AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Information Technologies

Bartosz Ziółko, Jakub Gałka

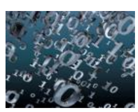
Technology

- Pragmatic purpose (Value)
- Methods
- Tools
- Processes
- Knowledge
- Production, processing of goods
- Technical, non-technical

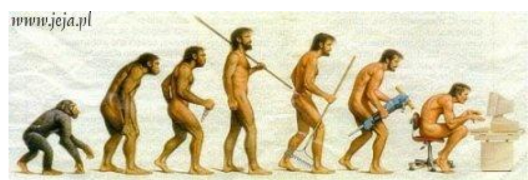
What IT is?

Generally, integration of a few technologies: hardware, software, telecommunications

- applied for collecting information,
- analysis,
- processing,
- security,
- managment,
- transfering (serving) information.



Homo sapiens, homo communicans



Information society

„An **information society** is a society in which the creation, distribution, diffusion, use, integration and manipulation of information is a significant economic, political, and cultural activity. ”

-Wikipedia

Information

IT Areas and context

- Information Theory
- Hardware – computing, access, storage, IoT
- Software – enterprise, consumer, embedded
- Internet – data access, communication, data generation, service platform

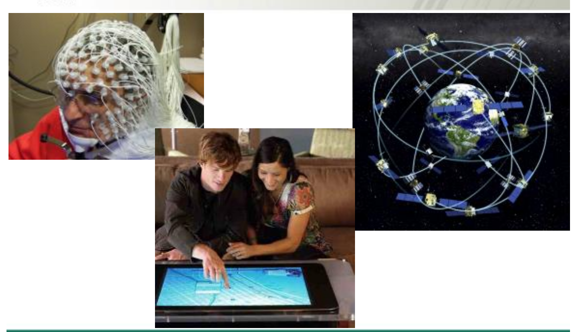
IT Areas and context

- IT Businesses, Business Models, SaaS, PaaS
- Innovation, Startups, Intellectual Property
- Internet of Things, Industry 4.0
- Cloud, Big Data, AI
- API, API Economy
- Social Aspects, Opportunities, Threats

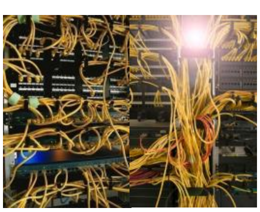
Lectures

- Aspects of information society and information technologies
- Information storage and processing
- Computer hardware
- Computational and programming techniques
- Telecommunications, multimedia, human - computer interaction and IT
- Test and student presentations

Why IT ?

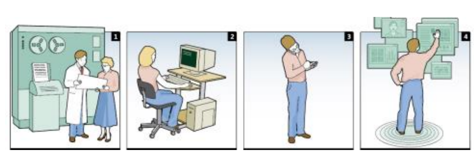


Why IT?



- All major companies depend on IT services (banks, telecommunications, media, insurance, trade, etc.),
- IT specialists earn in average more then other specialists,
- Even more will depend on IT and on people who work with IT in near future.

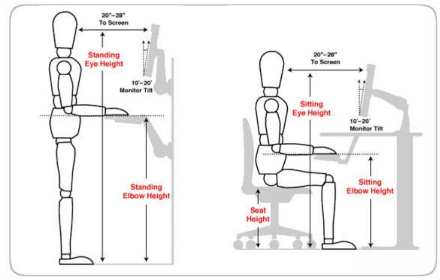
2020 – the era of 1 user and 100 computers



Topics

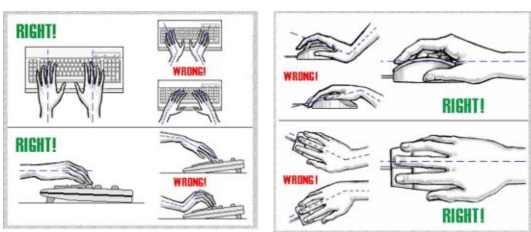
- Computer ergonomics
- Fundamentals of health and safety while working with a computer and on the campus
- Copyrights and patents
- Searching for information in order to self-educate and problem solving
- Ethical principles in the academic community and among the engineers
- Different types of word processors
- The principles of good multimedia presentation

Computer ergonomics



17

Computer ergonomics



18

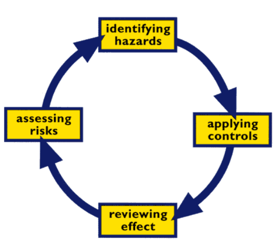
Health and safety

You should be trained regarding H&S for each laboratory class you attend

Law dictates that your future employer is responsible for your safety and health

20

Health and safety



21

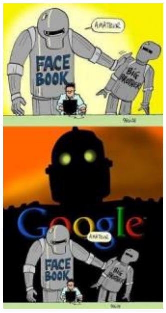
Health and safety

Several students and staff members were seriously or even fatally wounded on AGH university properties

22

AGH

Privacy policies in Internet



23

AGH

Copyright ©

is a set of exclusive rights granted by the law of a jurisdiction to the author or creator of an original work, including the right to copy, distribute and adapt the work.

prawa autorskie vs. autorskie prawa majątkowe

books, maps, charts, engravings, prints, musical compositions, dramatic works, photographs, paintings, drawings, sculptures, motion pictures, computer programs, sound recordings, choreography and architectural works.

24

AGH

Presentation
IT areas and contexts

- Algorithms, Data Structures, Programming Paradigms
- Databases, Communication protocols, formats and standards
- Hardware – computing, storage, transmission
- Cloud, Big Data, Machine Learning AI
- API, API Economy
- Internet of Things, Edge Computing
- IT Businesses, Business Models, SaaS / PaaS
- Innovation, Startups, Intellectual Property, OpenSoftware
- Project Management, Teams, Methodology, Tools (Gitlab)
- Social Aspects, Opportunities, Threats

AGH

Project

- Web Page
  - CSS/HTML5
  - Wordpress
  - PHP / SQL, JavaScript
  - UX Design – Content, SEO
  - UX Design - Graphics (stock / CC)
- Mobile Application
- IT-related YouTube Video – how to make it viral?

AGH

34% of US GDP is generated by intelectual property trade.

AGH

Fundamentals in patent law and strategy

Based on Jeffrey Schox

What are the types of intellectual property?
What are the sections of a patent?
What is the patent process?
Why should we file for a patent?
What is an invention?
Who is an inventor?
When should we file?
Where should we file?

What are the types of intellectual property?

Trademarks	$2 000
Utility Patents	$20 000
Design Patents	$2 000
Copyrights	$200
Trade secrets	$2

Sections of a patent

Front page	–	bibliographic
Description	–	technical
Claims	–	legal

Patent process

1. File patent application.
2. Await examination.
3. Negotiate with patent office.

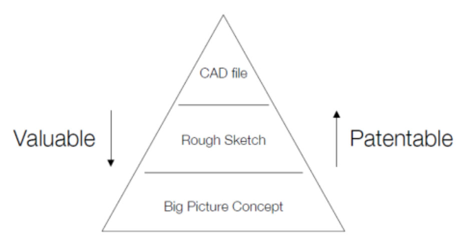
Why should we file for a patent?

Why should we file for a patent?

- Encourage investment.
- Protect competitive advantage.
- Avoid patent litigation.

What is an invention?

What is an invention?



Who is an inventor?

Who is an inventor?

„Someone who contributed to the original conception of the claimed invention.”

When should we file?

Where should we file?

Where should we file?

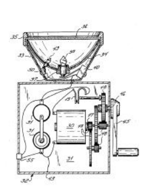
- Made and used / sold
- US, EU, CN, JP, KR, AU

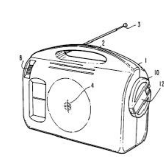
Requirements of a patent

Requirements of a patent

- Useful
- Novel
- Not an obvious combination of known inventions

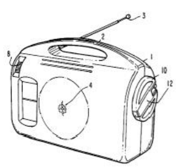
Is this invention patentable?





AGH

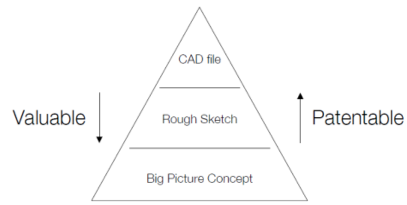
Is this invention patentable?



AGH

Summary

An invention may be patentable if claims can be written that distinguish the invention from previous inventions.



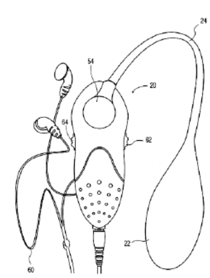
AGH

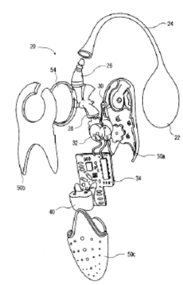
Infringement of a patent

Whoever makes, uses or sells any patented invention within a particular region during a particular timeframe.

AGH

Would this product infringe?





AGH

Claim 11 of U.S. Pat. No. 5,917,310 states:
A portable electric device comprising:
- a generator having a rotatable shaft;
- a gear train … coupled to said rotatable shaft;
- a mechanical energy storage device….;
- a handle coupled to said source, whereby rotating said handle adds mechanical energy to said source;
- a power control circuit coupled to said generator and configured to prevent an uncontrolled release of said mechanical energy from said source by controlling a voltage output of said generator; and
- a radio coupled to said power control circuit.

AGH

Claim 11 of U.S. Pat. No. 5,917,310 states:
A portable electric device comprising:
- a generator having a rotatable shaft;
- a gear train … coupled to said rotatable shaft;
- a mechanical energy storage device….;
- a handle coupled to said source, whereby rotating said handle adds mechanical energy to said source;
- a power control circuit coupled to said generator and configured to prevent an uncontrolled release of said mechanical energy from said source by controlling a voltage output of said generator; and
- a radio coupled to said power control circuit.

Summary

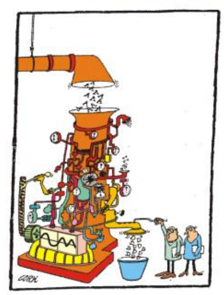
A product (or method) may infringe a patent if it contains all of the elements of a single claim of the patent.

is this patentable?
does it infringe?

