

01

Základy informatiky

VÝUKOVÝ TEXT PRO STUDENTY INFORMAČNÍCH TECHNOLOGIÍ

Marek Lučný, SŠPU OPAVA 2015

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).

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 označovány 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...).



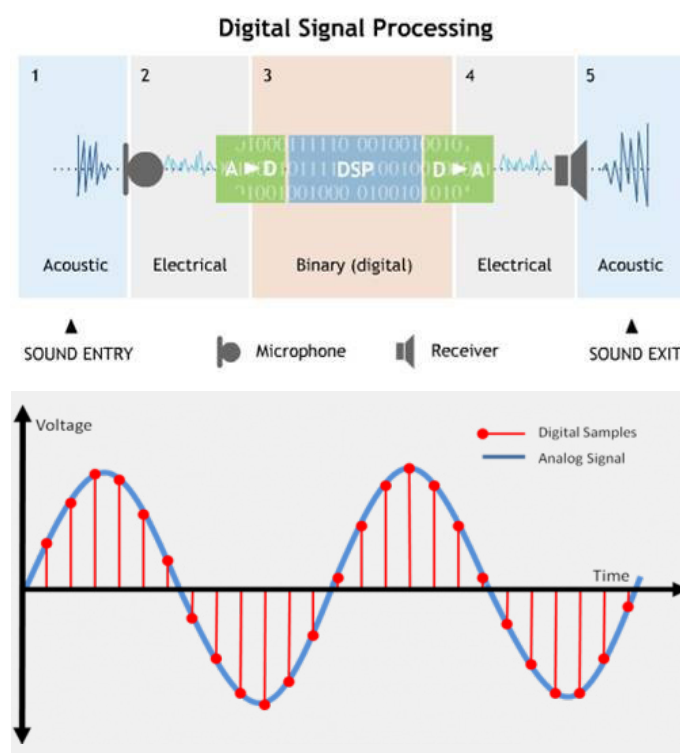
Rozdíl mezi daty a informacemi je možné vysvětlit na příkladu identifikačního průkazu. Průkazka obsahuje textová, číselná, grafická data. Izolovaná data sama o sobě nemají význam, teprve jejich začlenění do průkazky vytváří smysluplnou informaci o daném člověku.

V případě zmíněných fyzikálních jevů lze přirozené změny stavu vyjádřit a zaznamenat v podobě spojitě 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 pásky podél hlavy se magnetické částice uspořádají tak, aby odpovídaly směru magnetického pole. Při přehrávání pásky 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.



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}

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 0xFF) 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 (číslíky) binárního čísla představují 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 číslíky vyjadřující jednotlivé řády osmičkového (šestnáctkového) čísla:

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

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.



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 sada, 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 dálkopisu. 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 *bratři 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 pojmut 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 dálkopisu 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 (**řídících 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 `á`; pro znak á).

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	+127	2^8
short	16 b	-32 768	+32 767	2^{16}
int	32 b	-2 147 483 648	+2 147 483 647	2^{32}
longint	64 b	$-9,223 \cdot 10^{18}$	$+9,223 \cdot 10^{18}$	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 $3,4 \cdot 10^{38}$, číslo s nejmenší absolutní hodnotou ještě různé od nuly je přibližně $1,4 \cdot 10^{-45}$. U čísel s plovoucí řadovou čárkou je však důležitějším údajem přesnost zobrazení, která je u typu float 7 až 8 cifer. Znamená to, že vícečíslemá 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 double může dosahovat hodnoty až $1,8 \cdot 10^{308}$, 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:

$$1 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{11} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 15\,343$$

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**

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 $1 - \text{komprimovaný} / \text{původní} \cdot 100 \%$, tj. $1 - 2/10 \cdot 100 = 80 \%$).

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.

KOMPRESĚ 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 formátů 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 uspořádaně komprimované) a při přehrávání je dekódují do původní podoby (např. Divx, Xvid, H266, 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

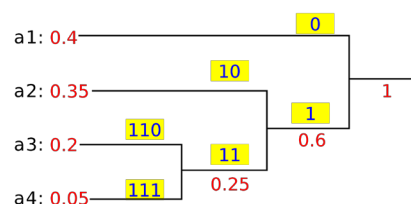
Kódování RLE (Run-Length Coding). Jednoduchá a pro velkou třídu dat i efektivní metoda vychází z předpokladu, že ve vstupních datech, se opakují „sousední“ hodnoty. Tehdy do souboru zapíšeme informaci o počtu opakujících se hodnot a poté hodnotu samotnou.

vstupní data	AAAhooooj
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 nejmenší binární číslo.

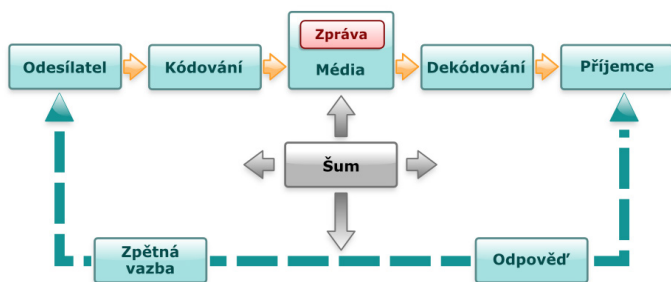


J

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 dekodování sdělení na straně příjemce by měla být odesílateli odeslána zpráva (zpětná vazba), která umožní posoudit, zda sdělení bylo přeneseno správně.

