Úvod do informačních technologií

ostatní témata

Martin Trnečka

Katedra informatiky
Univerzita Palackého v Olomouci

Pixel

- bezrozměrná jednotka
- rozlišení = počet pixelů
- ppi = počet pixelů na palec, dpi = počet bodů na palec
- 1920px
  - monitor 24 palců (šířka 20 palců), rozlišení 1920px $\rightarrow \frac{1920}{20} = 96$ppi
  - tisk 8,3 palců (formát A4), $\rightarrow \frac{1920}{8,3} \approx 231$ppi
  - tisk 11,7 palců (formát A3), $\rightarrow \frac{1920}{11,7} \approx 164$ppi
- referenční pixel, nezávislý pixel

Barevný model

- reprezentace barev v počítači
- barevné modely popisují barvu pomocí různých částí (složek)
- aditivní (sčítání složek, více $\rightarrow$ světlejší)
- subtraktivní (odčítání složek, více $\rightarrow$ tmavší)

RGB

- aditivní model
- displeje
- barva = červená (R), zelená (G) a modrá složka (B)
- složka reprezentována (obvykle) jako 8-bitové číslo
- každá složka 0–1, (v případě 8-bit. reprezentace 0–255)
- RGBA = RGB + průhlednost
- zápis: hexadecimální, `rgb()` (web)

CMY(K)

- subtraktivní
- tiskárny
- barva = tyrkysová (C), fialová (M), žlutá (Y), černá (K)
- složka v %
- prakticky: problém s přechodem od RGB k CMYK

HSV, HSL

- alternativní reprezentace RGB modelu
- HSV barva = odstín (H), sytost (S), jas (V)
  - nejvíce odpovídá lidskému oku
- HSL barva = odstín (H), sytost (S), světlost (L)
- odstín = 0–360 (poloha na barevném kole), ostatní %
- populární v grafických nástrojích

Reprezentace obrazu v počítači

- obraz $\rightarrow$ intuitivní chápání
- obrazová funkce $z = f(x, y)$
- převod obrazu do diskrétní reprezentace
  - vzorkování = diskretizace x a y
  - kvantování = diskretizace z
- rozlišení obrazu
- počet barev

Uložení obrazu

- typy obrázků
  - bitmapové
  - vektorové
- komprese
  - bezztrátová
  - ztrátová

Uložení obrazu: Příklady formátů

- JPEG, ztrátová komprese, míru lze nastavit, ideální pro fotografie, nebezpečí kumulativní ztráty kvality, příliš vysoká komprese → čtverečkované oblasti, nepodporuje průhlednost, progresivní JPEG
- PNG, určeno pro webovou grafiku (náhrada za GIF), různé typy barevných palet (8, 24 a další), bezztrátová komprese, progresivní PNG, vhodné pro obrázky obsahující velké plochy stejné barvy
- WebP, určeno pro webovou grafiku ztrátová i bezztrátová komprese, animace, progresivní WebP
- AVIF, ztrátová komprese, slabší podpora, nepodporuje progresivní
- TIFF bezztrátová komprese, původně určeno pro tisk
- ukázka: `https://squoosh.app/`

Odbočka: Metadata

- dodatečné informace
- Exif formát

Vektorová grafika

- obraz složen ze základních objektů (bod, přímka, polygon, kružnice, křivka)
- přesný popis
- libovolné zvětšení a zmenšení
- obvykle úspornější než bitmapová grafika, ale náročnější práce (vytváření, editace)
- může obsahovat bitmapovou grafiku
- formáty: SVG, EPS, PDF

SVG

- značkovací jazyk
- původně určený pro webovou grafiku → lze vnořit do HTML, modifikovat pomocí CSS a JS
- rozšířený a populární formát