Topics

- Fundamentals of information
- Basic methods and formats of data in a computer
- Analogue, digital, discrete signals and their conversion
- Sampling, quantization, Nyquist frequency
- Spectrum, bandwidth
- Shannon-Kotielnikov theory and aliasing
- Number systems, representations of real and negative numbers in computers
- Boolean logic, gates and De Morgan's laws

Bits, data, information



Information Theory

„I wrote this one [a letter] a bit longer, because I did not have time to write it shorter" – Blaise Pascal (1623–1662), Lettres Provinciales

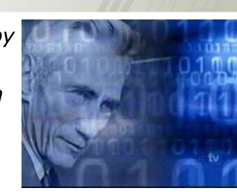
Information theory characterises recording, storing and using information in a mathematical way. There are two main aims (opposite ones):

- to save a carrier,
- to keep the information safe during transmission.

Algorithms operating on numbers, are much easier and faster then those operating on words.

Information measure

Information is measured by a probability of an event related to that information
– Claude E. Shannon



Entropy

Less likely events carry more information

Information measure

- A unit of information is such its amount which we gain after realising that one of two equally probable events occured.
- Message ***m,*** which can occur with probability ***p(m),*** contains

$$h(m) = \log_2\left(\frac{1}{p(m)}\right)$$

units of information.

- Let us define a unit of information to be a **bit (*binary digit)*** if its source can transmit only one message, which contains

$$h(m) = \log_2\left(\frac{1}{1}\right) = 0$$

bits of information.

after Zbigniew Bem

Bit

Simpler, bit is the amount of information that can be stored by a digital device or other physical system that can usually exist in only two distinct states, traditionally called 0 and 1.

Byte and Word

1	0	0	1	1	0	0	0

Bit

MSB - most significat bit

Byte

LSB - least significat bit

Byte is a sequence of 8 bits used in computers as a basic unit, on which the operations are conducted.

1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0

Word

Word is a sequence of 8, 16, 32 or 64 bits.

Bytes and words

In IT we often need 16 digits.
A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

Word size	The biggest number
8 bits	255
16 bits	65535
32 bits	4 294 967 295
64 bits	18 446 744 073 709 551 615

Letters as numbers – ASCII codes

ASCII Code Chart

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

- Wikipedia

Unicode

UTF-8 (8-bit Unicode Transformation Format) is a variable-length character encoding for Unicode. UTF-8 can represent every character in the Unicode character set, and is backward-compatible with ASCII.

The dominant character encoding for files, e-mail, web pages and software that manipulates textual information.

ISO 8859-1 Latin - 1

Standard for Western European fonts.

Very similar to Windows-1252.

Table 2 – Code table of Latin alphabet No. 1

b4 b3 b2 b1		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
b8		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
b7		0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
b6		0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
b5		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
0 0 0 0	00			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð	0
0 0 0 1	01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ	1
0 0 1 0	02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò	2
0 0 1 1	03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó	3
0 1 0 0	04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô	4
0 1 0 1	05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ	5
0 1 1 0	06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö	6
0 1 1 1	07			'	7	G	W	g	w			§	·	Ç	×	ç	÷	7
1 0 0 0	08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø	8
1 0 0 1	09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù	9
1 0 1 0	10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú	A
1 0 1 1	11			+	;	K	[	k	{			«	»	Ë	Û	ë	û	B
1 1 0 0	12			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü	C
1 1 0 1	13			-	=	M	]	m	}			SHY	½	Í	Ý	í	ý	D
1 1 1 0	14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ	E
1 1 1 1	15			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ	F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

ISO 8859-2

Central and Eastern European fonts

ISO/IEC 8859-2:1999

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	*Znaki kontrolne*															
1x																
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x	*Nieużywane*															
9x																
Ax	NBSP	Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź	SHY	Ž	Ż
Bx	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
Cx	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
Dx	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
Ex	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
Fx	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	˙

Decimal numeral system

The system is positional

$a_j \dots a_4 a_3 a_2 a_1 a_0 . a_{-1} a_{-2}$

where a is a symbol, and j is an exponent of the base r

number → $a_j \dots a_1 a_0 . a_{-1} a_{-2}$

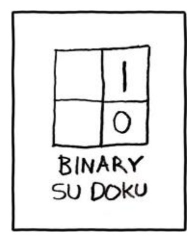
$a_j \cdot r^j + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2}$

Example:

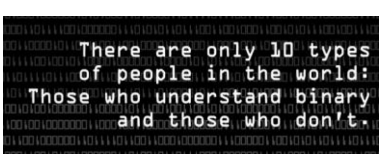
$4321 = 4 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0 =$

$= 4000 + 300 + 20 + 1 = 4321_{10}$

after Wojciech Kucewicz



Representing numbers



Binary Numeral System

It uses only 2 symbols

0 and **1**

The number of combinations is 2^n for **n** digits.

Example:

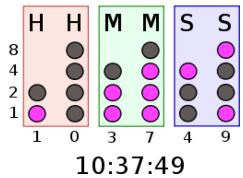
1000011100001_2 :

$1 \cdot 2^{12} + 0 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^{9} + 0 \cdot 2^{8} + 1 \cdot 2^{7} + 1 \cdot 2^{6} + 1 \cdot 2^{5} + 0 \cdot 2^{4} + 0 \cdot 2^{3} + 0 \cdot 2^{2} + 0 \cdot 2^{1} + 1 \cdot 2^{0}$

$= 4096+128+64+32+1= 4321_{10}$

After Wojciech Kucewicz

Binary watch



Hexadecimal system

Hexadecimal number system uses 16 symbols:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

The combination of numbers is 16^n for **n** digits.

Example:

$10E1_{16}$

$\mathbf{1 \bullet 16^3} + 0 \bullet 16^2 + 14 \bullet 16^1 + 1 \bullet 16^0$

$= 4096+224+1= 4321_{10}$

Numbers in different systems			
Decimal	Binary	Octal	Hexadecimal
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Binary systems for +/- integers

Sign and magnitude $A_{SM} = -1^{b_{n-1}} \cdot \sum_{i=0}^{n-1} b_i \cdot 2^i$

U1 (One’s complement) $A_{U1} = -b_{n-1} \cdot (2_{n-1} - 1) + \sum_{i=0}^{n-2} b_i \cdot 2^i$

U2 (Two's complement) $A_{U2} = -b_{n-1} \cdot 2_{n-1} + \sum_{i=0}^{n-2} b_i \cdot 2^i$

You can easily find the one's complement of a binary number by inverting the number (changing 1s into 0s and 0s into 1s). To determine the two's complement of a number, first take the ones's complement of the number and then add 1 to this number.

Binary systems - comparison

Sign & Magintude: -5 + 2 1101
-5: 1101, 2: 0010 +0010

U1: (invert bits) 1010
-5: 1**010**, 2: 0010 +0010
1100 = - 3

U2: (add 1 to U1) 1011
-5: 101**1**, 2: 0010 +0010
110**1**=(-4**+1**)$_{10}$=-3

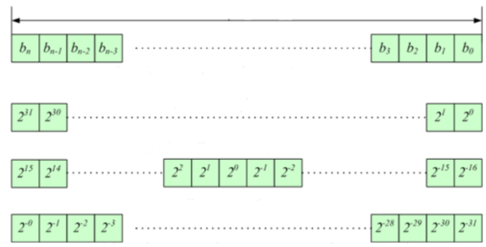
Comparision of binary integer systems

Type	Binary notification			Byte		16-bit word		32-bit word	
	min	zero	max	min	max	min	max	min	Max
NBC	00..0	00..0	11..1	0	255	0	65535	0	4294967295
U2	10..0	00..0	01..1	-128	127	-32768	32767	-2147483648	2147483647
SM	11..1	00..0 10..0	01..1	-127	127	-32767	32767	-2147483647	2147483647
U1	10..1	00..0 10..0	01..0	-127	127	-32767	32767	-2147483647	2147483647
Biased	00..0	01..1	11..1	-127	128	-32767	32768	-2147483647	2147483648

Fractions in binary

Fraction	Decimal	Binary	Fractional Approx.
1/1	1 or 0.9999...	1 or 0.1111...	1/1
1/2	0.5	0.1	1/2
1/3	0.333...	0.010101...	1/4+1/16+1/64...
1/4	0.25	0.01	1/4
1/5	0.2	0.00110011...	1/8+1/16+1/128...
1/6	0.1666...	0.0010101...	1/8+1/32+1/128...
1/7	0.142857142857...	0.001001...	1/8+1/64+1/512...
1/8	0.125	0.001	1/8
1/9	0.111...	0.000111000111...	1/16+1/32+1/64...
1/10	0.1	0.000110011...	1/16+1/32+1/256...
1/11	0.090909...	0.0001011101000101101...	1/16+1/64+1/128...
1/12	0.08333...	0.00010101...	1/16+1/64+1/256...
1/13	0.07692376923...	0.000100111011000100111011...	1/16+1/128+1/256...
1/14	0.0714285714285...	0.0001001001...	1/16+1/128+1/1024...
1/15	0.0666...	0.00010001...	1/16+1/256...
1/16	0.0625	0.0001	1/16

Binary real numbers



Floating point standard

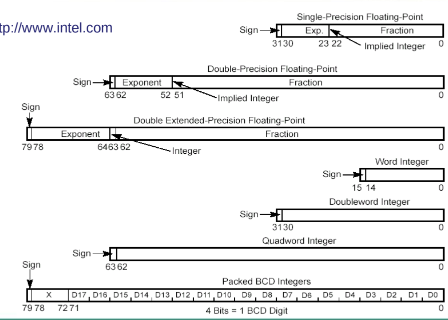
Floating point describes a system for representing numbers that would be too large or too small to be represented as integers.

significant digits × base^{exponent}

- The base for the scaling is normally 2, 10 or 16.
  It can support a much wider range of values.
- Fixed-point representation (7 decimal digits with 2 decimal places), can represent the numbers 12345.67, 123.45, 1.23 and so on.
- Floating-point representation with 7 decimal digits could in addition represent 1.234567, 123456.7, 0.00001234567, 1234567000000000, and so on.

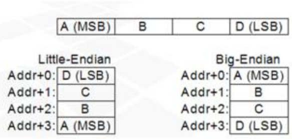
Types of data in Intel x87 *Floating Point Unit*

http://www.intel.com



Data in computer memory

- A size of a word can be larger then single byte — for example modern computers can operate on 64-bit words. It means that a particular instruction can modify 64 bits at once.
- The memory architecture is addresed in a binary way.

