएफ-8 (UTF-8- Unicode Transformation Format-8)-यूटीएफ-8 फॉर्मेट में समस्त यूनिकोड अक्षरों को एक, दो, तीन या चार बाइट के कोड में बदला जाता है।
- यूटीएफ-16 (UTF-16)-इस फॉर्मेट में यूनिकोड अक्षरों को एक या दो शब्दों (1 शब्द = 16 बिट) के कोड में बदला जाता है। अतः इसे Word Oriented Format भी कहते हैं।
- यूटीएफ-32 (UTF-32)-इस कोड में समस्त अक्षरों को दो शब्दों (Words) यानी 32 बिट के यूनिकोड में बदला जाता है।

क्या आप जानते हैं?

Microsoft Word प्रोग्राम में Insert Symbol कमांड का प्रयोग कर किसी Symbol या Character के लिए प्रयुक्त Unicode देख सकते हैं।

9. बुलियन अलजेबरा (Boolean Algebra)

बाइनरी संख्या पद्धति में प्रयोग किये जाने वाले गणित, जिसमें केवल दो चर (Variable), 0 और 1 का प्रयोग किया जाता है, बुलियन अलजेबरा कहलाता है। इसका उपयोग कम्प्यूटर में प्रयुक्त लॉजिक सर्किट (Logic Circuit) को सरल बनाने के लिए किया जाता है।

9.1. लॉजिक गेट (Logic Gate) : यह एक इलेक्ट्रॉनिक परिपथ (Circuit) है जो एक या अधिक इनपुट लेकर मानक आउटपुट देता है। कम्प्यूटर में सभी परिपथ का निर्माण लॉजिक गेट से ही किया जाता है। कम्प्यूटर में स्थित बाइनरी डाटा को लॉजिक गेट की सहायता से ही प्रोसेस किया जाता है। किसी लॉजिक गेट का Truth Table यह बताता है कि इनपुट की विभिन्न संभावनाओं के लिए लॉजिक गेट का आउटपुट क्या होगा। प्रत्येक लॉजिक गेट को एक विशेष चिह्न (Symbol) द्वारा दर्शाया जाता है। AND, OR तथा NOT गेट Basic Logic Gate हैं। अन्य लॉजिक गेट हैं—NAND, NOR, XOR, XNOR.

9.2. और गेट (OR Gate) : OR गेट का प्रयोग बुलियन जोड़ (+) के लिए किया जाता है। इसे लॉजिकल एडिशन (Logical Addition) कहते हैं जिसे '+' चिह्न या 'OR' ऑपरेटर द्वारा निरूपित किया जाता है। इसमें कोई भी इनपुट 1 होने पर आउटपुट 1 होता है। आउटपुट 0 तभी होता है जब सभी इनपुट 0 हों। यह

समानान्तर में जुड़े दो या अधिक स्विच की तरह है। कोई भी स्विच ऑन (1) होने पर आउटपुट सिग्नल प्राप्त होगा।

क्या आप जानते हैं?

NAND तथा NOR गेट यूनिवर्सल बिल्डिंग ब्लॉक (Universal Building Block) कहलाते हैं क्योंकि ये किसी भी प्रकार के कम्प्यूटर परिपथ के निर्माण में सक्षम हैं।

A

इनपुट

आउटपुट

B

A

B

$C = A + B$ या $A \text{ OR } B$

OR गेट का चिह्न

इनपुट		आउटपुट
A	B	$C = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table of OR Gate

9.3. एण्ड गेट (AND Gate) : एण्ड गेट का प्रयोग बुलियन गुणा (.) के लिए किया जाता है। इसमें आउटपुट 1 तभी होता है जब सभी इनपुट 1 हों। किसी भी इनपुट के शून्य होने पर आउटपुट 0 होता है। इसे सीरीज में लगे दो या अधिक स्विच की तरह समझा जा सकता है। इसे लॉजिकल गुणा (Logical Multiplication) कहा जाता है। इसे निरूपित करने के लिए * चिह्न या 'AND' ऑपरेटर का प्रयोग किया जाता है।

इनपुट

आउटपुट

A B

A

B

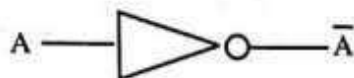
$C = A . B$ या $A \text{ AND } B$

AND Gate Symbol

इनपुट		आउटपुट
A	B	$C = A . B$
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table of AND Gate

9.4. नॉट गेट (NOT Gate) : यह इनपुट के विपरीत आउटपुट देता है। इनपुट 1 होने पर आउटपुट 0 तथा इनपुट 0 होने पर आउटपुट 1 होता है। इसे इनवर्टर (Inverter) या काम्प्लीमेंट (Complement) ऑपरेशन भी कहते हैं। इसे निरूपित करने के लिए '-' चिह्न या 'NOT' ऑपरेशन का प्रयोग किया जाता है।

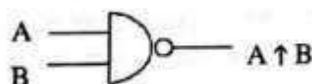
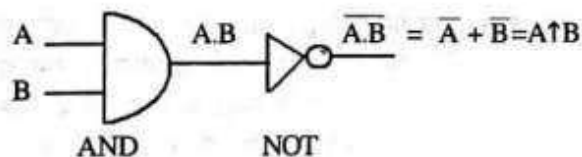


NOT Gate Symbol

इनपुट	आउटपुट
A	\bar{A}
0	1
1	0

Truth Table of NOT Gate

9.5. नैन्ड गेट (NAND Gate) : यह एक पूरक एंड गेट (Complementary AND Gate) है जो AND गेट के विपरीत परिणाम देता है। यह AND गेट के साथ जुड़े NOT गेट (AND+NOT) की तरह कार्य करता है। इसमें किसी भी इनपुट के शून्य होने पर आउटपुट 1 होता है। आउटपुट शून्य तभी होता है जब सभी इनपुट 1 हो। NAND ऑपरेशन को '↑' चिह्न द्वारा दर्शाते हैं।

