

## 1. ZÁKLADY POČÍTAČOVÉ GRAFIKY

### 1.1.1 Úvod



**Cíl** Po prostudování této kapitoly budete znát

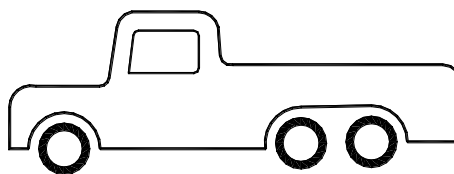
- význam počítačové grafiky pro praxi
- základní prostředky počítačové grafiky



**Výklad**

Význam počítačové grafiky se zvýšil s rozvojem a možnostmi výpočetní techniky. Technický záznam obrazu umožňuje předávat informace graficky a nikoliv jiným způsobem (například tabulkou, kde je popsán popis funkce nebo situace). Rozdílnost vnímání předání informací ukážeme na následujícím obrázku.

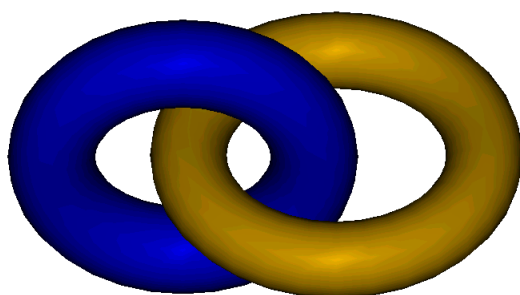
**AUTO**



Obr. 1.1.

Při pohledu na obrázek 1.1 si můžeme přečíst pojem „AUTO“ a prohlédnout si snímek auta.

**Anuloidy**

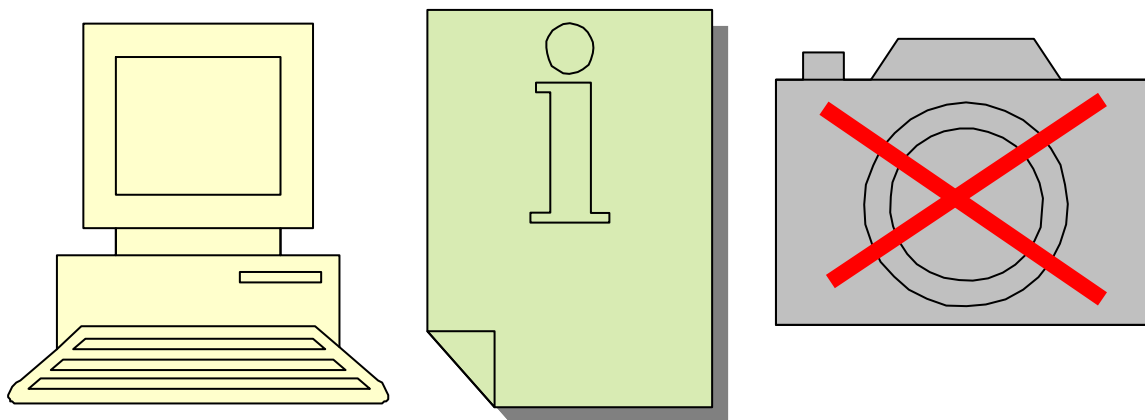


Obr. 1.2

Při přečtení pojmu „AUTO“ není nic řečeno o typu nebo barvě auta. Při pohledu na obrázek auta je ihned vidět, že jde o nákladní auto, resp. barvu i další informace lze z obrázku vyčíst. Záleží pouze na kvalitě obrázku. Nerozhoduje však pouze technická kvalita obrazu. Nezanedbatelné je i vyvolání určitého estetického dojmu při vnímání „přenosu“ informací. Shlédnout snímek určitého předmětu

je daleko příjemnější a snazší než pročítat popis. Při vnímání určitého snímku vidíme i jeho vlastnosti a vztah k okolí. Pohled na snímek dává informaci i o poloze apod. I při velmi podrobném popisu obrázku č. 1.2 nám pouhý pohled na tento obrázek řekne více.

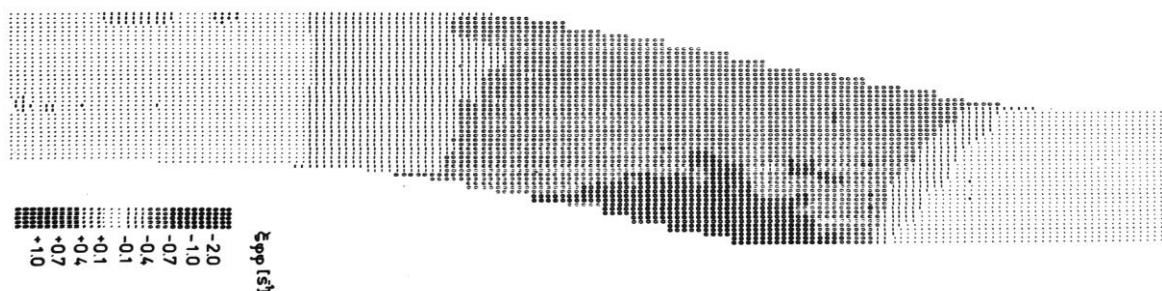
V současné době pracují i textové editory v grafickém módu a tak je umožněno do textu nejen vkládat obrázky vytvořené v jiném prostředí, ale i určité obrázky přímo vytvářet. Tím se značně rozšířila informační možnost vytvářeného tiskového dokumentu.



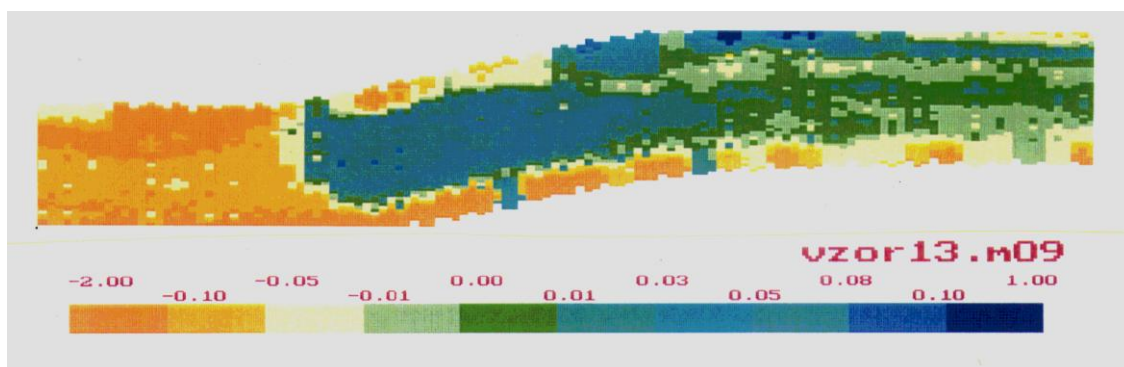
Obr. 1.3

Na uvedeném obrázku 1.3 je příklad informačních obrázků, které obsahuje přímo textový editor. Informace, které obrázky poskytují jsou zřejmé. Grafický modul textového editoru umožňuje kreslit i jednoduché entity (úsečky, vektory, kružnice a pod.). Grafický mód umožnil využití velkého množství fontů, které počítačová grafika poskytuje textovým editorům.

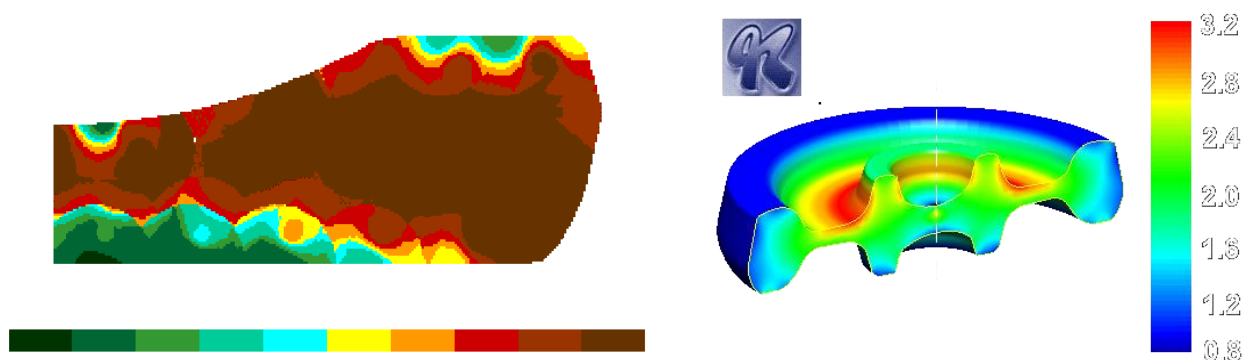
Z hlediska využití se počítačová grafika dělí do několika skupin. Vliv na toto dělení byl dán nemalou měrou technickými možnostmi výpočetní techniky. V době, kdy byly k dispozici převážně znakové tiskárny ať rychlořádkové nebo psací stroje připojené k počítači jako terminály, byly vytvářeny obrazy pomocí těchto výstupních zařízení přetiskem znaků. Takto byl vlastně vytvářen **pseudografický** obraz. Tento nebarevný obrázek pořízený na znakové tiskárně (obr. 1.4a) nahradil tabulku čísel, která udávala např. hodnoty napětí při deformaci materiálu. Obrázek udává lepší celkový přehled v daném prostoru, než příslušná tabulka. Z dnešního hlediska se může zdát vytváření takovýchto grafických obrazů nekvalitní nebo nedůstojné grafiky. Není tomu tak ani dnes, kdy jsou k dispozici plotry, barevné tiskárny, které jsou rychlejší, technicky dokonalé a pod.