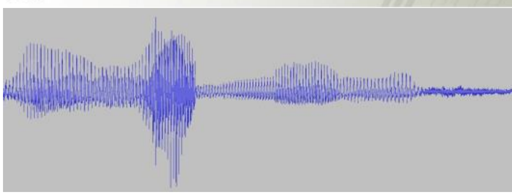


SPARC, Motorola 68000, PowerPC 970, IBM System/360, Siemens SIMATIC S7

Intel x86, AMD64, DEC VAX More natural for a computer then human

Data and signals

Sounds

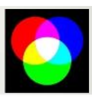


For a computer it is a sequence of numbers representing temporary acoustic pressure sampled with some frequency (8 kHz, 48 kHz, …)

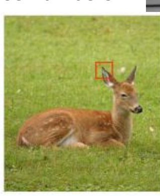
Black and white images

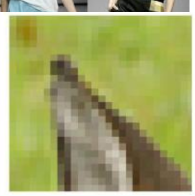
Images can be represented as a matrix of points (pixels).

Colour images

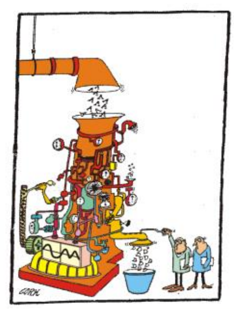


Colour images are represented by 3 matrices. All of them have the same size, equalled to the size of a screen. Each pixel is represented by three numbers.





Analog to Digital

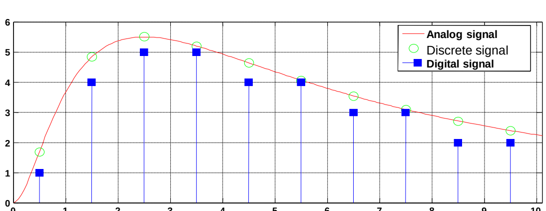


Analog signal
can have any values.
Real world is analog.
They are represented by differentiable functions (continuous).

Discrete signal
is defined for particular points in time, usually because of sampling.
Typical for analysis of real world.

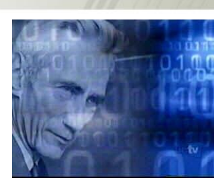
Digital signal
is discrete and in addition can take only particular values.

Example



Kotielnikov – Shannon Theory

Котельников 1933

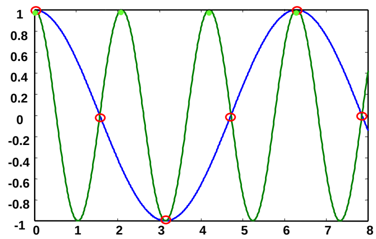


Shannon 1949

They provided mathematical conditions (and proved them) for digital-to-analog being possible.

$$\frac{1}{\Delta t} \overset{df}{=} f_s \geq 2 f_{max},$$

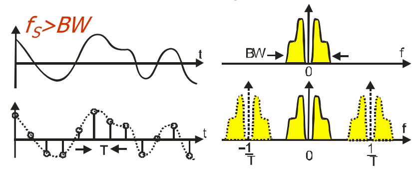
Can a sampled signal be restored to analog one?



Aliasing

Nyquist Frequency

f_s – sampling frequency - ratio of information
BW – BandWidth of a signal

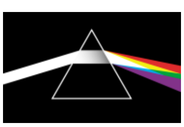


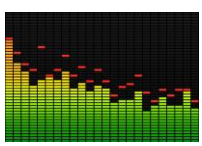
Harry Nyquist (1889-1960)

„The number of independent pulses that could be put through a telegraph channel per unit time is limited to twice the bandwidth of the channel". Nyquist: *Certain topics in Telegraph Transmission Theory* (1928).

What a spectrum (spectre ?) is ?

- It may present any signal in a frequency domain (Hz)
- Allows better analysis for signal processing and transmission

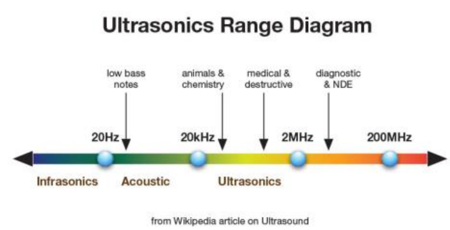




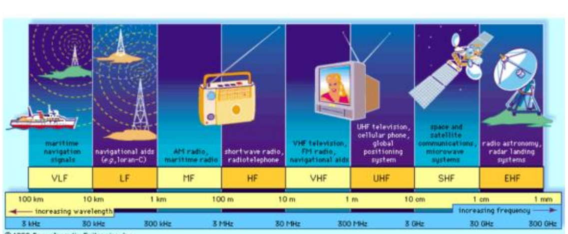
Bands

- Transmission band f_T
- Signal band f_S

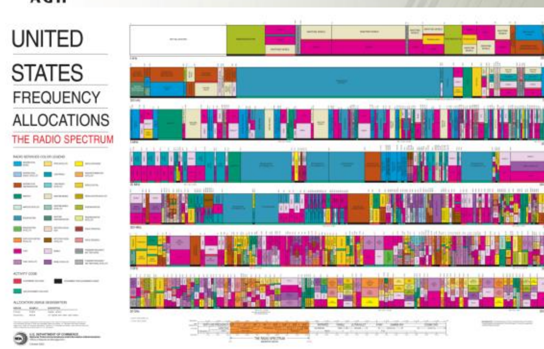
$f_T > f_S$



Transmission bands



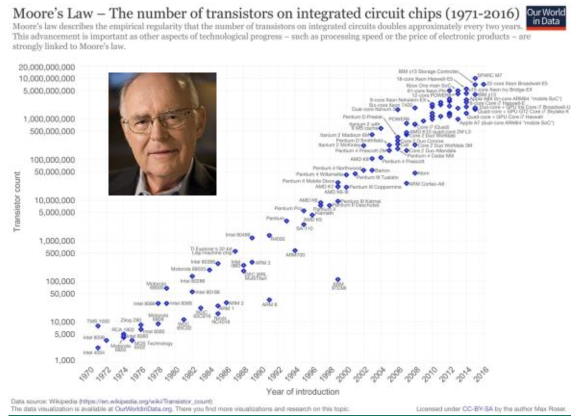
Frequency is limited, so expensive



AGH

Introduction to computer hardware

- Moore's law
- The first computing devices
- Turing machine
- Von Neumann architecture
- The development of personal computers
- Computer memory types
- Elements of a computer
- Computer peripherals



AGH

How many are a billion ?

billion = 1000 * 10^6

It is a lot, but how many actually?

A pile of money notes would have

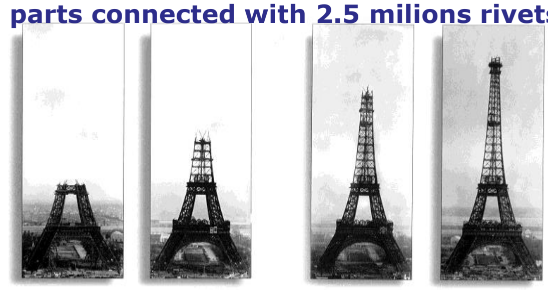
76.2 km of height

1 billion seconds are 32.5 years!

AGH

Comparison

The Eiffel Tower consists of 18 084 parts connected with 2.5 milions rivets



after *Software defined radio*, Frideric Harris

AGH

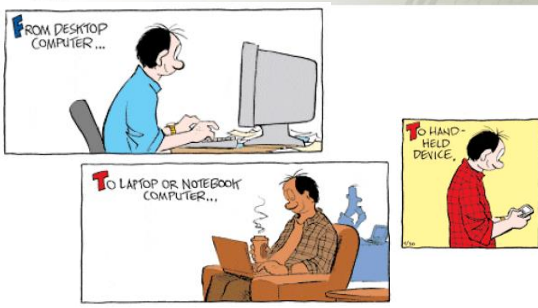
Do we eat a lot of rice ?

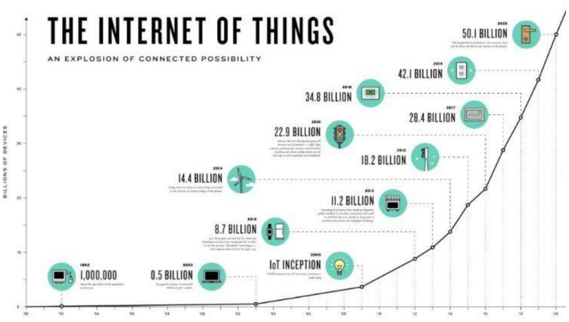
There are more transistors being produced then rice grains all over the world!

Intel i7 silicon wafer

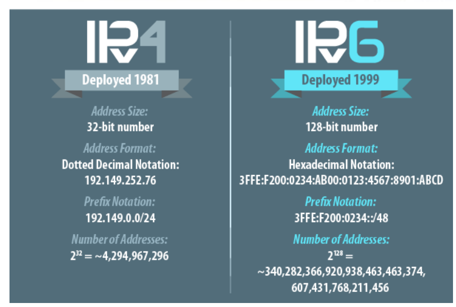
Long Grain Jasmine Rice

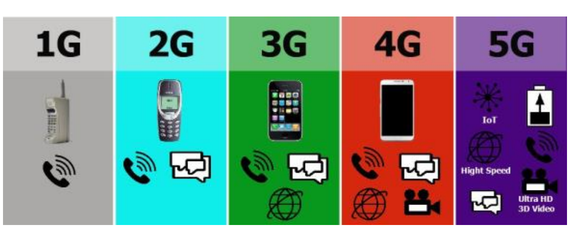
Evolution of computers



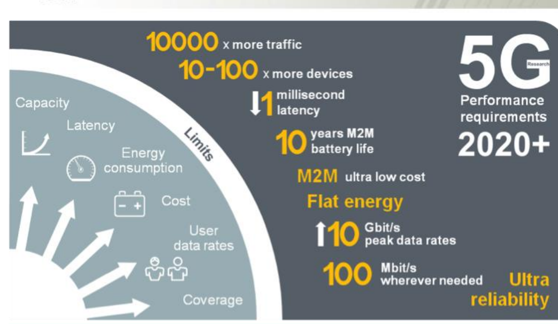


IPv4 & IPv6

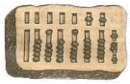




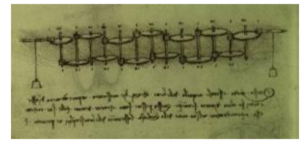
- `80s `91 `01 `09 2020+ ?



History of computations




Codex Madrid, XVI Century


Logarithms

John Napier of Merchiston (1550 – 1617)

Napier’s bones

Slide rule

Invented in 1632 by Williama Oughtreda.
Improved by E. Wingate, S. Patridge and A. Mennheim.

In common use up to 80’.