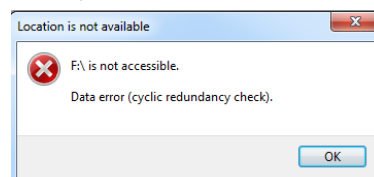
KONTROLNÍ SOUČET

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 buď pouze v jediném bitu, projeví se to i v odlišnosti kontrolních otisků.



02

Programy a data

VÝUKOVÝ TEXT PRO STUDENTY INFORMAČNÍCH TECHNOLOGIÍ

Marek Lučný, SŠPU OPAVA 2015

Soubor (*file*) je množina souvisejících dat uložených na diskové jednotce. Název souboru je obvykle tvořen jménem (ve Windows až 254 znaků) a příponou, která naznačuje **typ souboru**. K dalším vlastnostem souboru patří jeho velikost, autor (vlastník), datum vytvoření, datum změny, atributy, přístupová práva apod.

K přehlednější práci se soubory slouží systém **adresářů** či složek (*directories, folders*). Adresáře tvoří na disku typicky hierarchickou strukturu (*tree - strom*); základem je tzv. **kořenový adresář** (*root*) označený symbolem „/“ (Unix, Linux), nebo „\“ (Windows). Každý adresář může obsahovat soubory i **podadresáře** (*subdirectories*).

Cesta (*path*) vyjadřuje umístění souboru vždy směrem od nejvyššího adresáře: `C:\XAMPP\mysql\bin\mysql.exe`. Pro operace se soubory využíváme někdy **zástupné znaky** (*wildcards*):

- symbol `*` nahrazuje v názvu libovolný počet znaků (např. `*.*`);
- symbol `?` nahrazuje jeden libovolný znak (např. `s?s.txt`).

Programové soubory obsahují instrukce, podle nichž procesor ve spolupráci s dalšími hardwarovými komponentami plní konkrétní úlohy. Mohou být uloženy v již zkompileované binární podobě (nejčastěji ve **strojovém kódu** pro danou platformu), nebo ve **zdrojovém kódu**, ob-

sahujícím příkazy určitého programovacího jazyka. Mezi programové soubory můžeme zařadit také **skripty**, soubory instrukcí v textové podobě, které ke svému spuštění vyžadují speciální program - **interpret**.

Větší programy jsou tvořeny **programovými knihovnami** (*runtime libraries*), speciálními soubory s připravenými funkcemi, které programy využívají pro svůj běh. Ve Windows to jsou např. soubory *DRV* a *SYS* (obsahující zejména ovladače čili drivery, ale i systémové funkce), nebo také **DLL** (*dynamic linking library* - dynamicky linkované knihovny), které mohou být podle potřeby za běhu (dynamicky) připojeny k jednomu i více programům.

Některé soubory s tzv. **spustitelnou příponou** (např. `.exe` a `.com` ve Windows, `.sh` v Linuxu) slouží ke spuštění programu - aktivaci procesu v rámci operačního systému. Za **rezidentní programy** (služby, démony) označujeme ty procesy, které jsou spuštěny automaticky po iniciaci operačního systému (např. antivirové programy).

Datové soubory obsahují data různého charakteru (číselné, textové, grafické, zvukové údaje). K jejich zpracování se používají konkrétní programy, a proto jsou často přípony těchto datových souborů asociovány (propojeny) s určitou aplikací (např. *DOCX* s programem MS Word, nebo *XLSX* s programem MS Excel).

Programové vybavení (*software*) je podmínkou využití technického vybavení (*hardware*). V raných dobách počítačové éry byly programy vytvářeny vždy pro konkrétní hardware. V okamžiku, kdy počítače začaly být produkovány ve větších sériích, objevila se i potřeba vývoje speciálních **systémových programů**, které by obstaraly základní operace s hardwarem a nabídly prostředí pro běh aplikací, tj. programů určených pro konkrétní účely.

Základním programovým vybavením počítačů (a jim podobných zařízení) jsou **operační systémy** (*OS - Operation System*). Jejich nejdůležitější funkcí je zajištění řízení hardware. To má na starosti **jádro systému**, které tvoří subsystémy pro správu procesoru, paměti, diskových médií a dalších zařízení. Jádro pro své operace vyžaduje instalaci ovladačů jednotlivých zařízení (*drivers*) a s jejich pomocí zpřístupňuje technické vybavení běžícím procesům.

Nezbytnou součástí operačních systémů je **aplikační rozhraní** (*API - Application Programming Interface*), které nabízí programátorům prostředky pro vývoj aplikací pro daný systém. **Uživatelské rozhraní** slouží k ovládání OS a k spouštění aplikací. Zatímco starší OS se ovládaly častěji prostřednictvím **příkazového řádku** (*CLI - Command Line Interface*), moderní systémy poskytují **grafické uživatelské rozhraní** (*GUI - Graphic User Interface*) založené na oknech a grafických prvcích.