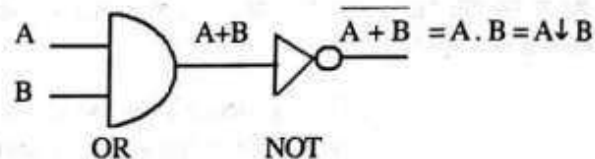


NAND Gate Symbol

A	B	$C = A \uparrow B = A \text{ AND } B$
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table of NAND Gate

9.6. नॉर गेट (NOR Gate) : यह पूरक ऑर गेट (Complementary OR Gate) है जो OR गेट के विपरीत परिणाम देता है। यह OR गेट से जुड़े NOT गेट (OR + NOT) गेट की तरह कार्य करता है। इसमें आउटपुट 1 तभी होता है जब सभी इनपुट 0 हो। किसी भी इनपुट के 1 होने पर आउटपुट 0 होता है। NOR ऑपरेशन को '↓' चिह्न द्वारा दर्शाते हैं।



NOR Gate Symbol

A	B	$C = A \downarrow B (A \text{ NOR } B)$
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table of NOR Gate

गेट का नाम	क्रिया	आउटपुट
OR	जोड़ (+)	कोई भी इनपुट 1 होने पर आउटपुट 1 होता है। आउटपुट 0 तभी होता है जब सभी इनपुट 0 हो।
AND	गुणा (*)	सभी इनपुट 1 होने पर ही आउटपुट 1 होता है।
NOT	विपरीत (A)	इनपुट 1 होने पर आउटपुट 0 तथा इनपुट 0 होने पर आउटपुट 1 होता है।
NAND	AND+NOT	सभी इनपुट 1 होने पर आउटपुट 0 होता है। किसी भी इनपुट के शून्य होने पर आउटपुट 1 होता है।
NOR	OR+NOT	कोई भी इनपुट 1 होने पर आउटपुट 0 होता है। सभी इनपुट 0 होने पर आउटपुट 1 होता है।

वस्तुनिष्ठ प्रश्न

(Objective Question)

- कम्प्यूटर की स्मृति सामान्य तौर पर किलोबाइट या मेगाबाइट के रूप में व्यक्त की जाती है। एक बाइट बना होता है—
(IAS/Pre/2000)

(a) आठ द्विआधारी अंकों का

(b) आठ दशमलव अंकों का

(c) दो द्विआधारी अंकों का

(d) दो दशमलव अंकों का

Ans. (a)

व्याख्या : कम्प्यूटर मेमोरी का सबसे छोटा भाग बिट (Bit) कहलाता है जो द्विआधारी अंक (Binary Digit) का संक्षिप्त रूप है जिसे 0 या 1 से व्यक्त किया जाता है। आठ बिट के समूह को बाइट (Byte) कहा जाता है जो कम्प्यूटर मेमोरी में किसी एक अक्षर या कैरेक्टर को दर्शाता है। 1024 बाइट एक किलोबाइट के तथा 1024 किलोबाइट एक मेगाबाइट के बराबर होता है।

2. एक किलोबाइट बराबर होता है (Uttanchal PSC/Pre/2005
Uttanchal PSC/Mains/2002)

- (a) 1000 बाइट्स (b) 1024 बाइट्स
(c) 1042 बाइट्स (d) 1 किलोग्राम बाइट्स

Ans. (b)

व्याख्या : कम्प्यूटर मेमोरी को द्विआधारी पद्धति में मापते हैं। इस कारण एक किलोबाइट = 2^{10} बाइट = 1024 बाइट होता है।

3. कम्प्यूटर भाषा में एक मेगाबाइट में कितने बाइट होते हैं—
(MPPSC/Pre/2005)

- (a) 1,00,000 (b) 10,00,000
(c) 10,24,000 (d) 10,48,576

Ans. (d)

व्याख्या : 1 मेगाबाइट (MB) = 2^{10} किलोबाइट
= 1024 किलोबाइट
= 1024×1024 बाइट
= 2^{20} बाइट
= 10,48,576 बाइट

4. आठ बिटों के समूह को कहते हैं (Utt/PSC/Mains/2002)

- (a) निबल (b) बाइट
(c) वर्ड (शब्द) (d) किलोबाइट

Ans. (b)

व्याख्या : 4 बिट = 1 निबल

8 बिट = 1 बाइट = 2 निबल

2^{10} बाइट = 1024 बाइट = 1 किलोबाइट

5. एक बाइट से कितने मूल्य निरूपित किये जा सकते हैं—
(SBI/Clk/2008)

- (a) 4 (b) 16
(c) 64 (d) 256

Ans. (d)

व्याख्या : एक बाइट में आठ बिट होते हैं। एक बिट 0 या 1 से बना होता है। इस तरह, एक बाइट से कुल $2^8 = 256$ मूल्यों को निरूपित किया जा सकता है।

6. इनमें से कौन कम्प्यूटर में प्रयुक्त अंक पद्धति है—

- (a) द्विआधारी (Binary)
(b) आक्टल (Octal)
(c) हेक्साडेसिमल (Hexadecimal)
(d) उपर्युक्त तीनों

Ans. (d)

व्याख्या : हालांकि कम्प्यूटर केवल द्विआधारी अंक पद्धति को समझ सकता है, पर परिवर्तन में सरलता के लिए आक्टल और हेक्साडेसिमल अंक पद्धति का भी प्रयोग किया जाता है।

7. बिट (Bit) क्या है ?