A shot from „Apollo 13”

Wilhelm Schickard (1592-1635)

Based on Neper’s bones
Used by Kepler

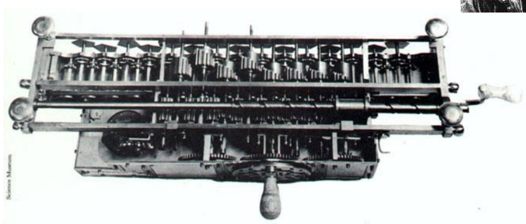
4 basic arithmetic operations

Blaise Pascal (1623-1662)

Made to help in tax calculations


Gottfied Leibniz (1646 – 1716)

Binary system
1st multiplying mechanical machine

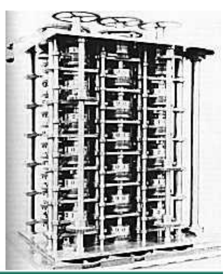


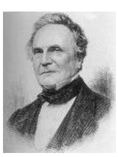
Joseph-Marie Jacquard (1752-1834)

1st programmable (!) machine
loom - weaving

Charles Babbage (1791 - 1871)

- Concept of a programmable computer
- Differential computations
- 31-digit precision
  - how many bits?
- Analytical machine concept

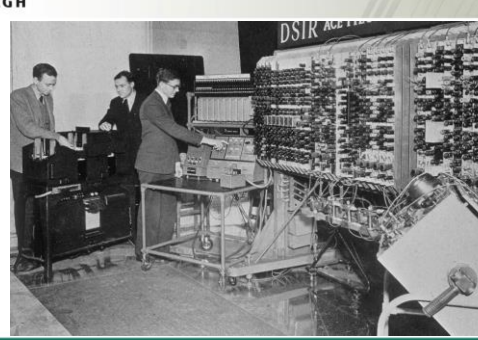




Herman Hollerith (1860-1929)

- Applying electricity to computations
- Punched card
- Hollerith built machines under contract for the Census Office

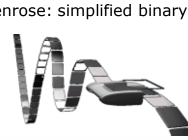
Alan Turing



Alan Turing (1912-1954)

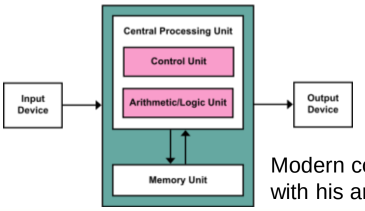
- In 1936 revolutionary work on the theory of computation machines and algorithms.
- Turing declared that his machine is able to conduct any algorithm. It was never disproven.

- Infinite tape
- Each field – N states
- M states of machine
- Atom instructions
- Penrose: simplified binary version



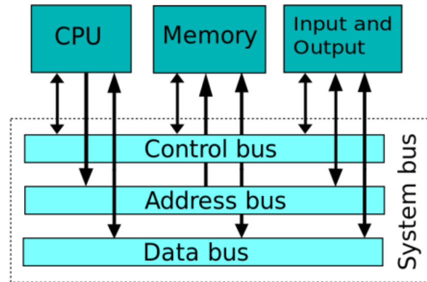
John von Neumann (1903-1957)

1946 - **EDVAC project (Electronic Discrete Variable Automatic Computer)**



Modern computers are built with his architecture

Developments of Von Neumanns architecture

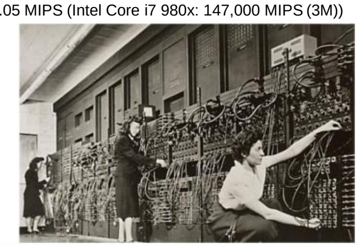


Harvard computing architecture

- Physically separate signals and storage for code and data memory.
- It is possible to access program memory and data memory simultaneously.
- DSP, uC

ENIAC, 1945-55

- Parallel computing
- Separate memory and computing units
- 27-30 metric tons, 20000 vacuum tubes
- 0.05 MIPS (Intel Core i7 980x: 147,000 MIPS (3M))



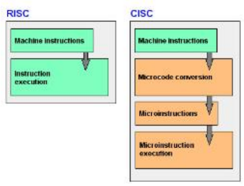
1947 – transistor (William Bradford Shockley, J. Bardeenem and W.H. Brattainem) – Nobel prize
1953 – IBM 650, first mass production computer (using punched cards)

1955 - Bell Telephone Labs made the first computer based on transistors
1956 – First hard drive (IBM)

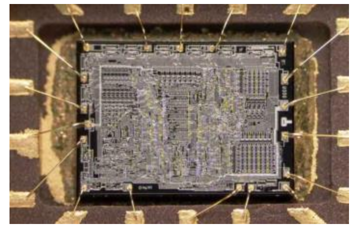
What is it?

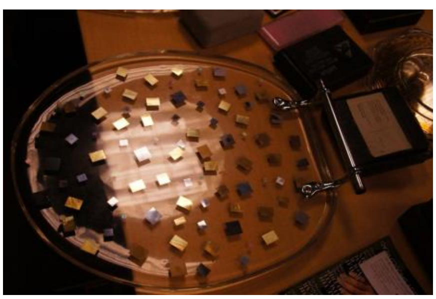
CISC vs RISC architecture

- Complex Instruction Set Computer
- Reduced Instruction Set Computer

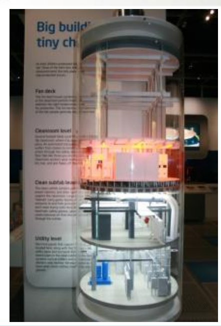


1972 - Intel 8008 (200 kHz) - 1st 8-bit processor





Intel museum



Intel museum

Jan Czochralski (1885-1953)

- Czochralski proces (1915)
- Crystal growing method
- Used in semiconductor industry

Microsoft 1975

BASIC for Atari
1977 - Microsoft provides BASIC to almost every new microcomputer: Apple, Commodore, Radio Shack, …

more then 1000 computers all over the world

ZX Spectrum

+ amazing software library
+ its design
+ graphic possibilities
+ its success
+ its price

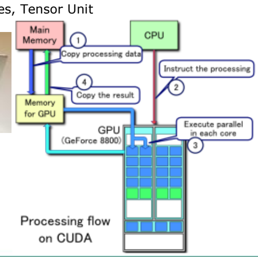
- The rubber keyboard
- Poor sound

Amiga 500

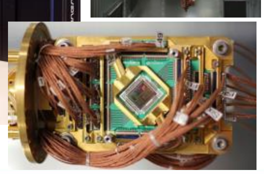
A computer with 16- bit procesor 68000 (1987)

GP-GPU

- General Purpose – Graphical Processing Unit
- CUDA computing (Compute Unified Device Architecture), OpenCL
- TESLA 1 (128 cores)
- TESLA T4 (TURING arch) up to 5120 cores, Tensor Unit



Quantum computers D-WAVE



Quantum computing

- Data is stored as quantum states
- Qubit – neither 0 nor 1
- quantum superposition of probabilities of 0 and 1

- Quantum algorithm - design of quantum evolution of the system
- Parallel problem solving (like #cracking, NP problems, etc)

- System is unstable (uncoherence)

IBM Q – available as API

IBM Q Experience is an online platform that gives users in the general public access to a set of IBM's prototype quantum processors via the Cloud

Random-access memory (RAM)

Integrated circuits that allow stored data to be accessed in any order (i.e., at random).

„Random" refers to the idea that any piece of data can be returned in a constant time, regardless of its physical location and whether or not it is related to the previous piece of data.

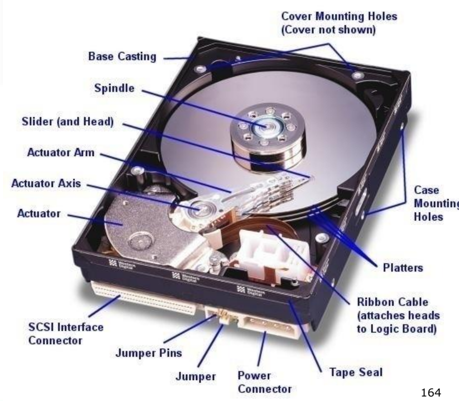
Volatile – requires power

Read-only memory (ROM)

It is mainly used to distribute firmware (software that is very closely tied to specific hardware, and unlikely to require frequent updates).

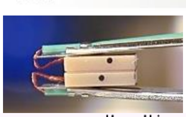
Non-volatile

EPROM - can be erased and re-programmed multiple times. (electrically erasable programmable read-only memory)

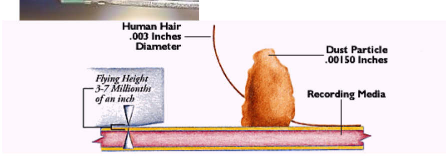


164

Hard disk



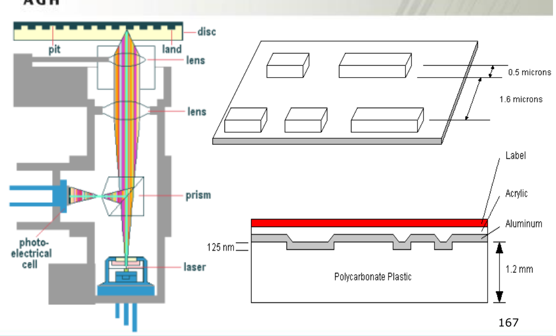
A pair of mated head sliders with their platter removed. You can see that the tension of the head arms has caused them to press against each other.



This illustration gives you some idea of just how small the flying height of a modern hard disk is (and today's hard disks have flying heights significantly lower than 3-7 millionths of an inch!

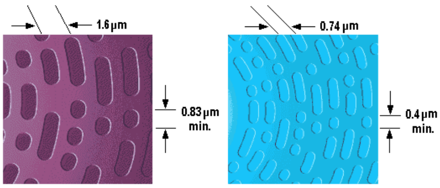
166

CD



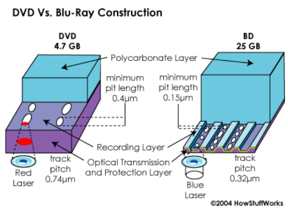
167

DVD Digital Versatile Disc



Blu-Ray

25 GB (single-layer)
50 GB (dual-layer)



Blu-ray Disc™

Flash Memory

- Based on EEPROM
- Developed by Toshiba
- Uses NAND (eg. USB, SSD) or NOR gates
- Metal-oxidated Field Effect Floating Gate Transistors (MOSFET)

- Developed by Toshiba
- Can be written or erased in blocks
- Flash cells get older after each write

- Samsung expects to present 100TB SSD in 2020 (8TB now @ CES Las Vegas)

Introduction to computational and programming techniques

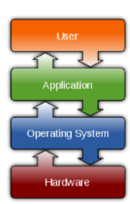
- Operational systems
- Parallel processing
- Supercomputing

Software

- Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to produce bigger and better idiots. So far, the universe is winning.
  Rick Cook, *Wizardry Compiled*

Software

Software is the collection of computer programs and related data that provide the instructions telling a computer what to do.



