01

Základy informatiky

VÝUKOVÝ TEXT PRO STUDENTY INFORMAČNÍCH TECHNOLOGIÍ

Marek Lučný, SŠPU OPAVA 2015

A Informatika a informační technologie

Slovem **informatika** označujeme vědní obor, který se zabývá získáváním, zpracováním a využitím informací. Počátky oboru sahají do poloviny 20. století, kdy vznikly první elektronické počítače a začala skutečná informační revoluce. Informatika se zabývá nejen obecnými teoretickými otázkami spojenými s informacemi, ale prostřednictvím řady aplikovaných oborů řeší i praktické problémy.

K nejvýznamnějším oblastem aplikované informatiky patří:
Umělá inteligence (*AI – Artificial Inteligency*). Pokouší se napodobit intelektuální činnosti člověka pomocí počítače (rozpoznávání řeči, tvorba hypotéz).
Expertní systémy. Pomáhají odborníkům rozhodovat v oblastech, kde je zapotřebí brát v úvahu velké množství údajů (např. diagnóza nemoci). Do expertního systému se vloží výsledky mnoha různých vyšetření a systém na základě své báze znalostí a složitých algoritmů nabídne možné diagnózy.

Kybernetika a robotika se zabývá konstrukcí strojů, které samostatně reagují na podněty. V průmyslové praxi se již celkem běžně využívají průmyslové roboty, experimentálně jsou vyvíjeny humanoidní stroje.
Počítačová simulace umožňuje zkoumat neexistující objekty (které je možné nasimulovat, matematicky popsat), případně objekty, které lze jen obtížně zkoumat přímo (vznik hvězdy, průběh tlaků v centru hurikánu atd.).
Telekomunikace. Zkoumá a řeší vzdálený přenos informací, výsledkem jsou internet, bezdrátová komunikace, mobilní telefony.

Technické (hardware) i programové (software) prostředky, které jsou pro moderní zpracování informací využívány, bývají souhrnně označovány jako **informační a komunikační technologie** (*ICT - Information and Communication Technologies*).

B Data, informace, signál

Data jsou fakta získaná z reality, např. naměřené fyzikální veličiny (teplota, tlak, hmotnost), historické údaje, fotografie, zvukové či filmové záznamy apod.

Pojmem **informace** jsou odborně označována již vyhodnocená a smysluplně uspořádaná data do podoby, která má určitý význam. Díky informacím si vytváříme poznatky o okolním světě a na základě informací se můžeme rozhodovat o svém jednání.

Informace má nehmotnou povahu. Pro její uložení, přenos a zpracování je používán **signál** – fyzický nositel informací.
V primitivní podobě může být signál např. uzel na provázku, vyvěšená vlajka na stěžni apod. Pro strojové zpracování informací se jako signály využívají různé fyzikální veličiny, které mění svůj stav (elektrické napětí, magnetická polarizace, elektromagnetické vlnění, zvuková vlna, tlak plynu nebo tekutiny…).

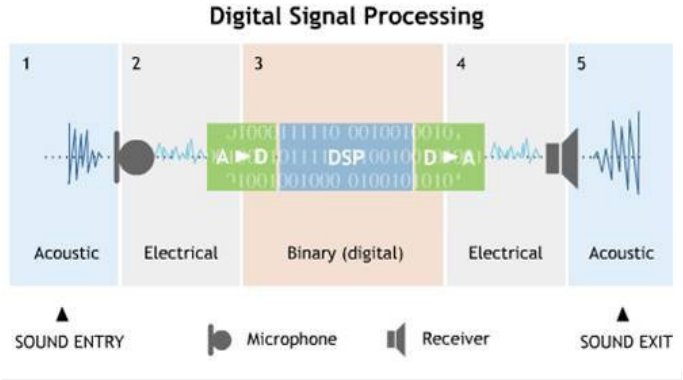
Rozdíl mezi daty a informacemi je možné vysvětlit na příkladu identifikačního průkazu. Průkazka obsahuje textová, číselná, grafická data. Izolovaná data sama o sobě nemají význam, teprve jejich začlenění do průkazky vytváří smysluplnou informaci o daném člověku.

