Information Representation; Storage Devices, File sizes

Department of Computer Science
City University of New York, Graduate Center

Lecture 5: Information Representation; Storage Devices, File sizes

September 20 , 2010

Outline

Outline

What is a bit

- A bit is the smallest unit of memory. bit=binary digit. "Wires" are either on or off - corresponding to 1 or 0. All data in a computer is represented by patterns of bits.
- A group of 8 bits is called a byte. Since each bit can be either 0 or 1, there are 256 different bit patterns that can be represented with 8 bits.

ASCII

- ASCII is a standardized scheme for representing characters in patterns of 7 bits. (2**7=128, more than enough for upper and lower case, digits and punctuation). Since we use bits in groups of 8, extra bit can be used for error-checking.
- A document that contains plain text only (such as a Notepad file) is called an ASCII file or a text file. Each character of text is stored as one ASCII pattern, in one byte of memory. So a file containing 20 lines of text, with 100 characters per line, would be stored in 2000 bytes.

ASCII

- ASCII is a standardized scheme for representing characters in patterns of 7 bits. (2**7=128, more than enough for upper and lower case, digits and punctuation). Since we use bits in groups of 8, extra bit can be used for error-checking.
- A document that contains plain text only (such as a Notepad file) is called an ASCII file or a text file. Each character of text is stored as one ASCII pattern, in one byte of memory. So a file containing 20 lines of text, with 100 characters per line, would be stored in 2000 bytes.

ASCII

- ASCII is a standardized scheme for representing characters in patterns of 7 bits. (2**7=128, more than enough for upper and lower case, digits and punctuation). Since we use bits in groups of 8, extra bit can be used for error-checking.
- A document that contains plain text only (such as a Notepad file) is called an ASCII file or a text file. Each character of text is stored as one ASCII pattern, in one byte of memory. So a file containing 20 lines of text, with 100 characters per line, would be stored in 2000 bytes.

Outline

Binary Files

- Other data can be stored in other ways.
- Files that contain data that is not plain text (e.g. Word documents which contain formatting information) are not stored as plain ASCII files.
- But the information is still stored in some type of binary format. They are called binary files. (What happens when you try to open a Word document in Notepad? Sometimes you see garbage characters on the screen, because those bytes don't correspond to ASCII codes.)

Binary Files

- Other data can be stored in other ways.
- Files that contain data that is not plain text (e.g. Word documents which contain formatting information) are not stored as plain ASCII files.
- But the information is still stored in some type of binary format. They are called binary files. (What happens when you try to open a Word document in Notepad? Sometimes you see garbage characters on the screen, because those bytes don't correspond to ASCII codes.)

Binary Files

- Other data can be stored in other ways.
- Files that contain data that is not plain text (e.g. Word documents which contain formatting information) are not stored as plain ASCII files.
- But the information is still stored in some type of binary format. They are called binary files. (What happens when you try to open a Word document in Notepad? Sometimes you see garbage characters on the screen, because those bytes don't correspond to ASCII codes.)

Binary Files

- Other data can be stored in other ways.
- Files that contain data that is not plain text (e.g. Word documents which contain formatting information) are not stored as plain ASCII files.
- But the information is still stored in some type of binary format. They are called binary files. (What happens when you try to open a Word document in Notepad? Sometimes you see garbage characters on the screen, because those bytes don't correspond to ASCII codes.)

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

File Sizes

- The size of a file = number of bytes stored in the file. For plain ASCII text files, the size of the file = number of characters.
- Word processing documents are larger because of the extra formatting information that is part of the file.
- 1KB = 1,024 bytes = 2**10 bytes (example above, a file of 20 lines of text, about 100 chars per line, would be about 2 KB)
- 1MB = 1,024 KB = 2**20 bytes (about 1,000 pages of text, each page 20 lines of 100 chars, would be about 2MB) RAM is usually measured in MB.
- floppy disk can store 1.44MBm which is usually enough for several short text files.