- Application software
- Middleware controls and coordinates distributed systems
- Programming languages and tools
- System software
- Testware
- Firmware is treated like hardware and run by other software programs
- Device drivers

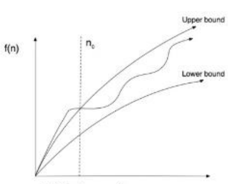
BIOS (Basic Input/Output System)

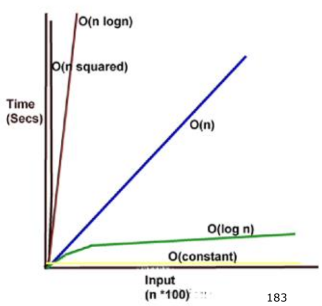
- Firmware
- the first code run by a PC when powered on (booting)
- the primary function of the BIOS is to load and start an operating system
- to initialise and identify system devices such as the video display card, keyboard and mouse, hard disk, CD/DVD drive and other hardware
- locates software held on a peripheral device (designated as a 'boot device'), such as a hard disk or a CD, and loads and executes that software, giving it control of the PC.

180

Operating systems

Big O notation



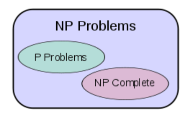


183

Complexity of problems

The abbreviation **NP** refers to "nondeterministic polynomial time„

Intuitively, **NP** is the set of all decision problems for which the 'yes'-answers have efficiently verifiable proofs of the fact that the answer is indeed 'yes'. More precisely, these proofs have to be *verifiable* in polynomial time ($T(n) = O(n^k)$ for some constant k by a deterministic Turing machine.

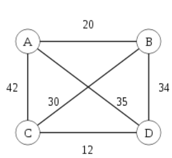


For example, an algorithm that runs for 2^n steps on an input of size n requires superpolynomial time (more specifically, exponential time).

! (Factorial)

184

Traveling salesman problem (NP)



185

FLOPS, MIPS

- **FLOPS** (FLoating point Operations Per Second) – a unit of measuring computer computation efectiveness. It is a number of floating point operations per second.
  - 1 MFLOPS
  - 1 GFLOPS
  - 1 TFLOPS
  - 1 PFLOPS
- **MIPS** (Million Instructions Per Second) – a measure of CPU effectiveness. It describes the number of millions of fixed point operations per second made by the particular unit.
- A similar one is Milion Operations per Second (**MOPS**).

192

What are supercomputers used for?

- Simmulation of new chemical particles, catalysts, etc.
- Designing new medicaments,
- Studying proteins,
- Geological calulations,
- Simmulations in physics, especially nuclear,
- Virtual experiments,
- Linguistic calculations,
- Network simmulations.

193

Supercomputer – Cray, 50 MFLOPS

CRAY-1 in EPFL (École Polytechnique Fédérale de Lausanne)

194

Supercomputer – Cray-2, 1.9 GFLOPS

195

Supercomputer – Cray-XC50

TFLOPS

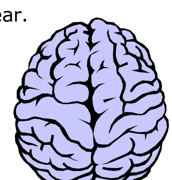
Swiss National Supercomputing Centre (CSCS) Switzerland	Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	387,872	21,230.0	27,154.3	2,384

196

TOP 500

- A website presenting a list of 500 most efficient computers was established in 1993 year:
  - Computational power is measured using LINPACK benchmark.
  - It is updated 2 times per year.
- www.top500.org

We estimate, that a brain is able to conduct around quadrillion (10^{15}) operations per second.



197

Cyfronet Prometheus

Cyfronet - Prometheus

- The most powerful supercomputer in Poland

			Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
130	National Computational Infrastructure National Facility (NCI-NF) Australia	**Raijin** - Hybrid Cluster PRIMERGY/x3550 M5/PowerEdge/Rackable/Rackable, Xeon , Infiniband EDR, NVIDIA Tesla P100, Xeon Phi Fujitsu / Lenovo / Xenon	87,224	1,676.2	3,801.4	
131	Cyfronet Poland	**Prometheus** - HP Apollo 8000, Xeon E5-2680v3 12C 2.5GHz, Infiniband FDR, NVIDIA Tesla K40 HPE	55,728	1,670.1	2,348.6	808
132	Government China	**01** - ThinkSystem Flex SN550, Xeon Gold 6150 18C 2.7GHz, Infiniband FDR Lenovo	30,960	1,668.2	2,674.9	

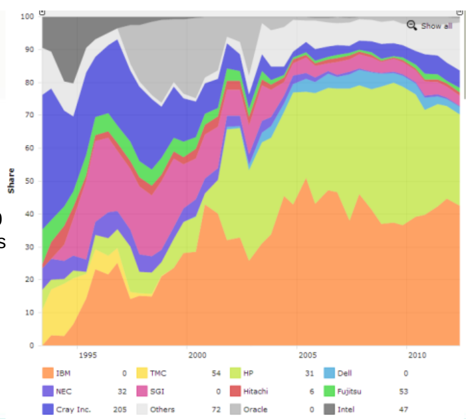
Cyfronet Prometheus Running CentOS 7

Model	#Server	CPUs	Cores/ node	Clock	RAM/ node	comments
HPE ProLiant XL730f Gen9	2160	2x Intel Xeon E5-2680v3	24	2,50 GHz	128 GB	
HPE ProLiant XL750f Gen9	72	2x Intel Xeon E5-2680v3	24	2,50 GHz	128 GB	2x Nvidia Tesla K40 XL
HPE ProLiant DL360 Gen10	2	2x Intel Xeon Gold 6128	12	3,4 GHz	768 GB	
HPE ProLiant DL360 Gen10	1	2x Intel Xeon Gold 6128	12	3,4 GHz	1536 GB	

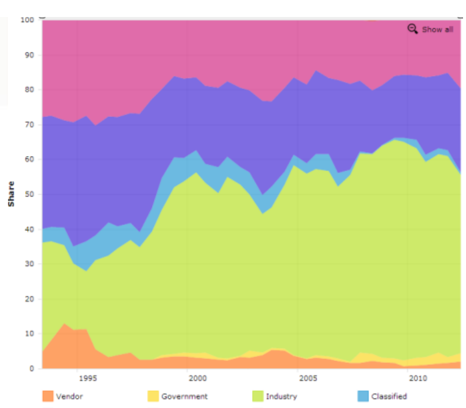
Rank	Site	System	Cores	(TFlop/s)	(TFlop/s)	(kW)
1	DOE/SC/Oak Ridge National Laboratory United States	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	2,397,824	143,500.0	200,794.9	9,783
2	DOE/NNSA/LLNL United States	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM / NVIDIA / Mellanox	1,572,480	94,640.0	125,712.0	7,438
3	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10,649,600	93,014.6	125,435.9	15,371
131	Cyfronet Poland	**Prometheus** - HP Apollo 8000, Xeon E5-2680v3 12C 2.5GHz, Infiniband FDR, NVIDIA Tesla K40 HPE	55,728	1,670.1	2,348.6	808
132	Government China	**01** - ThinkSystem Flex SN550, Xeon Gold 6150 18C 2.7GHz, Infiniband FDR Lenovo	30,960	1,668.2	2,674.9	

Used to be 35th, in 2015

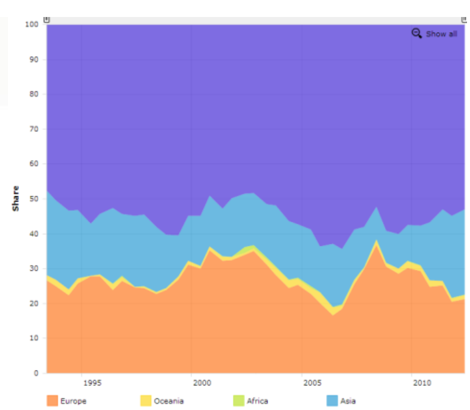
TOP 500 providers

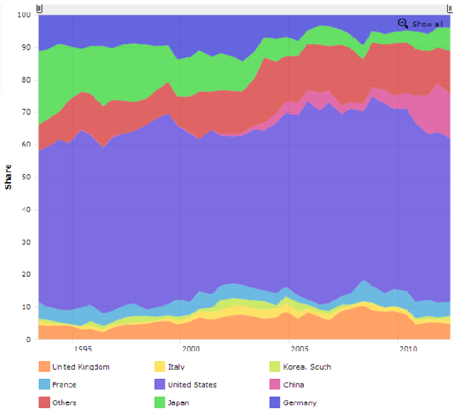


segments



continents





countries

Computer clusters

- Applications with databases are not easly paralelised,
- Clusters are architectures highly integrated, localised in one building,
- Used for safety,
- And for efficiency as well.

Columbia, (2004) supercomputer,
20 clusters SGI Altix ,
10240 CPU in total

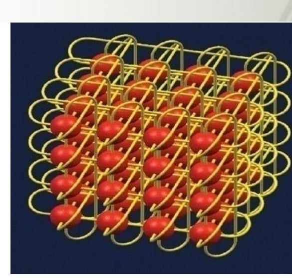
216

Clusters

- Efficient PCs as nodes,
- GNU/Linux,
- Paraller processing software (MPI, PVM).

217

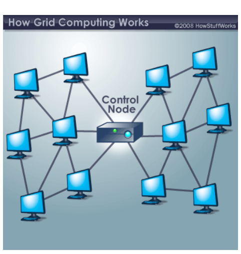
Clusters



BlueGene/L uses a three-dimensional (3D) torus network

218

Grid computing

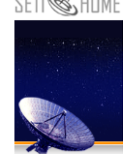


Many computers in a network to solve a single problem at the same time.

219

First virtual constructs similar to grids

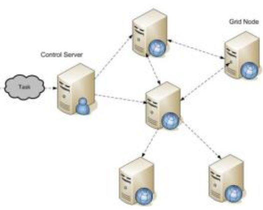
- GIMPS (Great Internet Mersenne Prime Search)
- SETI@home (Search for Extra-Terrestrial Intelligence)
- Berkeley Open Infrastructure for Network Computing http://boinc.berkeley.edu/



220

AGH

Grid computing



Requires the use of software that can divide and farm out pieces of a program to as many as several thousand computers.