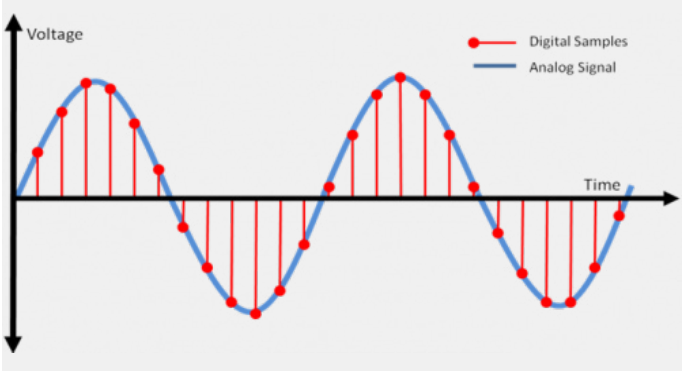
C Analogový a digitální signál

V případě zmíněných fyzikálních jevů lze přirozené změny stavu vyjádřit a zaznamenat v podobě spojité křivky – analogového signálu.
Analogový signál můžeme zpracovat pomocí příslušného analogového zařízení (například analogový magnetofon nahrávající zvuk na magnetickou pásku). Analogový signál věrně napodobuje původní signál, ovšem přenosem a kopírováním původní křivky dochází ke zkreslení (zvýšení šumu) a ztrátě kvality.

K analogovému záznamu zvuku je možné použít mikrofon připojený k analogovému magnetofonu. Zvukové vlny jsou zachyceny mikrofonem, který vstupní audiosignál převede na měnící se elektrické napětí. Elektrický signál je magnetofonem zesílen a snímán nahrávací hlavou. Při průchodu pásku podél hlavy se magnetické částice uspořádají tak, aby odpovídaly směru magnetického pole. Při přehrávání pásku indukuje magnetický záznam elektrický signál ve snímací hlavě; ten se po zesílení přehraje v reproduktoru.

Pro zpracování signálu na digitálním zařízení (např. počítač) je nutné provést jeho **digitalizaci** – vyjádření v číslech (*digitální = číselný*), nejčastěji v binární soustavě (sledu jedniček a nul). K digitalizaci se používá **A/D převodník** (*analog/digital convertor*). Původní křivka analogového signálu je „rozporcována“ na malé úseky (vzorkování) a pro každý z těchto vzorků je zaznamenána číselná hodnota (kvantování). Výsledkem je nespojitý **digitální signál**, který lze mnohonásobně kopírovat bez ztráty kvality.





D Jednotky v informatice

Bit (z anglického *binary digit - dvojková číslice*) je základní a současně nejmenší jednotkou informace. Značí se malým písmenem **b**, např. 16 b. Jeden bit reprezentuje logickou informaci typu ano/ne, kterou můžeme vyjádřit binárními číslicemi 0 a 1. V praxi se s bity setkáme např. při označení architektury systému (32bitový, 64bitový) nebo při určení tzv. **bitové hloubky** - tj. počtu bitů potřebných k uložení jednoho vzorku dat. Jednotkou přenosové rychlosti je **bit za sekundu** - *bit/s* (anglicky *bps – bit per second*); např. linka s přenosovou rychlostí 20 Mbit/s je schopna každou sekundu přenést 20 megabitů dat.

Byte (čti bajt, z anglického *byte* = *slabika*), je jednotka množství dat v informatice. Značí se velkým písmenem **B**, např. 1024 B. Zahrnuje osm bitů, tzn. osmiciferné binární číslo. Do jednoho bajtu je tak možné uložit celkem 256 různých hodnot ($2^8 = 256$). Jeden bajt je obvykle nejmenší objem dat, se kterým dokáže počítač (resp. procesor) přímo pracovat; v bytech a odvozených jednotkách se zpravidla udává kapacita (velikost) pamětí.

ODVOZENÉ JEDNOTKY

Pro bajty i bity se používají tradiční předpony soustavy SI jako kilo, mega, giga atd. (např. 5 GB, 2 Mb/s, 4 KB). Tyto předpony však mají odlišný význam: z technologických důvodů jsou velikosti některých počítačových pamětí obvykle násobkem mocniny dvou - kilobyte nepředstavuje přesně 1 000 bajtů, ale 2^{10} = 1 024 bajtů; podobně je to i u dalších odvozených jednotek.