- (a) एक अंक पद्धति
(b) कम्प्यूटर मेमोरी की सबसे छोटी इकाई
(c) एक इनपुट डिवाइस
(d) इनमें से कोई नहीं

Ans. (b)

व्याख्या : बिट (Bit) बाइनरी डिजिट (Binary digit) का संक्षिप्त रूप है। हालांकि बाइनरी एक अंक पद्धति है, पर बिट इसकी सबसे छोटी इकाई है।

8. कम्प्यूटर में शब्द की लम्बाई को मापा जाता है—

- (a) बाइट (b) किलोग्राम
(c) मीटर (d) इनमें से कोई नहीं

Ans. (a)

व्याख्या : शब्द की लम्बाई को बाइट में मापा जाता है। एक शब्द की लम्बाई 1 से 64 बाइट तक हो सकती है। शब्द की लम्बाई जितनी अधिक होगी, कम्प्यूटर का गणना समय उतना ही कम होगा।

9. कम्प्यूटर मेमोरी की क्षमता को मापते हैं—

- (a) बिट (b) बाइट
(c) मीटर (d) किलोग्राम

Ans. (b)

व्याख्या : कम्प्यूटर मेमोरी को बाइट में मापते हैं। अधिक मेमोरी को किलोबाइट, मेगाबाइट या टेराबाइट में मापते हैं।

10. बिट (Bit) का मतलब है—

- (a) बाइनरी डिजिट (b) बाइनरी नम्बर
(c) कम्प्यूटर एक भाग (d) इनमें से कोई नहीं

Ans. (a)

व्याख्या : बिट (Bit) बाइनरी डिजिट का संक्षिप्त रूप है जिसे Binary का Bi तथा Digit का t लेकर Bit बनाया गया है।

11. प्रत्येक कम्प्यूटर के की-बोर्ड के प्रत्येक कैरेक्टर की ASCII होती है जिसका पूर्ण रूप है— (UGC/Net/JRF/2007)

- (a) अमेरिकन स्टॉक कोड फॉर इन्फार्मेशन इंटरचेंज
- (b) अमेरिकन स्टैंडर्ड कोड फॉर इन्फार्मेशन इंटरचेंज
- (c) अफ्रीकन स्टैंडर्ड कोड फॉर इन्फार्मेशन इंटरचेंज
- (d) एडाप्टेबल स्टैंडर्ड कोड फॉर इन्फार्मेशन इंटरचेंज

Ans. (b)

व्याख्या : आस्की (ASCII) का पूर्ण रूप है—अमेरिकन स्टैंडर्ड कोड फॉर इन्फार्मेशन इंटरचेंज (American Standard Code for Information Interchange) यह अमेरिकन नेशनल स्टैंडर्ड्स इंस्टीच्यूट (ANSI) द्वारा जारी एक लोकप्रिय कोडिंग सिस्टम है।

12. सुपर कम्प्यूटर के लिए शब्द लंबाई की परास (Range) होती है— (RAJ/PCS/2008)

- (a) 16 बिट तक
- (b) 32 बिट तक
- (c) 64 बिट तक
- (d) 128 बिट तक

Ans. (c)

13. कम्प्यूटर में डाटा स्टोर करने तथा गणना करने के लिए किस नंबर सिस्टम का प्रयोग होता है— (SBI/Clk/2009)

- (a) दशमलव
- (b) आक्टल
- (c) बाइनरी
- (d) हेक्सा डेसिमल

Ans. (c)

14. पद एम बी (MB) प्रयोग किया जाता है— (Utt. PCS (P), 2008)

- (a) मैग्नेटिक बिट्स के लिए
- (b) मेगा बाइट्स (Mega Bites) के लिए
- (c) मेगा बिट्स के लिए
- (d) इनमें से कोई नहीं

Ans. (b)

15. दस लाख (1 मिलियन) बाइट लगभग होती है— (MPPSC (P), 2008)
(SBI/Clk/2008)

- (a) गीगा बाइट
- (b) किलो बाइट
- (c) मेगा बाइट
- (d) टेरा बाइट

Ans. (c)

व्याख्या : 1 मेगा बाइट = 2^{20} बाइट

= 1,048,576 बाइट

= लगभग 1 मिलियन बाइट

= लगभग 10 लाख बाइट

16. निम्नलिखित में से स्टोरेज का सबसे बड़ा यूनिट कौन-सा है— (IBPS (Clk), 2011)

- (a) GB
- (b) KB
- (c) MB
- (d) TB
- (e) इनमें से कोई नहीं

Ans. (d)

17. सबसे बड़े से सबसे छोटे के क्रम में निम्नलिखित में से कौन-सा सही है — (IBPS (Clk), 2011)

- (a) TB-MB-GB-KB
- (b) GB-TB-MB-KB
- (c) TB-GB-KB-MB
- (d) TB-GB-MB-KB
- (e) इनमें से कोई नहीं

Ans. (d)

व्याख्या : 1024 बाइट = 1 किलो बाइट (KB)

1024 KB = 1 मेगा बाइट (MB)

1024 MB = 1 गीगा बाइट (GB)

1024 GB = 1 टेरा बाइट (TB)

18. कम्प्यूटर संक्षिप्ताक्षर KB का सामान्यतः अर्थ होता है— (IBPS (PO), 2011)/Allahabad Bank, 2011)

- (a) की ब्लॉक
- (b) कर्नल बूट
- (c) किलो बाइट
- (d) किट बिट
- (e) इनमें से कोई नहीं

Ans. (c)

19. सबसे ज्यादा प्रयोग में आने वाला कोड कौन-सा है जो प्रत्येक कैरेक्टर को विशिष्ट 8 बिट कोड के रूप में निरूपित करता है— (IBPS/Clk, 2011)

- (a) आस्की (ASCII)
- (b) यूनिकोड (Unicode)
- (c) बाइनरी नंबर
- (d) ईबीसीडीआईसी (EBCDIC)
- (e) इनमें से कोई नहीं

Ans. (b)

20. कम्प्यूटर में एक निबल (Nibble) कितने बिट को निरूपित करता है— (IBPS/Clk, 2011)/(SSC, 2013)

- (a) 4
- (b) 8
- (c) 16
- (d) 32
- (e) 64

Ans. (a)



साफ्टवेयर

(Software)

