Lukáš Vykuka, I4E

21. - Digitalizace dat

1. Problematika digitalizace dat, kódování informací ve výpočetní technice

1.1. Problematika digitalizace dat

Digitalizace znamená převedení něčeho do digitální podoby. Zdroje všech běžných informací získávaných smysly (obraz, zvuk) jsou analogové, proto se musí digitalizovat. Dále to může být tištěný text (noviny, knihy), mluvené slovo nebo výsledky nějakého výzkumu. Jednotlivé oblasti využití počítače při digitalizaci bychom mohli rozdělit na:

- správa údajů – Vyšší přehlednost a dostupnost.
- uchovávání údajů – Počítač je velmi levné záznamové médium. Sdílení materiálů je jednodušší a levnější než kopírování pomocí klasických nosičů (papír, kazety, …).
- analytická činnost
- publikování – Mnoho možností publikace (CD-ROM, Internet...). Možnost využití nástrojů zjednodušující publikace (oprava pravopisu, tvorba obsahu publikace,....)

1.2. Kódování informací ve výpočetní technice

Vše v počítači jsou jen nuly a jedničky. Počítače používají tzv. dvojkovou soustavu – jsou to digitální zařízení. 1 bit je nejmenší jednotka informace (0 nebo 1), která říká, který ze dvou stejně pravděpodobných jevů nastal. Technicky lze realizovat pomocí elektrického signálu: 0 – není napětí, 1 – je napětí. Převedení do binární podoby provádějí vstupní zařízení. Těch je několik druhů:

- obrazová (digikamera, skener)
- zvuková (zvuková karta, analogově digitální převodník)
- znaková (klávesnice)

Tato zařízení převedou vstupní hodnoty do proudu bitů, se kterým se počítačům lépe pracuje. Při kopírování nebo přesunu dat v digitální podobě nedochází ke zkreslení nebo ztrátě dat.

2. Váhové (poziční) číselné soustavy (binární, oktalová, hexadecimální), vzájemné převody, základní operace v binární soustavě

2.1. Číselná soustava

Číselná soustava je způsob reprezentace čísel. Podle způsobu určení hodnoty čísla z dané reprezentace rozlišujeme dva hlavní druhy číselných soustav

- poziční číselné soustavy - hodnota každé číslice je dána její pozicí v sekvenci symbolů. Každá číslice má touto pozicí dánu svou váhu pro výpočet celkové hodnoty čísla.
- nepoziční číselné soustavy - způsob reprezentace čísel, ve kterém není hodnota číslice dána

jejím umístěním v dané sekvenci číslic. Tyto způsoby zápisu čísel se dnes již téměř nepoužívají a jsou považovány za zastaralé. Přesto se i nadále používá u římských čísel.

2.2. Desítková (decimální, dekadická) číselná soustava

Desítková soustava zahrnuje číslice 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Pro člověka je nejpřirozenější, protože má 10 prstů :)

2.2.1. Převod z různých soustav

Pro převod z různých soustav do desítkové se nejvíce hodí substituční metoda (je vhodná i pro vzájemné převody mezi dvojkovou, osmičkovou a šestnáctkovou soustavou). Nehodí se pro převod z desítkové do ostatních.

Převod čísel z různých soustav do soustavy desítkové:

- nejprve se číslo, které chceme převést, vyjádří v původní číselné soustavě polynomem
- následně se, již v aritmetice cílové soustavy, spočtou mocniny čísel, každá mocnina se vynásobí hodnotou příslušné číslice a vše se sečte

Například převod čísla 1101101 z dvojkové soustavy do desítkové:

$1*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 64 + 32 + 0 + 8 + 4 + 0 + 1 = 109$

^ polynom ^ vynásobené mocniny čísel

2.3. Dvojková soustava

Dvojková soustava je založená na mocninách čísla 2 a zapisujeme ji číslicemi 0 a 1. S dvojkovou soustavou se můžete nejčastěji setkat ve výpočetní technice, neboť její dva symboly (0 a 1) odpovídají dvěma stavům elektrického obvodu (vypnuto a zapnuto).

