

Przetłumaczono na język: [czeski](#) ▼

Pokaż oryginał

Opcje ▼



czeski ▼

Technologia Tłumacz

Slezská univerzita



Katedra technologie

Ústav aplikované informatiky

MAGISTERSKÁ PRÁCE

Téma:

Emulace tiskárny a plotru v grafickém režimu

Provádí: Vedoucí:

Ryszard Czekaj, prof. dr hab. v . Dariusz Badura

Paweł Szymik konzultant:

mgr Ireneusz Gocinak

BOROVICE 1996

Úvod

Kromě tisku jednoduchých textů vám moderní tiskárny umožňují používat různé škálovatelné fonty a dokonce umísťovat na stránku grafické obrázky. Plotry, stejně jako tiskárny, jsou řízeny speciálními sekvencemi, které lze v mnoha případech považovat za jazyky pro řízení tisku. Proud řídicích kódů spolu s daty k tisku lze odeslat na tiskárnu nebo uložit do souboru na disk počítače. Takový soubor obsahující data pro tiskárnu nebo plotr se nazývá tiskový soubor. Tento soubor lze později odeslat do zařízení připojeného k jiné počítačové stanici. Tiskový soubor obsahuje řídicí sekvence, specifické pouze pro konkrétní typ zařízení, které jej později správně interpretuje.

Cílem této práce je vyvinout počítačový program emulující provoz vybrané tiskárny a plotru, který umožní grafickou prezentaci tiskového souboru na obrazovce počítače.

Při dosahování tohoto cíle byly učiněny následující předpoklady:

- existuje tiskový soubor vybraných tiskáren a plotrů v odpovídajícím formátu pro ně
- je znám model tiskárny nebo plotru, pro který byl tiskový soubor vytvořen
- tiskový soubor může obsahovat jednoduchá textová data a grafický obrázek
- vyvinutý program bude prezentovat obrázek obsažený v souboru co nejvěrněji
- program umožní tisk dat na jiné tiskárně, než pro kterou byl tiskový soubor připraven
 - k sestavení programu se doporučuje použít grafické prostředí Windows 3.xx a balík programovacích nástrojů Borland C++ .

Technické aspekty provozu tiskárny a plotru

1. Klasifikace tiskáren

1.1. Jehličkové tiskárny (nárazové tiskárny)

V jehličkové tiskárně je obraz vytvářen tiskovou hlavou vybavenou řadou kolíků (obvykle 9, 24 nebo 48), které narážejí na pásku nasycenou inkoustem, když se tisková hlava pohybuje tam a zpět před papírem. Jehly jsou uspořádány v jedné nebo několika vertikálních řadách.

Další, méně známá technologie, nazvaná Comb Matrix nebo Shuttle Matrix, používá vodorovnou řadu jehel (optimálně jednu jehlu pro každý vodorovný znak), protože jedna řada bodů může znak lépe reprodukovat. V případech, kdy není tolik jehel jako znaků ve vodorovných čarách, hřeben se mírně pohybuje dopředu a dozadu, aby se vytiskly potřebné body. Pomocí této techniky se papír může posouvat téměř nepřetržitě, a proto se často používá ve vysokorychlostních jehličkových tiskárnách. Tato technologie je mnohem efektivnější než tisk jednotlivých symbolů.

Dopadové jehličkové tiskárny jsou všestranné, tj. kolíkovou matici lze upravit pro tisk různých písem a dokonce i grafiky.

Jehličkové tiskárny lze použít k tisku kopií spolu s originálem, protože obrázky jsou vytvořeny naražením pásky inkoustu. Tyto tiskárny používají nekonečný papír, ačkoli většina modelů může používat také jednotlivé listy a štítky.

Vzhledem k metodě tisku jsou jehličkové tiskárny výrazně hlučnější než ostatní modely (i v tzv. tichém režimu).

Rychlost tisku, obvykle udávaná v cps (znaků za sekundu), se liší v závislosti na použitých fontech (Draft/LQ) a různých hodnotách rozlišení cpi (bodů na palec) [16], [15], [8].

1.2. Tiskárny a psací stroje Daisywheel

V tiskárnách typu daisywheel se výtisk vytváří pomocí bubnu, na kterém jsou vyražena všechna písmena a symboly. Hlava se pohybuje tam a zpět po papíru a elektromagnet začne pracovat, když je buben umístěn pro požadované písmeno.

Tiskárny Daisywheel byly oblíbené, protože jejich kvalita tisku byla lepší než u dřívějších modelů, jako jsou devítipixelové jehličkové tiskárny.

Jejich hlavní nevýhody jsou: pomalá rychlost tisku, omezený počet symbolů na válci a chybějící schopnost vytvářet grafiku.

Tiskárny Daisywheel jsou typem nárazové tiskárny. To vám umožní vytisknout kopii současně s

originalem. Tisk je možný na nekonečný papír i na jednotlivé listy a štítky.

Rychlost tisku se obvykle udává v cps (znaků za sekundu) [8].

1.3. Inkoustové tiskárny

Inkoustové tiskárny fungují podobně jako jehličkové tiskárny, které vytvářejí tisk uspořádáním bodů na papíře. V inkoustových tiskárnách se kapky inkoustu stříkají přímo na papír. Mnohem hustší vzorek bodů než u jehličkových tiskáren umožňuje kvalitnější výtisky.

U novějších modelů inkoustových tiskáren není inkoust vystřikován elektrostatičticky nebo pomocí malých čerpadel, ale přeměňuje se na páru zahříváním a pod tlakem expandující páry se ukládá na papír.

Inkoustové modely nejsou nárazové tiskárny, takže kopie se tisknou samostatně. Tisk je možný na řezaný papír jako u kopírky. Některé modely vyžadují papír speciální kvality pro lepší tisk, ačkoli pro většinu tiskáren stačí standardní papír.

Některé tiskárny mohou používat nekonečný papír a mohou dokonce tisknout štítky, pokud jsou vyrobeny z papíru, který dobře absorbuje inkoust.



rys.A HP DeskJet 600C

Inkoustové tiskárny jsou tišší a mnohem rychlejší než nárazové tiskárny. Parametr rychlosti tisku se udává v cps (znaků za sekundu), někdy také u různých písem v ppm (stránkách za minutu).

Přestože jsou inkousty těchto tiskáren rozpustné ve vodě, značně se liší z hlediska smáčení. To může být důležité, pokud má být tisk použit pro prezentace v

prostředí s vysokou vlhkostí.

Některé typy inkoustů nejsou odolné vůči vodě. Mezi takové inkousty patří: všechny inkousty Canon, DEC 520ic black a inkousty TI Micromarc. Černý inkoust HP DeskJet (obr. 1) je však téměř stejně voděodolný jako laserový tisk. Černé a barevné inkousty Epson se zdají být nejvíce voděodolné: písmena se rozptýlí, ale při vystavení vlhkosti se nerozmazávají.

Barevné (inkoustové) výtisky bohužel nejsou tak trvanlivé jako fotografie. Ty, které jsou odolné vůči světlu, jsou často zničeny vlhkostí, zatímco jiné rychle vyblednou na jasném světle. Trvanlivost potisku lze prodloužit laminováním. Moderní barevné inkoustové tiskárny se bohužel nepoužívají dostatečně dlouho na to, aby otestovaly životnost nejnovějších inkoustů [8], [10], [11].

1.4. Laserové tiskárny a LED tiskárny

Laserové a LED modely nejsou nárazové tiskárny, takže kopie musí být vytištěny samostatně. Obvykle používají jednotlivé listy papíru. Mohou tisknout štítky, které vydrží teplo při vypalování toneru.

Pro tisk v laserové tiskárně se speciální válec elektrostatičticky nabije a kresba stránky se vytvoří vybitím válce laserovým paprskem nebo čarami LED (Light Emitting Diode) v příslušných bodech. Toner přilne k vybitým bodům. Tyto body jsou zrcadlovým obrazem místa, kde se médium objeví na konečném

výtisku.



Obr HP LaserJet 4V

Papír prochází koronovým drátem, který mu dodává elektrický náboj opačný k náboji bodů na válci, pak prochází přes válec, ze kterého je toner zadržován na jeho povrchu. Nakonec se přesune pod topné těleso, které spojí toner s papírem.

Většina současných laserových tiskáren nemá koronový drát a papír je elektrifikován drátem. Laserové a LED tiskárny vytvářejí velmi jasné, přesné výtisky s rozlišením nad 300 dpi (dokonce dosahující až 1200 dpi). Některé tiskárny používají různé techniky, jako je technologie Resolution Enhancement, k ovládání rozestupu bodů za účelem vyhlazení obrysů.

Laserové a LED tiskárny jsou relativně tiché. Tiskový výstup laserových a LED modelů se měří v ppm (stránkách za minutu) v režimu kopírování (tj. obraz stránky, který je již v paměti tiskárny, se co nejrychleji zkopíruje na papír) [8], [10], [11].

1.5. Barevné tiskárny

Barevné tiskárny používají k vytvoření barevného výtisku několik metod.

Bez ohledu na metodu použitou k vytvoření barevného tisku existují tři různé způsoby, jak určit barvy:

RGB (červená-zelená-modrá): standardní metoda používaná ve spojení se zařízeními, jako jsou monitory a televize. Jde o aditivní proces míchání barev (bílá se získá smícháním všech tří barev stejné intenzity).

CMYK (cyan-magenta-yellow-black): metoda využívaná zejména v tisku pro komerční účely. Jedná se o subtraktivní proces míchání barev (černá barva se získá smícháním všech tří barev se stejnou intenzitou nebo přidáním další černé barvy, která vytvoří sytější, tmavší černou).

Rozklad: Metoda, při které se barvy ve skutečnosti nemíchají, ale umístěním bodů různých barev do různých stínovaných oblastí se vytvoří dojem, že obraz používá více než čtyři barvy. Tato metoda je podobná metodě vytváření stupňů šedi nastavením více nebo méně černých bodů.

Barevné jehličkové tiskárny používají barevnou pásku. Inkoustové tiskárny obsahují barevné inkoustové kazety. Některé z nich vytvářejí černou barvu mícháním barev (HP DeskJet 600 C obr.1), jiné mají samostatnou černou inkoustovou kazetu (HP DeskJet 660 C).