Obr. 1.4a



Obr. 1.4b

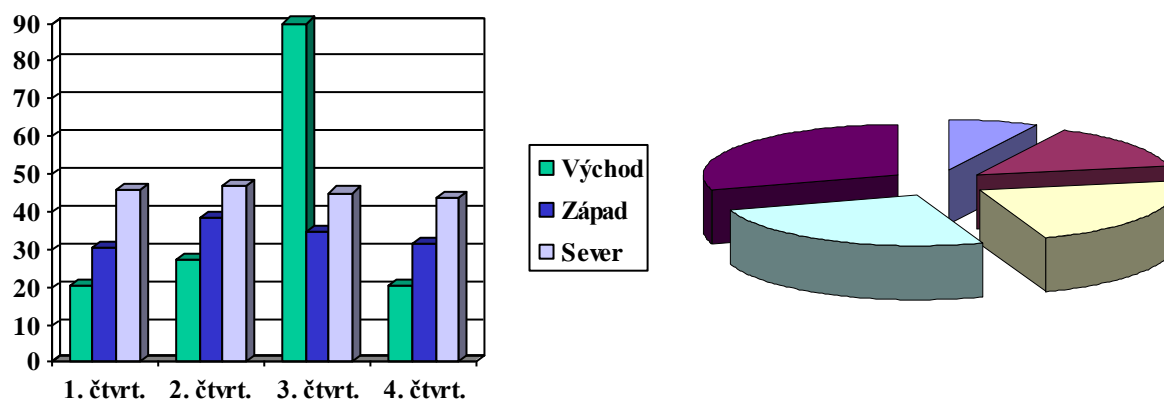


Obr. 1.5

Na obrázku č. 1.4b je ukázka výstupu barevnému znázornění intenzity deformace při válcování materiálu. Z obrázku je patrné, kde je deformace největší. Intenzita deformace je odstupňována níže vyznačenou stupnicí. Je zřejmé, že tento obrázek nedává přesnou informaci o deformační síle v určitých souřadnicích průřezu. Ale dává celkový pohled, který by byl méně přehledný v tabulce, byť sebezpřesnější. Na obrázku 1.5 je ukázka barevného vyznačení měřených vlastností ve 2D a 3D prostoru.

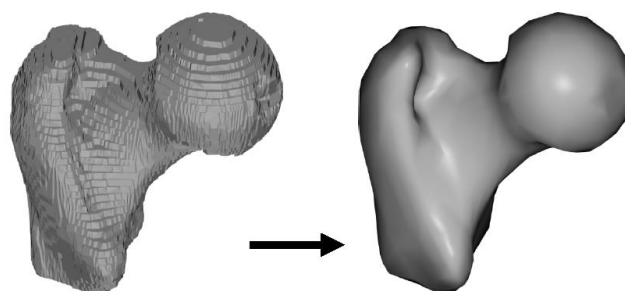
Tam kde nejde o přílišnou přesnost pseudografikou lze zobrazovat matematické funkce, které jsou výstupem programů. Grafický výstup, kde znaková políčka obsahují grafické znaky jiné než písmena se nazývá **semigrafika**.

Nepřehlédnutelné využití počítačové grafiky je v obchodní oblasti. Jde o tzv. **obchodní grafiku**. Na obrázku 1.6 je ukázka sloupcového a kruhového grafu.



Obr. 1.6

Využití počítačové grafiky v medicíně je nezastupitelné. Technické možnosti umožní zobrazit skutečný stav 3D objektu. Na obrázku 1.7 je snímek pořízený tomograficky a upraven metodami počítačové grafiky.



Obr. 1.7

Dosud jsme se seznámili s některými možnostmi zobrazování dat. Tyto možnosti jsou dány vyspělostí techniky, což je patrné z uvedených obrázků. Tyto obrázky byly pořízeny v rozmezí let 1985 až 2005. Je tedy možné částečně porovnat vyspělost zobrazovací techniky v této době. Kvalita obrázku je samozřejmě závislá na technice. To znamená velikosti paměti a rychlosti počítače. I když obojí se mnohonásobně zvětšilo, současným požadavkům stále nevyhovuje. Zvětšující se nároky na techniku způsobují finanční náročnost a kvalitu vzdělání uživatele. Tato náročnost je vyvážena výhodami, které počítačová technika umožňuje. Jde především o archivaci a aktualizaci výsledků. Jestliže výkresy byly uloženy v papírové formě, byla manipulace a archivace velice náročná. Aktualizace těchto výkresů byla téměř nemožná. Současná archivace v digitální formě umožňuje snazší archivaci i aktualizaci.

## 1.1.2 Programové vybavení pro počítačovou grafiku

O významu grafické komunikace již bylo zmíněno v úvodu 1.1 (zřetelnost a srozumitelnost a pod.). S rozvojem výpočetní techniky vznikla potřeba "normalizovat" určité entity tak, aby při narůstání dalšího počtu programových balíků tyto měly alespoň stejný popis základních grafických prvků - entit (úsečky, kruhové oblouky, kružnice, elipsy, polygony, generování znaků a pod.). A tak vznikly programové balíky, které tyto entity v daném systému a daném HW vykreslí. Pro větší názornost prvky počítačové grafiky využívají i programy, které nejsou pro grafiku programovány. Od operačních systémů až po databáze je grafika využívána jako vhodnější komunikační prostředek. Je však velice špatný názor, že počítače používané pro řešení geometrických úloh pouze nahrazují kreslicí "papírovou" techniku. Konečná grafická podoba pořízená počítačem, kterou dostaneme jako výstup z připojeného výstupního zařízení. Např. obrazovka, tiskárna, plotr apod. Velice důležitým výstupem je výstupní soubor, který můžeme poskytnout prakticky okamžitě na jiné výstupní zařízení. To umožňuje řešit složité a náročné grafické úlohy, aniž bychom u počítače vlastnili drahé výstupní zařízení.

### 1.1.3 Porovnání tradičního způsobu řešení geometrických úloh a použití počítače.

Do programového vybavení je nutno zahrnout vše, co je v PC dostupné. Tedy knihovny, moduly a systémy. Vše to je možné pro řešení geometrických problémů využít. Tím se liší využití počítačů pro řešení geometrických úloh od tradičního postupu, kdy řešitel byl odkázán na kreslicí pomůcky tužku a papír. Využití programového vybavení má podstatný vliv i na způsob řešení úlohy. Výhoda je v tom, že řešitel nemusí daný problém nejprve zobrazit na papír a posléze úkol řešit. Může se soustředit na prostorové řešení bez ohledu na zobrazení ať zadání či výsledku. To za něj udělá grafický systém, ve kterém řešitel pracuje. Výrazně se však liší zadávání vstupních dat. Při zadávání tradičním způsobem řešitel viděl nakreslené vstupní prvky a mohl ihned posoudit, zda je úloha dobře zadána. Při zadávání geometrických úloh digitálním způsobem, je nutno dle požadavků zadavatele vstupní data kontrolovat. Jako ukázkou uvedu jednoduchý příklad.

**Příklad:** Sestrojte průsečíky dané kružnice a dané přímky.

Při zadání kružnice středem a poloměrem je třeba kontrolovat souřadnice středu a to jejich počet (2D, 3D) a poloměr  $r \neq 0$ . Jestliže je kružnice  $k$  zadána body  $A$ ,  $B$  a  $C$ , je nutno

kontrolovat počet bodů i jejich souřadnice. Jestli nejde o body totožné po případě, jestli neleží na přímce. Taktéž vlastní řešení konstrukce průsečíku přímky a kružnice má postup odlišný.

Na obrázku 1.8 jsou zadány tři různé přímky  $o$ ,  $m$  a  $l$  dané body  $K$ ,  $L$ ,  $M$ ,  $N$ ,  $O$  a  $P$ . Z pohledu tradičního řešení je zřejmá poloha přímek  $o$ ,  $m$  a  $l$  a kružnice  $k$ . Je tedy zřejmé jakou polohu jednotlivá přímka zaujímá vzhledem ke kružnici  $k$ . Dle toho je vybráno řešení. Při řešení na počítači musíme nejprve tuto polohu zjistit. A to tak, že určíme vzdálenost zadané přímky od středu kružnice  $k$ . Vzdálenost přímky od středu kružnice je dána vzdáleností průsečíku dané přímky a kolmice na danou přímku středem kružnice.

Poloha přímky  $l$ : úsečka  $/SD_l/ > r$ , úloha nemá řešení.

Pro polohu přímky  $m$  platí:  $/SD_m/ = r$ , je přímka  $m$  je

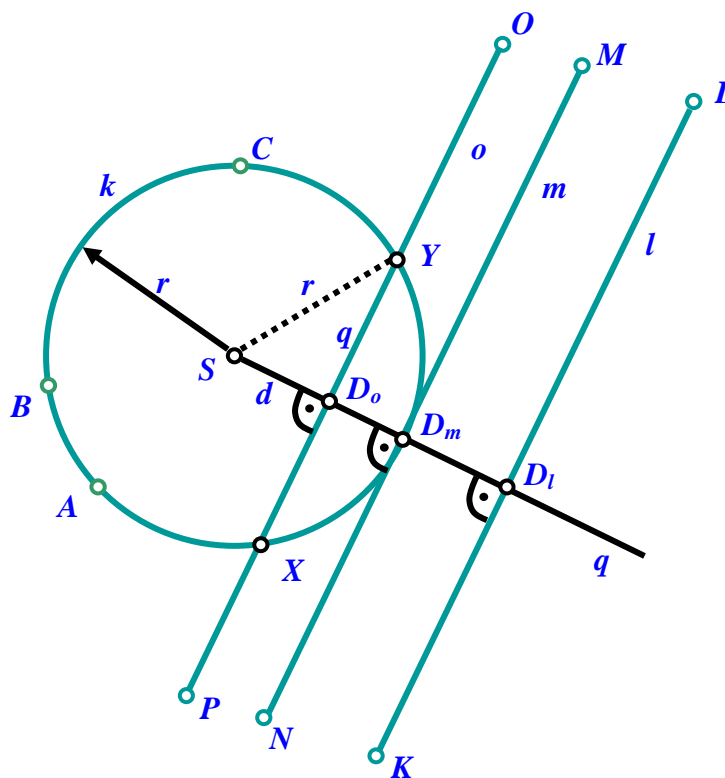
tečnou kružnice  $k$  a bod  $D_m$  je společným bodem přímky  $m$  a kružnice  $k$ .