2.3.1. Převod do dvojkové soustavy

Spočívá v neustálém dělení tohoto čísla dvojkou - číslo vydělíme 2 a pokud zůstane zbytek (1), bude hodnota 1. Pokud nebude zbytek, bude hodnota 0. Výsledek zapisujeme odspodu. Příklad převodu desítkového čísla 173:

173 : 2 = 86 (1)

86 : 2 = 43 (0)

43 : 2 = 21 (1)

21 : 2 = 10 (1)

10 : 2 = 5 (0)

5 : 2 = 2 (1)

2 : 2 = 1 (0)

1 : 2 = 0 (1)

Výsledek je 10101101

2.4. Osmičková (oktalová) číselná soustava

Osmičková soustava může obsahovat cifry 0, 1, 2, 3, 4, 5, 6 a 7. Stejně jako dvojková soustava funguje na principu mocnin, ale tentokráte čísla 8.

2.4.1. Převod do osmičkové soustavy

Jednoduší to bude předvést na převodu nějakého čísla, např. 594 do oktalové soustavy. Nejprve si zjistíme největší mocninu osmi, která se vejde do čísla 594, a tou je číslo 512 (8^3). 512 se do 594 vejde pouze jednou, takže první číslice bude 1. Dále odečteme 512 od 594 a dostaneme číslo 82 a následuje odečtení mocniny 8^2 (64) – ta se vejde do 82 taky jen jednou (druhá číslice je také 1). Po odečtení získáme číslo 18 a následovat bude odečtení mocniny 8^1 (8), ovšem 8 se vejde do 18 dvakrát, tudíž 3. číslice bude 2 a po odečtení nám zbude 2. Dvojku budeme dělit mocninou 8^0 (1), takže 4. číslice bude rovněž 2. Výsledkem je tedy, že decimální číslo 594 je číslem 1 122 v oktalové soustavě.

2.5. Šestnáctková (hexadecimální) číselná soustava

Šestnáctková soustava zahrnuje číslice 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 a znaky A (10), B (11), C (12), D (13), E (14), F (15) a primárním číslem je 16 (respektive opět jeho mocniny).

2.5.1. Převod do šestnáctkové soustavy

Převod čísla 6 540 do hexadecimální soustavy. Opět si najdeme největší mocninu, která se do čísla 6540 vejde a to je 16^3 (4 096) – to se tam vejde pouze jednou, takže 1. číslice bude 1. Po odečtení dostanete 2 444 a dělit budete mocninou 16^2 (256). Dvě 256 se do 2 444 vejde dokonce devětkrát, takže 2. číslice bude 9. Teď si zjistíte zbytek, a to 2 444 - 9 * 256 = 140. A 140 vydělíte mocninou 16^1 (16) a dostanete se na 3. číslici, na 8. Následuje výpočet 140 - 8 * 16 = 12. Dvanáctka už není dělitelná šestnácti a v hexadecimální soustavě ji reprezentuje písmeno C (což je 4. „číslice“). Decimální číslo 6 540 tedy zapíšeme v hexadecimální soustavě jako 198C. Vysvětlovat převod hexadecimálního čísla na číslo decimální opět není, myslím, nutné.

2.6. Operace v binární soustavě

2.6.1. Aritmetické operace

2.6.1.1. Sčítaní

Přenos jedničky do vyššího řádu je generován nabude-li součet hodnoty 2 (1+1).

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

2.6.1.2. Odčítaní

Provede se převedením na součet menšence a dvojkového doplňku menšitele. Dvojkový doplněk se

získá negací menšitele a přičtením 1 k nejnižšímu bitu. Z výsledku se škrtne nejvyšší bit.

Příklad: 0110 – 0010

Negace menšitele je 1101, po přičtení 1 vyjde dvojkový doplněk 1110. Poté se přičte k menšenci: 0110 + 1110 = 0100.