72 / 184

1. परिभाषा

साफ्टवेयर (Software) : साफ्टवेयर प्रोग्रामों, नियम व क्रियाओं का वह समूह है जो कम्प्यूटर सिस्टम के कार्यों को नियंत्रित करता है तथा कम्प्यूटर के विभिन्न हार्डवेयर के बीच समन्वय स्थापित करता है, ताकि किसी विशेष कार्य को पूरा किया जा सके। इस तरह, साफ्टवेयर वह निर्देश है जो हार्डवेयर से निर्धारित कार्य कराने के लिए उसे दिए जाते हैं। साफ्टवेयर हार्डवेयर को यह बताता है कि उसे क्या करना है, कब करना है और कैसे करना है। साफ्टवेयर कम्प्यूटर का वह भाग है जिसे हम छू नहीं सकते। अगर हार्डवेयर इंजन है तो साफ्टवेयर उसका ईंधन। साधारणतः प्रोग्राम (Program), अप्लिकेशन (Application) और साफ्टवेयर (Software) एक ही चीज को इंगित करते हैं।

जब हार्डवेयर किसी कार्य को पूरा करने के लिए साफ्टवेयर प्रोग्राम के निर्देशों का अनुपालन करता है तो इसे प्रोग्राम run या execute करना कहा जाता है।

2. साफ्टवेयर के प्रकार (Types of Software)

साफ्टवेयर को मुख्यतः तीन भागों में बांटा जा सकता है—

1. सिस्टम साफ्टवेयर (System Software)
2. एप्लिकेशन साफ्टवेयर (Application Software)
3. यूटिलिटी साफ्टवेयर (Utility Software)



चित्र संख्या 8.1

3. सिस्टम साफ्टवेयर (System Software)

प्रोग्रामों का समूह जो कम्प्यूटर सिस्टम के मूलभूत कार्यों को संपन्न करने तथा उन्हें कार्य के लायक बनाए रखने के लिए तैयार किए जाते हैं, सिस्टम साफ्टवेयर कहलाते हैं। यह कम्प्यूटर तथा उपयोगकर्ता के बीच मध्यस्थ का कार्य करता है। सिस्टम साफ्टवेयर के बिना कम्प्यूटर एक बेजान मशीन भर ही रह जाता है। सिस्टम साफ्टवेयर एक तरफ तो कम्प्यूटर हार्डवेयर से जुड़ा होता है तो दूसरी तरफ अप्लिकेशन साफ्टवेयर से। सिस्टम साफ्टवेयर अप्लिकेशन साफ्टवेयर के लिए पृष्ठभूमि तैयार करता है। कोई भी अप्लिकेशन साफ्टवेयर सिस्टम साफ्टवेयर को ध्यान

में रखकर ही तैयार किया जाता है।

सिस्टम साफ्टवेयर के प्रमुख कार्य हैं—

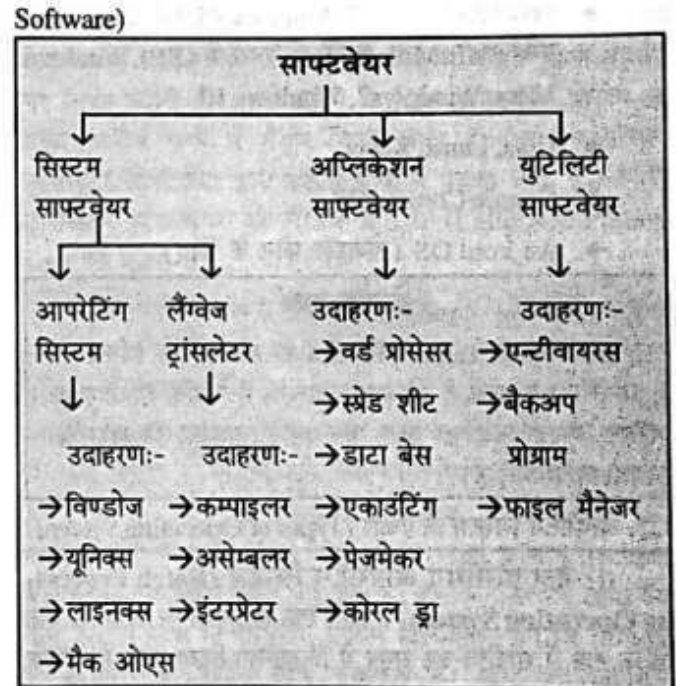
- (i) विभिन्न हार्डवेयर संसाधनों का नियंत्रण, समन्वय तथा महत्तम उपयोग सुनिश्चित करना।
- (ii) अप्लिकेशन साफ्टवेयर के लिए पृष्ठभूमि तैयार करना।
- (iii) पेरिफेरल डिवाइसेस का समन्वय तथा नियंत्रण करना।
- (iv) उपयोगकर्ता, अप्लिकेशन साफ्टवेयर तथा हार्डवेयर के बीच मध्यस्थ का काम करना।

सिस्टम साफ्टवेयर के उदाहरण हैं—

डॉस (DOS), विन्डोज (Windows), युनिक्स (Unix), मैसिन्टास (Macintosh) आदि।

सिस्टम साफ्टवेयर को मुख्यतः दो भागों में बांटा जाता है—

1. ऑपरेटिंग सिस्टम साफ्टवेयर (Operating System Software)
2. लैंग्वेज ट्रांसलेटर साफ्टवेयर (Language Translator Software)



3.1. ऑपरेटिंग सिस्टम (Operating System) : ऑपरेटिंग

सिस्टम प्रोग्रामों का वह समूह है जो कम्प्यूटर सिस्टम तथा उसके विभिन्न संसाधनों के कार्यों को नियंत्रित करता है तथा हार्डवेयर, अप्लिकेशन साफ्टवेयर तथा उपयोगकर्ता के बीच संबंध स्थापित करता है। यह विभिन्न अप्लिकेशन प्रोग्राम के बीच समन्वय भी स्थापित करता है। ऑपरेटिंग सिस्टम के बिना हार्डवेयर किसी अप्लिकेशन प्रोग्राम को क्रियान्वित नहीं कर सकता। अधिकांश ऑपरेटिंग सिस्टम के साथ कुछ अप्लिकेशन साफ्टवेयर जैसे—Video Player, Web Browser, Calculator आदि पहले से ही बने होते हैं।

