LIÈGE université
GIGA

Data representation & storage

GIGA Doctorate School

Mohamed Ali Bahri, Ir Ph.D.

Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

Bits & bytes

- Bit (for "binary digit") =
  - a basic unit of information used in computing and digital communications.
  - can have only one of two values $\rightarrow$ physically represented with a two-state device.
  - most commonly represented as either a 0 or 1
- Byte =
  - a unit of digital information
  - most commonly consists of eight bits,
  - representing a binary number

Bytes

Originally,

- number of bits used to encode a single character of text in a computer
- hardware dependent
- convenient as power of 2 $\rightarrow$ values from 0 to 255

Now

- *de facto* standard for smallest amount of “memory unit”
- 32- or 64-bit ‘words’, built of four or eight bytes
- aka. “octet”, symbol ‘o’,

Memory size

Expressed in binary vs. decimal base

Name	Binary	Decimal	Discrepancy
Kilo-byte (kB)	2^10 = 1.024 o	1.000	2,4%
Mega-byte (MB)	2^20 = 1.048.576 o	1.000.000	4,8%
Giga-byte (GB)	2^30 = 1.073.741.824 o	1.000.000.000	7,4%
Tera-byte (TB)	2^40 = 1.099.511.627.776 o	1.000.000.000.000	9,9%
Peta-byte (PB)	2^50 = 1.125.899.906.842.674 o	1.000.000.000.000.000	12,6%

Note: **3 orders of magnitude** between peta/tera/giga/mega/kilo!

Transfer speed

Typical bandwidth

- RAM, ~10Gb per second
  → 1Gb of data in ~ 0,1 second
- Hard drive, ~0,5Gb per second
  → 10Gb of data in ~ 20 second
- Network, ~100Mbps = ~ 0,1GB per second
  → 1Tb of data in ~ 10.000 seconds = ~2.8 hours !!!

USB type	Speed
USB 1.1	1.5 Mo/s
USB 2	60 Mo/s
USB 3.2 G1	640 Mo/s
USB 3.2 G2	1,25 Go/s
USB 4	5 Go/s

Data transfer can be a bottle neck!

Note: here Gb = "giga bits"

Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

Character

= letter, digit, or punctuation

- with 1 byte, 1 simple character, aka. ‘char’, from ASCII (“American Standard Code for Information Interchange” from the 1960’s)
- 127 characters: 10 digits, 26 letters in lower & upper case, punctuation + formatting codes.
- Limited to English…

USASCII code chart

b_4	b_3	b_2	b_1	Row ↓ / Column →	0 (000)	1 (001)	2 (010)	3 (011)	4 (100)	5 (101)	6 (110)	7 (111)
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Character

= letter, digit, or punctuation

- UTF-8 from Unicode (or Universal Coded Character Set) Transformation Format – 8-bit

- UTF-8 extended up to 4 bytes,

  → more possibilities but more complicated

  → extension to more (non-)characters (math symbols, arrows,…) and alphabets (Greek, Chinese,…)

  → most common for WWW and emails encoding

https://en.wikipedia.org/wiki/UTF-8

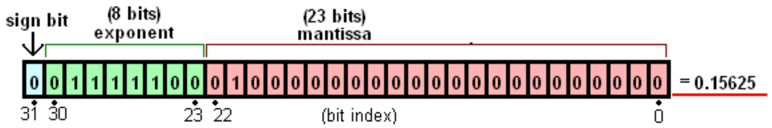
Integer, signed or unsigned numbers

- with 1 byte,
  - ‘int8’, values $\in$ [-128 127]
  - ‘unit8’, values $\in$ [0 255]
- with 2 bytes,
  - ‘int16’ or ‘short’, values $\in$ [−32,768   32,767] i.e. $[-(2^{15})\ \ 2^{15} - 1]$
  - ‘uint16’, values $\in$ [0 65,535] i.e. $[0\ \ 2^{16} - 1]$
- with 4 bytes,
  - ‘int32’ or ‘long’, values $\in [-(2^{31})\ \ 2^{31} - 1]$
  - ‘uint32’, values $\in$ [0 4,294,967,295 i.e. $[0\ \ 2^{32} - 1]$
- with 8 bytes,
  - …

Typically use for pixel intensity coding!