2.6.1.3. Násobení

Postupuje se stejně jako v desítkové soustavě: postupné sčítaní násobků posunutých o jedno místo vlevo.

0 * 0 = 0

1 * 0 = 0

0 * 1 = 0

1 * 1 = 1

Příklad: 0010 * 0101

```
   0010
  *0101
   0010
  0000
 0010
0000
0001010
```

2.6.1.4. Dělení

„Pokusím se dodat později.“

2.6.2. Logické operace

Viz. materiály na programování.

3. Kódování znaků

Znak je reprezentován jedním bytem (v případě Unicode i více byty). Tento byte může vyjadřovat celé nezáporné číslo. Ke každému číslu z množiny, kterou může byte vyjádřit, je přiřazen určitý znak. Této asociaci určitého znaku k určitému kódu se jinak říká kódování, sadě těchto asociací ke všem možným kódům pak znaková sada (kódová tabulka).

3.1. ASCII

ASCII je anglická zkratka pro American Standard Code for Information Interchange, tedy americký standardní kód pro výměnu informací. V podstatě jde o kódovou tabulku která definuje znaky anglické abecedy, a jiné znaky používané v informatice. Jde o historicky nejúspěšnější znakovou sadu, z které vychází většina současných standardů pro kódování textu přinejmenším v euro-

americké zóně. Tabulka obsahuje tisknutelné znaky: písmena, číslice, jiné znaky (závorky, matematické znaky (+-*/% …), interpunkční znaménka (,.:; …), speciální znaky(@$~ …)), a řídící (netisknutelné) kódy, které byly původně určeny pro řízení periferních zařízení (např. tiskárny nebo dálnopisu). Kód ASCII je podle původní definice 7-bitový, obsahuje tedy 128 platných znaků. Pro potřeby dalších jazyků a pro rozšíření znakové sady se používají osmibitová rozšíření ASCII kódu, která obsahují dalších 128 kódů. Takto rozšířený kód je přesto příliš malý na to, aby pojmul třeba jen evropské národní abecedy. Pro potřeby jednotlivých jazyků byly vytvořeny různé kódové tabulky, význam kódů nad 127 není tedy jednoznačný. Systém kódových tabulek pro národní abecedy vytvořila například organizace ISO.

3.2. Unicode/UCS (Universal Character Set)

Je 31 bitová znaková sada všech existujících abeced a často používaných technických znaků. Vychází z ASCII, používá se ale 31 bitů pro identifikaci znaků, aby bylo možné podporovat vícejazyčné texty. Jediný font tak může obsahovat podporu pro mnoho abeced, ale také různé typografické varianty znaků např. kapitálky, iniciály atd.

Původní cíle standardu Unicode byly tyto:

- univerzálnost - Kapacita znakové sady musí být dostatečná k zahrnutí všech znaků, které by mohly být použity při výměně textů - zejména ty, které už byly definovány v hlavních mezinárodních, národních a průmyslových znakových sadách.
- efektivita - Text, poskládaný z posloupnosti znaků o konstantní šířce, je velmi jednoduchý na zpracování; software nemusí uchovávat stav, dávat pozor na speciální escape sekvence nebo prohledávat text dopředu či zpět kvůli identifikaci znaků.
- jednotnost - Konstatní šířka znaků dovoluje efektivní třídění, hledání, zobrazování a editaci textů.
- Jednoznačnost - Kterákoli 32 bitová hodnota reprezentuje v jakémkoli kontextu stejný znak.

Přirozené kódování znaků Unicode/UCS do 2 nebo 4 bajtů se nazývá UCS-2 a UCS-4. S řetězci uloženými ve formátu UCS-2 nebo UCS-4 je spojeno několik problémů:

- Uložení textu je několikanásobně náročnější na paměť.
- Některé bajty v řetězci mohou obsahovat binární nuly, které mají zvláštní význam v některých programovacích jazycích.
- Některé bajty mohou obsahovat znaky, které mají zvláštní význam pro operační systém (např. „/“, „\“).