Odvozená jednotka	Značka	Velikost v B	Mocnina
kilobyte	**KB**	1 024	2^{10}
megabyte	**MB**	1 048 576	2^{20}
gigabyte	**GB**	1 073 741 824	2^{30}
terabyte	**TB**	1 099 511 627 776	2^{40}
petabyte	**PB**	1 125 899 906 842 624	2^{50}
exabyte	**EB**	1 152 921 504 606 846 976	2^{60}
zettabyte	**ZB**	1 180 591 620 717 411 303 424	2^{70}

E Číselné soustavy a informatika

Na rozdíl od lidského světa čísel, v němž hraje prim desítková (dekadická) soustava, vládne v digitálním světě **soustava dvojková** (*binární*), která odpovídá i dvěma základním stavům elektronických obvodů:

stav VYPNUTO	stav ZAPNUTO
v obvodu **není napětí**	v obvodu **je napětí**
vyjádřeno číslem **0**	vyjádřeno číslem **1**
logická NEPRAVDA (**false**)	logická PRAVDA (**true**)

V počítačové praxi se používají také soustavy **osmičková** (*oktalová*) a **šestnáctková** (*hexadecimální*). V hexadecimální soustavě jsou chybějící číslice (nad hodnotu 9) vyjadřovány pomocí abecedních znaků (tj. 10 = A, 11 = B, 12 = C, 13 = D, 14 = E, 15 = F). Hexadecimálně se zapisují například adresy v paměti (např. 00H - 1FH), barvy v HTML a CSS (*color="#FFFFFF"*, *color: #FFFFFF*), kódy ve znakových sadách (*0x00* až *0x7F*) či fyzické MAC adresy síťových zařízení (např. *f4:08:a2:36:cc:87*).

Převody mezi číselnými soustavami

Převeďte desítkové číslo 56 do dvojkové, osmičkové a šestnáctkové soustavy

Pro převod do dvojkové (binární) soustavy dělíme číslo zapsané v desítkové soustavě základem m = 2 a zapisujeme zbytky (tj. v tomto případě 0 nebo 1):

56 : 2 = 28	28 : 2 = 14	14 : 2 = 7	7 : 2 = 3	3 : 2 = 1	1 : 2 = 0
zbytek 0	zbytek 0	zbytek 0	zbytek 1	zbytek 1	zbytek 1

Zjištěné zbytky uspořádáme v obráceném pořadí - tj. zprava doleva - a dostaneme výslednou binární podobu čísla: $(56)_{10} = (111000)_2$

Podobně můžeme postupovat i v případě převodu desítkového čísla do osmičkové nebo šestnáctkové soustavy. Změní se pouze základ dělení: v případě osmičkové soustavy dělíme osmi (m = 8), v případě šestnáctkové šestnácti (m = 16). Zbytky opět zapisujeme zprava:

$(56)_{10}=(70)_8$	56 : 8 = 7	7 : 8 = 0	$(56)_{10}=(38)_{16}$	56 : 16 = 3	3 : 16 = 0
	zbytek 0	zbytek 7		zbytek 8	zbytek 3

Převeďte binární číslo 11010110 do desítkové soustavy

Převod z dvojkové do desítkové soustavy můžeme urychlit, když si uvědomíme, že jednotlivé řády (číslice) binárního čísla představuji k-tou mocninu základu - tedy 2. Stačí poté sečíst pouze ty mocniny, které jsou v binárním čísle zastoupeny jedničkou:

1	1	0	1	0	1	1	0	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	214

Pro zjednodušení převodu z dvojkové soustavy do osmičkové a šestnáctkové si můžeme uvědomit, že hodnotu čísla v osmičkové soustavě tvoří vždy trojice dvojkových čísel a hodnotu čísla v šestnáctkové soustavě vždy čtveřice dvojkových čísel. Po rozdělení binárního čísla na trojice (respektive čtveřice) stačí převést dvojková čísla na číslice vyjadřující jednotlivé řády osmičkového (šestnáctkového) čísla:

$(111000)_2$	111	000	$(111000)_2$	0011	1000
$(70)_8$	7	0	$(38)_{16}$	3	8

F Kódování informací

Kód představuje pravidlo, jak převést jednu množinu znaků na jinou; např. jednotlivým alfanumerickým znakům se přiřadí čísla (viz kódování ASCII), *morseovka*, *Braillovo slepecké písmo* atd. Množina všech přípustných znaků se nazývá **abeceda** (např. Morseho telegrafická abeceda). Při zpracování informací pomocí digitální techniky se používá **binární abeceda**, která obsahuje pouze dva znaky (0,1). S kódy v různých podobách se v informatice setkáváme na každém kroku. Zajímavými příklady grafických kódů jsou *čárové kódy* a *QR kódy*.