An algorithm example

- "A picture is worth 1,000 words" - Actually, computer scientists would say that it is worth more! - 1,000 words, at an average of 5 chars per word = 5,000 chars = about 5KB. That's only enough for a very, very tiny picture. Most graphics on the web are over 30KB!
- (Digression into how graphics are stored; high-resolution vs. low-resolution -> tradeoff of image quality vs. storage space)
- 1GB = 1,024 MB = 2**30 bytes
- Dell recently announced that is making floppy drives optional - other storage devices that can store larger quantities of data: Zip disks, Jaz drives, CDs, DVDs. graphics (picture), music and video files are very large.
- File compression techniques are important. Less space will be needed

An algorithm example

- "A picture is worth 1,000 words" - Actually, computer scientists would say that it is worth more! - 1,000 words, at an average of 5 chars per word = 5,000 chars = about 5KB. That's only enough for a very, very tiny picture. Most graphics on the web are over 30KB!
- (Digression into how graphics are stored; high-resolution vs. low-resolution -> tradeoff of image quality vs. storage space)
- 1GB = 1,024 MB = 2**30 bytes
- Dell recently announced that is making floppy drives optional - other storage devices that can store larger quantities of data: Zip disks, Jaz drives, CDs, DVDs. graphics (picture), music and video files are very large.
- File compression techniques are important. Less space will be needed

An algorithm example

- "A picture is worth 1,000 words" - Actually, computer scientists would say that it is worth more! - 1,000 words, at an average of 5 chars per word = 5,000 chars = about 5KB. That's only enough for a very, very tiny picture. Most graphics on the web are over 30KB!
- (Digression into how graphics are stored; high-resolution vs. low-resolution -> tradeoff of image quality vs. storage space)
- 1GB = 1,024 MB = 2**30 bytes
- Dell recently announced that is making floppy drives optional - other storage devices that can store larger quantities of data: Zip disks, Jaz drives, CDs, DVDs. graphics (picture), music and video files are very large.
- File compression techniques are important. Less space will be needed

An algorithm example

- "A picture is worth 1,000 words" - Actually, computer scientists would say that it is worth more! - 1,000 words, at an average of 5 chars per word = 5,000 chars = about 5KB. That's only enough for a very, very tiny picture. Most graphics on the web are over 30KB!
- (Digression into how graphics are stored; high-resolution vs. low-resolution -> tradeoff of image quality vs. storage space)
- 1GB = 1,024 MB = 2**30 bytes
- Dell recently announced that is making floppy drives optional - other storage devices that can store larger quantities of data: Zip disks, Jaz drives, CDs, DVDs. graphics (picture), music and video files are very large.
- File compression techniques are important. Less space will be needed

An algorithm example

- "A picture is worth 1,000 words" - Actually, computer scientists would say that it is worth more! - 1,000 words, at an average of 5 chars per word = 5,000 chars = about 5KB. That's only enough for a very, very tiny picture. Most graphics on the web are over 30KB!
- (Digression into how graphics are stored; high-resolution vs. low-resolution -> tradeoff of image quality vs. storage space)
- 1GB = 1,024 MB = 2**30 bytes
- Dell recently announced that is making floppy drives optional - other storage devices that can store larger quantities of data: Zip disks, Jaz drives, CDs, DVDs. graphics (picture), music and video files are very large.
- File compression techniques are important. Less space will be needed

Outline

- Data is transmitted at speeds that are measured in terms of bps- bits per second. The time it takes to download a file depends on the size of the file and the speed of the transmission.
- When you connect to the Internet, slowest point is usually the connection from home. Modems - 28.8K - 28,800bps, 56K (although phone lines really don't go faster than about 40-45K), other faster means such as cable modem, ISDN.

- Data is transmitted at speeds that are measured in terms of bps- bits per second. The time it takes to download a file depends on the size of the file and the speed of the transmission.
- When you connect to the Internet, slowest point is usually the connection from home. Modems - 28.8K - 28,800bps, 56K (although phone lines really don't go faster than about 40-45K), other faster means such as cable modem, ISDN.

Outline

Representations

- Text files use ASCII and can be read by text word processors such as Notepad and Word. Pictures use pixels (bitmap) and can be read by a graphics program.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Representations

- Text files use ASCII and can be read by text word processors such as Notepad and Word. Pictures use pixels (bitmap) and can be read by a graphics program.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

- Unicode is an extension of ASCII, which uses 2-bytes instead of one, but allows for many different characters, so can represent characters from different foreign languages that use other character sets such as Russian, Japanes, Hebrew, Arabic.
- Files that are stored using Unicode will require more memory.
- Within programs be stored in different ways. Characters values are stored as ASCII, (or Unicode). In order to allow arithmetic operations, numbers cannot be stored as ASCII.