Na počátku vývoje moderních OS byl víceúlohový a víceuživatelský systém **UNIX**, který se od 70. let používal v sálových počítačích i minipočítačích. Jeho principy využily i další OS. V 80. letech minulého století si vzájemně konkurovaly systémy firmy *Microsoft* (**MS DOS**), instalované na počítačích IBM PC, a firmy *Apple* (**Mac OS**), která rozvinula myšlenky vývojářů společnosti *Xerox* a začala jako první používat grafické uživatelské rozhraní. To se stalo také základem **MS Windows**, jenž hrál prim především na přelomu tisíciletí. V té době už získával stále větší popularitu volně šiřitelný systém **Linux**, pojmenovaný podle svého tvůrce *Linuse Torvaldse* a staršího vzoru UNIXu. Zatímco operační systémy Windows dominují především na osobních počítačích v domácnostech i kancelářích, různé linuxové distribuce jsou velmi často instalovány na síťových serverech.

Současné OS rozdělujeme nejčastěji podle typů zařízení, pro která jsou určeny. Existují **systémy pro síťové servery** (např. *MS Windows 2012 Server*, *Linux*, *Unix*), **pro běžné osobní počítače** - desktopy, notebooky, netbooky (např. *MS Windows*, *Linux Ubuntu*, *OS X*), **pro mobilní zařízení** (*Android*, *Symbian*, *Windows Phone*, *iOS*) i speciální **RTOS** (*Real-Time Operating System* - *systém reálného času*), které zajišťují provoz „chytrých“ zařízení, jakými jsou robotické kamery využívané pro sportovní přenosy, měřicí přístroje na orbitálních stanicích nebo jednotky pro řízení různých průmyslových procesů. Zatímco u desktopového systému je možné pozdržet dokončení některých operací a provádět paralelně jiné operace, v případě RTOS je nezbytné provádět operace bez prodlevy a s vysokou spolehlivostí v reálném čase.

K systémovému software můžeme kromě operačních systémů zařadit ovladače hardwarových zařízení, firmware a aspoň zčásti i tzv. utility.

Ovladače zařízení (*device drivers*). Speciální programy, které zajišťují komunikaci OS s určitým technickým vybavením. Bez potřebného ovladače nemůže zařízení správně fungovat. Standardní ovladače pro běžná zařízení, jakými jsou flash disk, myš, klávesnice nebo digitální fotoaparáty, bývají součástí moderních OS. Instalaci zařízení ve Windows usnadňuje standard **Plug and Play** (*PnP - „připoj a hrej“*), který zajistí automatickou detekci nově připojeného hardware a případně i automatické vyhledání potřebného ovladače. Není-li driver nalezen přímo v OS, musí být použit externí instalátor ovladače z disku dodaného se zařízením (např. z instalačního CD k tiskárně), nebo stažený z internetových stránek, nejlépe přímo od výrobce. Nevhodně zvolený ovladač může v krajním případě způsobit nefunkčnost zařízení, někdy i celého OS.

Firmware. Programové vybavení, které slouží pro řízení nějakého vestavěného systému (*embedded system*). Firmware řídí například kalkulačky, počítačo-

vé komponenty (pevný disk, router...), mobilní telefon, digitální fotoaparát atd. Dříve byla aktualizace firmwaru realizována nahrazením paměťového média obsahujícího firmware (např. výměna ROM paměti), dnes může být novější verze firmware přepsána díky používaným flash pamětím. Základním firmwarem v počítači je **BIOS** (*Basic Output-Input System*), který zajišťuje především start počítače a umožňuje konfiguraci základní desky i připojeného hardware.

Utility. Jsou menší pomocné programy, které umožňují vykonávat servisní operace potřebné k fungování počítače. Patří k nim komprimační programy, diagnostické a testovací programy, utility pro kontrolu a optimalizaci operačního systému, různé konverzní (převodní) programy, které umí přehledně zobrazit obrázky, převést je z jednoho formátu na jiný atd. Řada utilit je integrální součástí OS. Příkladem může být *Správce úloh* ve Windows (*Task Manager*), který umožňuje nejen správu spuštěných programů a služeb, ale i sledování využití systémových zdrojů (procesoru, paměti). Dalšími užitečnými utilitami jsou nástroje pro defragmentaci (*defrag*) nebo kontrolu disku (*chkdsk*). Některé utility lze spouštět z příkazového řádku (*ping*, *ipconfig* apod.).

Applikace jsou uživatelské programy určené pro jeden nebo více specifických účelů. Pro spouštění aplikací je nezbytná přítomnost základního software - operačního systému.

ZÁKLADNÍ SKUPINY APLIKACÍ

Kancelářské balíky. Symbolický „balík“ programů obvykle tvoří textový procesor, tabulkový procesor, nástroj na tvorbu prezentací, databázový systém, někdy i groupwarové nástroje (systémy organizování času, zajišťování schůzek), grafické editory apod. Vedle komerčních *Microsoft Office* či *Apple iWork* existují také balíky *OpenOffice* nebo *WPS Office*, dostupné zdarma a pro řadu platform.