Grid computing can be thought of as distributed and large-scale cluster computing and as a form of network-distributed parallel processing.

221

AGH

Examples of grids

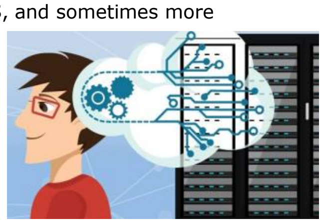
222

AGH

PL Grid

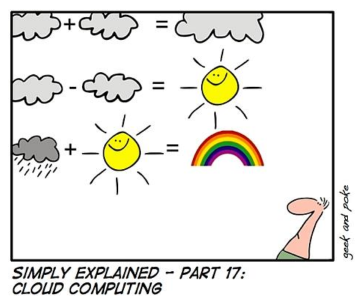
- Integrates Polish supercomputers
- For researchers (and students)
- +5PFLOPS, and sometimes more

PL GRID



AGH

Cloud computing ? ? ?



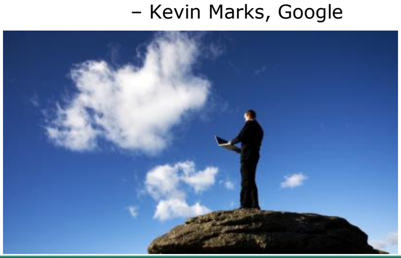
229

AGH

Cloud computing

„We didn't care where the messages went… the cloud hid it from us."

– Kevin Marks, Google



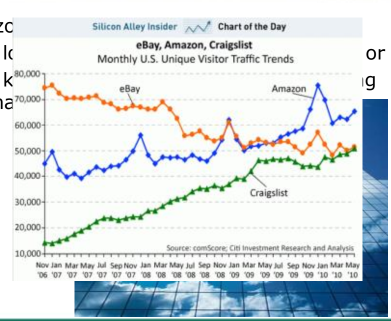
230

AGH

How did it start?

Amazo

- To lo … or
- To k … g Xma



231

How did it start?

Amazon had a problem:
- To loose potential buyers during Xmas … or
- To keep computers being used only during Xmas

232

How did it start?

They solved it by renting out computers when they don't use them, but physically keeping them. To achieve it, they had to be able to do things Amazon wouldn't expect people may need.

233

Cloud computing

- computing / data access paradigm
- services and data reside in shared resources in scalable data centers,
- services and data are accessible by any authenticated device over the Internet (API).
- one the most significant trends today
- reduction of costs

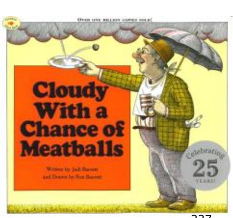
234

Most popular clouds

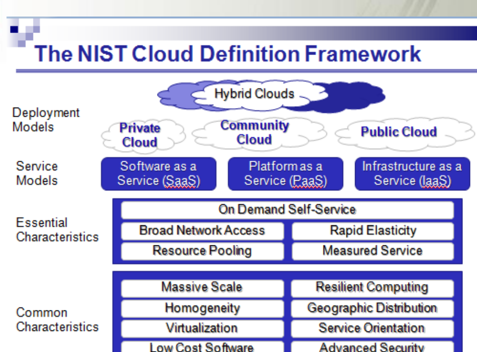
amazon web services
Baidu Cloud
Google Cloud
Microsoft Azure
IBM Bluemix
Alibaba Cloud
intel Cloud Technology
NVIDIA GPU CLOUD

Essential Cloud Characteristics

- On-demand self-service
- Broad network access
- Resource pooling
  - Location independence
- Rapid elasticity
- Measured service



237



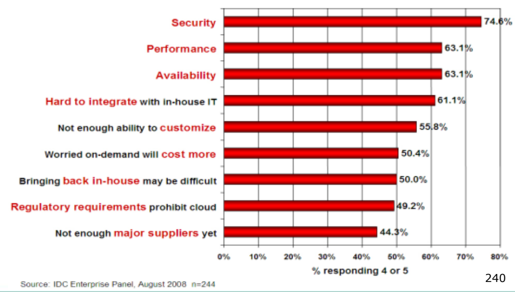
238

Clouds are in containers

- Clouds can be quickly built using shipping containers, pulled by trucks, and parked near electric utilities and rivers.
- Medium sized data center requires 50 mega-watts and evaporates 4M Litre of "chilled" fresh water / day. Conditioned air is also needed. Also, this does not consider bandwidth concerns.
- Result: Physical targets

239

Q: Rate the challenges/issues ascribed to the 'cloud'/on-demand model
(1=not significant, 5=very significant)



Source: IDC Enterprise Panel, August 2008 n=244

240

Possible effects of cloud computing

- Small enterprises use public SaaS and public clouds and minimise growth of data centers.
- Large enterprise data centers may evolve to act as private clouds.
- Large enterprises may use hybrid cloud infrastructure software to leverage both internal and public clouds.
- Public clouds may adopt standards in order to run workloads from competing hybrid cloud infrastructures.

241

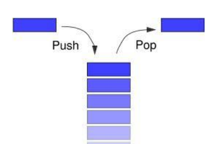
Data structures & Databases

Data structures

- Stack
- Queue
- List
- Tree

243

Stack



http://www.csanimated.com/animation.php?t=Stack

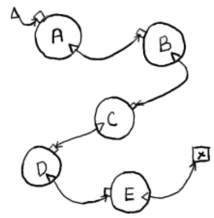
244

Queue

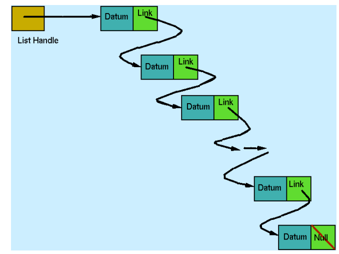
http://www.csanimated.com/animation.php?t=Queue

245

List



Linked List



http://www.csanimated.com/animation.php?t=Linked_list

246

Tree

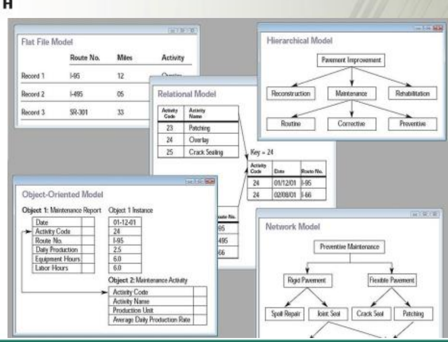
Tree

247

Databases

248

Database models



249

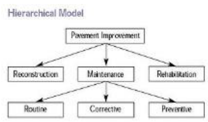
Flat files

- good for non-technicians
- simple to open and interprete
- encoding issues
- hard to navigate
- no indexing
- loose schema
- low expressiveness

250

Hierarchical model

In a hierarchical model, data is organised into a tree-like structure, implying a single upward link in each record to describe the nesting, and a sort field to keep the records in a particular order in each same-level list.



251

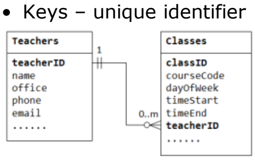
Relational model

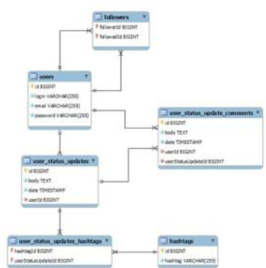
The basic data structure of the relational model is the table, where information about a particular entity (say, an employee) is represented in rows and columns. Thus, the "relation" in "relational database" refers to the various tables in the database; a relation is a set of rows. The columns enumerate the various attributes of the entity (the employee's name, address or phone number, for example), and a row is an actual instance of the entity (a specific employee) that is represented by the relation. As a result, each row of the employee table represents various attributes of a single employee.

252

Relational model

- Tables (rows of columns)
- Relation – set of rows
- Rows - entities
- Columns – atributes
- Keys – unique identifier





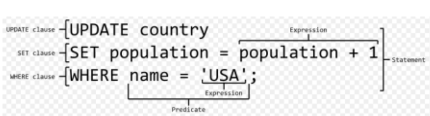
Key in a relational model

Tables can also have a designated single attribute or a set of attributes that can act as a "**key**", which can be used to **uniquely identify** each row in the table.

ID
Email
PESEL
...

254

SQL (Structured Query Language)



SELECT *
FROM Book
WHERE price > 100.00
ORDER BY title;

255

SQL Database engines

- SQLite
- PostgreSQL
- MS SQL
- Oracle

Relational database vs. object oriented databases

- well known to developers
- strict schema
- support for transactions, concurrency, disaster recovery, etc.
- SQL
- client-server architecture

Think of a library (with books) or an encyclopedia. You don't need all the information, but you might need any piece of it.

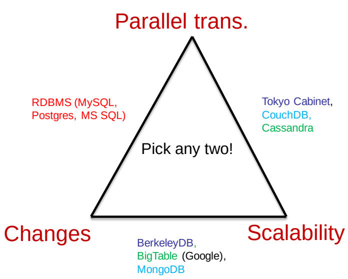
257

SQL vs noSQL

SQL – Structured Query Language – relational data bases (MySQL, Ms SQL, Oracle, SQLite ...)