Pouze poloha přímky  $o$  má dva průsečíky  $X$ ,  $Y$  přímky  $o$  a kružnice  $k$  ( $/SD_o/ = d < r$ ).

Využití počítače nám však umožňuje řešit úlohu mnoha způsoby. "Tradiční" řešení - matematické je řešení soustavy dvou rovnic - kvadratické a lineární. Při tomto řešení však provedeme analýzu řešení znovu. Což je zbytečné. Z obrázku je zřejmé, že hledané body  $X$  a  $Y$  jsou body přímky  $o$  vzdálené od bodu  $D_o$  o délku  $q$ , kde  $q = \sqrt{r^2 - d^2}$ .

Z uvedeného postupu je zřejmé, že analýza řešení geometrických problémů je rozdílná od tradičního postupu a velice závislá na znalostech řešitele a to odborných i programátorských.

Daleko větší rozdíl v postupu řešení je při řešení úkolů ve 3D prostoru. Zde řešitel nemusí znát žádnou zobrazovací metodu, aby mohl problém převést do roviny (2D). Řešitel má k dispozici dostatečný zobrazovací program (systém), ve kterém může daný problém řešit.



Obr. 1.8

Opět ale stejný požadavek - vedle odborných znalostí musí umět tento program ovládat. Uživatel může pro prostorové řešení využít i jiných programových modulů, které jsou v počítači implementovány.

Tradiční způsob řešení geometrických úloh vyžadoval zobrazení problému v rovině. Pro řešení rovinných úloh zde většinou nebyl problém. Zadávání souřadnic však bylo omezeno přesností kreslicích pomůcek a pečlivostí řešitele. Řešení prostorové úlohy (3D) však bylo prováděno v rovině, kdy daný problém bylo nutno do roviny transformovat. K tomu bylo nutno dokonale znát zobrazovací metodu a byla nutná prostorová představivost řešitele. Výsledek řešení nebyl vždy zcela názorný a k interpretaci a pochopení byla taktéž nutná prostorová představivost uživatele.

Řešení pomocí počítače se liší od tradičního hned při zadávání vstupních dat. Jak již bylo výše naznačeno je nutno kontrolovat jednak správnost načítaných hodnot a to co do obsahu tak významu. Vstupní data je nutno uchovat v původní formě, jak je zadal uživatel. Úpravu vstupních dat (formát) je možno provést pro výpočty a konstrukce. Pro výstupní informaci uživateli je však vhodné převést hodnoty v jednotkách zadaných uživatelem.

Pro prostorové řešení úlohy je možno využít i procedur, které prostorový problém řeší, aniž by daná úloha byla zobrazena. Čímž se liší počítačové řešení od tradičního řešení, kdy vlastně každý úkon bylo nutno zobrazit.

Při požadavku grafického výstupu je možno zaměřit se na účel řešeného příkladu. Je vhodné přepočítat výstupní údaje dle původně zadaných jednotek. Taktéž vlastní zobrazení řešení je možno dle požadavku zobrazit ve vhodném měřítku.

#### **1.1.4 Výhody a nevýhody použití počítače.**

**Nevýhody** použití počítače při řešení geometrických úloh.

a) znalosti potřebné k využití počítače ( nejde o programování )

- ◆ ovládání počítače
- ◆ znalost grafického systému (grafického editoru)
- ◆ znalost základních procedur grafických knihoven
- ◆ znalost (alespoň základních) principů počítačové grafiky (zobrazení entit, transformační řetězce jak pro transformace v rovině a prostoru, tak pro zobrazení 3D na 2D)
- ◆ řešení úlohy z hlediska algoritmů počítače.

b) malé cvičení prostorové představivosti.

**Výhody:**

- ◆ zvýšení atraktivnosti výuky
- ◆ zapojení širší oblasti pro řešení geometrických problémů (numerické metody apod.) ⇒ širší pohled na řešení úlohy
- ◆ univerzální (komplexnější) výstup
- ◆ snadný přenos dat na jiná média a výstupní zařízení
- ◆ snadná aktualizace a archivace výsledků.

**Kontrolní otázky 1.1.**

1. Úloha a význam počítačové grafiky v různých oborech.
2. Uveďte výhody, nevýhody použití počítačů pro řešení geometrických úkolů.
3. Uveďte rozdíl při řešení geometrických úkolů tradičním způsobem - bez použití počítačů a s použitím výpočetní techniky.
4. Co je pseudografika, semigrafika ?

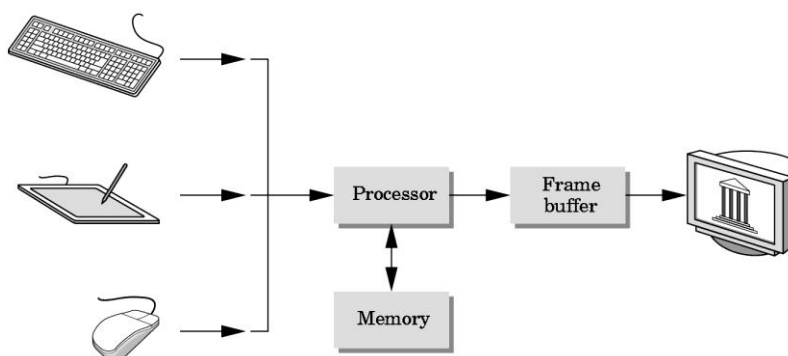
**1. 2. Technické prostředky**

**Cíl** Po prostudování této kapitoly budete znát

- sestavy interakčního grafického systému
- skladbu a základní technické vstupní a výstupní zařízení počítačové grafiky

**Výklad**

Následující obrázek 1.9 je schématem standardní sestavy interakčního grafického systému. Je zde počítač typu PC, který je vybaven grafickým procesorem – grafickou kartou,



Obr. 1.9



kteřá umožňuje rastrovou obrazovku adresovat na jednotlivé pixely. Popis základních typů grafických karet je v odstavci 1.3.1. Mezi standardní vybavení nemusí patřit plotr ani grafické vstupní zařízení, jako je scanner. Plotr může být úspěšně nahrazen grafickou tiskárnou. Lepší (finální) grafický výstup lze pořídit na pracovišti, které takové zařízení vlastní.

Popis jednotlivých částí schématu následuje.

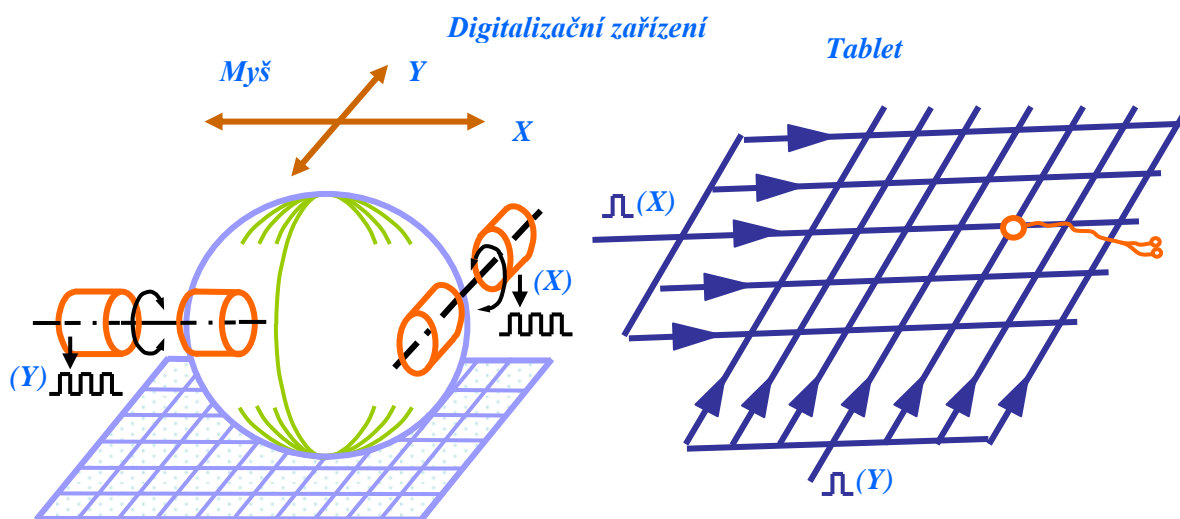
### 1. 2. 1. Vstupní a výstupní zařízení

Vstupní zařízení dělíme do několika tříd dle funkčnosti zařízení.

- lokátor - poskytuje identifikaci polohy v rovině i v prostoru;
- valuátor - poskytuje číselnou hodnotu;
- zařízení pro výběr grafického prvku - je schopné identifikovat grafický prvek;
- klávesnice - pro vstup znaků.

Vstupní zařízení jako např. **tablet, digitizér, pisátko, myš, světelné pero, scanner, kamera.**

Na obrázku 1.10 je schéma snímání polohu myši a tabletu.



Obr. 1.10

Výstupní zařízení .

- a) dočasný záznam - **obrazovka** - bodová - grafická .
- b) trvalý záznam - **plotry** - kreslicí stoly  
- **tiskárny** .

Výhody plotrů - přesnost, velikost kresby, možnost volby různých druhů papírů.

Nevýhody plotrů - pomalost, velké a drahé zařízení, pouze originál (bez kopie).

Výhody tiskáren - rychlost „kresby“, levnější, kopie.

Nevýhody tiskáren - možnost volby pouze určité velikosti a kvality papíru.

## 1. 2. 2. Monitory a displeje - fyzikální principy

### Katodová obrazovka (CRT monitor)

Elektronová tryska (v případě barevného obrazu jsou tři pro 3 základní barvy RGB, jejichž vzájemným mícháním se sestavuje barevný obraz) je zdrojem elektronového paprsku vyslaného ke stínítku obrazovky. Paprsek je pak vychylován v horizontálním i vertikálním směru (putuje zleva doprava, pak návrat zpět, posun o jeden řádek níže a celý proces se znovu opakuje). Takto se vykreslí na obrazovce celý obraz, poté se paprsek vrátí do výchozí polohy a celý děj se opakuje s novým obrazem. Působením vnějších vlivů je paprsek rozostřen. Proto před dopadem na luminofory prochází paprsek ještě maskou, která jeho tok usměrní. V případě barevného obrazu musí maska přesně vedle sebe usměrnit 3 paprsky současně. Jejich kombinací se vytváří jeden barevný bod.