Grafické aplikace. Oblast grafiky je velmi široká; grafické programy umožňují vytvoření grafických dokumentů přímo v počítači, skenování papírových předloh i zachycení obrazu digitálním fotoaparátem. Dominantní postavení v této oblasti zaujímá profesionální software firmy *Adobe* (např. *Adobe Photoshop*, *Adobe Illustrator* aj.). Mezi oblíbené multiplatformní opensource grafické aplikace patří např. rastrový editor *GIMP* nebo vektorový *Inkscape*. Svěbytné oblasti počítačové grafiky tvoří programy pro tvorbu animací, rozsáhlé konstrukční systémy **CAD** (*Computer Aided Design*) nebo programy **DTP** (*Desktop Publishing*) určené pro tvorbu tiskovin.

Multimediální programy. Umožňují vytváření, zpracování a prezentaci multimediálních, tedy především audiovizuálních dat na počítači. Kromě nejrozšířenějších multimediálních přehrávačů zde můžeme zařadit řadu programů pro editaci digitálního zvuku nebo videa, programy pro tvorbu multimediálních prezentací, výukové programy, ale také rozsáhlou oblast počítačových her.

Programy pro komunikaci v počítačové síti. Slouží pro správu počítačových sítí, vzdálené řízení počítače, přenos a přístup ke vzdáleným souborům a vzdálený tisk, pro připojení vnitřní podnikové sítě (intranet) na síť celosvětovou (Internet) přes poskytovatele Internetu. K základním uživatelským (klientským) aplikacím patří webové prohlížeče, klienty elektronické pošty a groupware (software pro skupinovou práci), prostředky online komunikace (*IM* - *Instant Messaging*) a jiné.

Informační a databázové systémy. Informační systémy (*IS* - *information systems*) představují komplexní programové vybavení určité instituce nebo firmy. Základem IS jsou obvykle tzv. *datové sklady* (*data warehouse*) založené na databázových systémech. Podnikové informační systémy bývají označovány zkratkou *ERP* (*Enterprise Resource Planning*) a jsou tvořeny celou řadou dílčích programových modulů (např. moduly pro výrobu, logistiku, správu majetku, účetnictví ...). K lídrům v oblasti informačních i databázových systémů patří velké firmy jako *Oracle*, *Microsoft*, *IBM*, *SAP*. Kromě komplexních IS jsou vyvíjeny mnohé specializované aplikace pro podporu výrobních procesů, ekonomických transakcí a nejrůznějších jiných pracovních činností.

Malware a antivirové programy. Zatímco malware je počítačový program určený ke vniknutí nebo poškození počítačového systému, cílem antivirových programů (např. *AVG*, *AVAST*, *ESET*, *Kaspersky Antivirus* atd.) je chránit počítačové systémy před tímto nežádoucím softwarem.

Integrovaná vývojová prostředí. Obvykle jsou označovány zkratkou *IDE* (*Integrated Development Environment*) a používají je programátoři k vývoji aplikací. Tradičně je tvoří editor s různými pomůckami pro psaní programového kódu, ladicí prostředky, překladače a další nástroje zefektivňující práci vývojářů.

Instalace programu. Program, který si koupíme, dostaneme na CD/DVD disku, nebo si ho stáhneme z Internetu. Po vložení instalačního CD/DVD disku se většinou okamžitě spustí instalační program (díky souboru *autorun.inf*), v případě softwaru staženého z Internetu musíme instalační program spustit sami (ve Windows nejčastěji poklepnutím na soubory *install.exe* či *setup.exe*). Průvodce instalací (*wizzard*) si často vyžádá sériové číslo a souhlas s licenčními podmínkami, zeptá se, kam (do jaké složky na disku) se má program uložit a kam se má umístit zástupce programu, případně žádá další informace. Potom instalační program nahraje všechny soubory, ze kterých se program skládá, na pevný disk počítače, provede nutnou konfiguraci (v systémovém registru Windows, konfiguračních souborech) a nakonec vytvoří zástupce programu pro jeho snadné spuštění. K instalaci většiny programů je nutné mít práva správce.

Odinstalace programu. Úplné odebrání programu bývá paradoxně často složitější, zvláště v systémech Windows. Při instalaci si program nahraje své komponenty do mnoha složek včetně systémových, sdílí některé soubory s jinými programy (dynamické knihovny DLL), vytvoří záznamy v tzv. registru, při svém běhu vytváří pomocné soubory. Program, zajišťující odinstalování aplikace, často neví přesně, kde všude jsou programové soubory uloženy a všechny je nesmaže. V řadě případů tak dochází k hromadění zbytečných a nepoužívaných souborů. Pro jejich odstranění lze používat některé speciální utility – např. *RegCleaner* (program pro čištění registrů). Při podobných operacích je však nutné postupovat velmi obezřetně, aby nedošlo k odstranění některé důležité součásti systému nebo používaného programu! V linuxových systémech je lépe vyřešeno oddělení systémových a uživatelských souborů, úplné odstranění nepotřebného programu je proto pravděpodobnější.