Termální tiskárny používají film se čtyřmi barvami vosku. Tiskárna podává papír čtyřikrát, jednou pro každou barvu. Když se každá barva přenesne na papír, vytvoří se plně barevný obraz.

Někteří výrobci nedávno představili barevné laserové tiskárny se čtyřmi samostatnými zásobníky toneru, jeden pro každou primární barvu.

Barevné laserové tiskárny obvykle používají čtyři barvy toneru: tři hlavní subtraktivní barvy (azurová, žlutá a červená) a černou.

purpurova a žlutá) a černou.

Viditelné spektrum je spojité, takže viditelné barvy můžete pouze aproximovat smícháním primárních, subtraktivních barev (smícháním azurového, purpurového a žlutého toneru ve skutečnosti vznikne druh šedé, nikoli výrazná černá). Přidáním černého toneru získáte více barev a mnohem živější černou. U všech barevných laserových tiskáren se barevný rozsah vytváří smícháním tonerů jedním ze dvou základních způsobů:

1) V tiskárnách s kontinuálním tónováním můžete změnit množství toneru pro každou barvu u každého bodu. Tyto tiskárny jsou poměrně drahé, ale také poskytují reprodukce téměř fotografické kvality. U 32bitové barevné tiskárny s kontinuálním tónováním může mít každý pixel 4 294 967 296 různých kombinací tonerů. Není nutné tisknout všechny, když například kombinace výrazné černé s C, M a Y vypadá identicky a uživateli tak vzniká pouze plýtvání tonerem. Navíc přechod z 24bitové RGB palety na 32bitovou CMYK paletu obvykle není zřejmý, takže uživatel má ve skutečnosti přístup k 16 777 216 barvám.

Ne všechny barvy, které lze zobrazit na obrazovce v paletě RGB, jsou vytvořeny smícháním čtyř tonerů. Některé z nich budou mimo měřítko a lze je dosáhnout jejich nahrazením tisknutelnou barvou, která je vnímána podobně. Stejným způsobem bude možné tisknout barvy, které nelze reprodukovat na obrazovce RGB.

2) V tiskárnách s odstupňovaným tónem nelze změnit množství toneru na barvu v každém pixelu. Jsou relativně levnější, ale jejich kvalita tisku je mnohem nižší, zejména při reprodukci skutečných obrázků. Každá ze čtyř barev může nebo nemusí být přítomna v každém pixelu, takže každý bod může mít pouze 16 různých kombinací toneru. Navíc černá barva smíchaná s jakoukoli jinou barvou se bude jevit jako černá, takže 8 z těchto kombinací bude vypadat stejně a každý pixel se může ve skutečnosti jevit jako 9 různých barev. Barvy, které nelze přímo znázornit, jsou simulovány ditheringem, což způsobuje, že lidské oko vnímá přechodné barvy stejným způsobem jako polotónové novinové fotografie. Tisk není možný v 16 777 216 barvách reprezentovaných 24bitovým RGB. Tisknout lze pouze 9 různých barev, takže je nutné použít proces rozkladu pro zobrazení 24bitových barev. Rozklad umožňuje zobrazení mezilehlých barev způsobem srovnatelným se spojitým tónem. Tiskárny s odstupňovaným tónováním lze dodávat s 24bitovými daty, aniž by byla ovlivněna schopnost smíchat různá množství každého toneru pro stejný barevný rozsah v každém bodu [8].

1.6. Jiné typy tiskáren

Existují další tiskové technologie, např. technologie řádkové tiskárny, používané zejména v systémech sálových počítačů. Tisk se provádí pomocí rotujícího válce nebo řetězu obsahujícího všechny symboly.

Když je použit rotační válec, všechny sloupce obsahují kompletní sadu znaků, válec se otáčí ve směru posunu papíru a papír je zasunut mezi válec, když je symbol, který se má tisknout, ve správné poloze. Jedna otáčka válce vytiskne celý řádek textu.

Při použití rotujícího řetězu se na řetězu několikrát opakuje sada symbolů. Návrhář tiskárny určuje

počet opakování každého symbolu v závislosti na tom, jak často se vyskytuje v textu. V americké angličtině je sekvence ETAON... Pokud se často používají symboly, pak je tisk celého textu rychlejší. a řetěz se otáčí kolmo ke směru pohybu papíru a papír se zacvakne mezi pás a řetěz, když je symbol, který se má tisknout, ve správné poloze. Variantou řetězové tiskárny je Belt Printer. Kvalita tisku je podobná jako u řetězové tiskárny, s nižší spotřebou energie a menší hlučností. Standardní pásová tiskárna používá vysoce kvalitní řetězovou sadu. Na tiskárnách IBM 3203 a 1403 každý tah vytiskne tři symboly z řetězce. Použití řetězce slov je záměrné: tři symboly na modulu následují jeden za druhým během otáčení řetězce.



Obr. C IBM 3203

V pásové tiskárně jsou symboly uspořádány samostatně na prstech (podobně jako symboly na sedmikráskovém kolečku) umístěných na gumovém pásu. Stejně jako u řetězové tiskárny se ne všechny symboly opakují se stejnou frekvencí. Na rozdíl od válcových a řetězových tiskáren je pás zacvaknutý mezi papírové pružky.

Rychlost tisku se u těchto tiskáren obvykle udává v lps (řádky za sekundu) nebo dokonce pps (stránky za sekundu). Pro zvýšení rychlosti tisku se doporučují pásové tiskárny, řetězové tiskárny a rotační válcové tiskárny (až 70

lps).

Existují také termální jehličkové tiskárny, které umožňují tisk symbolů a grafiky (např. čárových kódů) zahřátím papíru (tečky): zahřátá místa zčernají.

V oblasti tvorby tisku se používá několik dalších exotických technologií, jako je hliníkem potažený papír používaný společností Sinclair a technologie jiskrového výboje. Jsou však velmi vzácné a většinou zastaralé [8••].

2. Klasifikace plotru v

2.1. Deskové plotry

Deskové plotry jsou zařízení s poměrně velkými rozměry, blízkými maximálnímu formátu výkresu. Například plotr formátu A1 nebo A0 vypadal jako obrovský stůl, na kterém byly elektrostaticky připevněny listy pauzovacího papíru. Značky se pohybovaly po ocelových kolejničkách na desce a kreslení i toho nejmenšího prvku trvalo velmi dlouho, protože značky musely cestovat celou cestu tam a zpět po kolejničkách [11].

2.2. Perové plotry (nové)

Mnohem lepší jsou perové plotry (obr. 4). Tato zařízení zabírají mnohem méně místa a list pauzovacího papíru je připevněn pouze ve dvou bodech.

Princip činnosti lze nejjednodušeji představit jako rám s napnutým pauzovacím papírem pohybujícím

se po válečcích ve vertikální rovině, zatímco vozík s pery se pohybuje v horizontální rovině. Celkové



kresba s Perovský plotr

rozměry perového plotru bylo nutné přizpůsobit výhradně šířce formátu. Jednalo se o mnohem menší, tišší a rychlejší zařízení.

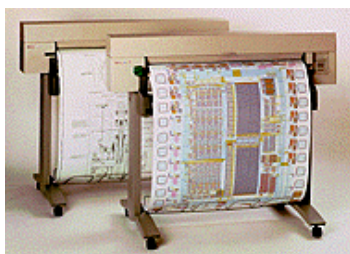
Rychlost práce se neustále zvyšovala díky softwarové optimalizaci pohybu vozíku per.

U perových plotrů nebylo možné překonat následující nevýhody: barevné možnosti byly omezeny na výměnu per a vyplňování polí se omezilo na jejich šrafování. Kreslení bitmap bylo možné pouze teoreticky. Kvalita kresby závisí na řadě faktorů, jako je kvalita inkoustu a pera a rychlost pohybu pera. U složitých, vícevrstevných kreseb větších formátů docházelo k nepřesnostem v řádu milimetrů, způsobených hromaděním mechanických odchylek posuvu pera.

Vývoj perových plotrů (laboratoře Houston Instruments) vedl nakonec ke vzniku plotru se snímací hlavou. Instalace takové hlavy přemění plotr na velkoformátové skenovací zařízení [11].

2.3. Inkoustové plotry

Rastrové inkoustové plotry byly vyvíjeny souběžně s perovými plotry. Toto zařízení funguje tak, že stříká kapičky inkoustu na pauzovací papír, podobně jako inkoustová tiskárna. Inkoustové plotry jsou mnohem přesnější a snáze se používají a efekty jejich práce jsou jasnější.



výkres E DesignJet 350 C

V dnešní době mnoho společností uvádí na trh velkoformátové rastrové plotry. Nejnovějším produktem na tomto trhu je Hewlett-Packard DesignJet 750C . Jedná se o barevný plotr formátu A0 s extrémně vysokým rozlišením skutečných 600 dpi černobíle. Jako většina moderních produktů Hewlett-Packard jde o zcela bezúdržbové zařízení. Po vložení role papíru a inkoustových kazet na jejich místa lze plotr zapnout. DesignJet (obr. 5) se již sám

postará o správnou funkci všech trysek, o správné umístění papíru, zkontroluje parametry média a přizpůsobí se danému formátu.

Původním příspěvkem Hewlett-Packard k vývoji koncepce rastrového plotru bylo použití tzv. pigmentový inkoust. Doposud byl hlavním argumentem proti inkoustovým plotterům fenomén vysychání trysek. Společnost HP zajistila, aby produkt sám zkontroloval, zda není nějaká tryska ucpaná, a pokusí se poruchu sám opravit . Pouze použití pigmentového inkoustu přinese radikální efekty a přispěje ke zlepšení kvality tisku. Tento inkoust je na rozdíl od jiných suspenzí pevné látky. Kazetu s takovým inkoustem již nelze znovu naplnit, tzv. refilami .

Účinek použití pigmentového inkoustu spočívá ve zvýšení jasnosti nakreslené čáry nebo povrchu. Vrstva hmoty nastříkaná na papír jej chrání před vlhkostí a zvyšuje trvanlivost a odolnost kresby.