V následující části je uvedena možná klasifikace obrazovek. Předem uvádím, že dle dostupných materiálů neexistují žádná přesvědčivá objektivní a všeobecná kritéria upřednostňující jednu technologii proti druhé.

Podle způsobu usměřňování barevných paprsků lze katodové obrazovky rozdělit takto:

- Katodová obrazovka s bodovou maskou (Delta uspořádání). Jedná se o nejstarší typ barevné obrazovky se stínící maskou. Elektronové trysky jsou umístěny do vrcholů rovnostranného trojúhelníku. Jednou z hlavních nevýhod je deformace obrazu v okrajích obrazovky, která se u invarových obrazovek eliminuje klenutím obrazovky.
- Katodová obrazovka s aperturní mřížkou (Trinitron)  
Tyto obrazovky patří mezi ploché (Trinitron firmy Sony, Diamondtron firmy Mitsubishi). Masky jsou vytvořeny ze svislých drátků, zpevněných dvěma příčnými drátky. Výhodami jsou vyšší jas, kontrast bez ztráty ostrosti, deformace obrazových bodů je eliminována snížením vzdálenosti jednotlivých drátků. Velkou nevýhodou je naopak měkkost masky, která tak snadno podléhá deformacím (např. magnetické pole reproduktorů dokáže masku trvale poškodit).
- Katodová obrazovka se šterbinovou maskou (CromaClear)  
Hybridní obrazovka firmy NEC. Jedná se o kombinaci trinitronového obrazu s osvědčenou trvanlivostí invarové obrazovky s bodovou maskou. Používá se maska s elipticky tvarovanými luminofory. Díky nim by se mělo zvýšit vodorovné a svislé rozlišení, což je výhodné zvláště pro aplikace pracující s textem.

### 1.2.3. Ploché displeje LCD (Liquid Crystal Displays) - displeje z tekutých krystalů

- pasivní TN displeje (Twisted Nematic)

První ploché displeje. Tekuté krystaly jsou umístěny mezi dvěma skleněnými deskami, na nichž jsou elektrody a polarizační filtr. Jestliže je na elektrodách napětí, molekulární tyčinky (Twisted Nematic) se díky svým optoelektrickým vlastnostem jinak prostorově orientují a polarizují světlo. Barevný účinek určuje výška LC vrstvy. Protože světlo na své cestě tekutými krystaly ztrácí na intenzitě a trpí LCD slabým kontrastem.

- pasivní STN displeje (Super Twisted Nematic)

Úhel natáčení tyčinek se z původních  $90^\circ$  zvětší na  $240^\circ$ . Znatelně se zlepšil kontrast, ale zvětšil se rovněž barevné zkreslení (dopadající světlo se láme více nebo méně dle různých vlnových délek každé barvy).

- DSTN displeje (Double Super Twisted Nematic)

Dvě vrstvy STN na sobě. Jedna vrstva (aktivní) je elektricky ovlivněna tak, že se tyčinky přestanou natáčet. V druhé (pasivní) se tyčinky stále otáčejí o  $240^\circ$  proti směru hodinových ručiček. Dopadající světlo se v této vrstvě lomí stejně jako u STN panelů, druhá vrstva tuto chybu opět koriguje. Výsledkem je lepší kontrast než u předchozích typů.

- TFT displej (Thin Film Transistor)

LC vrstvy jsou osazeny drobnými tranzistory, přičemž každý z těchto tranzistorů řídí jeden obrazový bod.

Výhody displejů TFT:

- urychlení reakční doby (rychlost překreslení obrazu)
- pohyby na displeji bez stínů
- bez barevných chyb
- menší spotřeba proudu
- menší tloušťka
- velký kontrastní poměr

Nevýhody displejů TFT:

- komplikovaná výroba
- vysoká cena.
- Fero LCD (Feroelektrický displej)

LCD buňky nahrazeny feroelektrickými, které jsou schopny zachovávat obraz až do nového impulsu bez překreslování a zároveň reagují rychleji než LCD buňky. Tato technologie přináší lepší obraz než STN, ale výroba je nákladná a navíc displeje jsou těžké, tzn. pro přenosné počítače nepoužitelné.

#### 1.2.4. Plazmové displeje (PD)

Objevily se v polovině 80- tých let 20. století. V těchto obrazovkách je směs plynů neonu a argonu, kterou elektrické pole přiměje k vyzařování. Barva vzniká přimíšením svítících látek, které se aktivují ultrafialovým zářením plazmy. Nevýhodou je nedostatečná kvalita obrazu (reflexe poškozují kontrast a reprodukci barvy). Výhodami jsou nízké pořizovací náklady a dlouhá životnost. Předpokládá se jejich uplatnění spíše jako televizních než počítačových monitorů.

#### 1.2.5. Plazmatronové (PALC) (Plazma Addressed Liquid Crystal)

(firma Sony). Jedná se o kombinaci techniky LCD a plazmy. Pomocí přesně dávkovaného výboje plynů se zapínají a vypínají tekuté krystaly. Celkový obraz se pak skládá z asi 450 horizontálních plazmových kanálů. Výhodou je jednoduchá stavba pro velké a lehké obrazovky, cenově výhodná výroba, dobrá obrazová kvalita. Předpoklad je, že se nejprve uplatní v televizních přijímačích.

#### 1.2.6. Technická data monitorů

Úhlopříčka - běžné velikosti obrazovek monitorů (15", 17", 19", 21" , 23"), u některých monitorů se ještě udává skutečně viditelná plocha rozlišení - obraz na monitoru se skládá z bodů (pixelů) s určitou roztečí, ty se udávají v horizontálním a vertikálním směru (např. 800x600 - 600 řádků a každý s 800 body)

Optimální velikost rastru pro 15" 800 x 600

pro 17" 1024 x 768

pro 19" 1152 x 870

pro 21" 1280 x 1024; 1600 x 1200

Rozteč bodů (Dotch Pitch) - vzdálenost mezi body (proužky nebo elipsami) stejné barvy.

Čím vyšší rozlišení, tím nižší vzdálenost.

Standart pro obrazovky : s bodovou maskou 0,27 mm a méně.

Trinitron a CromaClear 0,25 mm a méně. Frekvence - řádková (horizontální) [kHz] - udává řádky za sekundu.

Obnovovací (vertikální) [Hz] - udává kolikrát se nakreslí (obnoví) obraz za sekundu.

Obnovovací frekvence od 72 Hz se považuje za neblíkové zobrazení. Při vyšších obnovovacích frekvencích obraz sice neblíká, ale je obětována kvalita obrazu. Aby totiž vedle sebe mohl být zobrazen bílý a černý bod je třeba, aby elektronový paprsek přešel od nulové k plné intenzitě za velmi krátkou dobu. Jestliže nemá dost času, přechod mezi těmito body

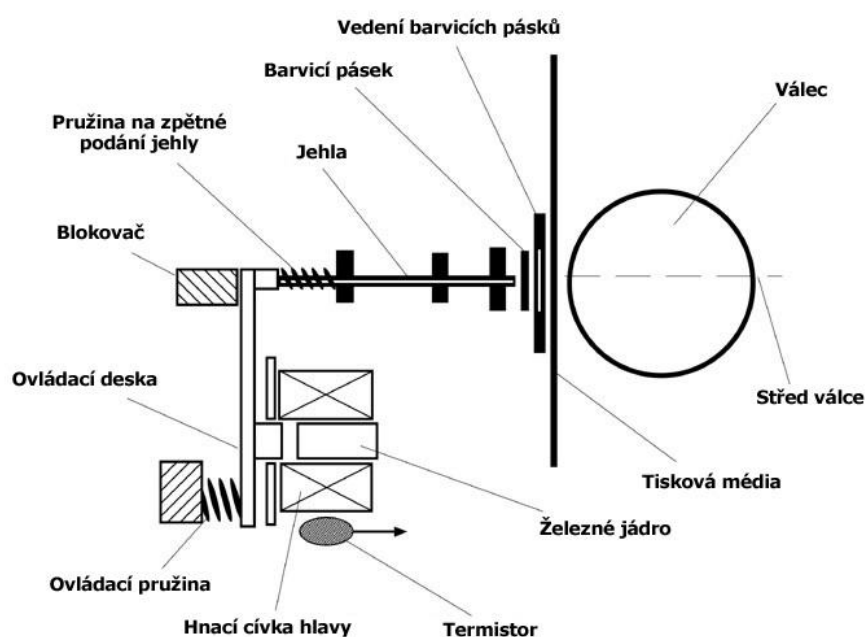
bude šedivý. Řešením pak je zvolit nejnižší obnovovací frekvenci při níž už nepozorujeme blikání - optimální 85 Hz.

### 1.3. Tiskárny

Typy tiskáren

- s typovým kolečkem
- jehličkové
- inkoustové
- termotransferové
- laserové

#### 1.3.1 Jehličkové tiskárny

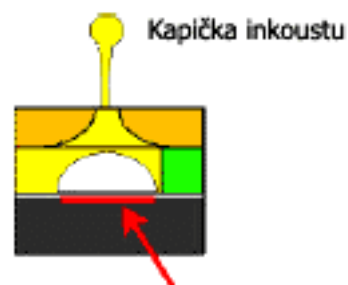


Obr. 1.11 Schéma jehličkové tiskárny

Vlastnosti jehličkových tiskáren:

- 9, 18, 24 jehliček
- barevný tisk (7 barev)
- nízké náklady
- až 400 stran/hod.
- až 6 průklepů
- snížení hlučnosti (nižší hlučnost)
- práce v nepříznivém prostředí (prašné,

vlhké apod.)



expandující inkoust vlivem tepelného ohřevu

Obr. 1.12

Jehličkové tiskárny se používají tam, kde není kladen důraz na kvalitu tisku.