Lokalizace. Lokalizace je překlad programu, v našem případě do češtiny. Většina významných softwarových firem pochází z anglicky mluvících zemí a originály programů tedy existují v angličtině. Dnes je však zcela běžné, že tyto firmy mají místní zastoupení, která zajišťují prodej programů, ale také je před prodejem lokalizují - přeloží do češtiny. Lokalizace je drahá a časově náročná činnost, protože se musí přeložit nejen volby nabídek a obsah dialogových oken, ale také nápověda k programu a návod k programu (manuál), případně vytvořit dělicí slovník pro české dělení slov a český korektor pravopisu.

Platforma a kompatibilita. V souvislosti s určitým typem OS i typem procesoru používáme pojem **platforma** - např. Windows, Linux, OS X. Používaný program musí být určen pro operační systém, který je na počítači nainstalován; pod jiným systémem většinou nepracuje (např. program určený pro MS

Windows nebude pracovat pod Linuxem a naopak). Mnoho programů, zvláště z oblasti svobodného software (*free software*), má proto verze určené pro několik operačních systémů – jsou **multiplatformní**. Problémy se spuštěním programu však mohou nastat i na jedné platformě v souvislosti s různými verzemi OS i rozdíly mezi 32bitovými a 64bitovými systémy (například nový program určený pro MS Windows může spolehlivě fungovat v novějších verzích Windows, ale nemusí pracovat pod starší verzí stejného OS). Při pořizování programu si proto vždy musíme ověřit, zda je **kompatibilní** (ve funkční shodě) s námi používaným operačním systémem. V nových systémech Windows je v případě problémů možné využít nastavení na kartě *Kompatibilita*, která je součástí panelu *Vlastností* pro zástupce dané aplikace.

Verze programů. U počítačových programů je důležitá jeho verze, která se označuje číslem za názvem programu. Nový program může být například označen jako verze 1.0. Některé dodatečně zjištěné dílčí nedostatky nebo bezpečnostní problémy řeší *aktualizace programu* (dnes nejčastěji přes Internet), **update**. Aktualizace jsou zakáznicím, kteří si zakoupili určitou verzi programu, obvykle poskytovány bezplatně. Pro přehlednost jsou označovány buď datem vydání, nebo symbolickým číselným vyjádřením podverze – např. 1.1 nebo 1.23 apod. Povyšení na novou verzi (např. 2.0), **upgrade**, přináší významnější změny ve vývoji programu a v případě komerčního software je třeba vyšší verzi znovu zaplatit. Bývá pravidlem, že nové verze mají nové funkce a lepší ovládání, ale také vyšší nároky na výkon počítače. Před provedením upgrade je tedy nutné zjistit doporučenou (případně minimální) konfiguraci počítače, aby bylo možné program na daném počítači možné vůbec používat. Opačný postup, tedy použití nižší verze programu (např. kvůli nekompatibilitě s instalovanými OS nebo nedostatečnou konfigurací počítače), se nazývá **downgrade**.

Registrace programu je zaevidování jeho uživatele do databáze výrobce programu. Registrace je nepovinná a přináší většinou několik výhod:

- nárok na upgrade programu za sníženou cenu; uživatel se často včas dozví o časově omezených nabídkách výhodného nákupu upgradu nebo dalších programů od stejné firmy;
- někteří výrobci programů posílají svým registrovaným uživatelům materiály (časopisy, prospekty), kde se dozvíte zkušenosti jiných uživatelů s programem, bývají zde informace o nových verzích programu, případně o vzniklých doplňcích daného programu;
- u některých programů po registraci vzniká nárok na využití zákaznické linky v případě potíží s programem.

Uživatel nemůže koupit program jako celek se všemi právy k němu, ale může získat pouze **licenci** (oprávnění) k jeho užívání. Z toho plyne, že nesmí žádným způsobem upravovat toto dílo, pokud k tomu autor nedá svůj výslovný souhlas. Licence je většinou nepřenosná, a proto není možné tuto licenci někomu dalšímu prodat nebo darovat.

Multilicence. Při koupi programu běžně získává uživatel oprávnění provozovat program na jednom počítači. V případech hromadného využití programu (firmy, školy ...) stačí koupit jedno balení programu a k němu vícenásobnou licenci pro další počítače, která bývá cenově výhodnější variantou.

Komerční software. Často se používá rovněž označení **proprietární software** nebo také software s uzavřeným kódem (*closed source*). Jeho autor upravuje licenci (typicky **EULA** - *End-User-License-Agreement*) či jiným způsobem možnosti jeho používání. K takovému software nejsou zpravidla k dispozici volné zdrojové kódy či v nich nelze svobodně dělat úpravy a výsledné dílo distribuovat. Je šířen za úplatu a bývá omezen počet instalací, přenositelnost licence či právo modifikace produktu.

OEM (*Original Equipment Manufacturer* = původní výrobce zařízení). Touto zkratkou se označují výrobky, které jsou určeny k zabudování jako komponenty (součástí) do jiných zařízení. Jedná se například o programy, které jsou nabízeny pouze současně s hardwarem za sníženou cenu.