ČÁROVÝ KÓD

Čárový kód se skládá z tmavých čar a ze světlých mezer, které se čtou pomocí specializovaných čteček - **snímačů čárových kódů** (laserových i digitálních). Některé typy čárových kódů mohou kódovat pouze číslice, jiné i písmena a speciální znaky. **Jednodimenzionální kódy** (1D) mají omezenou kapacitu a obvykle kódují numerický nebo alfanumerický řetězec, který je klíčem k identifikaci označeného předmětu do nějaké externí databáze. **Dvoudimenzionální kódy** (2D) vzhledem k vyšší kapacitě obvykle obsahují veškerou potřebnou informaci o označeném předmětu. Nejběžnější čárové kódy, používané pro zboží v obchodech, mají označení **EAN 13** a **EAN 8**.

QR KÓD

Zkratka *QR* pochází z anglického „*quick response*“ (česky „rychlá odpověď“). V praxi stačí vyfotit QR kód mobilním telefonem a **QR čtečka** poté rozšifruje znaky kódu a přeloží je na žádané informace (text, vizitku, internetový odkaz).

K vytvoření vlastního QR kódu je možné využít některý z online generátorů. QR kód je tvořen **geometrickou vrstvou** (slouží k přesné lokalizaci geometrických pozic, kde mají být čteny informační bity) a **informační vrstvou** (každý černý bod kóduje binární jedničku a bílý bod binární nulu). Dokáže pracovat také s určitým zkreslením a kompenzovat ztrátu části dat.

QR kód je možné využít v řadě oblastí – od propagace, přes poskytování rychlých informací v galeriích či muzeích až po moderní platby.

G Znakové kódy

Každému znaku je přiřazen určitý číselný kód a podle svých kódů jsou znaky uspořádány do tabulky; hovoříme také o tzv. **znakové sadě** nebo kódové stránce (*CP – code page*).

ASCII (*American Standard Code for Information Interchange*) je historicky nejúspěšnější znaková sadu, z které vychází většina současných standardů pro kódování textu. Kód ASCII byl podle nejstarší definice sedmibitový, obsahoval tedy $2^7 = 128$ platných znaků (*standard ASCII*). Součástí standardní tabulky jsou tisknutelné znaky (písmena, číslice, matematické znaky, interpunkční znaménka, speciální znaky…), a řídicí (netisknutelné) kódy, určené původně např. pro řízení tiskárny nebo dálnopisu. Pro potřeby dalších jazyků a pro rozšíření znakové sady se používají osmibitová rozšíření ASCII kódu (*extended ASCII*); jeden znak je v tomto případě uložen v 1 bajtu a celá tabulka obsahuje $2^8 = 256$ znaků. V českém prostředí vzniklo hned několik vzájemně nekompatibilních osmibitových kódovacích tabulek, které nahrazovaly některé méně používané znaky z rozšířené části tabulky ASCII znaky z české abecedy. Nejvíce se používalo kódování vytvořené firmou Microsoft pro středoevropské jazyky označované *Windows-1250*, v prostředí OS Linux se naopak prosazoval standard *ISO 8859-2*, ve starších dokumentech je možné narazit i na kódování *Latin II* nebo *bratří Kamenických*.

Cílem nového standardu **Unicode** bylo vytvořit univerzální znakovou sadu, která by odstranila mnohá omezení osmibitového kódování, a výrazně tak zjednodušila používání znaků na počítačích. Především bylo nutné zvýšit kapacitu znakové sady tak, aby dokázala pojmout všechny znaky využívané při výměně textů – především ty, které už byly definovány v hlavních mezinárodních, národních a průmyslových znakových sadách. Původní návrh standardu Unicode počítal s šestnáctibitovým kódováním, které mělo být dostatečné pro znaky běžně používaných abeced (*Basic Multilingual Plane - BMP*). Postupně však z koncepce Unicode vzešlo několik variant kódování, které mají ve svých názvech zkratku **UTF** (*UCS Transformation Format*) a které se vzájemně liší zejména nároky na paměť.

UTF-8

V současnosti patří k nejužívanějším kódování UTF-8, zejména pro svůj úsporný způsob využití paměti a kompatibilitu s původním ASCII kódem. V UTF-8 se znaky kódují různě dlouhou posloupností bajtů podle jejich pozice v Unicode. Tradiční znaky ASCII jsou tak kódovány pouze jedním bajtem, české znaky s diakritikou dvěma bajty a méně obvyklé znaky mohou být vyjádřeny třemi nebo čtyřmi bajty.