आपरेटिंग सिस्टम के मुख्य कार्य हैं—

- कम्प्यूटर चालू किये जाने पर साफ्टवेयर को द्वितीयक मेमोरी से लेकर प्राथमिक मेमोरी में डालना तथा कुछ मूलभूत क्रियाएं स्वतः प्रारंभ करना।
- हार्डवेयर और उपयोगकर्ता के बीच संबंध स्थापित करना।
- हार्डवेयर संसाधनों का नियंत्रण तथा बेहतर उपयोग सुनिश्चित करना।
- अप्लिकेशन साफ्टवेयर के क्रियान्वयन के लिए पृष्ठभूमि तैयार करना।
- मेमोरी और फाइल प्रबंधन करना तथा मेमोरी और स्टोरेज डिस्क के बीच डाटा का आदान-प्रदान करना।
- हार्डवेयर व साफ्टवेयर से संबंधित कम्प्यूटर के विभिन्न दोषों (errors) को इंगित करना।
- कम्प्यूटर साफ्टवेयर तथा डाटा को अवैध प्रयोग से सुरक्षित रखना तथा इसकी चेतावनी (Warning) देना।

कुछ प्रमुख आपरेटिंग सिस्टम के उदाहरण हैं—

- माइक्रोसाफ्ट डॉस (MS DOS)
- Apple का Mac OS
- माइक्रोसाफ्ट विण्डोज-Windows 95, 98, 2000, एमई (ME-Millennium), एक्स पी (XP), Windows Vista, Windows 7, Windows 10.
- Unix, Linux, Xenix
- Google Chrome OS
- Android OS (मोबाइल फोन के लिए)

रोचक तथ्य

लिनक्स (Linux) विण्डोज के समान एक शक्तिशाली आपरेटिंग सिस्टम है जो मुफ्त उपलब्ध है जबकि विण्डोज के लिए शुल्क चुकाना पड़ता है। इसके बावजूद लिनक्स का प्रचलन सीमित है।

3.2. आपरेटिंग सिस्टम के प्रकार (Types of Operating System)

(i) बैच प्रोसेसिंग आपरेटिंग सिस्टम (Batch Processing Operating System) : इसमें एक ही प्रकृति के कार्यों को एक बैच के रूप में संगठित कर समूह में क्रियान्वित किया जाता है। इसके लिए बैच मॉनीटर साफ्टवेयर का प्रयोग किया जाता है।

इस सिस्टम का लाभ यह है कि प्रोग्राम के क्रियान्वयन के लिए कम्प्यूटर के सभी संसाधन उपलब्ध रहते हैं, अतः समय प्रबंधन की आवश्यकता नहीं पड़ती।

परंतु, इसमें उपयोगकर्ता तथा प्रोग्राम के बीच क्रियान्वयन के दौरान कोई अंतर्संबंध नहीं रहता तथा परिणाम प्राप्त करने में समय अधिक लगता है। मध्यवर्ती परिणामों पर उपयोगकर्ता का कोई नियंत्रण नहीं रहता।

उपयोग : इस सिस्टम का प्रयोग ऐसे कार्यों के लिए किया जाता है जिसमें मानवीय हस्तक्षेप की आवश्यकता नहीं होती। जैसे-सांख्यिकीय विश्लेषण (Statistical Analysis), बिलप्रिंट करना, पेरोल (Payroll) बनाना आदि।

(ii) मल्टी प्रोग्रामिंग आपरेटिंग सिस्टम (Multi Programming Operating System) : इस प्रकार के आपरेटिंग सिस्टम में एक साथ कई कार्यों को सम्पादित किया जा सकता है। उदाहरण के लिए, किसी एक प्रोग्राम के क्रियान्वयन के बाद जब उसका प्रिंट लिया जा रहा होता है तो प्रोसेसर खाली बैठने के स्थान पर दूसरे प्रोग्राम का क्रियान्वयन आरंभ कर देता है जिसमें प्रिंटर की आवश्यकता नहीं होती। इससे क्रियान्वयन में लगने वाला कुल समय कम हो जाता है तथा संसाधनों का बेहतर उपयोग भी संभव हो पाता है। मल्टीप्रोग्रामिंग आपरेटिंग सिस्टम में प्रोसेसर कई प्रोग्रामों को एक साथ क्रियान्वित नहीं करता, बल्कि एक समय में एक ही निर्देश को संपादित करता है। एक निर्देश संपादित होने के बाद ही मेन मेमोरी में स्थित दूसरे कार्य के निर्देश को संपादित किया जाता है।

इसके लिए विशेष हार्डवेयर व साफ्टवेयर की आवश्यकता होती है। इसमें कम्प्यूटर की मुख्य मेमोरी का आकार बड़ा होना चाहिए ताकि मुख्य मेमोरी का कुछ हिस्सा प्रत्येक प्रोग्राम के लिए आवंटित किया जा सके। इसमें प्रोग्राम क्रियान्वयन का क्रम तथा वरीयता निर्धारित करने की व्यवस्था भी होनी चाहिए।

(iii) टाइम शेयरिंग आपरेटिंग सिस्टम (Time Sharing Operating System) : इस आपरेटिंग सिस्टम में एक साथ कई उपयोगकर्ता जिन्हें टर्मिनल (Terminal) भी कहते हैं; इंटरएक्टिव मोड में कार्य करते हैं जिसमें प्रोग्राम के क्रियान्वयन के बाद प्राप्त परिणाम को तुरंत दर्शाया जाता है। प्रत्येक उपयोगकर्ता को संसाधनों के साझा उपयोग के लिए कुछ समय दिया जाता है जिसे टाइम स्लाइस (Time Slice) या क्वांटम कहते हैं।

क्या आप जानते हैं ?