- Unicode is an extension of ASCII, which uses 2-bytes instead of one, but allows for many different characters, so can represent characters from different foreign languages that use other character sets such as Russian, Japanes, Hebrew, Arabic.
- Files that are stored using Unicode will require more memory.
- Within programs be stored in different ways. Characters values are stored as ASCII, (or Unicode). In order to allow arithmetic operations, numbers cannot be stored as ASCII.

- Unicode is an extension of ASCII, which uses 2-bytes instead of one, but allows for many different characters, so can represent characters from different foreign languages that use other character sets such as Russian, Japanes, Hebrew, Arabic.
- Files that are stored using Unicode will require more memory.
- Within programs be stored in different ways. Characters values are stored as ASCII, (or Unicode). In order to allow arithmetic operations, numbers cannot be stored as ASCII.

- The term binary is used to refer to all representations aside for ASCII.
- Graphic files are said to be stored as binary files.
- Programs stored in executable files are also said to be stored in a binary file.
- Numbers are stored using binary.
- binary value 01000001 : as ASCII for A : binary for the number 65

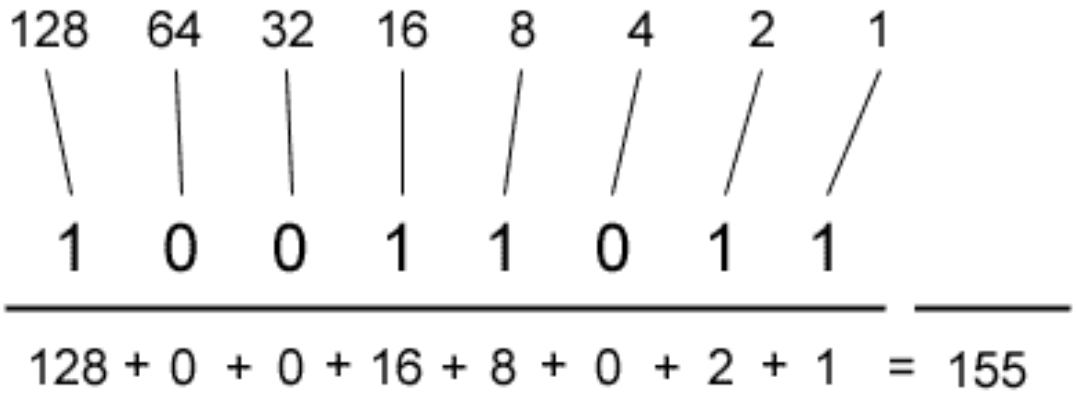
- The term binary is used to refer to all representations aside for ASCII.
- Graphic files are said to be stored as binary files.
- Programs stored in executable files are also said to be stored in a binary file.
- Numbers are stored using binary.
- binary value 01000001 : as ASCII for A : binary for the number 65

- The term binary is used to refer to all representations aside for ASCII.
- Graphic files are said to be stored as binary files.
- Programs stored in executable files are also said to be stored in a binary file.
- Numbers are stored using binary.
- binary value 01000001 : as ASCII for A : binary for the number 65

- The term binary is used to refer to all representations aside for ASCII.
- Graphic files are said to be stored as binary files.
- Programs stored in executable files are also said to be stored in a binary file.
- Numbers are stored using binary.
- binary value 01000001 : as ASCII for A : binary for the number 65

- The term binary is used to refer to all representations aside for ASCII.
- Graphic files are said to be stored as binary files.
- Programs stored in executable files are also said to be stored in a binary file.
- Numbers are stored using binary.
- binary value 01000001 : as ASCII for A : binary for the number 65

Converting a binary number to a decimal



List the powers of two from right to left. Start at 20, evaluating it as "1". Increment the exponent by one for each power. Stop when the amount of elements in the list is equal to the amount of digits in the binary number.

Thank you for your attention