Jemnozrná struktura inkoustu umožňuje dosáhnout rozlišení kolem 600 dpi.

Dalším zajímavým řešením použitým u modelu 750C je kompenzace vůle v převodech pojezdové dráhy. Odstraňuje efekt zubatých svislých čar, které v případě některých jiných zařízení, zejména při kreslení velkých prvků tenkými čarami, vytvářejí méně estetický vizuální efekt.

DesignJet 750C je extrémně tiché a rychlé zařízení. Pro jednoduché kresby sice nemůže konkurovat perovým plotterům, ale u složitých je předčí, s dobou kreslení A1 kolem 4 minut.

Velkou vymožeností je možnost vybavit plotr tzv. Karty HP JetDirect. Jedná se o síťovou kartu, díky které je plotr v síti vnímán jako zařízení dostupné všem zaměstnancům a nevyžaduje zřizování dalšího serveru. 8 MB RAM, která je standardně součástí DesignJet, umožňuje vytvářet velmi velké výkresy (dokonce o velikosti několika megabajtů).

Ovladače jsou navrženy tak, aby tisk probíhal v kontinuálním režimu v paměti počítače souběžně se zpracováním vektorů do bitmap. Další paměť je potřeba pouze tehdy, když systém rozšíříme o funkce PostScript [11].

3. Ovládací jazyky tiskárny a plotru

Moderní tiskárny dokážou vytvářet velmi složité výkresy, které někdy konkurují fotografiím nebo dokumentům přijatým z tiskárny. Způsob, jakým je datový tok z tiskárny odesílán, je sám o sobě formátem, ve kterém je obrázek uložen v souboru. Existují dva obecné typy formátů tiskových souborů pro tiskárny: rozšířené textové formáty a jazyky pro popis stránky.

3.1. Rozšířené textové formáty

Takové formáty zahrnují grafické informace do konvenčního textového datového toku. Vytiskne se prostý text a sekvence escape zavádí netextové prvky. Široce používaným formátem je Printer Control Language (PCL). Stala se standardem pro maloobjemové a středněobjemové laserové tiskárny [6], [15], [16], [8].

3.1.1. IBM Proprinter

Jazyk tiskárny IBM ProPrinter, původně používaný pro tiskárny IBM (IBM Graphics Printer 5152, IBM ProPrinter XL 4201/4202, IBM ProPrinter X24/XL24 4207/4208), je v současné době používán také mnoha jehličkovými tiskárnami, jako jsou Epson ESC/P a ESC/P2. Funkce se liší v závislosti na použité verzi nebo verzi zvolené během nastavení tiskárny (XL, XL24 nebo AGM jsou příklady takových verzí).

Ovládací jazyk IBM ProPrinter v tiskárnách od různých výrobců se může mírně lišit ve funkčnosti. Typickými rozdíly jsou další funkce, které nejsou součástí standardní výbavy.

3.1.2. Epson ESC/P, ESC/P2

ESC/P (standardní kód Epson)

ESC/P (standardní kód Epson).

ESC/P2 (standardní kód Epson, úroveň 2).

Jazyk tiskárny ESC/P byl vytvořen společností Epson pro použití v raných jehličkových tiskárnách. Dnes jej používají také inkoustové a laserové tiskárny Epson a mnoho dalších jehličkových tiskáren na trhu. ESC/P2 je vylepšením ESC/P, má např. nové funkce pro změnu velikosti písma, pro tisk rastrové grafiky atd.

Tiskárny ESC/P nebo ESC/P2 od různých výrobců se mohou svou funkčností mírně lišit. Tyto rozdíly jsou obvykle doplňkové funkce, které nejsou součástí původní verze Epson.

Informace o použité verzi ESC/P a ESC/P2 naleznete v dokumentaci k tiskárnám jednotlivých výrobců. Některé položky jsou krátkým seznamem dostupných funkcí s příklady v BASICu. Najdete zde také tabulky se sadami symbolů a tabulky s šířkami písma. Referenční příručka Epson ESC/P2 obsahuje seznam kódů v ESC/P a ESC/P2 a také úplný popis rozdílů v jednotlivých příkazech tiskárny. Poslední verze je ze srpna 1992.

3.2. Jazyky popisu stránky

Dalším přístupem k ovládání tiskárny je definovat zcela nový jazyk pro popis tisku. V minulosti bylo použito několik takových jazyků, ale PostScript se dnes stal standardem [8], [6].

3.2.1. PostScript

PostScript je jazyk pro popis stránek vytvořený společností Adobe Systems Inc. od počátku 80. let. Společnost Adobe byla založena v roce 1982 Dr. Johnem E. Warnockem a Dr. Charlesem M. Geschkem. Poskytuje pokyny pro popis informační stránky. Protože vyžadoval více paměti než většina jazyků pro popis stránek, byl prvním široce dostupným produktem, který ovládal větší počet písem a grafiky.

První verze publikovaná v roce 1985 se nazývá Level I, aktuální vylepšení se nazývá Level II (neplést s: PostScript verze 47.0 nebo 2011.110, ani s číslem na začátku jakéhokoli PostScriptového výstupu, jako %!PS-Adobe-3.0). Úroveň PostScript a rozšíření verze o interpret rozšiřuje možné operace.

V oběhu je několik PostScriptových klonů, ale kvůli ceně licenčního poplatku Adobe pro interpreta; nejznámější je GhostScript. Mezi další, zabudované přímo do laserových tiskáren nebo přizpůsobené pomocí přídavné kazety, patří: Phoenix Page, BrotherScript, Page Styler, True Image, Turbo PS, PDL a KPDL. Říká se o nich, že jsou zcela kompatibilní s PostScriptem, ale tato kompatibilita někdy praskne při načítání písma, při manipulaci s písmem (tj. vkládání metrické tabulky nebo nových symbolů) a při dalších operacích. Pomocí klonů ale není problém s tiskem jednoduchého textu nebo grafiky.

Sada symbolů PostScript se používá k psaní programu PostScript, nikoli symbolů tištěného písma PostScript.

Společnost Adobe doporučuje používat v PostScriptu pouze tisknutelnou podmnožinu symbolů ASCII: mezery, tabulátorů a symbolů CR a LF. PostScript nezakazuje použití symbolů mimo tuto sadu.

ale jejich použití může způsobit problémy s přenosem (např. přenos 8bitových symbolů přes 7bitovou sériovou linku na tiskárnu). Chcete-li reprezentovat 8bitové symboly mimo řetězec, použijte vzorec `\ddd`.

Tiskový soubor je ve skutečnosti PostScriptový počítačový program, který kreslí obrázek. PostScript se používá k provádění operací úzce souvisejících s generováním obrázků.

PostScriptové programy kreslí grafiku dvěma základními způsoby. Jednodušší způsob je nakreslit pixelovou mapu. Jeden ze základních operátorů v PostScriptu čte sekvenci pixelů a zobrazuje je v obdélníkové oblasti na stránce. Tento přístup je vhodný pro připojení k dokumentům obrázků naskenovaných nebo pořízených z obrazovky monitoru nebo jakýchkoli jiných obrázků, které již existují v bitmapovém formátu.

Alternativním způsobem použití PostScriptu je vytvoření výkresu ve vektorovém nebo metasouborovém formátu. Existují operátory pro kreslení čar, kružnic, křivek, obdélníků atd. a z těchto prvků lze sestavit grafiku.

Tyto dva přístupy můžete také kombinovat. Operátory čáry, obdélníku a křivky lze například použít k definování oblasti, která bude poté použita jako maska klipu pro ovládání zobrazení bitmapy [8], [11].

3.2.2. HP PCL a PJP

Jazyk PCL vytvořila společnost Hewlett-Packard pro potřeby svých tiskáren (laserových a inkoustových). Jazykové verze PCL jsou číslovány od 1 do aktuální verze 5e.

Stručná historie PCL (na základě technické referenční příručky pro jazyk tiskárny HP):

PCL 1 - Funkce tisku a prostoru je základem funkcí poskytovaných pro jednoduchou a pohodlnou pracovní stanici pro jednoho uživatele.

PCL 2 - Funkce EDP (Electronic Data Processing) je vylepšením PCL 1. Byly přidány univerzální funkce, tiskový systém pro více uživatelů.

PCL 3 - Funkce Office Word Processing je vylepšením PCL 2. Byly také přidány funkce pro zlepšení kvality tisku a zpracování kancelářských dokumentů (tiskárny řady HP DeskJet).

Formátování stránky PCL 4 je oproti PCL 3 vylepšením. Byly přidány nové možnosti tisku stránek (tiskárny: HP LaserJet, HP LaserJet IIP (PCL 4,5)).

PCL 5 - Funkce Office Publishing je vylepšením PCL 4. Mezi nové možnosti publikování patří HP-GL/2 font a grafické škálování (tiskárny HP LaserJet III, HP LaserJet 4 (PCL 5e)).

Verze PCL se liší funkčností (např. typy písem, bitmapová písma, škálovatelná písma (Intellifonts, True Type), metody komprese grafiky, podpora grafiky ze strany HP LaserJet III).

PCL je nejběžnějším jazykem tiskárny na současném trhu laserových tiskáren. Většina výrobců laserových tiskáren používá pro své tiskárny PCL 4 nebo PCL 5.

PJP (Printer Job Language) byl také vytvořen společností HP, aby poskytoval metodu pro změnu úrovně úlohy a čtení parametrů stavu mezi tiskárnou a hostitelským počítačem. PJP lze použít na začátku tisku k nastavení některých parametrů, jako je jazyk tiskárny (PCL, PostScript nebo jiné), rozlišení (300

nebo 600 dpi), počet kopií atd.

PJL se v současnosti používá v následujících tiskárnách HP : LaserJet IIISi , řada LaserJet 4, PrintJet XL 300 a DesignJet.

PJL se používá také v tiskárnách řady LaserJet 5 [11], [7], [10].

3.3. Jiné jazyky ovládání tiskárny

Na trhu existuje mnoho dalších jedinečných jazyků pro ovládání tiskáren. Následující výčet tedy není úplný (záměrem je pouze je zmínit, nikoli podrobně popisovat). Pořadí, ve kterém jsou jazyky uvedeny, nemá nic společného s jejich důležitostí na trhu.