आधुनिक कम्प्यूटर में मुख्यतौर पर मल्टी प्रोग्रामिंग आपरेटिंग सिस्टम का प्रयोग किया जाता है। विण्डोज (Windows) और लिनक्स (Linux) मल्टी प्रोग्रामिंग आपरेटिंग सिस्टम है जिनमें एक साथ कई प्रोग्राम चलाये जा सकते हैं।

इनपुट देने और आउटपुट प्राप्त करने के बीच के समय को टर्न अराउंड समय (Turn Around Time) कहा जाता है। इस समय का उपयोग कम्प्यूटर द्वारा अन्य उपयोगकर्ता के प्रोग्रामों के क्रियान्वयन में किया जाता है।

इस आपरेटिंग सिस्टम में मेमोरी का सही प्रबंधन आवश्यक होता है क्योंकि कई प्रोग्राम एक साथ मुख्य मेमोरी में उपस्थित होते हैं। इस व्यवस्था में पूरे प्रोग्रामों को मुख्य मेमोरी में न रखकर प्रोग्राम क्रियान्वयन के लिए आवश्यक हिस्सा ही मुख्य मेमोरी में लाया जाता है। इस प्रक्रिया को स्वीपिंग (Swapping) कहते हैं।

(iv) रीयल टाइम सिस्टम (Real Time System) : इस आपरेटिंग सिस्टम में निर्धारित समय सीमा में परिणाम देने को महत्व

दिया जाता है। इसमें एक प्रोग्राम के परिणाम का दूसरे प्रोग्राम में इनपुट डाटा के रूप में प्रयोग किया जा सकता है। पहले प्रोग्राम के क्रियान्वयन में देरी से दूसरे प्रोग्राम का क्रियान्वयन और परिणाम रुक सकता है। अतः इस व्यवस्था में प्रोग्राम के क्रियान्वयन समय (Response time) को तीव्र रखा जाता है।

इस आपरेटिंग सिस्टम का उपयोग उपग्रहों के संचालन, हवाई जहाज के नियंत्रण, परमाणु भट्टियों, वैज्ञानिक अनुसंधान, रक्षा, चिकित्सा, रेलवे आरक्षण आदि में किया जाता है। लिनक्स (Linux) आपरेटिंग सिस्टम रीयल टाइम आपरेटिंग सिस्टम का उदाहरण है।

(v) **एकल आपरेटिंग सिस्टम (Single Operating System)** : पर्सनल कम्प्यूटर के विकास के साथ एकल आपरेटिंग सिस्टम की आवश्यकता महसूस की गई जिसमें प्रोग्राम क्रियान्वयन की समय सीमा या संसाधनों के बेहतर उपयोग को वरीयता न देकर प्रोग्राम की सरलता तथा उपयोगकर्ता को अधिक से अधिक सुविधा प्रदान करने पर जोर दिया गया। एमएस डॉस (MS DOS-Microsoft Disk Operating System) एकल आपरेटिंग सिस्टम का उदाहरण है।

एकल मल्टी टास्किंग आपरेटिंग सिस्टम (Single User Multi Tasking Operating System) : इस प्रकार के सिस्टम में प्रोसेसर द्वारा एक साथ कई कार्य संपादित किए जाते हैं। इसमें प्रोसेसर अपना कुछ समय सभी चालू प्रोग्राम को देता है तथा सभी प्रोग्राम साथ-साथ संपादित होते हैं। इसमें अलग-अलग कार्यों की प्रगति का विवरण भी स्क्रीन पर देखा जा सकता है। यह टाइम शेयरिंग साफ्टवेयर का ही एक प्रकार है। माइक्रोसाफ्ट विंडोज (Microsoft Windows) सिंगल यूजर मल्टी टास्किंग साफ्टवेयर का उदाहरण है।

मल्टी प्रोसेसिंग सिस्टम (Multi Processing System) : इसमें एक साथ दो या अधिक प्रोसेसर को आपस में जोड़कर उनका उपयोग किया जाता है। इससे कार्य संपादित करने की गति में वृद्धि होती है। इसमें एक साथ दो अलग-अलग प्रोग्राम या एक ही प्रोग्राम के भाग क्रियान्वित किया जा सकता है। इसे पैरालेल प्रोसेसिंग (Parallel Processing) भी कहा जाता है।

(vi) **मल्टी यूजर आपरेटिंग सिस्टम (Multi User Operating System)** : इस आपरेटिंग सिस्टम का प्रयोग नेटवर्क से जुड़े कम्प्यूटर सिस्टम में किया जाता है। इसमें कई उपयोगकर्ता एक ही समय में कम्प्यूटर पर स्थित एक ही डाटा का उपयोग तथा उसका प्रोसेस कर सकते हैं। Unix, Linux, Window-7 आदि मल्टी यूजर आपरेटिंग सिस्टम के उदाहरण हैं।

(vii) **इम्बेडेड आपरेटिंग सिस्टम (Embedded Operating System)** : किसी उपकरण के भीतर स्थित प्रोसेसर के प्रयोग के लिए बना आपरेटिंग साफ्टवेयर इम्बेडेड आपरेटिंग सिस्टम कहलाता है। यह साफ्टवेयर प्रोसेसर के भीतर ही रॉम (ROM) में स्टोर किया जाता है। माइक्रोवेव, वाशिंग मशीन, डीवीडी प्लेयर, इलेक्ट्रॉनिक घड़ी आदि में इसका प्रयोग किया जाता है।