Demoverze a zkušební verze programů. Demoverze jsou „ostré“ programy, které mají zablokované některé funkce, často ukládání souborů na disk a tisk. Dodávají se zdarma, nebo za malý poplatek. V demoverzi je možné vyzkoušet funkčnost programu před jeho koupi. Zkušební (*trial*) verze většinou umožňují plnohodnotnou práci včetně ukládání i tisku, ale jen po určitou dobu. Po uplynutí této doby program nelze spustit ani znovu nainstalovat.

Shareware. S programem může uživatel pracovat a po určité době je povinen zaslat jeho autorovi uvedený poplatek. Poté smí program legálně užívat a autor k němu často pošle stručný manuál, případně i novou verzi programu.

Freeware. Doslova lze název této skupiny programů přeložit jako volné zboží. Program je možné používat a rozšiřovat zdarma, požaduje se pouze dodržování autorských práv. Nesmí se zahrnovat do komerčně šířených programů a nesmí být změněn. Někdy je jako freeware šířen i velmi kvalitní program a po jeho hromadném rozšíření se z něj může stát program šířený komerčně.

PUBLIC DOMAIN

Je to rozsáhlá oblast programů k volnému užití. Tyto programy lze používat, volně šířit i upravovat. Autoři se vzdávají některých svých práv k programu a nechají uživatele s ním nakládat podle libosti.

SVOBODNÝ SOFTWARE A GNU/LINUX

V současnosti vystupuje do popředí tzv. **svobodný software** (*free software*, nezaměňovat s *freeware*) šířený většinou pod licenci *Open Source* (nejčastěji GPL, viz níže). Ten se stává alternativou pro často finančně náročné komerční programy.

Myšlenku svobodného softwaru formuloval v polovině 80. let minulého století **Richard M. Stallman**, do té doby programátor na univerzitě v USA. Stallman založil projekt, jehož cílem bylo vytvořit svobodný unixový operační systém. Projekt byl pojmenován jako projekt **GNU** (*GNU is Not Unix*). Systém GNU brzy obsahoval všechny důležité komponenty s výjimkou jádra. To začal vytvářet **Linus Torvalds** v roce 1991 a časem dostalo název Linux. Kombinací systému GNU a jádra Linux pak vznikl **GNU/Linux** systém, který dnes díky jeho distribucím používají desítky milionů uživatelů.

OPEN SOURCE A LICENCE GPL

Open Source znamená doslova otevřený zdroj. Takové programy musí být vždy šířeny i s uvedením zdrojových textů, takže každý (kdo má patřičné znalosti) může přesně zjistit, jak takový program funguje. Každý ho také může vylepšit, opravit v něm chybu, použít ho ve svém programu, vždy však musí takový produkt opět distribuovat jako Open Source.

Přesné podmínky pak stanovuje licence **GPL** (*General Public Licence* - všeobecná veřejná licence). Nejrozšířenější takovou licenci je **GNU/GPL** (*GNU General Public Licence*). Prvními vývojáři byli nadšenci z prostředí především amerických univerzit, studenti i zaměstnanci, a dodnes je akademická půda silnou základnou Open Source. Mnoho komerčních firem platí své vývojáře za práci na Open Source a přispívá finančně celé komunitě. Vlády některých zemí (např. Číny a Japonska) sledují se znepokojením dominanci firmy Microsoft na poli softwaru a podporují vysokými částkami vývoj volně šiřitelného softwaru, který pak mohou jejich občané používat bez nákupu drahých komerčních systémů.

Program je většinou zbožím, které se prodává na volném trhu tak jako jakékoli jiné zboží. Oproti třeba technickým (hardwarovým) počítačovým dílům má však program určité specifické vlastnosti. Je chráněn tzv. **autorským zákonem**, který i počítačový program považuje za autorské dílo. Jeho tvůrci (firma, pro kterou pracují) k němu mají autorská práva, podobně jako autoři knih, hudební skladatelé atd.

Počítačový program je dnes asi nejčastěji kradeným zbožím, o primát s ním soutěží nelegálně stažená hudba a kopírované video. Softwarovým firmám kvůli počítačové kriminalitě unikají velké tržby za prodej jejich programů a uživatelé počítače se dopouštějí protiprávního jednání (oba dva - ten, který si program nelegálně nainstaluje, i ten, který mu to umožní). V zákoně na ochranu autorských práv je jasně uvedeno, že pachatelé takového jednání mohou být

odsouzeni k peněžitému trestu i k trestu odnětí svobody na několik let.

Kontrolu legálnosti software provádí specializované oddělení Policie ČR, žádná jiná organizace na to nemá právo. Je-li ve firmě nebo v domácnosti zjištěno užívání nelegálního softwaru, je vyčíslena jeho hodnota a dojde k finančním i trestním postihům odpovědných lidí, ale i firmy.