Floating-point ==> $Sign.2^{E}\,.\,F$

- Single-precision = 32 bits = 4 bytes
- wide dynamic range of values with “floating radix point”:



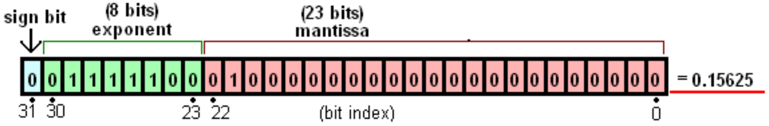
  - sign bit : 1 bit
  - exponent width: 8 bits
  - significand precision: 24 bits (23 explicitly stored)

$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\dots b_{23})_2-127} \times (1.b_{22}b_{21}\dots b_0)_2,$$

- Half-/double-precision with 16/64 bits = 2/8 bytes

Floating-point $\Rightarrow Sign.2^{E}.F$

- Single-precision = 32 bits = 4 bytes
- wide dynamic range of values with "floating radix point":
  - sign bit : 1 bit
  - exponent width: 8 bits
  - significand precision: 24 bits (23 explicitly stored)

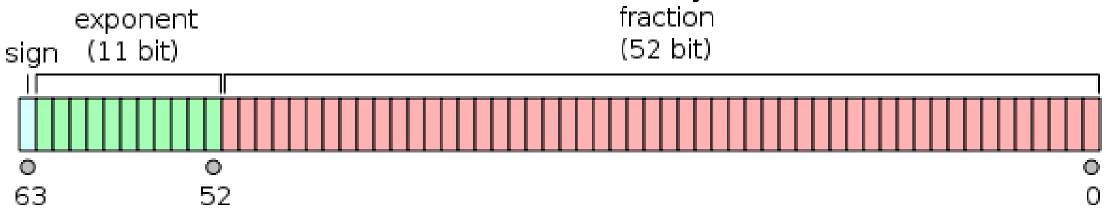


$\rightarrow \quad (-1)^{b_{31}} \times 2^{(b_{30}b_{29}\dots b_{23})_2 - 127} \times (1.b_{22}b_{21}\dots b_0)_2,$

- values up to $(2 - 2^{-23}) \times 2^{127} \approx 3.402823 \times 10^{38}$

Floating-point

- Half-/double-precision with 16/64 bits = 2/8 bytes

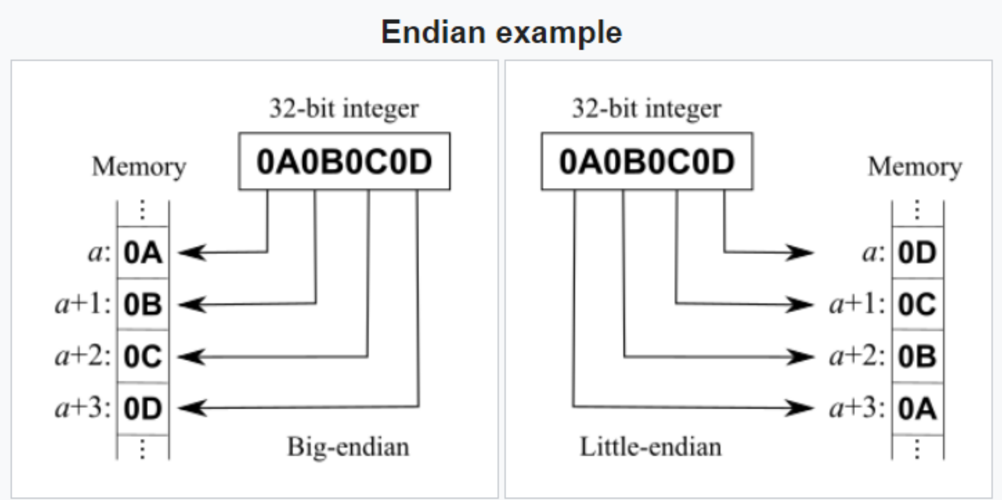


- Double-precision,
  - numbers between 10^{-308} and 10^{308}, with full 15–17 decimal digits precision.
  - smaller values up to about 5×10^{-324} (but some compromise needed)

- **Still limited (relative) precision, e.g. estimating `(v+1)-v` can be 0 !**

Endianness

- In which order should you interpret the bytes and bits?
- Differences
  - in software & hardware
  - in types of data (integer, float,…)
- Source of problems!



Big-endian Little-endian

https://en.wikipedia.org/wiki/Endianness

Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

Signal discretization

Some continuous values =

1. measured by some instrument, (image reconstruction), and
2. stored numerically

→ discretized value with **finite resolution**!