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" width="300" height="300"
  xmlns="http://www.w3.org/2000/svg">
  <rect width="100%" height="100%" fill="#016BAB" />
  <circle cx="150" cy="150" r="90" fill="white" />
  <text x="93" y="160" font-size="28" fill="#016BAB">inf.upol.cz</text>
</svg>
```

Písmo

- součást OS, lze přidávat
- bitmapové písmo
- outline písmo
  - založeno na křivkách
  - Type1, TrueType
  - třeba přizpůsobit pixelové mřížce
- rodiny písem
  - patkové písma (serif), např. Times, Georgia
  - bezpatkové písma (sans-serif), např. Arial, Verdana
  - stejná velikost písmen (monospace), např. Courier
  - kurzíva, např. Comic Sans
  - dekorativní (fantasy), např. Impact
- font znatelně ovlivňuje prožitek ze čtení
- typografie

Bezpečnost

- obecné požadavky (na systém)
  - data jsou přístupná pouze uživatelům, kteří na to mají nárok
  - data mohou měnit jen uživatelé, kteří na to mají nárok
  - ověření identity uživatele
- důvody porušení bezpečnosti

Bezpečnost z pohledu OS

- procesy a soubory mají vlastníka
- oprávnění v OS = řízení přístupu k procesům a souborům
- oprávnění (kdo, co, s čím)
- různé implementace
  - Access Control List (ACL) = seznam uživatelů a jejich práv (Windows)
  - Role-Based Access Control (RBAC)
  - přístupová práva pro vlastníka, skupinu a ostatní (UNIX)

Útoky

- hrubá síla
- trojský kůň, backdoors
- spoofing, phishing
- Denial of Service (DoS), Distributed Denial of Service (DDoS)
- malware (virus, rootkit, spyware, adware, ransomware)
- postraní kanál
- sociální inženýrství
- Cross-site scripting (XSS), SQL injection

Kryptografické hashovací funkce

- libovolný vstup, výstup pevné délky
- základní vlastnosti:
  - hash $f(x)$ je snadno spočitatelný pro libovolné x
  - drobná změna x způsobí velkou změnu $f(x)$
  - z hodnoty $f(x)$ je obtížné určit x
  - je těžké nalézt dvě různé x_1 a x_2 takové, že $f(x_1) = f(x_2)$
- příklad: MD5, SHA-1 (obě nejsou bezpečné), SHA-256
- `KMI/UDITE` $\rightarrow$ `2844012995373d81dbcd20d0bac97b34`
  `LMI/UDITE` $\rightarrow$ `10da7dc30e9c8b709545f813dcc55bfe`

Heslo

- silné heslo $\rightarrow$ určené počtem pokusů potřebných pro uhádnutí hesla
  - `https://www.purecloudsolutions.co.uk/how-long-will-it-take-to-hack-your-password/`
- obtížně zapamatovatelné heslo = bezpečnostní riziko
- únik hesla (k účtu) lze ověřit ve veřejných databázích
  - `https://haveibeenpwned.com/`
  - pozor na podvodné weby
- ukládání hesel
  - nikdy v textové podobě
  - obvykle tzv. sůl, tedy kryptografický hash $f(heslo + sul)$, sůl je pro každé heslo jiná

Šifrování

- zabránění čtení neoprávněnou osobou
- symetrické šifrování
  - symetrický klíč
  - problém výměny klíče
- asymetrické šifrování
  - asymetrický klíč
  - dva druhy klíčů: soukromý a veřejný, navzájem propojeny:

$$f_s(f_v(m)) = m = f_v(f_s(m))$$

  - z veřejného nelze odvodit soukromý
  - generování klíčů: např. RSA, eliptické křivky
- šifrování vs. utajení (např. steganografie)

Integrita zprávy a elektronický podpis

- integrita = zpráva nebyla změněna
  - kryptografické hashovací funkce
  - posílá se: zpráva, hash zprávy a ověřovacího klíče (zabránění podvržení zprávy)
  - běžně se používá pro ověření integrity datových souborů
- podpis = ověření autora
  - využití asymetrického šifrování
  - podpis zprávy $m = f_s(m)$
  - podpis je platný pouze pro zprávu $m \rightarrow$ integrita dat
  - výpočetní náročnost hashování $\rightarrow$ podpis hashe zprávy
  - certifikace veřejných klíčů $\rightarrow$ ověření vlastníka klíče certifikační autoritou (podepsání veřejného klíče)

Blockchain

- řetězec bloků (data, hash dat a hash předchozích dat) = datová struktura
- distribuované (P2P) a bezpečné uchovávání dat
- představeno v roce 1991 jako distribuovaná účetní kniha (DLT), 2009 Satoshi Nakamoto → Bitcoin protokol → hype
- v kostce:
  - problém v P2P: shoda na blockchainu (každý uzel může mít jiný)
  - nejdelší řetězec je považován za validní
  - teoretický útok: získat 51 % výkonu → nejdelší řetězec je vytvářen 1 entitou
  - řešení: vytvoření nového bloku je výpočetně náročné (různé metody, např. proof-of-work)
  - každé přidání bloku = potvrzení předešlých
  - změna v existujícím bloku zneplatní předešlé bloky
- použití: velmi populární, kryptoměny

Blockchain: Příklad Bitcoin

- blockchain = účetní kniha (kdo, komu, kolik), všechny transakce
- transakce, digitální podpis pro verifikaci
- blok = několik transakcí
- přidání bloku (těžba Bitcoinů)
  - umístění transakcí do bloku, přidání odkazu na předchozí blok, ověření pravidel protokolu
  - verifikace bloku (uhádnutí vstupní hodnoty SHA-256, každé 2 týdny se mění obtížnost, čas vždy cca 10 minut)
  - nejrychlejší vyhrává (získá odměnu)
  - odměna = poplatky + nové bitcoiny (cca každé 4 roky se zmenší na polovinu, 2140 dojdou)
- vylepšení: smart contracts
- kritika: mining-pools $\rightarrow$ centralizace, spotřeba elektrické energie, škálovatelnost

Verzovací systémy

- verzování software
- výhody: spolupráce na vývoji, evidence změn, ...
- lokální, centralizované a decentralizované (běžnější)
- například
  - Git
  - Mercurial
  - SVN
  - Apache Subversion

Git

- verzovací systém
- v roce 2005 vytvořil Linus Torvald pro verzování Linukového jádra
- Git (verzovací systém) $\neq$ GitHub (repozitář)
- poměrně komplikovaný systém
- ukládá stav všech souborů (na rozdíl od jiných, které ukládají seznam změn)
- data jsou uložena lokálně (repozitář) a nachází se ve třech stavech
  - modified (změněná neuložená data)
  - staged (data připravená k zapsání)
  - committed (zapsaná data)
- zapsání změn (vytvoření verze) = commit, obvykle obsahuje popis
- aktualizace lokálních dat ze vzdálených (pull)
- aktualizace vzdálených dat z lokálních (push)

Git: Větvení

- odklon od hlavní větve
- Git realizuje jako ukazatel
- aktuální větev (HEAD ukazatel)
- sloučení větví (merge), vytvoření nové verze spojující větve
- přeskládání (rebase), sloučení větví do jedné (ztráta větve)