Advanced Function Printing (AFP): používá se v IBM Mainframe pro stránkové tiskárny. Toto je prezentační funkce sady Mixed Object Document Content Architecture (MO:DCA), která je součástí IBM System Application Architecture.

Ve skutečnosti netisknete pomocí MO:DCA , ale používáte IPDS (Intelligent Printer Data Stream).

Informace zahrnují PTOCA (Print Text Object Content Architecture), GOCA (Graphic Object Content Architecture), IOCA (Image Object Content Architecture) a další.

IPDS je tiskový jazyk IBM SAA . Pracuje s řadou bitmapových písem, s jednoduchou základní grafikou a s bitmapovými obrázky. Vzhledem k jednoduchosti vymyšleného modelu jej lze použít k ovládání a obsluze vysokorychlostních laserových tiskáren.

Diablo 630: Zpočátku používané s rozetovými tiskárnami a psacími stroji. Toleruje pouze sekvence tabulátorů, řádkování a mezery mezi symboly, výběr atributů (tučné, dvojité přeškrtnutí, podtržení), horizontální pohyby v obou směrech, proporcionální mezery a automatické centrování a zarovnání - mimo jiné. Tento jazyk je někdy používán jinými výrobci jako základ pro jejich specifické emulace.

CaPSL : (Canon Printing System Language) byl předchozí standardní jazyk pro laserové tiskárny Canon . Dalším názvem, který se objeví, je LIPS (Laser-beam Image System). Za CaPSL se skrývá příběh – Canon vyrábí tiskové motory pro HP , ale neměl licenci na použití HP PCL ve svých vlastních tiskárnách. Proto vznikla potřeba najít vlastní jazyk tiskárny. Lasery Canon tradičně zahrnují emulace CaPSL, IBM ProPrinter, ESC/P a PostScript , ale ne PCL). Tato část smlouvy mezi Canonem a HP zřejmě vypršela, protože Canon nyní nabízí tiskárny s PCL 4 a PCL 5.

LIPS podporuje Diablo 630 (tovární nastavení pro příkazový režim), režim ISO (pro tisk textu a rastrové grafiky) a režim VDM (pro vektorovou grafiku a tisk symbolů).

RENO : Toto je standardní řídicí jazyk pro tiskárny Agfa (P400, P3400 atd.). RENO je typ jazyka popisu stránky. Jeho funkčnost je obrovská: kromě tisku textu s různými měřítky písma můžete kreslit čáry, vyplňovat ikony (Windows) vzory, používat programovací výrazy (if-then-else, repeat-until, set, use and print variables, push and pop operace), můžete načítat a tisknout vlastní symboly a přenášet data do paměti RAM tiskárny nebo na pevný disk či disketu, pokud je připojena.

Prescribe : Toto je jazyk pro popis stránky vytvořený společností Kyocera . Jeho výhodou je skutečnost, že jej lze zabudovat do jiné aktuální emulace tiskárny na zařízeních Kyocera . Tiskárny Kyocera podporují HP PCL 4, HP PCL 5 a systém KPS CL . Emulace ESC/P (režim LQ 850), IBM

Kyocera podporují HP PCL, klon HP-GLS nazvaný KC-GL, Epson ESC/P (težší LQ-630), IBM

ProPrinter X24E, Diablo 630 , obecnou emulaci řádkové tiskárny a volitelně KPDL , klon PostScript.

DEC: DEC má své vlastní jedinečné jazyky pro laserové tiskárny (LN03, LN06).

ANSI : Dan McGowan z Mannesmann Tally uvádí, že: Tiskárny Mannesmann Tally , které mají ANSI, poskytly základ pro zápis ANSI 3.64 . Je to poněkud volná specifikace pokrývající obecné periferní funkce. Většina tiskáren vyrobených v USA má všechny příkazy ANSI 3.64 týkající se funkcí tiskárny. Tiskárny vyráběné v Německu jsou série létajících sériových tiskáren. Mají MTPL (Mannesmann Tally Printer Language), který je založen na ANSI 3.64, ale obsahuje další příkazy [8].

3.4. Jazyky ovládání plotru

Název HP-GL (Hewlett -Packard Graphics Languages) odkazuje na jazyk původně používaný k ovládání plotrů Hewlett -Packard . Za datum vzniku tohoto jazyka je považován rok 1976, rok, kdy se na trhu objevily výše zmíněné plotry. Jak byla tato zařízení vylepšena, jazyk HP-GL byl obohacen o nové příkazy a jeho druhá verze je nyní označena jako HP-GL/2.

Jazyk HP-GL má také podporu pro vektorová písma, což v praxi znamená možnosti ovládání zařízení podobné těm, které poskytuje PostScript . Velká část jazyka HP-GL/2 byla začleněna do páté úrovně jazyka PCL , která obsahuje speciální příkazy, které přepínají ovládání z klasického PCL na HP-GL a naopak.

Syntaxe jazyka HP-GL/2 je jednoduchá. Všechny instrukce jsou dvouznačkové zkratky jejich názvů. Po zkratce následují parametry. Mohou být povinné nebo volitelné. Jako oddělovač mezi parametry lze použít mezeru nebo čárku. Preferovaným oddělovačem je čárka. Každá instrukce končí terminátorem. Terminátory mohou být: středník, první znak další instrukce nebo mezera [10], [7], [6].

Překládové systémy

1. Převod mezi typy souborů

Protože existuje mnoho formátů jazyků pro ovládání tiskárny, je samozřejmé, že je často potřeba převést jeden formát tiskového souboru na jiný. V závislosti na typu zdrojového a cílového formátu to může být triviální úkol nebo může být dokonce nemožné ho splnit. Obrázky uložené v tiskovém souboru mohou mít tvar . Může to být bitmapový obrázek, vektorový obrázek nebo dokonce prostý ASCII text . Vzhledem k jazykovému formátu, kterému tiskárna rozumí, může být nutné převést formát obrázku uložený v tiskovém souboru [6].

1.1. Bitmapa na bitmapu

Převod jednoho typu bitmapy na jiný je obvykle snadný. Jakmile se dostaneme přes detaily kódování souboru, máme co do činění s pixely, které jsou téměř vždy stejné, takže převod je jednoduchý.

Pokud je z více do méně popisného formátu, např. z barevného do černobílého nebo do odstínů šedi, jsou známy způsoby jeho převodu při zachování co nejlepší kvality obrazu.

1.2. Vektor do vektorového formátu

Základní otázkou při převodu mezi vektorovými formáty je přizpůsobit se mírně odlišné sémantice jednotlivých formátů a také do určité míry souřadnicovému systému. V nejjednodušším případě jsou příkazy překládány jedna ku jedné, např. příkaz k nakreslení čáry na stejný příkaz ve výstupním souboru. Problémy nastanou, když tyto dva formáty nemají odpovídající příkazy. Pokud původní formát obsahuje příkaz pro kreslení elipsy a cílový formát ne, musíte použít jednu z dostupných metod překladu. Elipsu můžete aproximovat kružnicí nebo mnohoúhelníkem vytvořeným z krátkých segmentů. Pokud cílový formát umožňuje samostatné měřítko os x a y, může být dobrou metodou nastavit na ně různé jednotky a nakreslit kružnici, která se změní na elipsu.

1.3. Vektorový formát na bitmapu

Zpracování obrázku z vektoru na bitmapu se nazývá rasterizace. Zahrnuje nalezení sady pixelů odpovídajících každému vektoru původního obrázku. Základní rasterizační algoritmus je znám od roku 1963, kdy byl publikován Bresenhamem ; Kruhy a oblouky lze kreslit stejnou metodou.

Tento typ převodu se zabývá stejnými problémy jako kreslení vektorového obrázku na rastrové obrazovce nebo laserové tiskárně, takže je často možné upravit kód zobrazení pro převod na bitmapu.

1.4. Bitmapa do vektorového formátu

Převod bitmapy na vektorový obrázek je mnohem složitější než kterákoli z předchozích transformací. V současné době existují uspokojivé algoritmy detekce hran. Lze je použít k nalezení čar v bitmapě, ale pouze v těch nejjednodušších případech. Problém hledání čar v naskenovaném obrázku (např. ve faxovaném dokumentu) zůstává nevyřešen.

2. Konverze programů

2.1. Nahrazení PostScriptu jinými standardy

Nejznámějším programem pro interpretaci PostScriptu je GhostScript . Vyvoláním GS s volbou -help můžete získat informace o dostupných zařízeních. GhostScript umí emulovat následující systémy: epon, eponc, necp6, laserjet, ljetplus, ejet2p, ljet3, paintjet , bj10e, djet500, djet500c, pjetxl, lbp8 (toto jsou názvy používané v programu). Do programu je také možné přidat nové ovladače v [8].

2.2. Změna jiných standardů na PostScript

V případě textu ASCII není změna ze standardního na PostScript složitá. Pro tento účel existuje několik veřejně dostupných nástrojů. Nejjednodušší z nich je program s názvem a2ps.

U textu bez ASCII (ISO 8859-1 nebo PC 437) nebo u sekvencí ovládání specifických pro tiskárnu může být převod komplikovaný. Chcete-li převést HP PCL na PostScript, použijte nástroj lj2ps , který je velmi užitečný v případě neproporcionálních písem (jako u HP LaserJet II). Problém se však komplikuje při převodu grafiky. Existuje také konvertor pro HPGL (hpgl2ps), ale je vzácný. Chcete-li změnit Epson na PostScript, použijte filtr epon2ps [8].

Aplikace Emulátor

Program EmuLator je určen pro prostředí Windows počínaje verzí 3.1. Volbu tohoto systému předurčila především jeho jednoduchost ovládání a komunikace s uživatelem prostřednictvím bohatého grafického rozhraní. Snadné použití programu je umožněno složitostí programování pomocí API (Application Programming Interface) – sady funkcí, které musí programátoři používat při psaní aplikací pro Windows. Windows API obsahuje více než 600 funkcí. Kromě výše uvedených nepříjemností má programování pro Windows několik výhod [13]:

System Windows poskytuje nezávislost na hardwaru. Stejný program může zobrazovat informace na různých monitorech (EGA, VGA atd.) a tisknout na různých tiskárnách, od jehličkových až po laserové.

