IT Fundamentals

Dr/Mai Ramadan Ibraheem

Lecturer at Information Technology Dept.,

Faculty of Computers and Information – KFS University

Outline

- Von Neumann architecture
- Machine level representation of data.
- Assembly level machine organization
- Hardware realizations of algorithms.
- Operating systems and virtual machines.
- Introduction to net-centric computing.

Grouping Bits

- A **bit** is the smallest unit of memory, and is basically a **switch**. It can be in one of two states, "0" or "1". These states are sometimes referenced as "off and on", or "no and yes"; but these are simply alternate designations for the same concept.
- Given that **each bit** is capable of holding **two possible** values, the number of possible different combinations of values that can be stored in n bits is $\mathbf{2^n}$. For example:

Grouping Bits

- 1 bit can hold $2 = 2^1$ possible values (0 or 1)
- 2 bits can hold $2 \times 2 = 2^2 = 4$ possible values (00, 01, 10, 11)
- 3 bits can hold $2 \times 2 \times 2 = 2^3 = 8$ possible values (000, 001, 010, 011, 100, 101, 110, or 111)
- 4 bits can hold $2 \times 2 \times 2 \times 2 = 2^4 = 16$ possible values
- 5 bits can hold $2 \times 2 \times 2 \times 2 \times 2 = 2^5 = 32$ possible values
- 6 bits can hold $2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^6 = 64$ possible values
- 7 bits can hold $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^7 = 128$ possible values
- 8 bits can hold $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ possible values
- n bits can hold 2^n possible values.

Bits vs. Bytes

- - A byte is simply 8 bits of memory or storage. This is the smallest amount of memory that standard computer processors can manipulate in a single operation.
- If you determine the number of bits of memory that are required, and divide by 8, you will get the number of bytes of memory that are required. Similar, to convert from bytes to bits, you must multiply by 8.

Multiplier Prefixes

- Memory requirements **can** become **huge**, and standard metric-system prefixes are utilized to keep the ultimate value manageable, **according** to the usual **metric system**:
- 1 kilobit (kb) or kilobyte (kB) = 1024 bits or 1024 bytes, respectively
- 1 megabit (Mb) or megabyte (MB) = 1024 kilobits or 1024 kilobytes, respectively

Memory Requirements for a Single Unit

- For **textual** data, a unit is a "**character**"; a file consists of a number of characters.
- For **picture** data, a unit is a "dot" or "**pixel**"; a file consists of a number of dots or pixels.
- For **sound** data, a unit is a "**sample**"; a file consists of a number of samples.
- For **video** data, a unit is a "**frame**"; a file consists of a number of frames

Standards for Representing Characters

- There are two standards for representing characters, **ASCII** (used most places) and **EBCDIC** (used only on some "mainframe" equipment of older design).
- In ASCII, for example,
- the character "3" is represented by 00110011,
- "A" by 00100001, "b" by 01100010,
- and "$" by 00100100.

Picture data - unit size

- is represented in “raster” or “bitmap” format – a rectangular array of dots, each with its own color.
- "dots" or "pixels" are identical.
- "dots" is associated with dots-per-inch (dpi), is a measure of resolution for scanners and printers, while "pixels" is associated with the working resolution of a computer monitor.
- The **memory requirement** for a single dot or pixel depends upon the *level of color* or *shade* resolution desired in the picture.

Picture data - unit size

- For *black-and-white* pictures: Line Art (black and white only) 1 bit/pixel
- 16 shade grayscale 4 bits/pixel
- 32 shade grayscale 5 bits/pixel
- 64 shade grayscale 6 bits/pixel
- 256 shade grayscale 8 bits/pixel

Picture data - unit size

- For *color pictures* 16 color (basic EGA (Enhanced Graphics Adapter) color) 4 bits/pixel
- 256 color (Basic VGA (Video Graphics Adapter) or 8 bit color) 8 bits/pixel
- 16 bit color (65,536 colors)

Picture data - unit size

- **Colors** are represented by **numbers**
- • For black and white: a *number* indicating how bright the dot is.
- • For color, *three number* indicating how bright the dot is (red, blue, and green).