Např..u pokladen, kontrolních výpisů – daňové doklady, výplatní pásky apod.

### 1.3.2 Inkoustové tiskárny

Inkoustové tiskárny se dělí na:

- termální - trysky vybaveny topným odporem
- piezoelektrická - před tryskou je umístěn piezoelektrický rezonátor.

Trysky obsahují topný odpor, který ohřeje inkoust a ten je expandován.

#### Piezoelektrická technologie inkoustových tiskáren

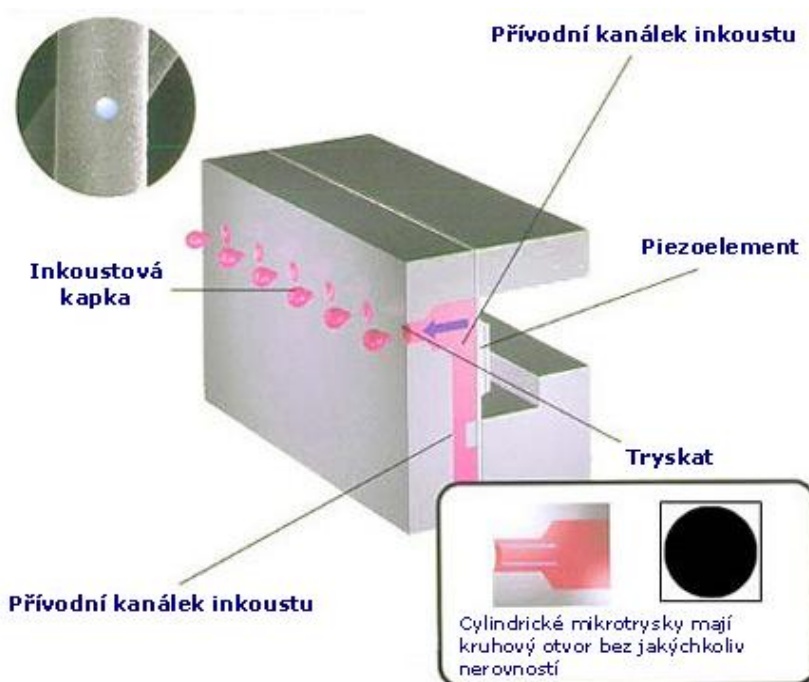


Schéma piezoelektrické technologie

Obr. 1.13

### 1.3.3 Laserové tiskárny

Na následujících obrázcích je znázorněn postup tisku na laserové tiskárně.

Laserovým paprskem je sejmut bodově řádek předlohy.

Tento sejmutý řádek je magneticky přenesen na bubnový válec, ze kterého je „obraz“ přenesen na papír.

Přenesení není jednoduché a pro stručný popis v této kapitole se spokojíme s následujícími schémata.

**Výhody laserových tiskáren:** kvalitní tisk, nehlukný provoz, oboustránkový tisk.

**Nevýhody laserových tiskáren:** velké pořizovací náklady, nákladný provoz

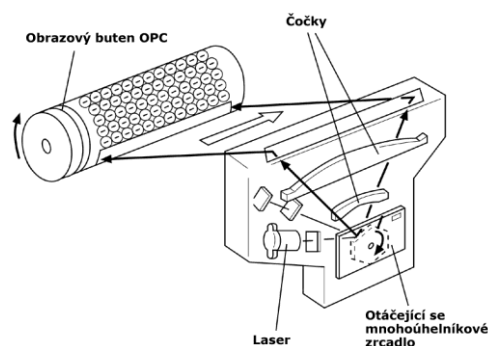


Schéma laserové tiskárny

Obr. 1.14

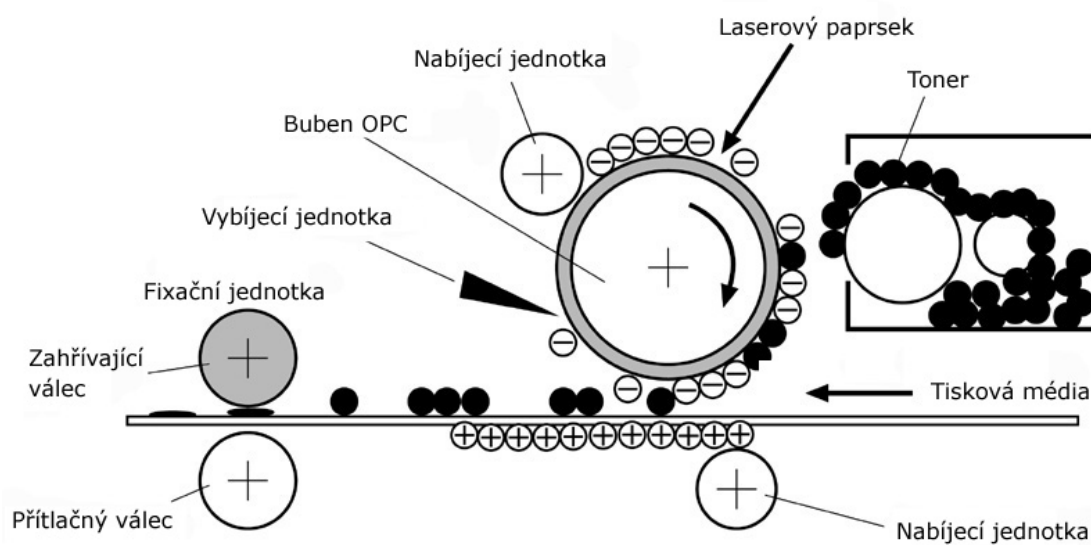


Schéma tisku laserové tiskárny

Obr. 1.15



## Shrnutí

Dříve - tiskárny s typovým kolečkem, psací stroje.

1964 první jehličková tiskárna

Nyní - jehličkové tiskárny (také barevné)

termotransferové tiskárny

inkoustové tiskárny

laserové tiskárny



## Kontrolní otázky 1.2 - 1.3

Charakterizujte vstupní a výstupní zařízení počítačové grafiky.

1. Uveďte rozdělení vstupních zařízení z hlediska aplikace.
2. Charakterizujte jednotlivá vstupní zařízení z hlediska vstupních dat.
3. Charakterizujte výstupní zařízení rastrové, vektorové (výhody, nevýhody).

## 1.4. Počítačová grafika pro osobní počítače řady PC.



**Cíl** Po prostudování tohoto odstavce budete znát

- princip práce v počítačové grafice ve srovnání s ostatními negrafickými pracemi
- rozdíl mezi prací na počítači řady PC a grafickou stanicí.



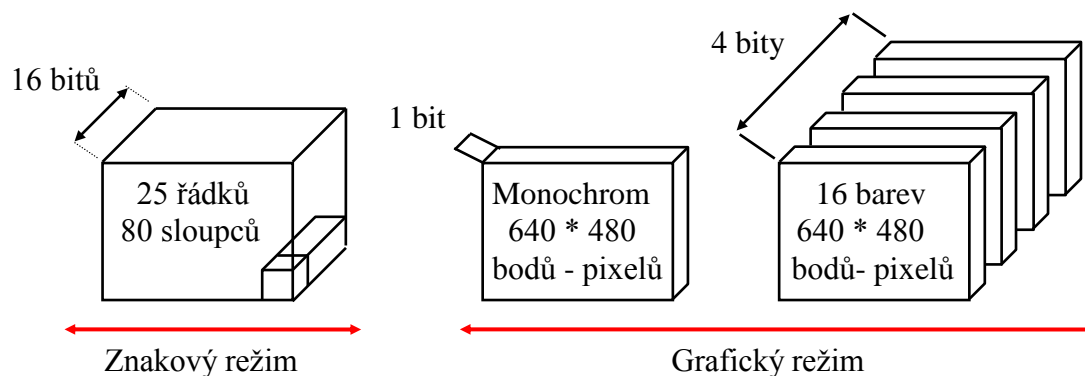
**Výklad**

Rastrová grafika je součástí osobních počítačů. Obrazová paměť je částí operační paměti. Pracuje obvykle ve dvou režimech - **textovém (znakovém) a grafickém**.

V obou je obraz generován do rastru stejným způsobem. U znakového režimu je ke generování alfanumerických znaků potřebný generátor znaků, který je obsažen v grafické kartě displeje. Pokud není přítomen takovýto generátor, je nutno v obrazové paměti mít pro každý znak nejen jeho kód, ale i jeho atributy (barva popředí, pozadí, blikání a pod.)

Osobní počítače typu PC počítají s tím, že je možno připojit libovolný terminál - barevný nebo černobílý. Je tedy nutno doplnit počítač o obrazovkový adaptér. Za tímto účelem existuje celá řada adaptérů.

Znárodnění obsahu rastrové paměti počítače.



Obr. 1.16.

### 1.4.1. Grafické karty

Operační systém MS DOS obsahuje část, která je označena BIOS (Basic Input/Output System), který poskytuje řadu služeb pro styk s technickými prostředky. To znamená i pro styk s obrazovým monitorem. Umožňuje nastavovat režim zobrazení, definovat kurzor, nastavit a číst polohu kurzoru, číst polohu světelného pera, nastavit paletu barev, rolovat okno v textovém režimu, číst a psát znak pod kurzorem, definovat množinu znaků pro generátor znaků a pod. Výhodou používání služeb operačního systému při tvorbě programů pro grafiku



je jeho přenositelnost mezi různými podsystémy na počítačích PC a PS/2. Je to však na úkor rychlosti. Není zde plně využíváno výhod grafických karet. Jestliže ale trváme na plném využívání operačního systému, musíme se s ním seznámit důkladně.

**MDA - (Monochrome Display Adapter)** adaptér pro monochromatický monitor, který zobrazí na 25 řádcích 80 znaků alfanumerického textu.