Z pohledu programátora nabízí Windows mnoho hotových prvků uživatelského rozhraní, jako jsou tlačítka na obrazovce, nabídky, dialogová okna, seznamy a textová pole.

Windows obsahuje rozsáhlé grafické rozhraní (GDI) pro zobrazování textu a grafiky. Toto rozhraní vám zejména umožňuje kreslit ve vašem vlastním souřadnicovém systému.

Zvoleným programovacím jazykem byl C++ pro nejnovější kompilátor v době vytvoření programu, Borland 4.52 [4]. Tento nástroj umožňuje objektově orientovaný přístup k navržené aplikaci [2] a umožňuje použití knihoven tříd OWL a CLASSLIB [1].

Primárním účelem knihovny tříd OWL (Object Windows Library) je poskytnout programátorům kompletní rámec pro aplikace Windows. Byl použit Borland's OWL 2.5, který umožnil rychlé vytvoření uživatelského rozhraní s atraktivním grafickým ovládáním.

Kontejnerová knihovna CLASSLIB společnosti Borland poskytovala výstupní objekty pro ukládání a zpracování dat.

1. Popis programu

Program je navržen tak, aby řešil problém interpretace grafického obrazu obsaženého v tiskovém souboru na obrazovce počítače. Velké množství jazyků pro ovládání tiskárny a plotru a různé formy jejich zápisu vyzve programátora k interpretaci zvoleného jazyka tiskárny. Tento přístup lze pozorovat například v programu GhostScript interpretujícím jazyk PostScript. Tato práce zahrnuje napsání programu, který bude interpretovat různé řídicí jazyky.

Předpokládá se, že tiskový soubor obsahuje různé příkazy způsobující grafické akce, jako je: tisk textu nebo kresby grafiky a sekvence měnící parametry tiskárny. Výsledky lze vidět na vytištěném listu papíru, takže je lze také prezentovat na obrazovce počítače. Jednotlivé řídicí jazyky způsobují stejnou akci pomocí různých kombinací kódů. Aby bylo možné simulovat činnost tiskárny nebo plotru, obsahuje program sadu základních operací prováděných těmito zařízeními a umožňuje provádění těchto operací.

Významným problémem bylo použití libovolného jazyka tiskárny programem způsobem, který byl uživatelsky přívětivý a zároveň umožňoval kdykoli přidávat nové ovládací jazyky. Emulátor otevře ovladač tiskárny, což je textový soubor v určeném formátu, a zkontroluje, zda neobsahuje chyby při zápisu. Tento soubor také obsahuje výchozí nastavení pro danou tiskárnu nebo plotr. Uživatel si může vytvořit ovladač pro každou tiskárnu sám pomocí některých operací obsažených v programu, aniž by bylo nutné program překompilovat. Pokud daná tiskárna používá specifické mechanismy, které nejsou obsaženy v probíraném programu, může šikovný programátor rychle rozšířit program o nové funkce.

Aplikace využívá mechanismy MDI (Multiple Document Interface), které umožňují prohlížení několika dokumentů současně. Program umožňuje interpretovat tiskové výstupy vždy pouze pro jeden typ tiskárny.

Výsledný efekt lze sledovat na obrazovce v měřítku 1:1 (s ohledem na nepřesnosti obrazovky monitoru), v celostránkovém náhledu tisku nebo vytisknout na libovolné tiskárně. Využitím hardwarové nezávislosti kontextu zařízení GDI vám program umožňuje převést známý formát tiskárny nebo plotru do formátu libovolného zařízení dostupného v systému Windows.

1.1. Konstrukce

Aplikace byla vytvořena na základě architektury MVC (Model-View-Controller), běžně používané v jazyce Smalltalk-80, s přihlédnutím ke specifickým možnostem knihovny OWL 2.5. Architektura MVC rozděluje aplikaci do tří vrstev:

1. Model znamená aplikační vrstvu, ve které jsou umístěny všechny objekty závislé na aplikaci. V EmuLatoru jsou to všechny objekty reprezentující datové vrstvy a operace na nich prováděné. Zejména se může jednat o programovatelný slovník umožňující překlad kódu v tiskovém souboru do grafických operací a třídu zpracovávající data tiskového souboru do grafického obrazu používaného v prezentační vrstvě.
2. Pohled je tzv. prezentační vrstva, která je zodpovědná za prezentaci dat. Prezentační vrstva přečte příslušné informace z aplikační vrstvy a zobrazí je na obrazovce. Tato vrstva se také zabývá okny grafického uživatelského rozhraní.
3. Ovladač je tzv. řídící vrstva, která zprostředkovává přenos informací ze vstupních zařízení (klávesnice, myš) do dalších dvou vrstev.

Tento model prochází během implementace drobnými úpravami a je podřízen novým technikám nalezeným v OWL 2.5, jako je Document/View, v závislosti na řešeném problému.

1.1.1. Kontrolní vrstva

1.1.1.1. Hlavní třída aplikace

třída emuApp: public TApplication - nejdůležitější aplikační třída, jejíž implementaci najdeme v souborech emuapp.cpp a emuapp.h, kde je definována i pomocná třída *TFileDrop*, která se zabývá manipulací s dokumenty načtenými metodou *drag and drop*. Třída *emuApp* je základem pro všechny objekty v aplikaci. Ve funkci *InitMainWindow* se vytvoří objekty tříd *emuMDIFrame* a *emuMDIClient*

, načtou se prostředky do grafických prvků programu (jako jsou ikony a nabídky), stavový řádek (*TStatusBar*) a panel nástrojů (*TControlBar*) se vloží do rámce *emuMDIFrame* , nastaví se atributy a prostorové prvky hlavního okna s *Entruetl* . Kromě zpracování standardních zpráv Windows je třída *emuApp* zodpovědná za následující úkoly pro danou aplikaci:

podpora systému nápovědy - načtení souboru nápovědy v běžném nebo kontextovém režimu;

vytvoření nového dokumentu a pohledu - při otevření nového dokumentu a vytvoření pohledu je uživatel informován, zda není načten soubor ovladače; při načítání nového souboru ovladače se všechna okna zavřou;

předání řídicí zprávy pro zobrazení dokumentu tiskového souboru (*emuView*) - třída *emuApp* předá aktuálnímu zobrazení *emuView* požadavek na přechod na další nebo předchozí stránku s informací o jejím aktuálním čísle v panelu nástrojů;

aktualizace informací ve stavovém řádku - třída *emuApp* informuje o aktuálně načteném ovladači tiskárny nebo jeho absenci a při výběru příkazu z nástrojové lišty nebo nabídky se zobrazí nápověda.

1.1.1.2. Třídy řízení aplikací MDI

třída *emuMDIFrame*: *public TDecoratedMDIFrame* (*emmdifm.cpp*, *emmdifm.h*) - obsahuje viditelné ovládací prvky a zpracovává všechny ovládací zprávy určené pro podřízená okna *emuMDICild* tak, že je nasměruje na *emuMDIClient*.

třída *emuMDIClient*: *public TMDIClient* (*mmdiclt.cpp*, *mmdiclt.h*) - je zodpovědný za správu objektů okna *emuMDICild* v reakci na požadavky z nabídky Window.

třída *emuMDIChild*: *public TMDIChild* (*mmdichld.cpp*, *mmdichld.h*) - definuje základní chování MDI okna obsahujícího ukazatel na objekt *TWindow* , který je viditelným prvkem zobrazení dokumentu.

1.1.1.3. Třída ovládání dokumentů v

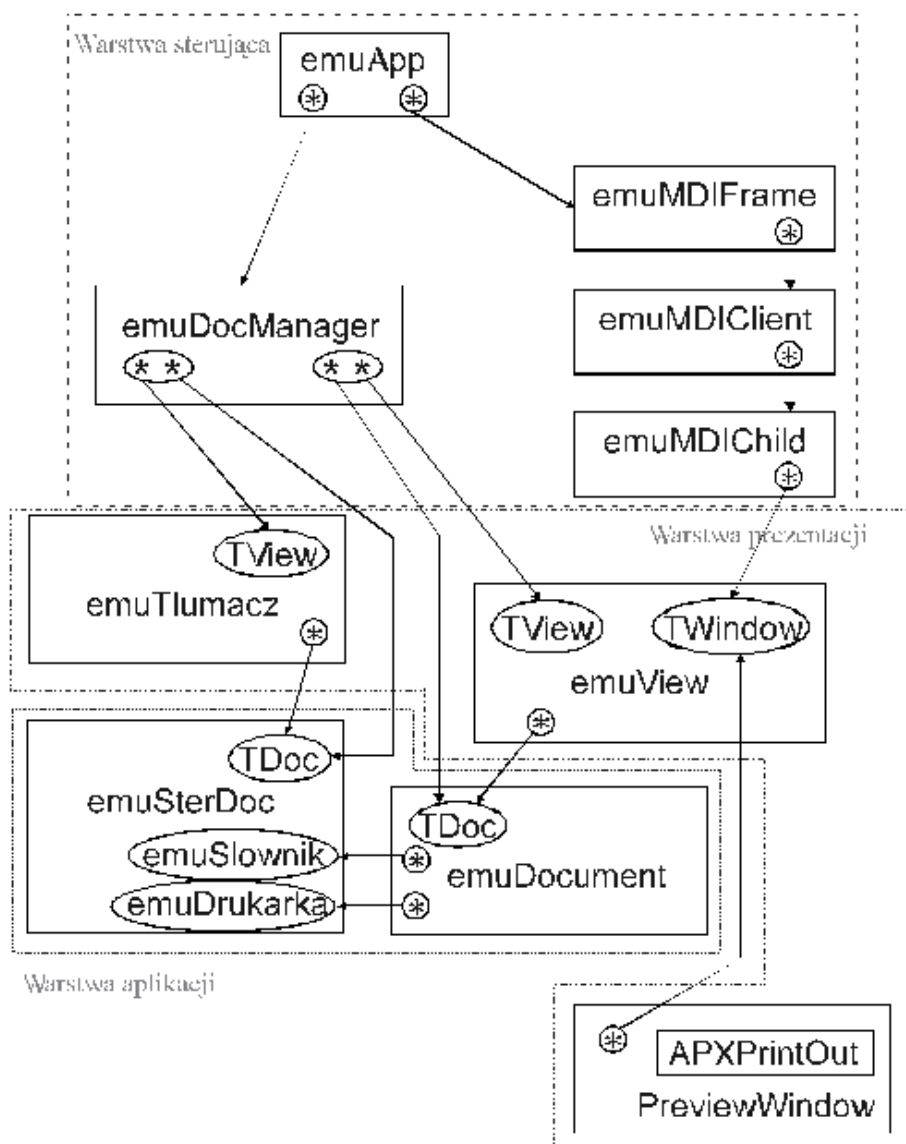
třída *emuDocManager*: *public TDocManager* (*emdocmgr.cpp*, *emdocmgr.h*) - objekt spravující seznam aktuálních registrovaných šablon Makro *DEFINE_DOC_TEMPLATE_CLASS* vytváří šablonu sdružující objekty dokumentu s pohledy, na základě kterých *TDocManager* zpracovává standardní příkazy nabídky Soubor a předává je příslušným dokumentům. Je také zodpovědný za zobrazení prvků uživatelského rozhraní pro výběr a zobrazení souborů.