SPECIÁLNÍ A ŘÍDICÍ ZNAKY

Tyto „neviditelné“ znaky, které jsou na začátku tabulky ASCII, byly původně určeny pro řízení dálnopisu či tiskárny. Dodnes se ve znakových řetězcích vyskytují:

SPC	*Space* - mezera, „prázdný znak“
HT	*Horizontal Tab* – tabulátor
LF	*Line Feed* - odřádkování
CR	*Carriage Return* - návrat vozíku

Ani pro používání těchto kódů neexistuje všeobecně přijímaný standard; například OS Unix používají pro odřádkování kód *LF*, OS firmy Microsoft používají kombinaci *CR+LF*, OS firmy Apple používají kód *CR*.

ESCAPE SEKVENCE

Označení **escape sekvence** se v informatice používá pro speciální skupiny znaků (**řídicích znaků**), začínající znakem *Esc* (kód 27), které umožňují pozměnit standardní chování terminálu, interpretu apod.

V programovacích jazycích nebo v příkazovém řádku se používají pro potlačení speciálního významu určitého znaku a počátečním escape znakem bývá nejčastěji obrácené lomítko (např. \“). Pro zalomení řádku se ve znakovém řetězci zadává escape sekvence *\n*.

Se speciálními sekvencemi znaků – **entitami** - se můžeme setkat i v jazyce HTML. Např. *&#n;* zobrazí znak s unikódovou hodnotou n, sekvence *á* zobrazí český znak *á* (v tomto případě byla hodnota znaku zadána dekadicky, pokud bychom ji chtěli uvést hexadecimálně, použijeme sekvenci *&#xn;* např. *á* pro znak *á*).

H Uložení čísel v paměti počítače

Stejně jako všechny ostatní informace musí být i číselné hodnoty uloženy v počítačové paměti v binární soustavě.
Číselné datové typy se rozdělují podle toho, zda slouží k ukládání celých, nebo desetinných čísel.
Čísla mohou být uložena jako záporná i kladná (*signed* - se znaménkem), nebo pouze jako nezáporná (*unsigned* - bez znaménka).
Pro každý datový typ je navíc předem vymezen počet bitů v paměti.

Nejjednodušší celočíselný typ **byte** zabírá v paměti právě 1 B (8 bitů), a protože se nabízí právě 256 (2^8) variací binárních jedniček a nul, může obsahovat buď nezáporné celé číslo (unsigned) v rozsahu 0 až 255, nebo čísla v rozsahu od -128 do +127 (varianta signed).

Využívané celočíselné typy:

Typ	Délka	Rozsah		Počet hodnot
byte	**8 b**	-128	+128	2^8
short	**16 b**	-32 768	+32 767	2^{16}
int	**32 b**	–2 147 483 648	+2 147 483 648	2^{32}
longint	**64 b**	$-9{,}223 * 10^{18}$	$+9{,}223 * 10^{18}$	2^{64}

Desetinná čísla bývají počítači zpracovávána jako **čísla s plovoucí řadovou čárkou**. V paměti jsou uložena v tomto seskupení bitů:

- první zleva je **znaménkový bit** (1 znamená záporné číslo);
- druhou skupinou je několik bitů chápaných jako celé číslo se znaménkem a označovaných jako **exponent**;
- třetí skupinou je několik bitů chápaných (a označovaných) jako **mantisa**.

Datový typ **float** (někdy také single nebo real, číslo v jednoduché přesnosti) zabírá 32 bitů paměti (31. bit je znaménkový, bity 30-23 tvoří exponent a bity 22-0 mantisu). V tomto datovém typu může být uloženo číslo s absolutní hodnotou až $3{,}4 * 10^{38}$, číslo s nejmenší absolutní hodnotou ještě různé od nuly je přibližně $1{,}4 * 10^{-45}$. U čísel s plovoucí řadovou čárkou je však důležitějším údajem přesnost zobrazení, která je u typu float 7 až 8 cifer. Znamená to, že víceciferná čísla je nutné považovat pouze za přibližná a nevhodné použití typu float může vést k neočekávaným a někdy i osudným chybám či odchylkám.