**CGA - (Color Graphics Adapter)** - monochromatický monitor, který zobrazí 25 řádků po 80-ti znacích. Podporuje ale i práci v alfanumerickém režimu (25 řádků, 80 sloupců) a to ve dvou grafických režimech. 640 bodů - 2 barvy nebo 320 bodů - 4 barvy. Nevýhodou adaptéru CGA je poměrně špatná kvalita alfanumerických znaků (díky malé rozlišitelnosti).

**EGA - (Enhanced Graphics Adapter)** uvedený na trh v roce 1985 poskytuje grafický režim pro barevný i monochromatický displej. Je schopen generovat signály pro barevné zobrazení 16 barev z 256 barev ve znakovém režimu nebo 640 \* 350 bodů v grafickém režimu. Navíc poskytuje grafický režim i pro monochromatický displej.

**HGC - (Hercules Graphics Card)** zastupuje další skupinu grafických karet firmy Hercules Computer Technology, která byla uvedena na trh v roce 1982. Umožňuje na monochromatický monitor zobrazovat dobře čitelný text i grafiku. Adaptér HGC+ (1987) umožňuje práci s 16-ti barvami z 64 barev. Všechny adaptéry Hercules pracují s rozlišením 720 \* 348 bodů.

**MCGA - (Multi Color Graphics Array)** je adaptér podobný adaptéru CGA, ovšem má lepší rozlišitelnost (640 \* 480 bodů) a má větší barevnost (256 barev z  $2^{18}$ ). Generuje analogové signály pro ovládání barev, což umožňuje širokou škálu barev.

**VGA - (Video Graphics Array)** má funkci adaptéru EGA s tím, že větší rozlišení (720 x 400 v textovém režimu, 640 x 480 v grafickém režimu a 256 barev z  $2^{18}$ ).

### 1.4.2. Grafické stanice

Základem grafické stanice je počítač, který je výkonnější od standardního PC. Grafické stanice se liší od PC jak HW vybavením, tak SW vybavením, které je zaměřeno na práci v počítačové grafice.

Výrazným rozdílem grafických stanic od využívání počítačů řady PC je:

- velká operační paměť ;
- víceprocesorový stroj - paralelní běh několika úloh;

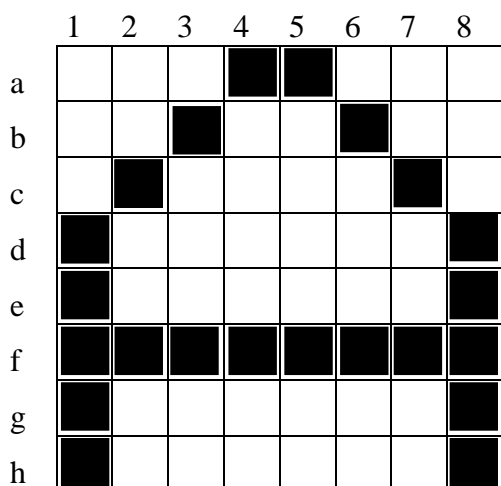
- větší vybavenost vstupními i výstupními perifériemi;
- víceobrazkový režim - jeden znakový displej, (minimálně) jeden grafický displej;
- komunikace na obrazovce s využitím vícenásobných oken;
- programová podpora pro určitý obor a zaměření stanice apod.
- rastrová grafika minimálně s 1 miliónem obrazových bodů.

Grafická stanice je po hardwarové stránce vybavena zařízením, ke kterému je výrobcem dodáno příslušné programové vybavení. V případě, že tomu tak není, je výrobcem dodán přesný popis zařízení a je na programátorovi, aby požadovaný software naprogramoval.

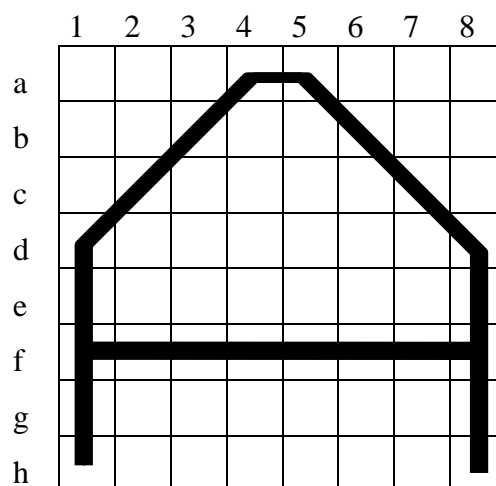
### 1.4.2. Generování znaků a grafických značek

a) **Rastrový obraz** znaku písmene A v rastru  $8 * 8$  je na obrázku číslo 1.17.

Takto vytvořený obraz znaku můžeme zvětšovat s modulem 2 a můžeme s ním točit v násobcích úhlu  $90^\circ$ . Otáčení o jiný úhel je problematické.



Obr. 1.17.



Obr. 1.18.

Existují generátory znaků, kde je možno vytvořit vlastní fonty pro různé rastry a různá programová vybavení.

b) **Vektorový obraz** (viz. obr. 1.18) stejného znaku potom bude vytvořen jako řada úseček.

Úsečky mohou být kresleny v pořadí:

1. line(x(h1),y(h1),x(d1),y(d1));
2. line(x(d1),y(d1),x(a4),y(a4));
3. line(x(a4),y(a4),x(a5),y(a5));
4. line(x(a5),y(a5),x(d8),y(d8));
5. line(x(d8),y(d8),x(h8),y(h8));
6. line(x(h8),y(h8),x(f8),y(f8)); {Překreslování úsečky !}
7. line(x(f8),y(f8),x(f1),y(f1));

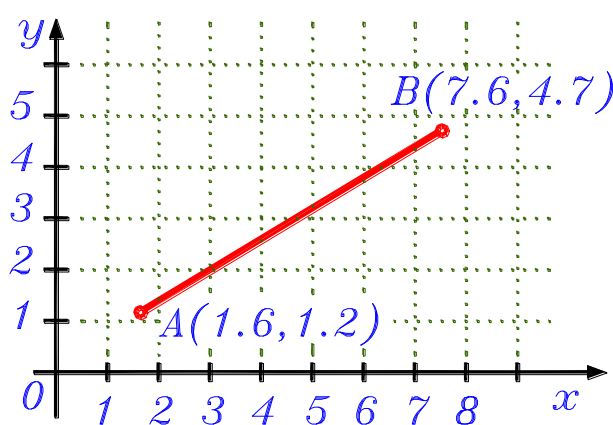
Jde vlastně o seznam vektorů, které budou kresleny vždy, kdy bude požadováno "vykreslení" znaku "A".

### 1.4.3. Generování úseček v rastru

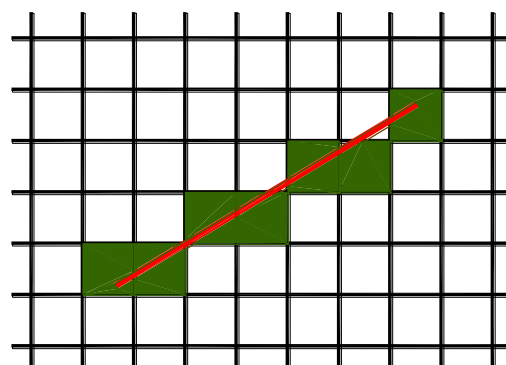
Úsečka je **základním** prvkem grafického obrazu. V rastru je ovšem třeba rozhodnout, kterými body (pixely) bude obraz úsečky aproximován. Podprogram pro aproximaci v pravoúhlém rastru musí být velmi racionální, protože bude velmi často volán. Takovýchto algoritmů existuje celá řada.

Úsečka a její obraz v rastrovém zařízení: Algoritmy vycházejí z rovnice přímky ve tvaru

$$y = m * x + b . \quad (1.1)$$



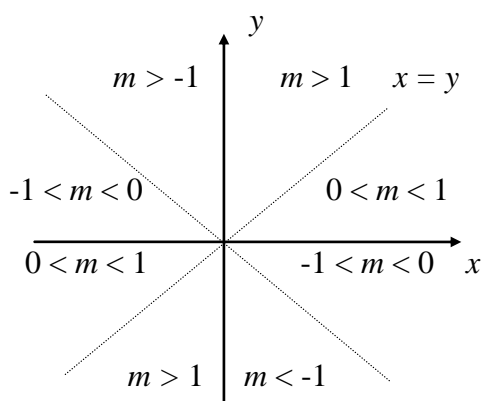
Obr. 1.19.



Obr. 1.20.

Na obrázku 1.19 je znázorněn reálný obraz úsečky zadané počátečním a koncovým bodem v reálných souřadnicích. Na obrázku 1.20 je obraz této úsečky v rastrovém zobrazení.

Nejčastější zadání úsečky je dvěma koncovými body  $(x_1, y_1)$  a  $(x_2, y_2)$ .



Obr. 1.21.

Pro toto zadání potom platí:  $y = m * x + b$ , kde  $m$  je směrnice a  $b$  je posun na ose  $y$ .

Směrnice úsečky

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}, \quad (1.2)$$

posun (úsek) na ose  $y$   $b = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}$ . (1.3)

Základem pro většinu algoritmů na kresbu úsečky je rovnice  $\Delta y = m * \Delta x$ . (1.4)

Na obrázku 1.21 jsou vyznačeny hodnoty směrnice  $m$  pro různé sklony úsečky. Mnohé algoritmy využívají právě tuto hodnotu proto, aby algoritmus, který byl vytvořen pro  $m > 1$  byl vhodnou úpravou (na příklad záměnou  $x$  za  $y$ ) použit.

#### 1.4.4. Algoritmus DDA

Přírůstkový algoritmus (DDA - **D**igital **D**ifferential **A**nalyzer) je algoritmus, který je založený na postupném přičítání přírůstků k oběma souřadnicím při výpočtu pozic jednotlivých pixelů zobrazujících úsečku. Jako ostatní, i tento algoritmus vychází z rovnice (1.2). Pro  $m < 1$  bude měnit souřadnice  $x$  o jedničku a určovat následující souřadnici  $y$  dle vztahu

$$y_{i+1} = y_i + m. \quad (1.5)$$

Celočíselná hodnota  $i$  se od hodnoty 1 zvyšuje po jedné až do dosažení koncového bodu dané úsečky. Osu  $x$  nazýváme pro tento případ **řídící osou**. Protože hodnota  $m$  je reálná, souřadnice  $y$  se zaokrouhluje na nejbližší celé číslo.