1.1.2. Prezentační vrstva

1.1.2.1. Třída prezentace dat ovladače tiskárny

třída *emuThucacz*: *public TView* (*emutlmcz.cpp*, *emutlmcz.h*) - objekt fungující jako prostředník při přístupu k datům obsaženým v dokumentu. *emuThucacz* umožňuje získat ukazatel na třídu slovníku (*emuSłownik*) vytvořenou v objektu dokumentu ovladače tiskárny (*emuSterDoc*). Jako jediný pohled v aplikaci nemá *emuThumacz* přidružené okno a je speciálně zpracován v hlavním objektu aplikace. Pro každý objekt dokumentu tiskového souboru (*emuDocument*) nastaví *emuApp* ukazatel na

emuTranslator (v jednu chvíli může existovat pouze jeden objekt *emuTranslator*).



Obr. F Hierarchie hlavních aplikačních tříd

1.1.2.2. Soubor tříd prezentace dat v tisku

class

emuView: public *TWindowView*

(emuv.cpp, emuv.h) - objekt zobrazení dokumentu pro tiskový soubor dědí vlastnosti dvou tříd OWL . Jako potomek *TView* podporuje přístup k datům

emuDocument .

Dědění z *TWindow* je grafické znázornění

dokumentu ve formě viditelného okna,

jehož ukazatel je předán objektu *emuMDIChild* . V reakci na požadavek na zobrazení další stránky předá

emuView objektu

emuDocument kontext zobrazení pro nakreslení obrázku tiskové stránky. Aby každá reakce pohledu na zprávu *WM_PAINT* nevedla ke spuštění celého mechanismu překladu, interpretace a vytvoření výkresu z tiskového souboru, je použit kontext metasouboru uložený ve formě objektu třídy *TMetaFilePict* .

třída *APXPrintOut*: public *TPrintout* (apxprint.cpp, apxprint.h) - představuje fyzický dokument odeslaný na tiskárnu. Objekt této třídy je zodpovědný za kreslení stránky na fyzické tiskárně nebo v okně náhledu tisku.

class *PreviewWindow*: public *TDecoratedFrame* (apxprev.cpp, apxprev.h) – vytvoří rámec náhledu tisku a podporuje zobrazení jedné nebo dvou stránek najednou v objektu *TLayoutWindow* . Je také odpovědný za tisk na fyzické tiskárně v důsledku výběru příkazu *Tisk* aplikace .

1.1.2.3. Jiné třídy

class emuEditView : public TEditView (emuedtvw.cpp, emuedtvw.h) - třída zobrazení spojená s jakýmkoliv typem souboru (přímo *TFileDocument*); umožňuje základní editaci, tisk textu a vyhledávací operace.

třída TBmpViewWindow (emuabout.cpp, emuabout.h) – používá se k zobrazení úvodní viněty programu jako bitmapy.

třída emuAboutDlg: public TDialog (emabtdlg.cpp, emabtdlg.h) - pomocí *pomocné třídy ProjectRCVersion* , která poskytuje informace o projektu, zobrazí dialogové okno o programu.

1.1.3. Aplikační vrstva

1.1.3.1. Třídy dokumentů ovladače tiskárny

třída emuSterDoc: public TFileDocument (emustrdc.cpp, emustrdc.h) - představuje objekt dat dokumentu a poskytuje způsob, jak je interpretovat pohledu. Obsahuje řadu metod pro manipulaci s fyzickým souborem na disku, včetně podpory streamů. Ke každému dokumentu lze přiřadit více pohledů, podporuje tedy komunikaci s nimi vytvářením seznamu aktuálních pohledů a zasíláním zpráv o jejich změnách. Dokumenty a pohledy mají seznam vlastností (*Vlastnosti*), které aplikace používá k rozhodování o zpracování dat. *emuSterDoc* má mezi svými vlastnostmi ukazatele na *objekty emuDictionary* a *emuPrinter*, které vytváří při otevírání dokumentu ovladače tiskárny .

class emuDictionary: public ContainerType (emuslwnk.cpp, emuslwnk.h) - tento objekt je odvozen od třídy kontejnerového slovníku vytvořené pomocí *makra typedef TDictionaryAsHashTable<AssociationType> ContainerType*; je sada asociací spolu s metodami pro jejich přidávání a vyhledávání. Asociace (*typedef TILAssociation<emuMyClass, GraphicalObject> AssociationType*;) je dvojice ukazatelů na objekty v *emuMyClass* a *GraphicalObject* . Objekt třídy *emuDocument* , který najde řídicí kód, provede grafickou operaci odvozenou z *GraphicalObject* . Když *emuSterDoc* vytvoří slovník , dynamicky se vytvoří a přidají pouze ty grafické objekty, které jsou potřebné pro interpretaci daným zařízením.

třída emuMyClass (emumycla.cpp, emumycla.h) – jediný kód tiskárny nebo plotru spolu s porovnávacím operátorem, který jej najde.

class GraphicalObject (emugrobj.cpp, emugrobj.h) - základní třída pro grafické operace v kontextu zařízení, ze které jsou virtuálně odvozeny objekty odpovídající jednotlivým řídicím kódům. Objekt této třídy neprovádí žádnou operaci, ale pro diagnostické účely vypisuje název operace, která má být provedena.

třída GraphObjHPGL: veřejný virtuální GraphicalObject (emughpgl.cpp, emughpgl.h) - základní třída pro grafické operace specifické pro plotr Od ní jsou odvozeny třídy pro konkrétní grafické operace: *třída HPGL_0xXXXX: veřejný virtuální GraphObjHPGL* , kde *0xXXXX* je jedinečné interní číslo aplikace pro danou operaci.

třída GraphObjPCL: veřejný virtuální GraphicalObject (emugopcl.cpp, emugopcl.h) - základní třída pro grafické operace specifické pro inkoustové a laserové tiskárny. Je zdrojem tříd pro konkrétní grafické operace: *třída PCL_0xXXXX: public virtual GraphObjPCL* , kde 0xXXXX je jedinečné interní číslo aplikace pro danou operaci.

třída GraphObjIBMPPro: veřejný virtuální GraphicalObject (soubor emugopro.cpp, emugopro.h) - základní třída pro grafické operace specifické pro jehličkové tiskárny. Je zdrojem tříd pro konkrétní grafické operace: *třída IBMPPro_0xXXXX: public virtual GraphObjIBMPPro* , kde 0xXXXX je jedinečné interní číslo aplikace pro danou operaci.

třída emuPrinter (emudrkrk.cpp, emudrkrk.h) - třída obsahující počáteční nastavení emulovaného zařízení, používaná i při vytváření grafického obrázku.

1.1.3.2. Tisk souborů Tříd dokumentů

třída emuDocument: public TFileDocument (emudcmnt.cpp, emudcmnt.h) - podporuje operace související s fyzickým tiskovým souborem. Na požádání *emuView* analyzuje soubor, hledá řídicí kódy v *emuDictionary* a provádí grafické operace s virtuální funkcí *Draw()* objektu odvozeného z *GraphicalObject* .

třída emuStrona (emustrona.cpp, emustrona.h) - třída podporující vytvoření kresby konkrétní stránky. *emuDocument* používá pole stránek (*typedef TArrayAsVector<emuPage> emuPages;*) k zapamatování pozice toku tiskového souboru a nastavení tiskárny (*emuPrinter*) pro každou stránku.

1.2. Rozvoj

Program poskytuje vynikající rámec pro rozšíření o nové funkce a moduly Pomocí techniky OWL Document/View lze vytvářet další typy pohledů a dokumentů pro stávající nebo nové objekty. Také lze snadno rozšířit moduly, které interpretují tiskové soubory. Při rozšiřování programu najdete užitečné komentáře ve zdrojových souborech a pomocných třídách, které nebyly zabudovány do finální verze programu.

1.2.1. Přidání objektu do pohledu v

Chcete-li přidat objekt pohledu, jednoduše definujte novou třídu odvozenou z *TView* , která používá metody obsažené ve vybraném objektu dokumentu. Pokud budou v zobrazení prezentace použity speciální datové formuláře, musí být definovány v objektu dokumentu. Objekt pohledu může být spojen s dokumentem, který má několik různých způsobů prezentace. Aby je program podporoval, musíte pro *emuDocManager* definovat vhodné šablony , např.:

```
DEFINE_DOC_TEMPLATE_CLASS(emuSterDoc, TSterListView, DocType7);
DocType7 __dvt7("Prohlížeč uživatelů", "*.emu", 0, "EMU", dtAutoDelete);
```

kde: *emuSterDoc* je existující třída dokumentu a *TSterListView* je třída zobrazení řadiče. Pohled se může objevit jako okno zobrazené na obrazovce nebo v jakékoli jiné formě interpretace dat obsažených v dokumentu.