Možnosti ochrany programů před nelegálním kopírováním. Výrobci programů chrání své programy tak, že se snaží kopírování přímo znemožnit nebo alespoň používají prvky, které usnadní usvědčit pachatele takového činu: *zadáání licenčního (sériového) čísla, aktivace přes Internet nebo telefon, hardwarový klíč* (malé technické zařízení, které se připojuje většinou k USB portu počítače, bez jehož přítomnosti v počítači program nefunguje, speciálně upravená chybná místa na CD/DVD, která způsobí chybu při kopírování).

Emulátor je program, který umožňuje běh počítačových programů na jiné platformě (architektuře, operačním systému), než pro kterou byly původně vytvořeny a kterou samy od sebe podporují. Typickým příkladem emulátoru je program umožňující běh videoher známých z herních konzol na běžném PC pod MS Windows nebo Linuxem. Jiným příkladem může být emulátor **DOSBox**, který emuluje starý operační systém MS DOS v prostředí novějších Windows, případně i na zcela odlišných platformách (Linux).

Virtualizace označuje v informatice postupy a techniky, které umožňují v počítači přistupovat k dostupným zdrojům jiným způsobem, než jakým fyzicky existují. Virtualizaci je možné realizovat na různých

úrovních - od celého počítače (tzv. *virtuální stroj*), po jeho jednotlivé hardwarové komponenty (např. virtuální procesory, virtuální paměť), až po konkrétní softwarové prostředí (virtualizace operačního systému).

Virtuální stroj je speciální software, který dokáže simulovat skutečný počítač (včetně jeho procesoru, paměti, disku, přídavných karet) a umožňuje instalaci OS i dalších programů. Podmínkou virtualizace je výkonný počítač a potřebný software - např. *Virtual PC* (Microsoft), *VirtualBox* (Oracle), *VMWare* apod.

Virtuální stroje se používají z bezpečnostních důvodů pro běh některých aplikací, aby tyto nemohly ohrozit hostitelský počítač a jeho operační systém. Často jsou využívány i pro testování nových aplikací. Velmi přínosná je virtualizace na úrovni serverů - tzv. **konsolidace serverů** nabízí dokonalejší využití hardware, oddělený běh síťových aplikací i vyšší bezpečnost dat.

Cloud computing je nový výpočetní model založený na poskytování služeb a programů uložených na internetových serverech, k nimž mohou uživatelé přistupovat pomocí klientských aplikací (nejčastěji webového prohlížeče) a používat je prakticky odkudkoliv. V případě placených služeb uživatelé neplatí za vlastní software, ale za jeho užití.

Veřejný cloud je určen pro velký nebo neomezený počet klientů (např. Skype, Seznam), zatímco **privátní cloud** je k dispozici pro omezený počet zákazníků. **Hybridní cloudy** pak kombinují veřejný i privátní přístup, navenek však nabízejí jednotné standardizované rozhraní.

Vedle každodenně používaných internetových služeb, jakými jsou internetové vyhledávače (Google, Bing, Seznam), webmail servery (Gmail, Hotmail, Seznam) nebo cloudová úložiště (OneDrive, GoogleDrive, iCloud, Dropbox, Uložto), může být cloud využit pro distribuované výpočty (Cloud Skype), provozování běžných uživatelských aplikací (Google Apps, Microsoft Office365, Adobe Creative Cloud), ale i pro systémy přístupné přes prohlížeče (eyeOS).

DISTRIBUČNÍ MODELY

Podle obsahu i způsobu nabídky cloudové služby rozlišujeme tyto modely:

IaaS (*Infrastructure as a Service* - infrastruktura jako služba). Poskytovatel služeb nabízí klientům celou infrastrukturu, typicky virtuální stroje. Hlavní výhodou

tohoto přístupu je, že se o veškeré problémy s hardwarem stará poskytovatel. Příkladem jsou *Amazon EC2*, *Rackspace Cloud* nebo *Microsoft Azure*.

PaaS (*Platform as a Service* - platforma jako služba). V tomto modelu jsou poskytovány kompletní prostředky pro podporu celého životního cyklu tvorby a distribuce webových. Toto řešení, vhodné pro vývojáře, zahrnuje různé prostředky pro vývoj aplikace (IDE, API), ale i pro její údržbu. Příkladem poskytovatelů PaaS jsou *Google App Engine*, *Force.com* nebo *Heroku*.

SaaS (*Software as a Service* - software jako služba). Aplikace je licencována jako služba pronajímána uživateli a přístupná odkudkoliv a kdykoliv. Příkladem může být známá sada aplikací *Google Apps* nebo v logistice známý systém *Cargopass*.

Výhody	Nevýhody
Není nutné udržovat SW a HW	Závislost na poskytovateli
Efektivní řízení, dostupnost dat	Nutnost připojení k Internetu
Jednoduché uživatelské rozhraní	Méně funkcí, horší stabilita
Zvýšení zabezpečení dat	Možné zneužití dat
Škálovatelnost - možnost okamžitého zvýšení výkonu i paměti	Náklady na migraci (přechod na novou platformu) - přeskolení, převod dat...
Růst produktivity práce ve firmách	Jiný právní řád poskytovatele a klienta

Zálohování programů a dat

Záloha nebo záložní kopie (anglicky backup) je kopie dat uložená na jiném datovém nosiči (nebo i místě). Záložní data jsou využívána v případě ztráty, poškození nebo jiné potřeby práce s daty uloženými v minulosti. Zálohování probíhá nepravidelně (např. v domácnostech), nebo pravidelně podle rozvrhu (např. ve firmách). Používají se různé strategie zálohování; volba správné strategie závisí na tom, kde je potřeba se zálohami pracovat velmi často, nebo je naopak požadována maximální délka archivace zálohovaných dat.