Z uvedených důvodů není formát UCS-2 a UCS-4 vhodný pro ukládání řetězců do souborů.

Tyto problémy řeší kódování UTF-8.

3.3. UTF-8

Je to způsob kódování řetězců znaků Unicode/UCS do sekvencí bajtů. Velmi sympatické je, že na obyčejný ASCII text je potřeba pouze jeden byte (prvních 127 znaků je shodných s ASCII tabulkou).

Cíle UTF-8 jsou zejména:

- Kompatibilita se staršími souborovými systémy. Souborové systémy zpravidla nepovolují v

názvech souborů nulový byte ani (zpětné) lomítko.

- Kompatibilita s existujícími programy. Zápis jakéhokoli znaku by neměl obsahovat ASCII, pokud znak původně v ASCII nebyl.
- Snadnost konverzí z/do UCS.

S Unicode textem zapsaným v UTF-8 lze zacházet stejně jako s obyčejným osmibitovým. Není divy, že se UTF-8 prosazuje jako univerzální formát pro výměnu dokumentů v Unicode. Formát UTF-8 odstraňuje všechny nevýhody nasazení Unicode.

4. Dokumenty a jejich formáty (textové, grafické, zvukové…), export a import

Formát dokumentu říká, jak jsou data uvnitř dokumentu uspořádány (vnitřní struktura), jak je ukládat. Jednotlivé formáty jsou mezi sebou v zásadě převoditelné. Každý převod sebou zpravidla nese určitou ztrátu informace a časové, případně finanční náklady spojené s převodem. Měli bychom volit formát, který je pokud možno rozšířený a snadno převoditelný. Formát dokumentu lze poznat podle koncovky ve jméně soubor. Avšak tu lze snadno změnit a přitom formát souboru (dokumentu) zůstane stejný. Další možností, jak zjistit formát souboru (dokumentu), je pomocí hlavičky. Ze souboru se přečte určitá sekvence bytů. Každý formát má tuto sekvenci jedinečnou, takže pak ji stačí porovnat s databází známých hlaviček a tím zjistit formát souboru (dokumentu).

4.1. Textové dokumenty

4.1.1. TXT

Formát .txt je nejjednoduší formát pro ukládání textu. Znaky jsou v něm jsou uloženy jako sekvence bytů, které udávají kód znaku v kódovací tabulce. Ať se text otevřete v jakémkoli programu na jakémkoli počítači, mělo by se objevit to samé. Problém může nastat při nesprávném nastavení kódování. Nejlepší volbou je nastavit UTF-8. Další nepříjemností je, že každý operační systém standartně ukončuje řádek jinak (toto chování jde často přenastavit v textovém editoru).

- MS Windows ukončuje řádek znakem CRLF neboli (\r\n)
- Unixové systémy znakem LF (\n)
- Mac ukončuje znakem CR (\r)

4.1.2. ODF

Je otevřený souborový formát určený pro ukládání a výměnu dokumentů vytvořených kancelářskými aplikacemi. ODF zahrnuje nejen textové dokumenty, ale i prezentace, tabulky, grafy a databáze. Je založen na XML. Formát OpenDocument je standardizován jako ISO. Otevřený znamená, že jeho specifikace je veřejně přístupná a formát může být použit libovolnou aplikací bez omezení. Účelem formátu OpenDocument je nabídnout otevřenou alternativu uzavřeným formátům, především od společnosti Microsoft (DOC, XLS a PPT využívané kancelářským balíkem Microsoft Office) a dále formátu Office Open XML.

4.1.3. DOC

Je to nativní formát aplikace MS Word. Různé verze Wordu ukládají dokumenty v různých variantách tohoto formátu. V zásadě by se dalo říci, že novější verze Wordu jsou navzájem kompatibilní. Soubor uložený ve Word 2000 lze otevřít ve Word 97 atp. Výhodou wordowského formátu je, že sebou nese všechny informace o formátování dokumentu a jeho relativní rozšířenost.