Pro úsečky, kde  $m > 1$ , se tento postup provádí obdobně s tím, že jsou zaměněny souřadnice  $x$  a  $y$ . Po jedné měníme souřadnici  $y$  a vypočítáváme hodnotu  $x$  dle vztahu

$$x_{i+1} = x_i + \frac{1}{m}. \quad (1.6)$$

Algoritmus je realizován následující procedurou, kde vstupními parametry jsou počáteční a koncové body  $(x_1, y_1)$ ,  $(x_2, y_2)$  úsečky. K počátečním hodnotám  $x_1$  a  $y_1$  jsou přičítány konstantní přírůstky až po dosažení koncového bodu. V následujícím výpisu textu procedury chybí, z důvodu přehlednosti, test na jednobodovou úsečku a svislost úsečky.

Procedura **Line\_DDA** vykreslí úsečku v cyklu  $k$ , kde **pocet\_bodu** je počet kroků cyklu. **Pocet\_bodu** je větší číslo v rozdílu příslušných souřadnic koncového a počátečního bodu úsečky.

```
procedure Line_DDA (x1, y1, x2, y2 : integer);
```

```
    var dx, dy, pocet_bodu, k          : integer;
```

```
        x_prirustek, y_prirustek, x, y : real;
```

```
begin
```

```
    dx := x2 - x1;                               {rozdíl na ose x }
```

```
    dy := y2 - y1;                               {rozdíl na ose y }
```

```
    if abs(dx) > abs(dy)                       {určení počtu kroků pro }
```

```

then pocet_bodu := abs(dx)           { kresbu úsečky }
else pocet_bodu := abs(dy);
x_pristek := dx / pocet_bodu;        { výpočet přírůsku pro x }
y_pristek := dy / pocet_bodu;        { výpočet přírůsku pro y }
x := x1; y := y1;
Put_pixel (round(x), round(y));      { kresba počátečního bodu }
for k := 1 to pocet_bodu do       { kresba "vnitřních" bodů úsečky }
  begin
    x := x + x_pristek;
    y := y + y_pristek;
    Put_pixel (round(x), round(y))
  end                                { cyklu k }
end;                                { Line_DDA }

```

### 1.4.5. Bresenhamův algoritmus

Zajímavější i známější je tzv. algoritmus **Bresenhamův**. Předpokládejme, že úsečka je dána koncovými body  $P_1 (x_1, y_1)$ ,  $P_2 (x_2, y_2)$  a o směrnici  $m$  (viz (2.2))

$$m = \frac{\Delta y}{\Delta x}, \quad \text{kde}$$

$$\Delta x = |x_1 - x_2|$$

$$\Delta y = |y_1 - y_2|$$
(1.7)

Předpokládejme, že platí  $|m| < 1$ .

Jestliže úsečka prochází bodem  $(x_i, y_i)$  rastru, pak nás zajímá, jestli po přírůsku na ose  $x$  o krok rastru do  $x_{i+1} = x_i + 1$  bude nová souřadnice  $y_{i+1} = y_i$  nebo  $y_{i+1} = y_i + 1$ .

Z obrázku a rovnice přímky dostaneme:  $y = m(x_i + 1) + b$  (1.8)

Zavedeme:

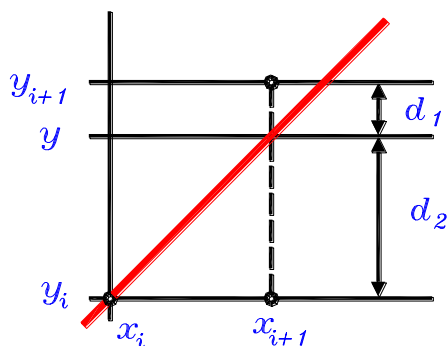
$$d_1 = y_{i+1} - y = y_i + 1 - m(x_i + 1) - b$$
(1.9)

$$d_2 = y - y_i = m(x_i + 1) + b - y_i$$
(1.10)

Z (1.9, 1.10) =>

$$\Delta d = d_1 - d_2 = 2m(x_i + 1) - 2y_i + 2b - 1$$
(1.11)

Hodnota  $\Delta d$  rozhoduje, který ze dvou pixelů leží blíže skutečné hodnotě úsečky a bude se kreslit.



Obr. 1.22

Zavedeme pomocný parametr

$$\begin{aligned} p_i &= \Delta x(d_1 - d_2) = 2\Delta y + x_i - 2\Delta x + y_i + c, \text{ kde} \\ c &= 2\Delta y + \Delta x(2b - 1). \quad \leftarrow \text{konstanta} \end{aligned} \quad (1.12)$$

Za  $(d_1 - d_2)$  dosazeno (1.11), za  $m$  dosazeno (1.7).

Jestliže  $p_i < 0$  at'  $y_{i+1} = y_i$ ,  
naopak při  $p_i \geq 0$  at'  $y_{i+1} = y_i + 1$ .

Na počítači budeme realizovat rekurzivní algoritmus, který je vyjádřen vztahem

$$p_{i+1} = f(p_i)$$

Platí:  $p_{i+1} = 2\Delta y x_i - 2\Delta x y_{i+1} + c$

$$p_{i+1} - p_i = 2\Delta y (x_{i+1} - x_i) - 2\Delta x (y_{i+1} - y_i)$$

$$x_{i+1} = x_i + 1 \quad \Rightarrow$$

$$p_{i+1} = p_i + 2 * \Delta y - 2 * \Delta x (y_{i+1} - y_i).$$

$$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$$

Hodnota poslední závorky je 0 nebo 1 podle předcházející hodnoty  $p_i$ .

To znamená, že  $p_{i+1} = p_i + \underbrace{2\Delta y - 2\Delta x}_{\text{konst2}}$  pro  $p_i \geq 0$

a

$$p_{i+1} = p_i + \underbrace{2\Delta y}_{\text{konst1}} \quad \text{pro } p_i < 0.$$

Pro vykreslení úsečky se směrnici  $|m| > 1$ , použijeme tentýž algoritmus s tím, že zaměníme souřadnice  $x$  a  $y$ .

**Program:**

**procedure** Line\_Bres ( $x1, y1, x2, y2$  : **integer**);

**var** dx, dy, konst1, konst2 : **integer**;

x, y, x\_koncové, predikce : **integer**;

**begin**

**if**  $x1 > x2$  {určení počátečního a koncového bodu}

**then begin**  $x := x2; y := y2; x\_koncove := x1$  **end**

**else begin**  $x := x1; y := y1; x\_koncove := x2$  **end;**

$dx := x\_koncove - x; \quad dy := \text{abs}(y1 - y2);$

$konst1 := 2 * dy;$

$konst2 := 2 * (dy - dx);$

$predikce := 2 * dy - dx;$

Put\_pixel ( $x, y$ ); {kresba počátečního bodu úsečky }

```

while x < x_koncove do           {cyklus pro kresbu vnitřních bodů }
  begin                             {úsečky, včetně koncového }
    x := x + 1;
    if predikce < 0                  {souřadnice y zůstává }
      then predikce := predikce + konst1
    else                             {souřadnice y se zvyšuje }
      begin y := y + 1; predikce:= predikce + konst2 end;
    Put_pixel(x, y)
  end                                 {while x < x_koncové}
end;                                 {Line_Bresenham}

```



### Kontrolní otázky 1.4.

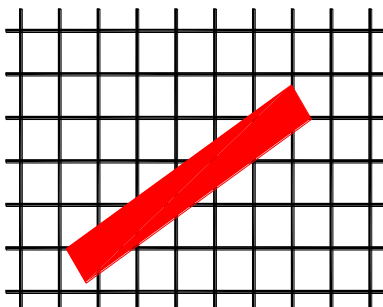
1. Jak jsou generovány úsečky (křivky) v rastrové grafice.
2. Vysvětlete princip DDA algoritmu pro generování úsečky.
3. Vysvětlete princip Bresenhamova algoritmu pro generování úsečky.

## 1.5. Vyhlazování čar

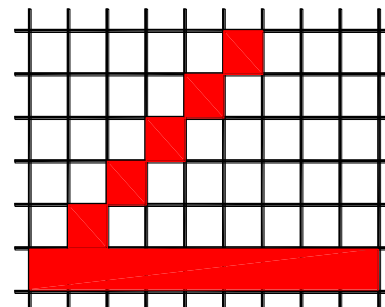


### Výklad

Ideální zobrazení čáry je na obrázku 1.23. Tohoto zobrazení však v rastrovém systému nelze dosáhnout. Je tedy používán pro rychlé zobrazení DDA algoritmus. Se vzrůstajícím přírůstkem ve směru osy x je vysvícován pouze jeden pixel plnou intenzitou sytosti barvy.

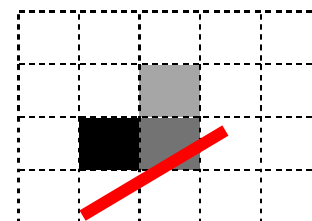


Obr. 1.23.



Obr. 1.24.

Při použití Bresenhamova algoritmu je možné vysvítit dva pixely nad sebou tak, že součet intenzit barvy těchto pixelů je roven plné intenzitě vysvícované barvy. Intenzita vysvícení jednotlivých pixelů je dána poměrem  $d_1$  a  $d_2$ . Viz. obr. 1.25. Hodnoty  $d_1$  a  $d_2$  jsou reciproké hodnoty intenzity vysvícení jednotlivých pixelů.