NoSQL - „not only SQL" - all other – **key-value store**, document store, graph DB, object DB, tabular ,… (BerkeleyDB, BigTable, CouchDB, Tokyo Cabinet, HODB …)

Choice of database engine



Relational
Key-Value
Tabular
Document

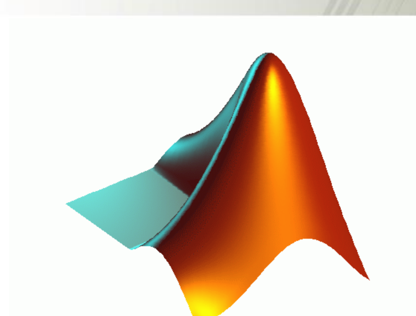
Calculations and programming

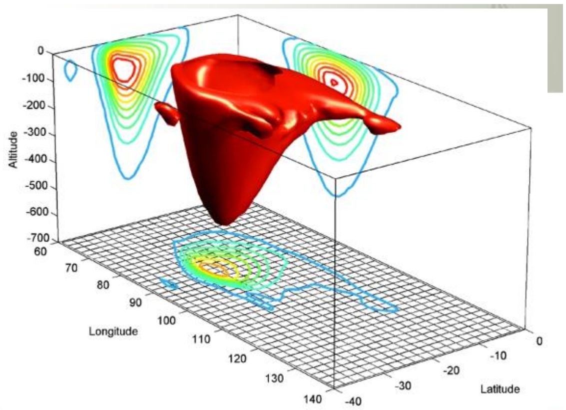
Calculations

- Mathematica
- Matlab
- Python
- C/C++

- GPU, CUDA
- FPGA

Matlab





Matlab code example

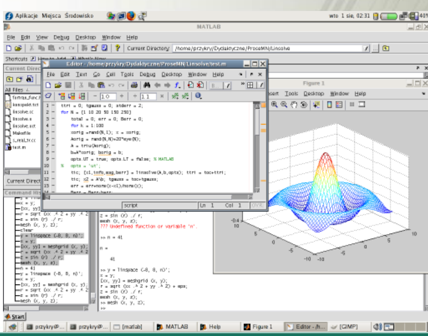
```
function [mean,stdev] = stats(vals)
% #codegen

% calculates a statistical mean and a standard
% deviation for the values in vals.

len = length(vals);
mean = avg(vals,len);
stdev = sqrt(sum(((vals-avg(vals,len)).^2))/len);
coder.extrinsic('plot');
plot(vals,'-+');

function mean = avg(array,size)
mean = sum(array)/size;
```

Matlab



Matlab Toolboxes

http://www.mathworks.com/products/product_listing/

http://www.mathworks.com/help/

Matlab alternatives:
Octave, Scilab, Python

Real computations should be in C/C++

- Matlab, and several other tools are designed for allow easy work and fast prototyping, but not for computational efficiency
- No compilation (interpreted languages)
- No computations on cache
- Doubly embeded loops are very slow comparing to compiled C/C++ software
- Matlab, etc. are for prototypes and research mainly

Graphics processing unit (GPU)

- is a specialized microprocessor that offloads and accelerates 3D or 2D graphics rendering from the microprocessor,
- very fast,
- very multithread,
- bad acess to memory
- nowadays often used for non-graphics calculations

GP-GPU Architecture features

- Many generic cores
- Less space for control logic
- Large registers
- Low-latency thread management
- Large numer of ALU per core
- Small cache per core
- High-bandwidth memory bus to ALUs

Nvidia and AMD

GP-GPU

- Compute Capability

GPU Computing Applications

Libraries and Middleware

cuDNN TensorRT	cuFFT, cuBLAS, cuRAND, cuSPARSE	CULA MAGMA	Thrust NPP	VSIPL, SVM, OpenCurrent	PhysX, OptiX, iRay	MATLAB Mathematica

Programming Languages

C	C++	Fortran	Java, Python, Wrappers	DirectCompute	Directives (e.g., OpenACC)

CUDA-enabled NVIDIA GPUs

	EMBEDDED	CONSUMER DESKTOP, LAPTOP	PROFESSIONAL WORKSTATION	DATA CENTER
Turing Architecture (Compute capabilities 7.x)	DRIVE/JETSON AGX Xavier	GeForce 2000 Series	Quadro RTX Series	Tesla T Series
Volta Architecture (Compute capabilities 7.x)	DRIVE/JETSON AGX Xavier			Tesla V Series
Pascal Architecture (Compute capabilities 6.x)	Tegra X2	GeForce 1000 Series	Quadro P Series	Tesla P Series
Maxwell Architecture (Compute capabilities 5.x)	Tegra X1	GeForce 900 Series	Quadro M Series	Tesla M Series
Kepler Architecture (Compute capabilities 3.x)	Tegra K1	GeForce 700 Series GeForce 600 Series	Quadro K Series	Tesla K Series

Programming GP-GPU concepts

- CUDA - Compute Unified Device Architecture
- OpenCL –Open Computing Language

- Kernel functions
- Blocks – kernels running in paralel
- Grid – set of blocks running in parallel
- Threads
- Streams
- Host / Device
- Shared Memory, Constant memory, …

GP-GPU scalability using SMs

- Array of Streaming Multiprocessors (SM)
- Multithread program is divided in to blocks of threads so more SMs=faster execution

Simple CUDA example in C

```c
#define N 10000000

void vector_add(float *out, float *a, float *b, int n) {
   for(int i = 0; i < n; i++){
      out[i] = a[i] + b[i];
    }
}

int main() {
   float *a, *b, *out;
   // Allocate memory
   a = (float*)malloc(sizeof(float) * N);
   b = (float*)malloc(sizeof(float) * N);
   out = (float*)malloc(sizeof(float) * N);
   // Initialize array
   for(int i = 0; i < N; i++){
      a[i] = 1.0f; b[i] = 2.0f;
    }

   // Main function
   vector_add(out, a, b, N);
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <cuda.h>
#include <cuda_runtime.h>

#define N 10000000
#define MAX_ERR 1e-6

__global__ void vector_add(float *out, float *a, float *b, int n) {
    for(int i = 0; i < n; i ++){
        out[i] = a[i] + b[i];
    }
}

int main(){
    float *a, *b, *out;
    float *d_a, *d_b, *d_out;

    // Allocate host memory
    a   = (float*)malloc(sizeof(float) * N);
    b   = (float*)malloc(sizeof(float) * N);
    out = (float*)malloc(sizeof(float) * N);

    // Initialize host arrays
    for(int i = 0; i < N; i++){
        a[i] = 1.0f;
        b[i] = 2.0f;
    }

    // Allocate device memory
    cudaMalloc((void**)&d_a, sizeof(float) * N);
    cudaMalloc((void**)&d_b, sizeof(float) * N);
    cudaMalloc((void**)&d_out, sizeof(float) * N);

    // Transfer data from host to device memory
    cudaMemcpy(d_a, a, sizeof(float) * N, cudaMemcpyHostToDevice);
    cudaMemcpy(d_b, b, sizeof(float) * N, cudaMemcpyHostToDevice);

    // Executing kernel
    vector_add<<<1,1>>>(d_out, d_a, d_b, N);
```

```
nclude <stdlib.h>
nclude <math.h>
nclude <assert.h>
nclude <cuda.h>
nclude <cuda_runtime.h>

efine N 10000000
efine MAX_ERR 1e-6

_global__ void vector_add(float *out, float *a, float *b, int n) {
  for(int i = 0; i < n; i ++){
      out[i] = a[i] + b[i];
  }

t main(){
  float *a, *b, *out;
  float *d_a, *d_b, *d_out;

  // Allocate host memory
  a   = (float*)malloc(sizeof(float) * N);
  b   = (float*)malloc(sizeof(float) * N);
  out = (float*)malloc(sizeof(float) * N);

  // Initialize host arrays
  for(int i = 0; i < N; i++){
      a[i] = 1.0f;
      b[i] = 2.0f;
  }

  // Allocate device memory
  cudaMalloc((void**)&d_a, sizeof(float) * N);
  cudaMalloc((void**)&d_b, sizeof(float) * N);
  cudaMalloc((void**)&d_out, sizeof(float) * N);

  // Transfer data from host to device memory
  cudaMemcpy(d_a, a, sizeof(float) * N, cudaMemcpyHostToDevice);
  cudaMemcpy(d_b, b, sizeof(float) * N, cudaMemcpyHostToDevice);

  // Executing kernel
  vector_add<<<1,1>>>(d_out, d_a, d_b, N);
```

```
  // Executing kernel
  vector_add<<<1,1>>>(d_out, d_a, d_b, N);

  // Transfer data back to host memory
  cudaMemcpy(out, d_out, sizeof(float) * N, cudaMemcpyDeviceToHost);

  // Verification
  for(int i = 0; i < N; i++){
      assert(fabs(out[i] - a[i] - b[i]) < MAX_ERR);
  }
  printf("out[0] = %f\n", out[0]);
  printf("PASSED\n");

  // Deallocate device memory
  cudaFree(d_a);
  cudaFree(d_b);
  cudaFree(d_out);

  // Deallocate host memory
  free(a);
  free(b);
  free(out);
}
```

GPU using interpreted languages

Matlab GPU example

```
A = gpuArray([1 0 1; -1 -2 0; 0 1 -1]);
e = eig(A);
```

Python GPU example

```
>>> import pycuda.gpuarray as gpuarray
>>> import pycuda.autoinit
>>> import numpy as np
>>> from skcuda import linalg
>>> linalg.init()
>>> a = np.array(np.random.rand(3,3), np.float32, order='F')
>>> a_gpu = gpuarray.to_gpu(a)
>>> w_gpu = linalg.eig(a_gpu, 'N', 'N')
>>> e = w_gpu.get()
```

Field Programmable Gate Array (FPGA)

- **FPGA** is an integrated circuit designed to be configured by the customer or designer after manufacturing—hence "field-programmable".
- Configuration is generally specified using a hardware description language (HDL)
- Reprograming is slow
- Calculations are fast

Programing languages

- 1 gen – machine languages
- 2 gen – assembler
- 3 gen – Lisp, Cobol, Fortran, C, C++, Java
- 4 gen – C++ - object oriented, Python, Java, C# and .Net, SQL
- 5 gen – eg. AI oriented approach

Gen 1 & 2 – Machine and Assembly

- Machine: binary (hex) code
- Assembly: Symbolic, Assembling
- Direct use of Registers and Instruction set
- Programming – writing the binary code directly into ROM

Machine code bytes	Assembly language statements
	foo:
B8 22 11 00 FF	movl $0xFF001122, %eax
01 CA	addl %ecx, %edx
31 F6	xorl %esi, %esi
53	pushl %ebx
8B 5C 24 04	movl 4(%esp), %ebx
8D 34 48	leal (%eax,%ecx,2), %esi
39 C3	cmpl %eax, %ebx
72 EB	jnae foo
C3	retl

Instruction stream

```
B8 22 11 00 FF 01 CA 31 F6 53 8B 5C 24
04 8D 34 48 39 C3 72 EB C3
```

3 gen – Fortran, Lisp

- Functional programming
- Used in computing
- Efficient description of functional tasks
- Reliable

- Fortran, Lisp

```
C AREA OF A TRIANGLE - HERON'S FORMULA
C INPUT - CARD READER UNIT 5, INTEGER INPUT
C OUTPUT -
C INTEGER VARIABLES START WITH I,J,K,L,M OR N
      READ(5,501) IA,IB,IC
  501 FORMAT(3I5)
      IF(IA.EQ.0 .OR. IB.EQ.0 .OR. IC.EQ.0) STOP 1
      S = (IA + IB + IC) / 2.0
      AREA = SQRT( S * (S - IA) * (S - IB) * (S - IC) )
      WRITE(6,601) IA,IB,IC,AREA
  601 FORMAT(4H A= ,I5,5H  B= ,I5,5H  C= ,I5,8H  AREA= ,F10.2,
     $13H SQUARE UNITS)
      STOP
      END
```

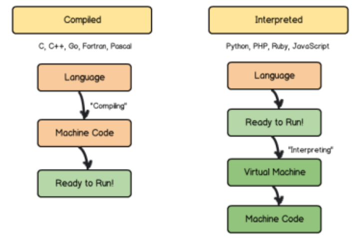
```
(defun get-max-value (list)
  (let ((ans (first list)))
    (do ((i 1 (1+ i)))
        ((>= i (length list)) ans)
      (when (> (nth i list) ans)
        (setf ans (nth i list))))))
```

4 –th generation

- Object Oriented Programming
- Garbage Collector
- (Interpreted languages)

- OOP C++
- Java
- Python
- …

Compiled vs Interpreted languages



Dynamic and static programing

```
int  data [10];



int *p;
int size;
cin >> size;
p = new int[size];
delete [] p;
```

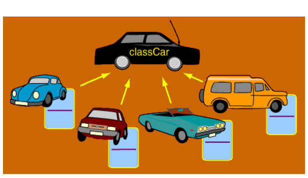
Garbage collector

- In classic approach a software engineer has to take care of memory usage,
- Non used variables have to be deleted to free memory,
- Risk of memory leakage,
- In Java and several other modern languages special hidden software is running all the time to take care of it – garbage collector,
- Extra computations, still not perfect, but allows to focus on other issues.