4.1.4. RTF

Rich text formát je jakýmsi pokusem o univerzální formát, který by s sebou oproti formátu TXT nesl také informace o formátování. Je dosti oblíben a používán.

4.1.5. HTML

Je to textový soubor, který obsahuje značky jazyka HTML. Ty jsou uspořádány tak, aby je mohl webový prohlížeč zobrazit jako formátovaný dokument. Tento formát slouží k publikování dokumentů na internetu. Každý lepší textový editor se chlubí možností ukládat text do formátu html.

4.1.6. XML

Jedná se o určitou obdobu formátu HTML s tím, že si sebou nese informace o struktuře dokumentu.

4.1.7. PDF

Zkratka anglického názvu Portable Document Format – Formát pro přenositelné dokumenty. Je souborový formát vyvinutý firmou Adobe pro ukládání dokumentů nezávisle na softwaru i hardwaru, na kterém byly pořízeny. Soubor typu PDF může obsahovat text i obrázky, přičemž tento formát zajišťuje, že se libovolný dokument na všech zařízeních zobrazí stejně. Pro tento formát existují volně dostupné prohlížeče pro mnoho platforem, nejznámějším je oficiální prohlížeč mateřské firmy Adobe – Adobe Reader.

4.2. Grafické dokumenty

Obrázky mohou být v počítači uloženy dvěma hlavními způsoby, jako:

- rastrový obrázek
- vektorový obrázek

4.2.1. Rastrové formáty (bitmapa)

V souboru jsou uloženy informace o barvě a jasu jednotlivých bodů obrázku (jako kdybychom si na něj položili mřížku a pak určovali jednotlivá políčka). Mezi hlavní charakteristiky bitmap patří počet zobrazovaných bodů a hloubka barev. Od tohoto údaje se pak odvíjí další vlastnosti závislé na formátu ve kterém je obrázek uložen, jako jsou rozměry obrázku, rozlišení, velikost souboru s obrázkem, barevný model, komprimace, přítomnost vrstev a objektů.

4.2.1.1. BMP

Nekomprimovaný formát. V prostředí MS Windows představuje jakýsi standardní typ obrázku.

Zabere hodně místa.

4.2.1.2. JPEG

Komprimovaný obrázek. Drtivá většina obrázků na internetu je uložena ve formátu JPEG. Při ukládání můžete nastavit úroveň ztrátové komprese na škále 1-100 (kde 1 = neztratí se žádná informace, 100 = maximální komprese a také max. degradace obrázku).

4.2.1.3. GIF

Komprimovaný obrázek. Na rozdíl od JPEG umí pracovat pouze s obrázky v paletě max. do 256 barev. Používá neztrátovou kompresi. Často se používá na internetu. Jeho výhodou je mimo jiné to, že si v něm můžete nastavit jednu z barev jako průhlednou.

4.2.1.4. PNG

Grafický formát určený pro bezeztrátovou kompresi rastrové grafiky. Byl vyvinut jako zdokonalení a náhrada formátu GIF, který byl patentově chráněný (LZW84 algoritmus). Oproti němu nabízí podporu 24 bitové barevné hloubky, lepší kompresi (algoritmus Deflate + filtry), 8 bitovou (obrázek může být v různých částech různě průhledný). Nevýhodou PNG oproti GIF je praktická nedostupnost jednoduché animace. PNG se stejně jako formáty GIF a JPEG používá na Internetu.

4.2.2. Vektorové formáty

V souboru jsou uloženy informace o tvaru a tloušťce jednotlivých čar, barvách výplní, atp. Obrázek se neskládá z jednotlivých bodů rastru jako u bitmapy, ale z objektů.