Numerical data/image

Function I associates a value v at discrete coordinates

(x, y, z, t, s) in a finite hyper-rectangle with regular sampling :

$$I: X \times Y \times Z \times T \times S \mapsto V: (x, y, z, t, s) \mapsto v$$

- $x \in \{0, \cdots, W-1\}$: horizontal coordinates (W = width)
- $y \in \{0, \cdots, H-1\}$: vertical coordinates (H = height)
- $z \in \{0, \cdots, T-1\}$: slice index (T = thickness).
- $t \in \{0, \cdots, L-1\}$: time in sequence/series (L = length).
- $s \in \{0, \cdots, S-1\}$: canal (S = number of samples).

Examples

$$I: X \times Y \times Z \times T \times S \mapsto V: (x, y, z, t, s) \mapsto v$$

- 2D image (pixels) : $\mathrm{I}(x, y) \Longrightarrow$ RX, photography, microscopy.
- 3D image (voxels) : $\mathrm{I}(x, y, z) \Longrightarrow$ CT, MRI, PET.
- 2D+t series : $\mathrm{I}(x, y, t) \Longrightarrow$ Ultra-sound, videos (frames).
- 3D+t image series : $\mathrm{I}(x, y, z, t) \Longrightarrow$ functional MRI (frames).
- Temporal signal : $\mathrm{I}(x, t, s) \Longrightarrow$ EEG, ECG.
- Multi-channel 2D image : $\mathrm{I}(x, y, s) \Longrightarrow$ microscopy.

Signal discretization

Some continuous values =

1. measured by some instrument, (image reconstruction), and
2. stored numerically

→ discretized value with **finite resolution**!

Two faces of "resolution" → Different file weight!

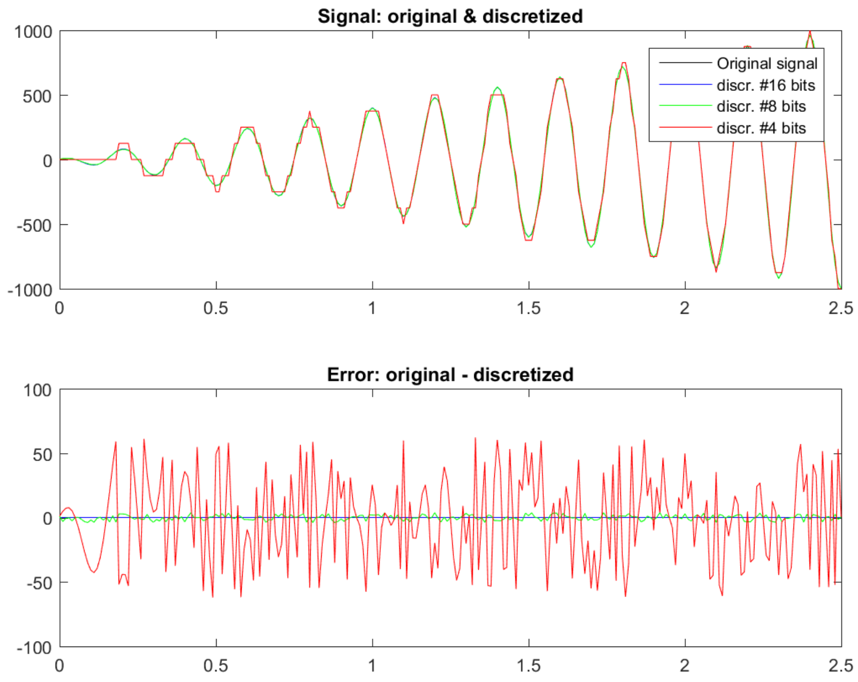
- time/space → sampling rate
- amplitude → encoding precision

Encoding precision

How is the value represented on disk?

- Integer vs. float?
- Number of bytes?