Datový typ **double** (také double precision, číslo ve dvojnásobné přesnosti) je uloženo v 64 bitech s přesností zobrazení 15 až 16 platných cifer. Absolutní hodnota čísla typu doble může dosahovat hodnoty až $1{,}8 * 10^{308}$, což je mnohem více než předpokládaný počet atomů v celém vesmíru.

Uložení záporného celého čísla typu short v paměti s využitím doplňku (inverze bitů)

	druhý bajt								první bajt							
obsah	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
bitu	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Desítková hodnota binárního čísla (bez znaménkového bitu) je:
$1 * 2^{13} + 1 * 2^{12} + 1 * 2^{11} + 1 * 2^9 + 1 * 2^8 + 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 15\,343$
Protože však jde o tzv. doplněk čísla, odečteme ho od čísla 32 768:
32 768 - 15 343 = 17 425
vzhledem k **znaménkovému bitu** připojíme znaménko „-“: **-17 425**

I Datová komprese

Komprese dat (také komprimace dat) je speciální postup při ukládání nebo transportu dat, jehož cílem je zmenšit velikost datových souborů, což je výhodné např. pro jejich archivaci nebo při přenosu přes síť s omezenou rychlostí (např. mobilní telefon komprimuje hovor pro přenos GSM sítí). Kódováním podle určitého **kompresního algoritmu** se ze souboru odstraňují redundantní (nadbytečné) informace.

Ztrátová komprese je procesem, během něhož jsou některé informace nenávratně ztraceny a nelze je zpět rekonstruovat. Používá se tam, kde je možné ztrátu některých informací tolerovat a kde je nevýhoda určitého zkreslení bohatě vyvážena velmi významným zmenšením souboru. Nejčastěji se ze ztrátovou kompresí setkáváme v případech záznamu zvuku a obrazu (videa), kdy lze počítat s tendencí lidského mozku vnímat realitu spojitě a uceleně bez ohledu na zanedbatelná zkreslení či ztráty detailů.

Bezeztrátová komprese obvykle není tak účinná jako ztrátová komprese, ovšem komprimovaný soubor je možné dekompresí rekonstruovat do původní podoby. To je nezbytná podmínka zvláště při přenášení počítačových dat, výsledků měření, textu apod., kde ztráta i jediného znaku může znamenat nenávratné poškození souboru.

Výsledky komprese můžeme exaktně vyjádřit buď **kompresním poměrem** (podíl objemu původních dat k objemu dat komprimovaných), nebo **účinností komprese** (procentuálně vyjádřená úspora objemu dat).

Například je-li původní soubor o velikosti 10 MB zkomprimován do nového souboru s velikostí 2 MB, bude kompresní poměr vyjádřen podílem 5 : 1 (pětkrát zmenšeno), zatímco účinnost komprese bude 80 % (vypočteme pomocí vzorce 1 – komprimovaný / původní * 100 %, tj. 1 – 2/10 * 100 = 80 %).

Kompresní poměr je ovlivněn volbou kompresního algoritmu i typem komprimovaných dat. Například nekomprimované skladby na Audio CD mají datový tok přibližně 1,35 Mb/s, zatímco datový tok už komprimovaných zvukových souborů ve formátu MP3 může nabývat hodnoty i 128 Kb/s; kompresní poměr je tedy přibližně 11 : 1 a účinnost komprese dosahuje zhruba 90 %. Pro bezeztrátovou zvukovou kompresi (např. formát FLAC) jsou v případě stejného objemu dat typické kompresní poměry okolo 2 : 1.

KOMPRESE V PRAXI

Archivace dat. Využívá speciálních archivních formátů, které se liší aplikací odlišných kompresních algoritmů i účinností komprese (zip, rar, tar, gzip, bzip2, 7zip); k nejznámějším archivačním programům patří WinZip, WinRAR, 7zip.

Ukládání grafických souborů. Ztrátová komprese je používána u formátu JPEG, zatímco v případě formátů GIF a PNG jde o bezeztrátovou kompresi.

Kódování audiovizuálních souborů. Pro co nejúčinnější přenos videa i zvuku jsou k dispozici tzv. kodeky (z anglického codec - složenina z počátečních slabik slov coder a decoder), speciální programy, které ukládají data do zakódované formy (obvykle velmi usporně komprimované) a při přehrávání je dekódují do původní podoby (např. Divx, Xvid, H236, Lame).

Komprese souborů a adresářů. Je umožněna moderními souborovými systémy, např. NTFS ve Windows.

