Character and String Representation

CS520

Department of Computer Science

University of New Hampshire

CDC 6600

- 6-bit character encodings
- i.e. only 64 characters
- Designers were not too concerned about text processing!

The table is from *Assembly Language Programming for the Control Data 6000 series and the Cyber 70 series* by Grishman.

TABLE OF DISPLAY CODES

Character	Code	Character	Code
A	01	0	33
B	02	1	34
C	03	2	35
D	04	3	36
E	05	4	37
F	06	5	40
G	07	6	41
H	10	7	42
I	11	8	43
J	12	9	44
K	13		
L	14	+	45
M	15	-	46
N	16	*	47
O	17	/	50
P	20	(	51
Q	21	)	52
R	22	$	53
S	23	=	54
T	24	(blank)	55
U	25	,	56
V	26	.	57
W	27	≡	60
X	30	[	61
Y	31	]	62
Z	32	:	63
		≠	64
		-	65
		∨	66
		∧	67
		↑	70
		↓	71
		<	72
		>	73
		≤	74
		≥	75
		¬	76
		;	77*

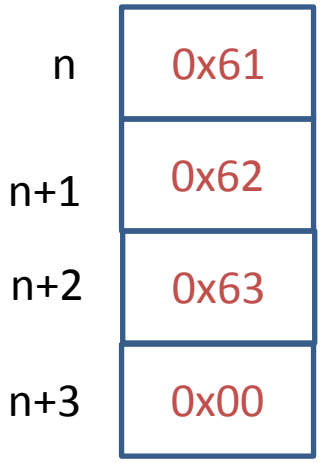
*Do not use the semicolon in COMPASS instructions

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

C Strings

- Usually implemented as a series of ASCII characters terminated by a null byte (0x00).
- `"abc"` in memory is:



Unicode

- The space of values is divided into 17 ***planes***.
- Plane 0 is the Basic Multilingual Plane (BMP).
  - Supports nearly all modern languages.
  - Encodings are 0x0000-0xFFFF.
- Planes 1-16 are supplementary planes.
  - Supports historic scripts and special symbols.
  - Encodings are 0x10000-0x10FFFF.
- Planes are divided into ***blocks***.

Unicode and ASCII

- ***ASCII is the bottom block in the BMP***, known as the Basic Latin block.
- So ASCII values are embedded “as is” into Unicode.
- i.e. 'a' is 0x61 in ASCII and 0x0061 in Unicode.

Special Encodings

- The Byte-Order Mark (BOM) is used to signal endian-ness.
- Has no other meaning (i.e. usually ignored).
- Encoded as 0xFEFF.
- 0xFFFE is a ***noncharacter.***
  - Cannot appear in any exchange of Unicode.
- So file can be started with a BOM; the reader can then know the endian-ness of the file.
- In absence of a BOM, Big Endian is assumed.

Other Noncharacters

- There are a total of 66 noncharacters:
  - 0xFFFE and 0xFFFF of the BMP
  - 0x1FFFE and 0x1FFFF of plane 1
  - 0x2FFFE and 0x2FFFF of plane 2
  - etc., up to
  - 0x10FFFE and 0x10FFFF of plane 16
  - Also 0xFDD0-0xFDEF of the BMP.

UTF: UCS* Transformation Format

- **UTF-8**
  - Encodes Unicode characters in 1-4 bytes.
  - ASCII gets encoded as 1 byte.
  - Dominant character encoding for the WWW.
- **UTF-16**
  - Encodes BMP characters in 2 bytes
  - Encodes non-BMP characters in 4 bytes.
- **UTF-32**
  - Fixed-sized representation of Unicode.

*Universal Character Set.

UTF-8

- Take the Unicode character and throw away the leading zero bits.*
- Count the remaining number of bits.
- **7 bits: 0xxxxxxx**
- **11 bits: 110xxxxx 10xxxxxx**
- **16 bits: 1110xxxx 10xxxxxx 10xxxxxx**
- **21 bits: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx**

*Overlong encodings are forbidden. Therefore there is a unique UTF-8 encoding for each Unicode character.

Errors in UTF-8

- Overlong encodings.
- An unexpected continuation byte.
- A start byte not followed by enough continuation bytes.
- A 4-byte sequence starting with 0xF4 that decodes to a value greater than 0x10FFFF.
- A sequence that decodes to a noncharacter.
- A sequence that decodes to a value in range 0xD800-0xDFFF.

UTF-16

- 1 UTF-16 code unit (2 8-bit bytes) for each BMP character.
- 2 UTF-16 code units for each non-BMP character (4 bytes in total).
  - 0x10000 is subtracted from the value, leaving a 20-bit number in the range 0x00000-0xFFFFF.
  - The top 10 bits are added to 0xD800 to give the first code unit, called the ***lead surrogate***.
  - The low 10 bits are added to 0xDC00 to give the second code unit, called the ***trail surrogate***.

Self-synchronizing

- 10 bits express values in the range 0x000-0x3FF.
- Lead surrogates will be in range 0xD800+0x000 to 0xD800+0x3FF (0xD800-0xDBFF).
- Trail surrogates will be in range 0xDC00+0x000 to 0xDC00+0x3FF (0xDC00-0xDFFF).
- Remember: values 0xD800-0xDFFF are not valid Unicode characters.
- UTF-16 BMP characters can be distinguished from UTF-16 non-BMP characters.
- **So you can tell where the Unicode character boundaries are in a UTF-16 stream.**

UTF-32

- Simply take the 21-bit Unicode value and add leading zero bits to extend it to 32 bits.
- Byte-order is an issue, like with UTF-16.