Object oriented programming - class

OOP is a programming paradigm that uses "objects"

- data structures
- data fields and methods
- interactions



OOP languages

- Java
- Python
- C++
- Objective-C
- JavaScript
- VB .NET
- Ruby

Main concepts

- Encapsulation
  - Class fields are private
- Abstraction
  - Class vs object (instance)
- Inheritance
  - New classes can share attributes from other classes
- Polymorphism
  - Same word means different things in different context

There is no best language!

The most modern languages like Java and C# are wrong for several applications, even C++ is not optimal.

In critical safety systems there is no place for dynamic allocation!
Every function has to be hard coded and memory granted at the start of the program.

Markup languages

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE recipe PUBLIC "-//Happy-Monkey//DTD Recip
"http://www.happy-monkey.net/recipebook/recipebook.

<recipe>

    <title>Peanut-butter On A Spoon</title>

    <ingredientlist>
        <ingredient>Peanut-butter</ingredient>
    </ingredientlist>

    <preparation>
        Stick a spoon in a jar of peanut-butter,
        scoop and pull out a big glob of peanut-butter.
    </preparation>

</recipe>
```

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
 <head>
  <title>sample</title>
 </head>
 <body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
 </body>
</html>
```

HTML

Markdown *.md

```
Heading
-------

## Sub-heading

Paragraphs are separated
by a blank line.

Two spaces at the end of a line
produces a line break.

Text attributes _italic_,
**bold**, `monospace`.

Horizontal rule:

---

Strikethrough:
~~strikethrough~~

Bullet list:

  * apples
  * oranges
  * pears

Numbered list:

  1. lather
  2. rinse
  3. repeat

An [example](http://example.com).
```

Heading

Sub-heading

Paragraphs are separated by a blank line.

Two spaces at the end of a line
produces a line break.

Text attributes *italic*, **bold**, `monospace`.

Horizontal rule:

Strikethrough:

~~strikethrough~~

Bullet list:

- apples
- oranges
- pears

Numbered list:

1. lather
2. rinse
3. repeat

An example.

Software testing and QA

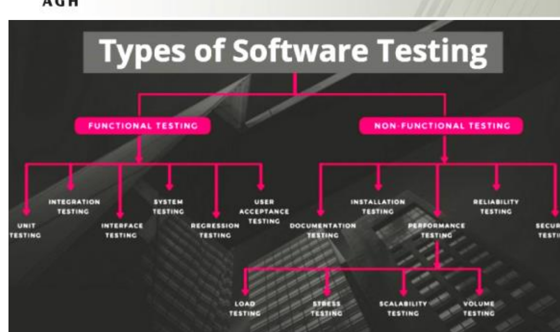
- Regular testing
- Results of even minor faults of important systems
- Critical safety systems tests
- Checking all scenarios
- Mathematical proofs of correctness

- It is the matter of the workflow rather than the code itself
- Quality Assurance - QA policy

Software testing

- What if we skip testing?
- No QA, no process
  cheap -> **technological debt** -> expensive

- Levels and Types of tests of Software (QA)
- Functional, Non-functional
- F: Unit, Integration, Acceptance
- NF: Performance, Security
- Black-box, White-box, Grey-box



AGH

Lifecycle of information systems development

AGH

Software development methodology

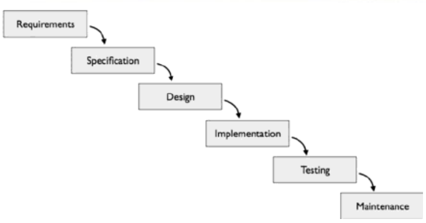
- Set of rules, tools, and policies that helps develop good software on time.

- Waterfall
- Lean
- Agile
- Scrum

- DevOps

AGH

Waterfall methodology



AGH

Agile methods vs traditional one

- Amount of documentation
- Rapid prototyping
- Changing roles in a team
- Meetings (including clients) vs documentation

AGH

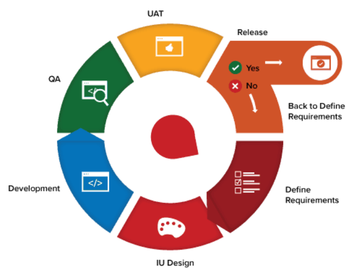
Lean development – focused on production process

AGH

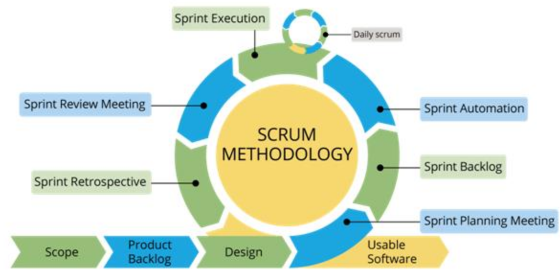
Manifesto for Agile Software Development

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile methodology – focused more on the development process



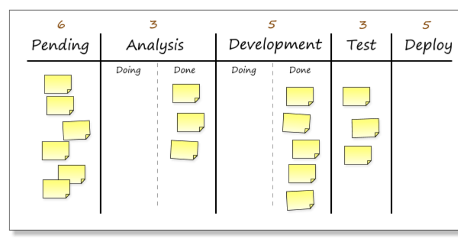
Scrum methodology

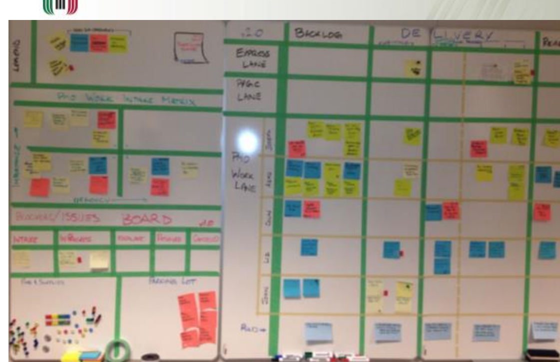


Scrum Rules

- The team, Scrum Master, Product Owner
- Sprint is 4 or less weeks long (2 weeks typical)
- No breaks between sprints
- Every sprint is same length
- Each sprint goal in shippable software
- Sprint planning (2hrs per week of sprint)
- Daily Scrum – same everyday (15 min max)
- Review -> feedback (product)
- Retrospective -> feedback (process)

Kanban

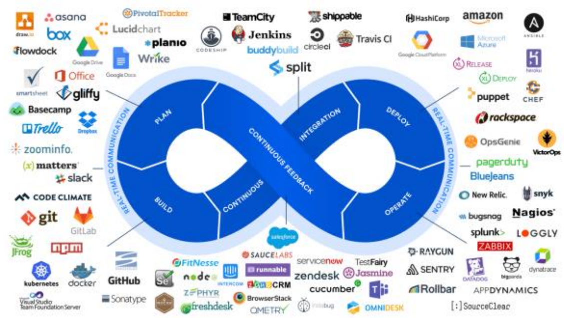




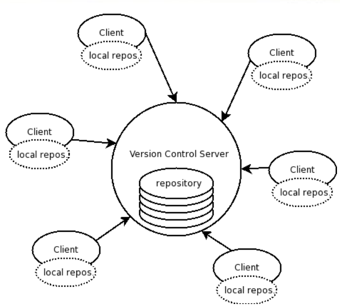
DevOps

- Integration of Dev, Ops, and QA
- Focuses on securing high quality development and delivery
- Continuous Integration / Continuous Delivery
- Workflow
  - Plan
  - Code
  - Build
  - Test
  - Package and Staging
  - Release, Deploy
  - Configure, Operate
  - Monitor

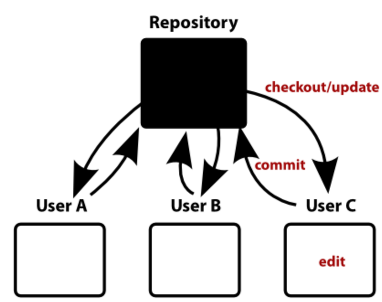
DevOps environment

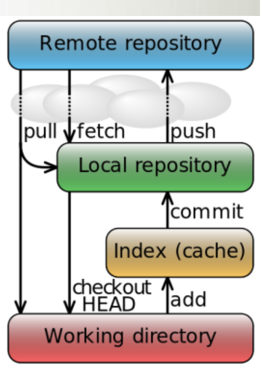


Version Control (Git, CVS and SubVersioN)

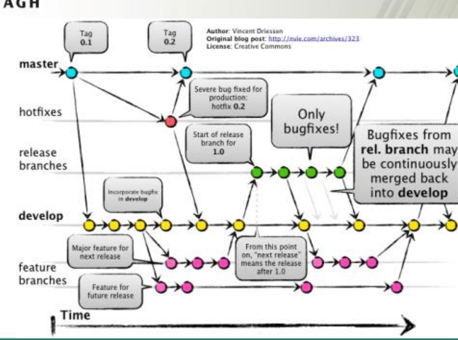


Version Control





Git workflow



Git commands

- $ git init
- $ git clone [url]
- $ git fetch
- $ git merge
- $ git push
- $ git pull