Speciální instalační soubory. Například soubory typu .msi (využívané k instalaci aplikací ve Windows) nebo .cab (komprimované soubory používané při instalaci systému Windows).

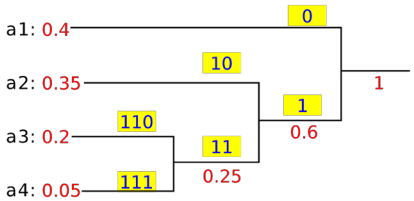
KOMPRESNÍ ALGORITMY

Kódování RLE (*Run-Length Coding*). Jednoduchá a pro velkou třídu dat i efektivní metoda vychází z předpokladu, že ve vstupních datech, se opakují „sousední" hodnoty. Tehdy do souboru zapíšeme informaci o počtu opakujících se hodnot a poté hodnotu samotnou.

vstupní data	AAAhoooo**j**
výstupní data	<3A>h<4o>j

LZW komprese. Bývá někdy označovaná jako slovníkově orientované kódování (*dictionary based encoding*). Při kompresi je vytvářen slovník s odkazy na nejčastěji se opakující bloky dat (např. slova v textu, části obrazu se stejnou barvou apod.).

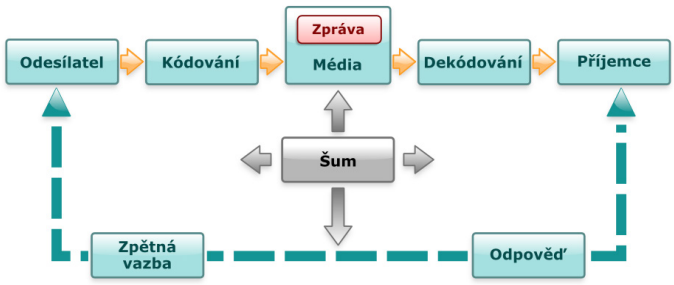
Huffmanovo kódování. Principem je vytvoření *binárního stromu* na základě znalosti frekvencí výskytu jednotlivých symbolů ve vstupním souboru. Nejčastější symbol má nejnižší binární číslo.



J Přenos informací

Počítač je stroj sloužící ke zpracování informací, umožňuje jejich ukládání, třídění, ale i přenos. Každé sdělení totiž vyžaduje komunikaci mezi zdrojem informace a jejím příjemcem; informace a komunikace proto spolu úzce souvisí, nelze je od sebe oddělit.

Obecný přenosový model komunikace ukazuje, jak se sdělení dostane od jeho zdroje až k příjemci:



Zdroj sdělení (jeho odesílatel) své sdělení zakóduje. Přenos sdělení probíhá mezi vysílačem a přijímačem pomocí domluveného sdělovacího kanálu. Z technického hlediska je důležitý dostatečný odstup přenášeného signálu od šumu, který je příčinou rušení signálu. Po dekódování sdělení na straně příjemce by měla být odesilateli odeslána zpráva (zpětná vazba), která umožní posoudit, zda sdělení bylo přeneseno správně.

KONTROLNÍ SOUČET

Aby se předcházelo přenosovým chybám, předává se v některých případech spolu s vlastní informací i tzv. kontrolní součet. Jedná se o doplňkovou informaci sloužící k ověření, zda je předaná informace úplná a zda při jejím přenosu nedošlo k chybě. Příjemce informace má díky tomu možnost spočítat vlastní kontrolní součet a porovnat ho s původním.

Parita. Levý krajní (paritní) bit obsahuje informaci o počtu jedničkových bitů ve slově (např. v jednom bajtu). Rozlišuje se buď *lichá parita* (součet jedničkových bitů včetně paritního musí být liché číslo), nebo *sudá parita* (výsledkem součtu je sudé číslo). V případě nesouladu je zřejmé, že během přenosu bajtu došlo k nějaké chybě.

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

CRC (*Cyclic redundancy check* = cyklický redundantní součet) je speciální hašovací funkce, používaná k detekci chyb během přenosu dat. *Hašovací funkce* je matematická funkce (resp. algoritmus), která může být provedena nad libovolně velkým souborem dat, ale její výstup je krátký řetězec znaků (*hash* neboli *kontrolní otisk*). V případě shody všech bitů v původním i přeneseném souboru je výsledný hash také shodný. Pokud se však dva soubory liší byť pouze v jediném bitu, projeví se to i v odlišnosti kontrolních otisků.