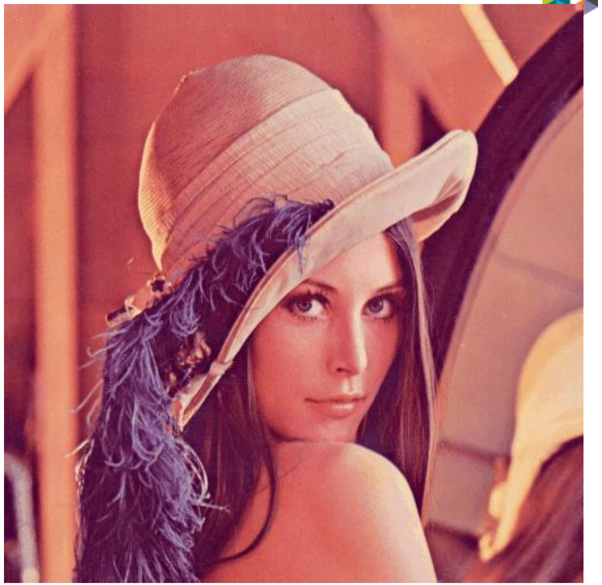
→ Different resolution



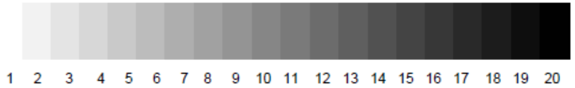
Colour depth

- Grey levels,
  e.g. 1 byte/pixel or “8bpp”
- Colour → 3 channels
  e.g. 3 bytes/pixel or “24bpp”

2 valeurs de quantification (1 bit) | 4 valeurs de quantification (2 bits) | 256 valeurs de quantification (8 bits)



échelle continue



échelle discrète

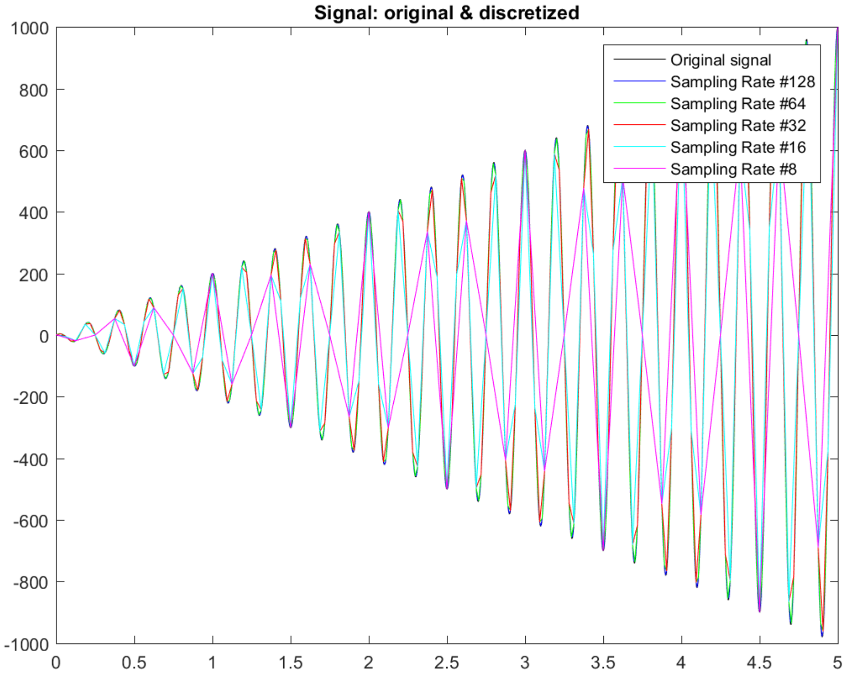
Sampling rate

How sparse/coarse are data sampled?

→ sampling rate

→ Nyquist theorem:

“Sampling Rate
> 2 x highest frequency of signal”



Example for 3D image

Consider a 3D image with 256 x 256 x 128 = 2^{23} voxels

- 1 `int16` per voxel → 16 Mb
- 1 `float32` per voxel → 32 Mb

Coloured image

→ 3 RGB values par voxel, e.g. 3 `int8` per voxel → 24 Mb

Resample at half the resolution, i.e. 128 x 128 x 64 voxels

→ divide sizes by 8

Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

File format

Open vs. closed file format:

- fully described vs. proprietary
- openly readable vs. requiring specific software
- community supported vs. software/company dependent

→ Stick to open format whenever possible

→ More flexibility to use with homemade software

The case of MS Word & Excel

Both are proprietary and cost €€€ + files are "binarized"

- Word & `.doc` files, replace by
  → 'MarkDown' (`.md`) files
  → open editor/reader, e.g. Typora (https://typora.io/, not free any more though)
- Excel & `.xls` files , replace by
  → 'comma-separated value' or 'tab-separated value' (`.csv`/`.tsv`) files
  → open editor/reader, e.g. CSVed (https://csved.sjfrancke.nl/)

Whenever possible and appropriate

DataComments.md - Typora

File Edit Paragraph Format View Themes Help

Some comments about the data.