1.2.2. Přidání objektu do dokumentu v

Každý objekt dokumentu je zodpovědný za načtení dat a jejich správné doručení do pohledů. Pokud má dokument více pohledů a jedno z nich umožňuje měnit data, je nutné zpracovat i zprávy předávající informaci o této skutečnosti dalším pohledům. Komunikace mezi pohledy a dokumenty může probíhat prostřednictvím zpráv, seznamů vlastností (*Prosperity*) nebo přímých odkazů. Další informace o interakci dokumentu s pohledy naleznete v dokumentaci kompilátoru Borland C++ 4.52 .

1.2.3. Rozšíření grafických funkcí

V programu EmuLator se při interpretaci tiskového souboru používá následující funkce:

```
int emuDocument::Draw( TDC& strDC, stránka emuPage* )
```

kde: *strDC* je kontext zařízení, ve kterém má být obrázek vytvořen; *stránka* je ukazatel na objekt obsahující aktuální nastavení stránky a kopii nastavení tiskárny (*emuPrinter*) z dříve vytvořené stránky nebo (v případě první stránky) nastavení zařízení načtené *emuSterDoc* . Pokud je v toku dokumentů nalezen řídicí kód, objekt *emuDictionary* vrátí ukazatel na asociaci, ze které se získá ukazatel na příslušný grafický objekt. Operace se provádí voláním virtuální funkce objektu odvozeného z *GraphicalObject* :

```
pokud (nalezen)
```

```
{
```

```
if (object = found->Value())
```

```
objekt -> Draw(strDC, stream, page);
```

Funkce *void GraphicalObject::Draw(TDC& dc, TInStream* is, emuStrona* str)* k provedení úlohy se volá s následujícími parametry:

TDC& dc - odkaz na kontext zařízení, na kterém by měla být provedena grafická operace;

TInStream je* - ukazatel na proud tiskového souboru, ze kterého lze získat další data pro provedení operace;

emuPage str* - ukazatel na objekt, kde najdete aktuální nastavení emulovaného zařízení (*emuPrinter *prints*) a v případě chyby nebo konce stránky byste měli nastavit jednu z hodnot *stavové proměnné* :

```
enum {
```

```
Chyba=0,
```

```
koncová stránka,
```

```
EndFile,
```

```
Další,
```

```
};
```

Chcete-li program rozšířit o nové grafické funkce, musíte vytvořit objekt odvozený virtuálně, přímo nebo nepřímo (např. jako objekty *GraphObjHPGL*) z třídy *GraphicalObject* , která provádí operaci pomocí některého z výše uvedených parametrů. Poté je třeba zvolit jedinečné interní číslo aplikace pro grafickou operaci (z rozsahu 0x0000 až 0xppemus : operace přidání objektu .

```

- - - - -
int
emuDictionary::AddItem(const string& mc, const unsigned long int & mv, const
string& mvs)
{
int vysledek=0;
přepínač (mv) {
případ 0xFFFF: {
AssociationType assoc( new emuMyClass(mc),
new GraphicalObject(mvs) );
vysledek=Add(assoc);
přerušeni;
}

```

Tuto funkci volá *emuSterDoc* při čtení souboru ovladače zařízení s následujícími parametry:

mc - řídicí kód emulovaného zařízení;

mv - interní aplikační kód pro grafické operace;

mvs - název operace používaný pro diagnostické účely, převzatý ze souboru ovladače.

1.2.4. Rozšíření vlastností emulovaných zařízení

Pokud grafická operace vyžaduje parametry, které nelze nalézt v definici *emuPrinter* (soubor *emudrkrk.h*), měly by být přidány do této třídy. Inicializace hodnoty probíhá v *bool emuSterDoc::GetPrinter(TInStream* is)*, kam by měla být přidána procedura čtení parametrů, např.:

```

if (key.contains("VELIKOST STRÁNKY")) // Velikost stránky
{
řádek += SkipAll(*is);
long x = GetLong(*is);
řádek++;
if (!is->good()) goto error;
řádek += SkipAll(*is);
long y = GetLong(*is);
řádek++;
if (!is->good()) goto error;
tiskárna->SetRStr(x*(metrické ? mm_pkt : cal_pkt),
y*(metrické ? mm_pts : inch_pts));
}
jiný

```

key je klíčové slovo nalezené v souboru ovladače tiskárny podle následující konvence:

!Velikost stránky

Poté se z textového proudu načtou všechny parametry. **je* pro ně vhodná funkce. Funkce *SkipAll(*is)* přeskakuje všechny znaky ve streamu, dokud nenarazí na znak ! na začátku řádku. *Řádek* znamená událost řádku, kde se objeví znak !. Konstanty *x*(metrické ? mm_pkt : cal_pkt)*

proměnné udává číslo řádku, kde se chyba vyskytuje. Konstrukce x ($metric : mm_prn : cal_prn$) umožňuje nastavit parametry v předdefinovaných jednotkách (aktuálně mm nebo palce).

1.2.5. Pomocné třídy

Při spuštění projektu můžete použít diagnostické třídy, které jsou součástí zdroje. Chcete-li to provést, musíte povolit zahrnutí kódu pro ladící program v možnostech projektu a změnit komentáře v souboru emuapp.cpp a aktivovat šablony pro tato zobrazení.

třída TDumpView : public TListBox, public TView (dumpview.cpp, dumpview.h) - umožňuje asociaci s jakýmkoliv typem dokumentu a zobrazuje přesně jeho obsah s hexadecimálními kódy jednotlivých bajtů v souboru;

třída TInfoView : public TWindowView (infoview.cpp, infoview.h) - umožňuje zobrazit seznam vlastností (*Prosperity*) libovolného dokumentu;

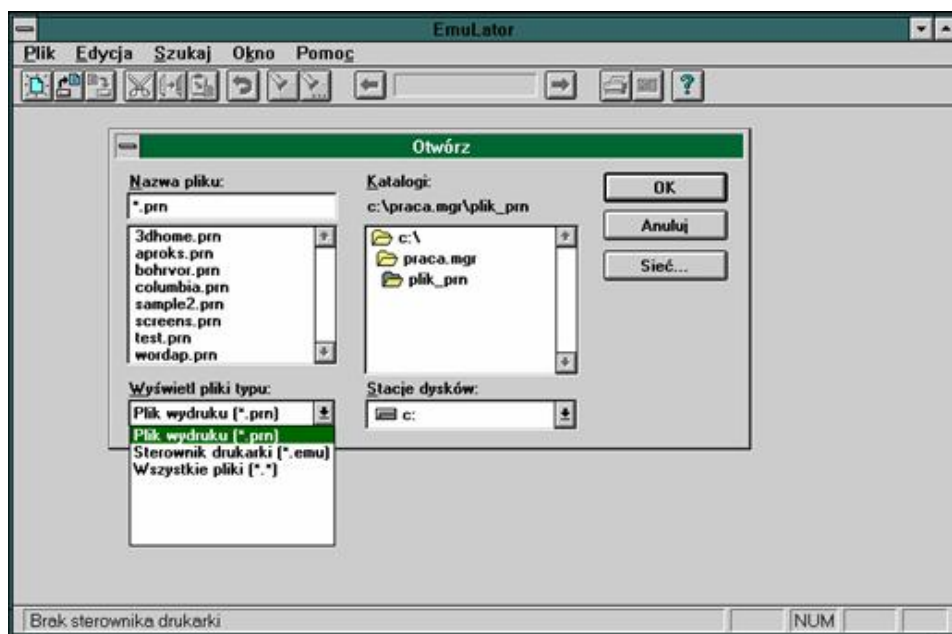
class TSterListView : public TListBox, public TView (strlistvw.cpp, strlistvw.h) - pohled vytvořený speciálně pro potřeby dokumentu třídy *emuSterDoc* zobrazující obsah slovníku inicializovaného pomocí souboru ovladače tiskárny.

Programová podpora

Návod k použití

Program Emulator musí být nainstalován na pevný disk vašeho počítače, než jej budete moci používat. Jeho instalační balíček je umístěn na HD 3,5" disketách. Proces instalace je typický pro prostředí Windows .

Po instalaci programu Emulátor má v okně Správce programů svou vlastní skupinu nazvanou Emulátor tiskárny a plotru . Hlavní programová ikona aplikace se jmenuje Emulátor .

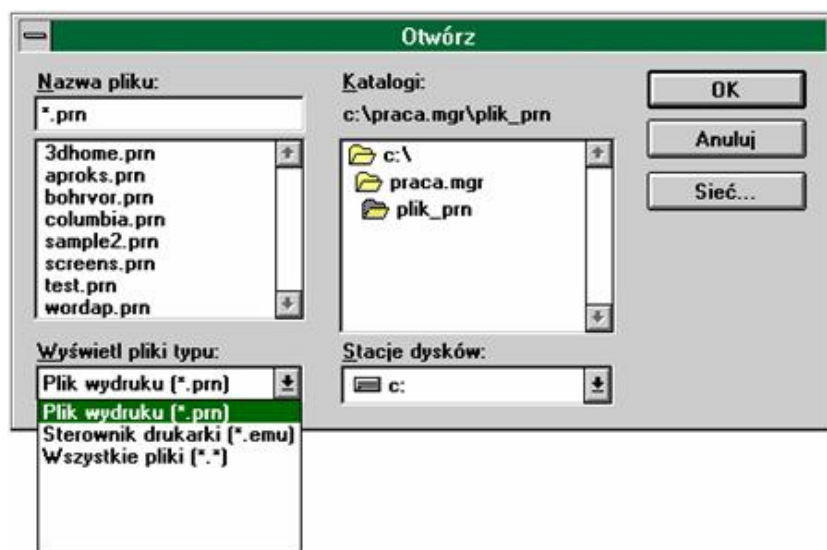


Obr. 6. Hlavní okno aplikace

Po spuštění aplikace se na obrazovce objeví okno aplikace s názvem Emulátor (obr. 7).