(viii) **ओपन/क्लोज्ड सोर्स आपरेटिंग सिस्टम (Open/Closed Source Operating System)** : ओपन सोर्स आपरेटिंग सिस्टम में साफ्टवेयर का केरनेल (Kernel) या सोर्स कोड (Source Code) सबके लिए उपलब्ध होता है और कोई भी अपनी आवश्यकतानुसार इसमें परिवर्तन कर उसका उपयोग कर सकता है। इस आपरेटिंग सिस्टम पर किसी का अधिकार नहीं होता और न ही उपयोगकर्ता द्वारा कोई शुल्क चुकाना पड़ता है। क्लोज्ड सोर्स आपरेटिंग सिस्टम में उसका सोर्स कोड गुप्त रखा जाता है तथा उपयोगकर्ता निर्धारित शुल्क चुकाकर ही इस साफ्टवेयर का उपयोग कर सकता है। Linux एक ओपन सोर्स आपरेटिंग सिस्टम है जबकि Windows माइक्रोसाफ्ट कम्पनी का क्लोज्ड सोर्स आपरेटिंग सिस्टम है। मोबाइल टेलीफोन में प्रयुक्त Google का Android OS ओपन सोर्स साफ्टवेयर है जबकि Apple का iPhone OS एक क्लोज्ड सोर्स आपरेटिंग सिस्टम है।

3.3. लैंग्वेज ट्रांसलेटर साफ्टवेयर (Language Translator Software) : कम्प्यूटर एक इलेक्ट्रॉनिक मशीन है जो केवल बाइनरी अंकों (0 तथा 1 या ऑफ तथा ऑन) को समझ सकता है। बाइनरी अंकों में लिखे निर्देश या साफ्टवेयर प्रोग्राम को मशीन भाषा (Machine Language) कहा जाता है। कम्प्यूटर मशीन भाषा में लिखे प्रोग्राम को समझ कर क्रियान्वित (run) कर सकता है। परंतु मशीन भाषा में प्रोग्राम या साफ्टवेयर तैयार करना कठिन काम होता है। साथ ही, प्रत्येक कम्प्यूटर प्रोसेसर की अपनी एक अलग मशीन भाषा होती है जो प्रोसेसर बनाने वाली कम्पनी पर निर्भर करती है। इससे बचने के लिए साफ्टवेयर प्रोग्राम को उच्च स्तरीय भाषा में तैयार किया जाता है तथा इसे लैंग्वेज ट्रांसलेटर साफ्टवेयर द्वारा मशीन भाषा में बदला जाता है। लैंग्वेज ट्रांसलेटर साफ्टवेयर को लैंग्वेज प्रोसेसर (Language Processor) भी कहते हैं।

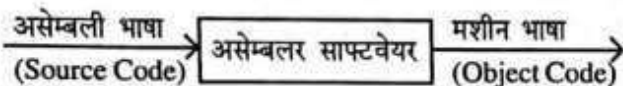
उच्च स्तरीय भाषा (High Level Language) आम बोलचाल की भाषा के करीब होती है। अतः इस भाषा में प्रोग्राम तैयार करना अपेक्षाकृत आसान होता है। साथ ही उच्च स्तरीय भाषा प्रोसेसर की कम्पनी तथा उसके मॉडल पर निर्भर नहीं करती। उच्च स्तरीय भाषा में तैयार प्रोग्राम को लैंग्वेज ट्रांसलेटर साफ्टवेयर द्वारा मशीन भाषा में परिवर्तित कर किसी भी कम्प्यूटर पर चलाया जा सकता है। उच्च स्तरीय भाषा में तैयार किया गया प्रोग्राम **सोर्स प्रोग्राम (Source Program)** या **सोर्स कोड** कहलाता है। जबकि ट्रांसलेटर साफ्टवेयर द्वारा मशीन भाषा में परिवर्तित प्रोग्राम **आब्जेक्ट प्रोग्राम (Object Program)** या मशीन कोड कहलाता है। सामान्यतः आपरेटिंग सिस्टम साफ्टवेयर निम्न स्तरीय भाषा (LLL) में लिखा जाता है जबकि अप्लिकेशन या यूटिलिटी साफ्टवेयर उच्च स्तरीय भाषा (HLL) में तैयार किया जाता है।

लैंग्वेज ट्रांसलेटर साफ्टवेयर तीन प्रकार के होते हैं—

- असेम्बलर (Assembler)
- कम्पाइलर (Compiler)
- इंटरप्रेटर (Interpreter)

3.3.1. असेम्बलर (Assembler) : यह एक साफ्टवेयर प्रोग्राम है जो असेम्बली या निम्न स्तरीय भाषा में लिखे प्रोग्राम को मशीन भाषा में परिवर्तित करता है। असेम्बलर साफ्टवेयर कम्प्यूटर निर्माता कम्पनियों द्वारा उपलब्ध कराया जाता है तथा हार्डवेयर या प्रोसेसर के प्रकार पर निर्भर करता है। अतः प्रत्येक प्रोसेसर का असेम्बलर प्रोग्राम अलग-अलग हो सकता है।

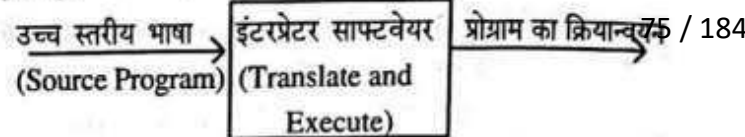
असेम्बलर साफ्टवेयर असेम्बली भाषा में लिखे प्रोग्राम के सोर्स कोड को मशीन या ऑब्जेक्ट कोड में बदलता है। यह मशीन कोड को एक स्थान पर इकट्ठा (Assemble) करता है तथा उसे कम्प्यूटर मेमोरी में स्थापित कर क्रियान्वयन (run) के लिए तैयार करता है।