4.2.2.1. SVG

Je značkovací jazyk a formát souboru, který popisuje dvojrozměrnou vektorovou grafiku pomocí XML. Formát SVG by se měl v budoucnu stát základním otevřeným formátem pro vektorovou grafiku na Internetu.

4.2.2.2. WMF, EMF

Grafické metaformáty WMF (Windows Metafile) a EMF (Enhanced Windows Metafile) byly navrženy firmou Microsoft pro její známé operační systémy Microsoft Windows.

4.3. Zvukové dokumenty

Digitalizace zvuku spočívá v tom, že se analogový zvukový záznam rozdělí na určitý počet vzorků, pro které se určí jaké zvukové vlastnosti v nich převládají. Množství vzorků a rozsah sledovaných vlastností pak určují kvalitu a také velikost daného zvukového záznamu. Do počítače můžete záznam uložit tak jak je (např. formát wav) nebo záznam komprimovat (MP3). Nahrávat zvuk do počítače můžeme z běžných hudebních CD, z kazet, z diktafonů atp. S vhodným softwarovým vybavením můžeme zvuk následně editovat - úprava hlasitosti, odstranění šumu, stříhání na kratší sekvence (např. určené k transkripci), případně přepisovat kousky signálu zvoleným zvukem (např. kvůli anonymizaci).

4.3.1. WAV

Formát wav ukládá jednotlivé vzorky bez dalších zvláštních úprav. Vzorků je samozřejmě mnoho a jsou velmi krátké - klasické zvukové CD např. má vzorkovací frekvenci 44,1 kHz (tedy 44 100 vzorků za vteřinu). Z nastavení, kterými ovlivňujete rozsah sledovaných hodnot vzorků, máte zpravidla možnost ovlivňovat, zda je záznam MONO nebo STEREO a zda jsou vzorky ukládány jako 8, 16, 32 ... bitové. Čím více bitů, tím více informací sebou údaj nese. Výhody: je to formát se kterým se dobře manipuluje, je vhodný např. pokud potřebujete digitalizovanou nahrávku nějakým způsobem editovat (hlasitost, redukce šumu).

4.3.2. MP3

Představuje tzv. ztrátovou kompresi dat. Stejně tak jako u formátu wav je jeho velikost závislá na vzorkovací frekvenci a rozsahu sledovaných hodnot. Kromě toho je ale záznam ochuzen o zvukové rozsahy a vlastnosti, které lidské ucho stejně neslyší, příp. nejsou pro slyšení důležité. Kvalita záznamu se potom určuje také kompresním poměrem - velká komprese = malý soubor a nízká kvalita zvuku, malá komprese naopak. Dalšími faktory, které ovlivňují kvalitu komprese jsou:

- použitý psychoakustický model, který (zjednodušeně řečeno) určuje co je a co není důležité pro naše ucho
- komprimační algoritmus

4.4. Audiovizuální dokumenty

Obsahují statické nebo pohyblivé obrázky, často doprovázené zvukem. Audiovizuální dokument je tvořen audiovizuálním kontejnerem (AVI, MP4, Matroska....). Tento kontejner obsahuje různé datové stopy (video, zvuk atd). Stopy jsou pomocí kodeku (Lame, Xvid....) za/dekodovány do/z nějakého formátu (MPEG, MP3, WMA....).

5. Funkce skeneru, OCR programy

5.1. Skener

Skener je hardwarové vstupní zařízení umožnující převedení fyzické 2D nebo 3D předlohy do digitální podoby pro další využití, většinou pomocí počítače. Existuje několik typů:

- čtečky čárových kódů
- ruční
- stolní
- bubnové
- filmové
- 3D

5.2. OCR

Neboli optické rozpoznávání znaků (z anglického Optical Character Recognition) je metoda, která umožňuje digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým

textem. Jinak řečeno je to technologie převodu textu uloženého v bitmapovém formátu do formátu textového. To není vůbec triviální záležitost. Text uložený v bitmapovém obrázku není chápán jako text, je to jen sada tmavých a světlých bodů v obrázku. OCR program tedy musí identifikovat v bitmapě různé tvary a porovnat je s předlohou a rozhodnout jaké písmenko, ten který shluk představuje. Novější trasovací programy (tak se jim také říká), pracují tak že dokument procházejí několikrát za sebou a při posledních průchodech už spolupracují se spelcheckerem (to co např. kontroluje pravopis v MS Word.). Převedený text je téměř vždy v závislosti na kvalitě předlohy třeba podrobit důkladné korektuře, protože OCR program nerozezná všechna písmena správně.