Overall ~79Gb: (~58k files & 208 folders)

- MSHS, 37Gb, 37 subjects
- MSPA, 40Gb, 40 subjects
- MSP FLAIR/mask, 2.5Gb, 40 subjects

MSPA:

possibly to exclude.
s08825. Rather visible movement artefacts. Poor positioning in scanner -> cerebellum out of FOV?
s00349. Some movement artefact + hyper-instensities (artefact) in orbito-frontal area for MT.
s00356. hyper-instensities (artefact) in orbito-frontal area for MT + small meningiome between the frontal hemispheres.

DataComments.md - Typora

File Edit Paragraph Format View Themes Help

```
## Some comments about the data.

Overall ~79Gb: (~58k files & 208 folders)
- MSHS, 37Gb, 37 subjects
- MSPA, 40Gb, 40 subjects
- MSP FLAIR/mask, 2.5Gb, 40 subjects


#### MSPA:


possibly to exclude.
**s08825**. Rather visible movement artefacts. Poor positioning in scanner -> cerebellum out of FOV?
**s00349**. Some movement artefact + hyper-instensities (artefact) in orbito-frontal area for MT.
**s00356**. hyper-instensities (artefact) in orbito-frontal area for MT + small meningiome between the frontal hemispheres.
```

Excel in Genetics

"Gene name errors are widespread in the scientific literature"

Abstract:

The spreadsheet software Microsoft Excel, when used with default settings, is known to **convert gene names to dates and floating-point numbers**. A programmatic scan of leading genomics journals reveals that **approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions**.

Ziemann et al., Genome Biology 201617:177

Structured data

Data as

- key/value pairs
- hierarchical structure

→ use 'JavaScript Object Notation', i.e. `.json`, files

Other option: YAML


```yaml
--- # The Smiths
- {name: John Smith, age: 33}
- name: Mary Smith
  age: 27
- [name, age]: [Rae Smith, 4]   # sequences as keys are supported
--- # People, by gender
men: [John Smith, Bill Jones]
women:
  - Mary Smith
  - Susan Williams
```


Example, `task-Nback_bold.json`


```json
{
    "RepetitionTime": 3.0,
    "EchoTime": 0.0003,
    "FlipAngle": 78,
    "SliceTiming": [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8],
    "MultibandAccellerationFactor": 4,
    "ParallelReductionFactorInPlane": 2
}
```

Data compression

Lossless:

- no data/signal lost $\rightarrow$ replace “patterns” with fewer bytes (RLE).
- 2-4x compression rate, depending on data
- e.g. ZIP, PNG, JPEG2000

Lossy:

- Removes some signal $\rightarrow$ irreversible loss!
- quality factor from 0 to 100 $\rightarrow$ >10x compression rate
- e.g. JPEG

Data compression

Very useful for quick (pre-)visualisation!

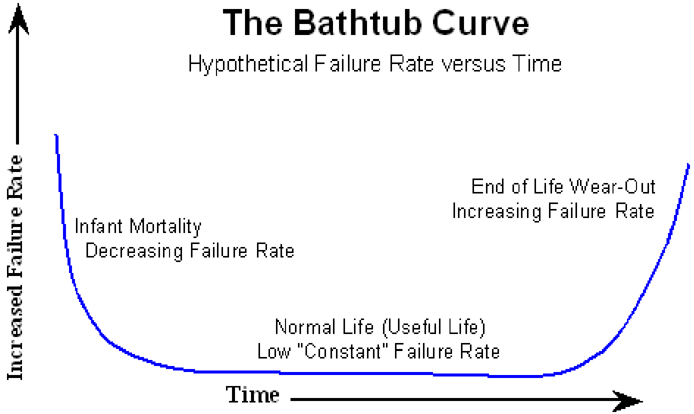
Program

- Bits & bytes
- Data format
- Signal discretization
- File format & compression
- Storage & Safety

Hard-disk drive

HDD = electromechanical data storage device:

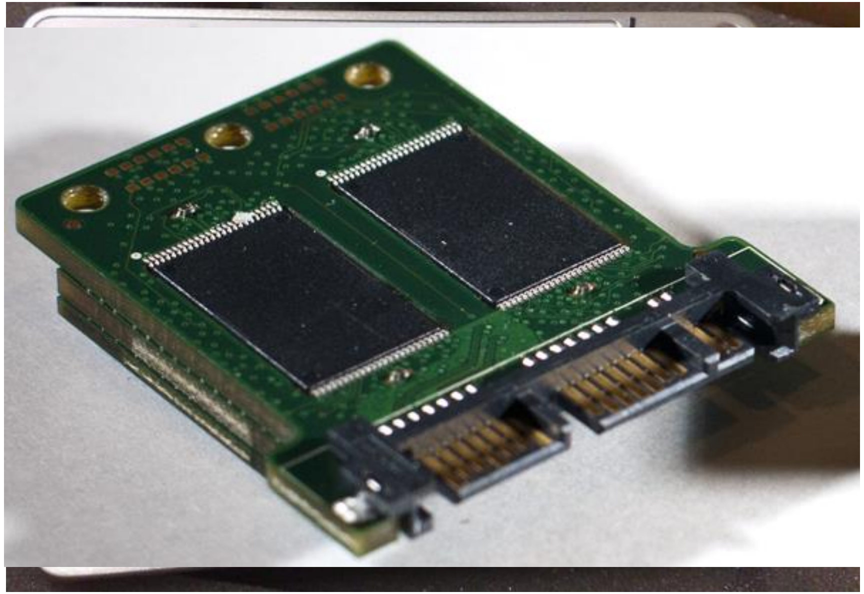
- magnetic storage to read/write data
- on one (or more rigid) rapidly rotating disks
- cheap and storage density increases (Moore's law)
- latency = ~a few ms,
- transfer rate up to ~1 Gb/s
- risk of failure increases with time but…



Solid-state drive

SSD = integrated circuit data storage device:

- non-volatile NAND flash memory to read/write data
- no mechanical or moving part
- latency < ms,
- transfer rate up to a few Gb/s
- compared to HDD
  - more expensive and more reliable
  - less power consumption



ULiège mass-storage

- Personal space → your own stuff
- Platform space → raw data access
- Team space → shared data & results

Keep in mind access time

→ no direct processing of data!

Backup vs. Archive

Backup

- copy of **current** data/system
- includes files which are currently being accessed/changed

→ Restoring data/system to a previous point in time, if they are lost or become corrupted

Archive

- store data/information to be kept for a long period of time
- includes files which should not be modified, accidentaly or purposely

→ Restoring the ‘original’ data/information, e.g. to re-analyse them

Local vs. Remote storage

Local, e.g. USB drive

- Cheap and easy
- Can be lost or corrupted with the rest of the computer

→ Better than nothing but not so safe!

Remote, e.g. institutional mass-storage

- More expensive (for the institution/users) and more constraining (network access)
- Little risk of losing anything (tapes, redundant disks, multi-sites,…)

→ Safest option, if available

For code, use versioning → more on Monday October 10!

References

- https://en.wikipedia.org/wiki/Bit
- https://en.wikipedia.org/wiki/Byte
- https://en.wikipedia.org/wiki/Character_(computing)
- https://en.wikipedia.org/wiki/ASCII
- https://en.wikipedia.org/wiki/UTF-8
- https://en.wikipedia.org/wiki/Integer_(computer_science)
- https://en.wikipedia.org/wiki/Single-precision_floating-point_format
- https://en.wikipedia.org/wiki/Endianness
- https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

References

- https://en.wikipedia.org/wiki/Markdown
- https://typora.io/
- https://en.wikipedia.org/wiki/Comma-separated_values
- https://en.wikipedia.org/wiki/Tab-separated_values
- https://csved.sjfrancke.nl/
- https://en.wikipedia.org/wiki/JSON
- https://en.wikipedia.org/wiki/YAML
- https://doi.org/10.1186/s13059-016-1044-7
- https://en.wikipedia.org/wiki/Run-length_encoding
- https://en.wikipedia.org/wiki/JPEG
- https://en.wikipedia.org/wiki/Hard_disk_drive
- https://en.wikipedia.org/wiki/Solid-state_drive


SUPER MARIO BROS.
SUPER MARIO BROS. 2
SUPER MARIO BROS. 3
SUPER MARIO WORLD
1985
1988
1990
1991
SUPER MARIO 64
SUPER MARIO SUNSHINE
SUPER MARIO GALAXY
SUPER SMASH BROS. for Wii U
1996
2002
2007
2014

LIÈGE université
GIGA

Thank you for your attention!

LIÈGE université
GIGA