Obr. 1.25

## 1.5. Atributy čar



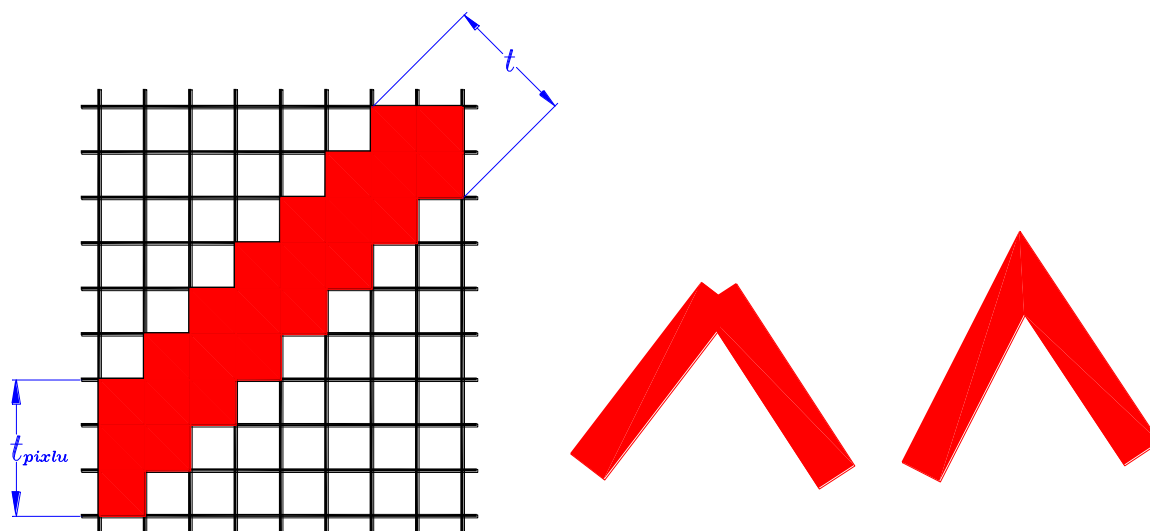
**Cíl** Po prostudování tohoto odstavce budete znát atributy čar, jejich význam a tvorba

### 1.5.1. Tloušťka čar



#### Výklad

Realizace tloušťky čar závisí na výstupním zařízení. Silné čáry na monitoru jsou realizovány vysvícením řady pixelů. Z následujícího obrázku je zřejmé, že tloušťka čar na terminálu je závislá na úhlu, který svírá kreslená úsečka s vodorovnou resp. svislou osou. Jednodušší realizace tloušťky čar na monitoru je možno vysvítit více pixelů nad resp. vedle



Obr. 1.26

sebe. Počet současně vysvícených pixelů je závislý na velikosti pixelu, úhlu kreslené úsečky a tloušťce čáry.

Výpočet pixelů v závislosti na tloušťce čáry na monitor lze vyjádřit vzorcem (pro první kvadrant) bude platit :

$$t_{pixl} = t \cdot \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{|\Delta x|}$$

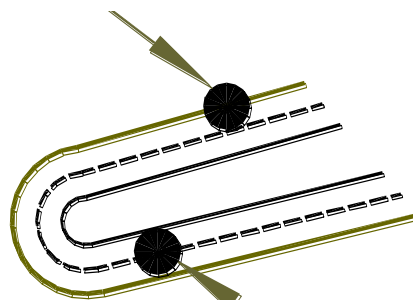
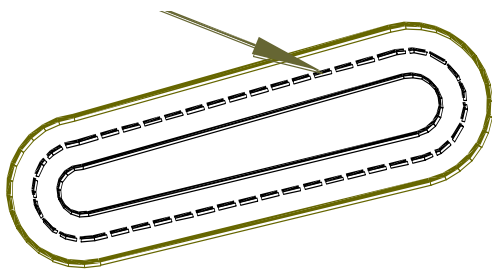
Některé grafické systémy realizují tloušťku čar vykreslením obrysu čáry a vyplněním vnitřních bodů - pixelů.

Realizace tloušťky čar na plotru je realizována tloušťkou kreslicího pera. Při vyplňování oblastí je pohyb pera rovnoběžně se souřadnicovou osou (nejčastěji s osou  $x$ ). Na obrázcích 1.27 a 1.28 je znázorněno vyplňování oblastí při respektování a nerespektování tloušťky pera.



ekvidistantní pohyb pera

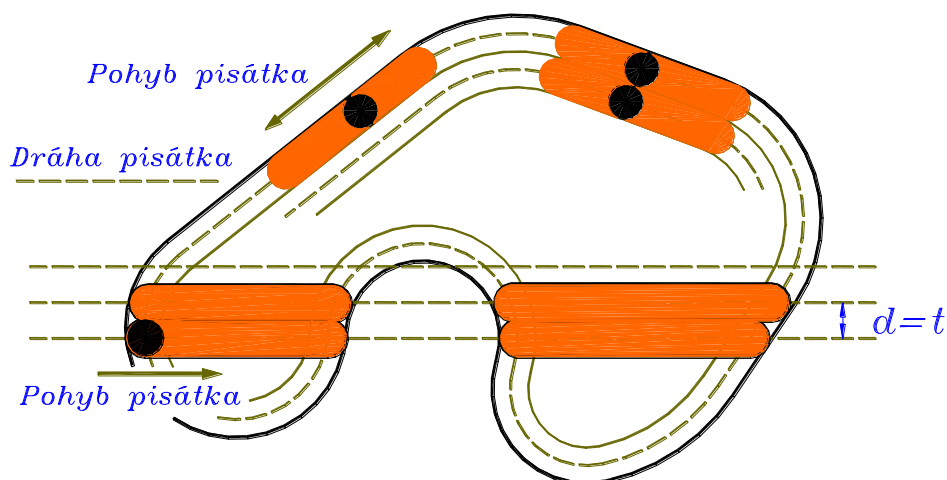
nerespektuje tloušťku pera



respektuje tloušťku pera

Obr. 1.27

Obr. 1.28



Obr. 1.29

Na obrázku 1.29 je znázorněn možný způsob vyplňování obrazců na plotru. Realizace tloušťky čar na bodových tiskárnách a terminálech je většinou realizována semínkovým vyplňováním. Vyplňování obrazců je obsahem následující kapitoly 2.

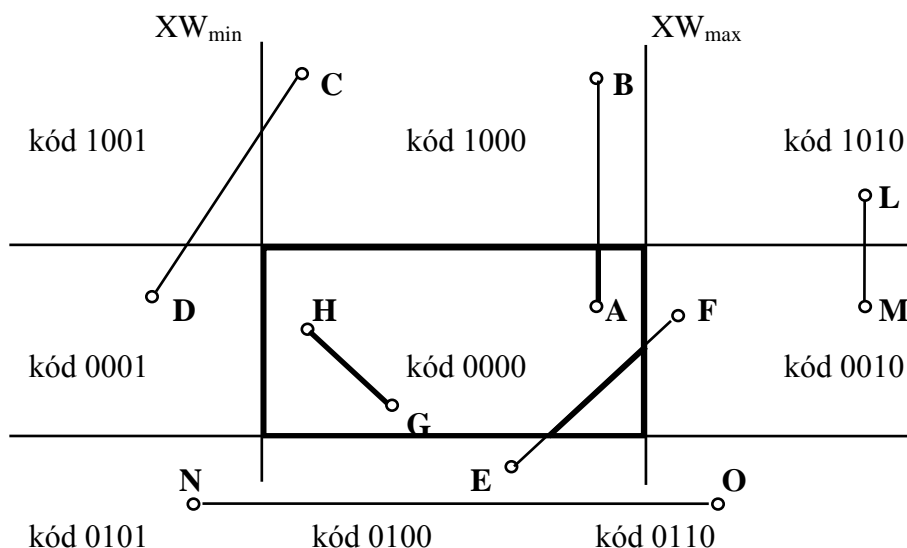
## 1.6. Algoritmus ořezávání

Při ořezávání v počítačové grafice je nutno rozlišovat zda pracujeme v normalizovaných grafických soustavách a nebo v nenormalizovaném - tedy vlastním - prostředí. V normalizovaném prostředí ( na př. GKS ) je k dispozici funkce systému, (GSCLIP-SET CLIPPING ) která při požadovaném ořezání u každé entity určí kód. Tímto kódem - 4 bity - jsou "označeny" koncové body úsečky. Tím se usnadní výpočet průsečíků s hranicí okna.

Na obrázku 1.30 následujícím je stručně uveden **Cohen - Sutherlandův** algoritmus pro ořezávání. Jsou zde označeny kódy jednotlivých oblastí ořezávajícího obdélníka. Úsečky,

kteří mají v kódu na stejné pozici 1, leží mimo ořezávající obdélník a není je tedy třeba brát v úvahu. Podobně úsečky, které mají u obou konců nulový kód leží celé ve výřezu.

Zbývá tedy vyšetřit úsečky : **AB** a **EF**. Pro "vyšetření" úseček existuje více metod. Tyto metody jsou stejné u normalizovaných i nenormalizovaných počítačových grafik. Jde vždy o co nejefektivnější zjištění průsečíků úseček s hranicí okna. Algoritmy pro ořezávání kružnic a kruhových oblouků jsou složitější. Nelze zde využít lineární interpolace, jako u úseček. Tím jsou tyto algoritmy výrazně pomalejší.



Obr. 1.30



### Kontrolní otázky 1.

1. Úloha a význam počítačové grafiky. Význam, výhody, nevýhody. Porovnejte rozdíl způsobu řešení geometrických úloh tradičním způsobem a s použitím výpočetní techniky. Nároky na hardware a software. Co je pseudografika, semigrafika, obchodní grafika?
2. Vstupní a výstupní zařízení počítačové grafiky. Charakterizujte základní kritéria jednotlivých zařízení.
3. Charakteristika počítačové grafiky pro počítače řady PC. 4. Generování úseček (křivek) v rastrové grafice. DDA algoritmus pro generování úsečky. Bresenhamův algoritmus pro generování úsečky.
5. Princip ořezávání obrazců (zoom).