5.3. OCR software

- OmniPage
- Adobe Acrobat
- Tesseract (pro více OS)

6. Komprimace (komprese)

Je to speciální postup při ukládání nebo transportu dat. Úkolem komprese dat je zmenšit datový tok nebo zmenšit potřebu zdrojů při ukládání informací. Zvláštními postupy – kódováním, které je dané zvoleným kompresním algoritmem – se ze souboru odstraňují redundantní (nadbytečné) informace. Komprese dat lze rozdělit do dvou základních kategorií:

- ztrátová komprese – Při kompresi jsou některé informace nenávratně ztraceny a nelze je zpět rekonstruovat. Používá se tam, kde je možné ztrátu některých informací tolerovat a kde nevýhoda určitého zkreslení je bohatě vyvážena velmi významným zmenšením souboru. Používá se pro kompresi zvuku a obrazu (videa), při jejichž vnímání si člověk chybějících údajů nevšimne nebo si je dokáže domyslet (do určité míry).
- bezeztrátová komprese – Obvykle není tak účinná jako ztrátová komprese dat. Velkou výhodou je, že komprimovaný soubor lze opačným postupem rekonstruovat do původní podoby. To je nutná podmínka při přenášení počítačových dat, výsledků měření, textu apod., kde by ztráta i jediného znaku mohla znamenat nenávratné poškození souboru.

6.1. Kompresní poměr

Je to podíl velikosti původních dat ku velikosti komprimovaných dat. Například při kompresi 10MB souboru do 2MB je poměr 10/2 = 5 (tj. 5 : 1 – pět ku jedné, pětkrát zmenšeno). Kompresní poměr je ovlivněn volbou kompresního algoritmu i typem komprimovaných dat.

6.2. Kompresní formáty

6.2.1. RAR

RAR je populární proprietární souborový formát pro kompresi dat a archivaci vyvinutý ruským programátorem Jevgenijem Rošalem. Autor WinRARu dal k dispozici zdrojové kódy dekomprimačního programu, díky tomu je možné RAR dekomprimovat i v jiných programech na různých platformách. Spolu s formátem 7z dosahuje RAR při kompresi velmi dobrých výsledků a nabízí řadu pokročilých funkcí (například vícesvazkové archivy), které ostatní formáty (například

rozšířenější ZIP) často nenabízí.

6.2.2. 7-Zip (7z)

7-Zip je komprimační program určený pro různé operační systémy. 7-Zip je svobodný software, vyvíjený Igorem Pavlovem a distribuován pod licencí GNU LGPL. Je konkurencí k známým programům jako WinZip a WinRAR. 7-zip používá přednostně kompresní algoritmus LZMA.

6.2.3. BZIP2

Je to svobodný komprimační algoritmus a program vyvinutý Julianem Sewardem. U většiny souborů pracuje bzip2 efektivněji než u ZIP. Na rozdíl od formátů jako RAR nebo ZIP, bzip2 neumí pracovat s více soubory, zkomprimovat dokáže pouze jeden soubor. Tento princip vychází ze základů unixu, kde program TAR spojí více souborů dohromady a bzip2 tento soubor pak zkomprimuje.

6.2.4. JPEG

Je to standardní metoda ztrátové komprese používané pro ukládání počítačových obrázků ve fotorealistické kvalitě. Formát souboru, který tuto kompresi používá, se také běžně nazývá JPEG.