Sound data - unit size

- air pressure variation using a sequence of numbers.
- (for humans vibrations ranging from 20 Hz (cycles per second) to over 20 kHz.
- Most CD-Quality recordings have range of upper 22.05 kHz represented as a 16-bit number.

Video data - unit size

- a series of pictures called frames.
- Video requires a *rapid sequence of pictures* (typically 24 frames per second) to provide *realistic animation*.

Determining how many "units" make up a file

- Alphanumeric files - number of units.
- Picture files - number of units.
- sound files - number of units.
- Video files - number of units.
- Final calculations, and the effect of **compression** on file size.

Alphanumeric files - number of units

- The number of characters **in a file** can be determined from *existing data*.
- As an example, the number of characters in a **book** can be determined by multiplying:
- Characters/book = (characters/line) × (lines/page) × (pages/book)

Alphanumeric files - number of units

- An **80-page book** with **50 lines** per page and **80 characters per line** would have
- (80 characters / line) × (50 lines / page) × (80 pages / book) = 320,000 characters
- Remember that *spaces* are characters, so half lines, half pages and blank pages need to be included.

Pictures files - number of units

- In picture files the number of dots or pixels make up the number of "units".
- Picture files are more likely to be specified by their **length and width**.
- To determine the number of units in this type of file, you simply *multiply the length by the width*.

Pictures files - number of units

- An example would be the **maximum resolution** of the camera.
- i.e. Olympus camera can produce picture files with a resolution of 2048 by 1536 pixels.
- (2048 pixels wide) × (1536 pixels long) = 3,145,728 pixels total.

Pictures files - number of units

- Another common practice is to provide a dot or pixel **density**.
- This is the case for most computer **printers and scanners**, densities using "dots per inch" (dpi).
- Typical resolutions are 300 dpi, 600 dpi, or 1440 dpi.
- To determine the total number of units in this type of file, you need to know the file's **overall size**.

Pictures files - number of units

- As an example, a **4-inch long by 6-inch** wide photo, which is scanned at a **resolution of 600 dpi**, will contain:
- (4 inch long) × (600 dpi) = 2400 dots long
- (6 inch wide) × (600 dpi) = 3600 dots wide
- (2400 dots long) × (3600 dots wide) = 8,640,000 dots total.

Sound files - number of units

- The number of **units** in sound files is based upon **time**.
- Samples are normally specified as **samples per second**, so you must also know the *total length of the recording*.
- The **number of samples** can then be calculated from the relationship:
- Total number of samples = (samples rate per second) × (total time in seconds)

Sound files - number of units

- Digital sampling requires a minimum of **two samples** per Hz. Since the **upper range** of human hearing is near **20,000 Hz**, CD-Quality recordings **range** near 22.05 kHz
- The **sampling rate for CD audio** is standardized at **44,100 samples per second**, and **16 bits** per sample.

Video files - number of units

- The number of units in a video file is based upon time.
- Samples are normally specified as frames per second (fps), thus the *total time* of the video file must also be known.
- The number of frames can then be calculated from the relationship:
- **total** number of frames = (frame **rate** per second) × (**total time** in seconds).

Video files - number of units

- Sampling of analog video requires a minimum of **10 fps** to provide low quality video, and at least **24 fps** for high quality. 30 fps is quite common.

Final calculations, and the effect of compression on file size

- As stated initially, **file size** is the product of "unit" size and *number of units that make up the file*.
- In the case of alphanumeric and picture files, this calculation determines the **uncompressed file size**.
- For sound and video files, if the sample or frame rate is provided, the calculation determines the **compressed file size**.

Final calculations, and the effect of compression on file size

- Compression is a process in which file size is **reduced** while maintaining all critical file components.
- There are many different compression techniques, most of which are optimized using specific "compression ratio".
- Compression ratio is described as:
- Compression ratio = (Original file size) / (Compressed file size)

End

Thank You