3.3.2. कम्पाइलर (Compiler) : यह एक लैंग्वेज ट्रांसलेटर साफ्टवेयर है जो उच्च स्तरीय भाषा (HLL) में तैयार किए गये प्रोग्राम को मशीनी भाषा में परिवर्तित करता है। कम्पाइलर पूरे प्रोग्राम को एक ही बार में अनुवादित करता है तथा प्रोग्राम की सभी गलतियों को उनके लाइन क्रम में एक साथ सूचित करता है। जब सभी गलतियाँ दूर हो जाती हैं तो प्रोग्राम संपादित हो जाता है तथा मेमोरी में सोर्स प्रोग्राम (Source Program) की कोई आवश्यकता नहीं होती। प्रत्येक उच्च स्तरीय भाषा के लिए अलग कम्पाइलर साफ्टवेयर होता है। कम्पाइलर उच्च स्तरीय भाषा के प्रत्येक निर्देश को मशीन भाषा निर्देश में संकलित (Compile) करता है। कम्पाइलर पूरे सोर्स प्रोग्राम या सोर्स कोड को ऑब्जेक्ट प्रोग्राम/कोड में बदलकर उसे मेमोरी में स्टोर करता है, परंतु उसे क्रियान्वित (run) नहीं करता। इसके पश्चात, प्रोग्राम को ऑब्जेक्ट कोड द्वारा क्रियान्वित किया जाता है। इस तरह, एक बार कम्पाइल हो जाने के बाद प्रोग्राम को क्रियान्वित करने के लिए कम्पाइलर साफ्टवेयर की जरूरत नहीं होती।

3.3.3. इंटरप्रेटर (Interpreter) : कम्पाइलर की तरह इंटरप्रेटर भी एक लैंग्वेज ट्रांसलेटर साफ्टवेयर है। इंटरप्रेटर साफ्टवेयर उच्च स्तरीय भाषा में तैयार किए गए प्रोग्राम को मशीनी भाषा में परिवर्तित कर उसे क्रियान्वित करता है। इंटरप्रेटर उच्च स्तरीय भाषा में तैयार किए गए प्रोग्राम के प्रत्येक लाइन को एक-एक कर (Line by line) मशीन भाषा में परिवर्तित करता है। यह प्रोग्राम के एक लाइन का मशीनी भाषा में अनुवाद कर लेने के पश्चात उसे क्रियान्वित (run या execute) भी करता है। यदि इस लाइन के क्रियान्वयन में कोई गलती हो तो उसे उसी समय इंगित करता है तथा संशोधन के बाद ही अगली लाइन को मशीन भाषा में परिवर्तित करता है। स्पष्टतः, इंटरप्रेटर का आउटपुट ऑब्जेक्ट प्रोग्राम न होकर साफ्टवेयर क्रियान्वयन का परिणाम होता है। अतः प्रत्येक बार साफ्टवेयर को क्रियान्वयन के दौरान इंटरप्रेटर से होकर गुजरना पड़ता है। इस कारण, इंटरप्रेटर साफ्टवेयर का मेमोरी में बना रहना आवश्यक होता है। कम्पाइलर की अपेक्षा इंटरप्रेटर साफ्टवेयर तैयार करना आसान होता है। चूंकि इंटरप्रेटर एक-एक लाइन कर प्रोग्राम की गलतियों को

इंगित करता है, अतः इंटरप्रेटर द्वारा प्रोग्राम में सुधार करना आसान होता है।



रोचक तथ्य

यूनक्स 'सी' (C) भाषा में लिखा जानेवाला पहला ऑपरेटिंग सिस्टम है। इससे किसी नए मशीन में इसका प्रयोग आसान हुआ।

3.3.4. कम्पाइलर (Compiler) और इंटरप्रेटर (Interpreter) में अंतर : वस्तुतः दोनों का कार्य उच्च स्तरीय भाषा (High Level Language) को मशीन भाषा में बदलना है। पर कार्य पद्धति के आधार पर दोनों में कुछ अंतर भी है—

कम्पाइलर	इंटरप्रेटर
(i) पूरे प्रोग्राम को एक साथ परिवर्तित करता है।	(i) प्रोग्राम को एक-एक लाइन कर अनुवादित करता है।
(ii) पूरे प्रोग्राम को मशीन भाषा में परिवर्तित कर सभी गलतियाँ एक साथ बताता है।	(ii) एक लाइन को मशीन भाषा में परिवर्तित कर उसकी गलतियाँ बताता है तथा उस दोष के दूर हो जाने पर ही आगे बढ़ता है।
(iii) प्रोग्राम को मशीन भाषा में परिवर्तित करता है, पर उसे क्रियान्वित नहीं करता।	(iii) प्रोग्राम को मशीन भाषा में परिवर्तित कर उसे क्रियान्वित भी करता है।
(iv) कम्पाइलर का आउटपुट मशीन भाषा का ऑब्जेक्ट कोड होता है।	(iv) इंटरप्रेटर का आउटपुट साफ्टवेयर क्रियान्वयन का परिणाम होता है।
(v) एक बार अनुवाद हो जाने के बाद प्रोग्राम को क्रियान्वित करने के लिए कम्पाइलर साफ्टवेयर की जरूरत नहीं होती।	(v) प्रोग्राम को क्रियान्वित करने के लिए साफ्टवेयर को प्रत्येक बार इंटरप्रेटर से होकर गुजरना पड़ता है। अतः हर बार, इंटरप्रेटर साफ्टवेयर की जरूरत पड़ती है।
(vi) अशुद्धियों को हटाने में धीमा होता है।	(vi) अशुद्धियों को हटाने में तीव्र होता है।
(vii) सम्पादन में कम समय लेता है।	(vii) सम्पादन में अधिक समय लेता है।

3.4. कुछ लोकप्रिय ऑपरेटिंग सिस्टम (Some Popular Operating System)

(a) एमएसडॉस (MS-DOS-Microsoft Disk Operating System) : यह 1981 में माइक्रोसाफ्ट व आईबीएम द्वारा संयुक्त रूप से तैयार किया गया एकल ऑपरेटिंग सिस्टम (Single