PROSTŘEDKY ZÁLOHOVÁNÍ

- **Specializovaný program** (např. ve Windows je součástí instalace).
- **Specializovaná zařízení** (hardware), která pracují automatizovaně.
- **Komplexní zálohovací systémy** umožňující efektivně zálohovat mnoho počítačů propojených počítačovou sítí, nebo naopak na mnoho počítačů propojených v síti data zálohovat (tzv. **úložný cluster**).

TYPY ZÁLOH

Nestrukturovaná. Úložištěm může být větší množství disket, CD, DVD medií s minimem informací o záloze; nejjednodušší, ale nevhodná pro větší firmy.

Úplná + inkrementální. Cílem je vytvořit více kopií zálohovaných dat; poprvé je provedena úplná záloha všech dat; poté je prováděna tzv. **inkrementální záloha** (tj. přírůstková, ukládány jsou pouze soubory, které se změnily od předešlé úplné nebo inkrementální zálohy); při obnově je nutné použít první úplnou i všechny následující inkrementální zálohy (náročné na paměťový prostor).

Úplná + rozdílová. Rozdíl oproti předešlé metodě je v tom, že po úplné záloze každá částečná záloha zachytí všechny soubory vytvořené nebo změněné od vytvoření úplné zálohy, třebaže některé už jsou obsaženy v předešlé částečné záloze; výhodou je, že obnova zahrnuje obnovení pouze poslední úplné zálohy a její překrytí poslední rozdílovou zálohou.

Zrcadlová + reverzní přírůstková. Tento model obsahuje zrcadlo reflektující stav systému po poslední záloze a historii přírůstkových záloh; výhodou je, že máme neustále k dispozici aktuální plnou zálohu a ukládáme pouze historii změn; metoda se nehodí pro přenosná media, protože každá záloha musí být provedena pomocí srovnání se zrcadlem.

Průběžná ochrana dat (žurnálování). Metoda využívá místo plánovaných periodických záloh okamžitý zápis každé změny do **žurnálu** změn (logu); provádí se ukládáním bytů nebo celých bloků dat místo ukládání celých změněných souborů; průběžný záznam změn v žurnálu umožňuje získat obraz dat v minulosti; naproti tomu prosté zrcadlení dat na druhý disk (např. RAID 1) stav v minulosti nezachycuje.

ZÁSADY ZÁLOHOVÁNÍ

- Postupy zálohování volíme v závislosti na konkrétní situaci (jak často dochází ke změně dat, jak jsou data cenná ...).
- Provádíme kontrolu záloh – umožňuje většina programů (např. použití kontrolního součtu).
- Pečlivě popisujeme zálohy (co obsahují, datum vytvoření).
- Pořizujeme zálohy z instalačních médií; originální média by měla být ihned po pořízení kopií uložena na bezpečném místě (včetně instalačních hesel a čísel!), vlastní instalace probíhá z pořízených kopií.
- Ukládáme zálohy na fyzicky různá místa – důležité zálohy by neměly být uloženy u počítače (požár, krádež...).
- Snažíme se zajistit důvěrnost dat (fyzicky, či aspoň zaheslováním zálohy).
- Volíme vhodná zálohovací média (CD, DVD, Flash...); médium volíme podle rychlosti zálohování (čtení), pořizovací a provozní ceny, spolehlivosti média, spolehlivost obnovení, doby uchovávání dat, kompatibilitu...

MOŽNOSTI ZÁLOHOVÁNÍ

Záloha adresářů a souborů. Provádíme ji různými způsoby - prostým kopírováním obsahu plánovanou synchronizací, zálohováním adresáře s komprimací obsahu. Pro běžnou zálohu můžeme použít řadu archivačních programů - např. *WinZip*, *WinRar*, *7zip* a jiné.

Záloha oddílu. Záloha obsahu celého diskového oddílu včetně instalovaného operačního systému (pokud jej oddíl obsahuje), instalovaných aplikací a veškerých dat; na rozdíl od souborové zálohy obsahuje také zaváděcí záznam a skryté části operačního systému, takže při obnově zálohy systémového oddílu je operační systém plně funkční a schopný bootovat (za předpokladu, že byl funkční v době tvorby zálohy).

Záloha disku. Záloha obsahu celého disku včetně rozdělení disku na jednotlivé oddíly (např. *Clonezilla*).

Záloha systému. Záloha funkčního stavu systému, která by umožňuje obnovení systému do původní podoby; je možné použít buď program nabízený samotným OS, nebo některý ze specializovaných programů (*Acronis*).