Obr. 8. Hlavní okno aplikace

Operace provozu programu jsou obvykle omezeny na několik základních:



obr. H Otevřené okno

Otevření dokumentu

Otevření

dokumentu obsahujícího tiskový výstup musí předcházet načtení ovladače zařízení, pro které byl soubor vytvořen. Ovladač se načte výběrem možnosti *Otevřít* z nabídky *Soubor* nebo kliknutím na příslušnou ikonu na panelu nástrojů. Po rozbalení dialogového

okna *Otevřít* nastavte typ souboru na *Ovladač tiskárny* s příponou *. emu a v dialogovém okně *Název souboru* zadejte název ovladače odpovídající vaší tiskárně. Název zařízení, pro které byl načten ovladač, se objeví ve stavovém řádku v dolní části okna (obr. 9).

Pokud je kód příkazu v ovladači špatně uložen, při pokusu o jeho načtení program zobrazí informaci o čísle řádku s chybně zadaným kódem.



Obr. I Stavový řádek s načteným ovladačem.

Program je připraven otevřít tiskový soubor. Postup je podobný výše uvedenému s tím rozdílem, že jako typ souboru, který chcete otevřít, byste měli vybrat *Tisk souborů* s příponou *. prn . Pokud se pokusíte znovu otevřít již načtený dokument, program o tom uživatele informuje a nedovolí vám to. Ke stejnému dokumentu je však možné přidat další pohledy; *Přidat* možnost zobrazení z nabídky *Okno* . Obsah tiskového souboru se zobrazí v okně MDI s názvem otevřeného dokumentu. Další (předchozí) stránka dokumentu (pokud existuje) se zobrazí výběrem možnosti *Stránka* v nabídce *Úpravy* nebo tlačítka se šipkou v panelu nástrojů (obr. 10).

Zavření dokumentu

Chcete-li zavřít otevřený dokument, vyberte možnost *Zavřít* v nabídce *Soubor* . Při změně zařízení, pro které se dokument zobrazuje, se automaticky zavřou všechna zobrazení (dokumenty).



obr. J Procházení stránek dokumentu

Náhled tisku

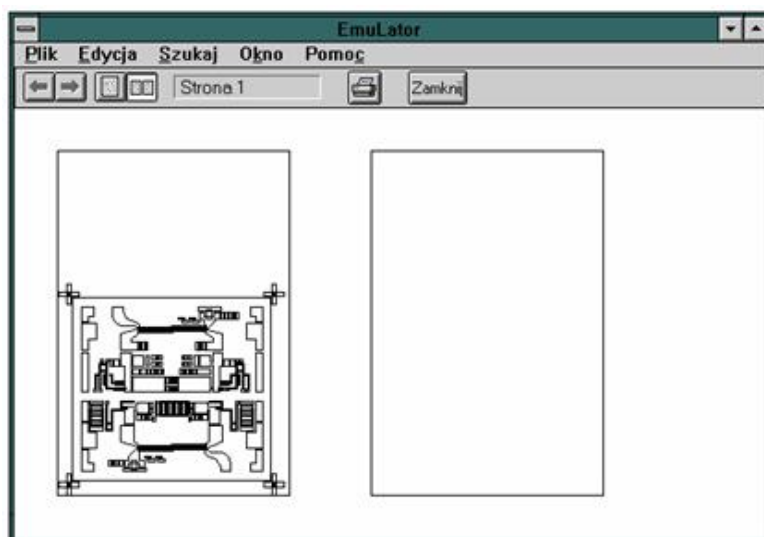
Chcete-li dokument zobrazit tak, jak bude vypadat na stránce po vytištění, vyberte možnost

Náhled *tisku* z

nabídky Soubor nebo příslušnou ikonu na panelu nástrojů. Obraz okna nového dokumentu zobrazuje jednu nebo dvě stránky tiskového souboru. Chcete-li procházet další (předchozí) stránky, klepněte na ikony šipek na panelu nástrojů (obr. 11).

Tisk dokumentu

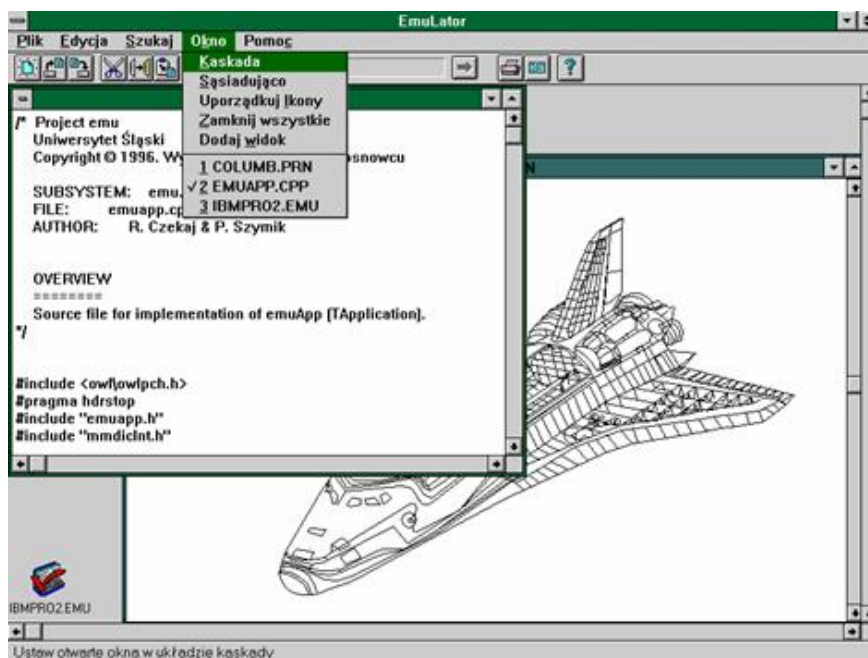
Jakmile dokument zkontrolujete, můžete jej vytisknout na zařízení připojeném k pracovní stanici. Tisk se provádí výběrem možnosti *Tisk* z nabídky *Soubor* nebo kliknutím na příslušnou ikonu na panelu nástrojů. Tisk bude proveden pomocí aktuálního nastavení tiskárny, které lze změnit ve volbě *Nastavení tiskárny* v nabídce *Soubor* .



Obr K Náhled tisku

Úprava
ovladače
tiskárny

Ovladače tiskárny lze upravit tak, aby vyhovovaly používanému zařízení. Musí být načteny v dialogu *Otevřít* s vybraným typem souboru *Všechny soubory (*.*)*. Operace úprav jsou povoleny prostřednictvím možností nabídky *Úpravy*, *Hledat*, *Okno* . Během úprav nelze ovladač načíst.



obr. L Úprava textových souborů

Získání
pomoci

Systém
nápovědy pro
aplikaci

Emulátor lze
vyvolat z
nabídky

Nápověda .

Umožňuje vám
získat pomoc
na základě

klíčového
slova nebo
tématu

vyhledávání.

Když stisknete

Shift+F1, ukazatel myši změní tvar, což vám umožní ukázat na ovládací prvek, o kterém potřebujete informace.

Popis ovladače

Formát souboru ovladače tiskárny byl vytvořen pro aplikaci Emulátor pomocí jednoduché konvence zápisu. Na rozdíl od jiných programů používá EmuLator ovladače, které nejsou kompilovatelné programy. Jsou uloženy ve formátu textového souboru, který nevyžaduje žádné další dodatečné operace. Program má zabudovanou kontrolu správnosti zápisu a při zjištění nesprávné hodnoty je uživatel informován, na jakém řádku k ní došlo. Počet uložených řídicích kódů tiskárny je v podstatě neomezený, stejně jako počet možných počátečních nastavení tiskárny nebo plotru.

Každý řádek začínající znakem w: ! , # , \$, % , modul pro čtení kódu řidiče považuje za komentář. Každý řádek začíná znakem ! je čtena inicializačním modulem nastavení tiskárny a každý řádek musí začínat znakem komentáře nebo obsahovat platná data. Od okamžiku, kdy je nalezen znak komentáře, program ignoruje následující znaky v proudu, dokud nenarazí na znak nového řádku.

První řádek ovladače je povinný, druhý bez komentáře je název emulovaného zařízení. Hlásí se ve stavovém řádku programu.

```
Soubor ovladače EMU
$ Soubor ovladače plotru:
Plotr HP-GL 2
$ pro program Emulátor
```

Popis příkazu je zkopírován obslužným programem proudu jako celý řádek.

Následující fragment obsahuje data pro jeden řídicí kód tiskárny.

```
# Příklad, jak popsat daný kód plotru:
2 2 bajty kódu ke čtení
0x2b 0xff řídicí kód zařízení
popis příkazu kódové slovo popis
0xffff programový kód emulátoru pro operaci (hexadecimálně od 0x0 do
# 0xffff)
```

Čísla lze zadávat libovolně podle konvencí jazyka C:

```
22. prosince
```

```
0x22 - HEX
```

```
022 - říjen
```

oddělené mezerou, např. 0x56 0x53

Zakázky jsou rozděleny do skupin. Rozdělení je zavedeno pouze za účelem zvýšení přehlednosti záznamu ovladače.

```
#####
# PŘÍKAZ KÓDŮ JAZYKA HP-GL2 #
# SKUPINOVÝ PŘÍKAZ V "0A" #
#####
```

Vlastní popis kódu příkazu následuje na následujících řádcích souboru ovladače. První parametr udává počet bajtů v kódu ovladače ke čtení. Další řádek obsahuje skutečné řídicí kódy zařízení. Za popisným řádkem je jedinečné číslo grafického provozu aplikace Emulátor . V níže uvedeném případě se jedná o jednobajtový návratový kód vozíku 0D, umístěný ve skupině 0A a označený číslem 11.

```
1
0x0D
návrat vozíku
0x0A11
```

Popis jednoho řídicího kódu tiskárny musí být oddělen od dalšího znakem komentáře, např.:

```
1
0x0A
LF line feed
0x0A01
#
2
0x56 0x53
volba rychlosti kreslení
0x131B
```

Seznam všech dostupných grafických operací s odpovídajícími čísly se nachází v adresáři ovladače tiskárny v souboru *emu_nrop.txt* .

Po popisu všech potřebných řídicích kódů tiskárny nebo plotru následuje popis výchozího nastavení emulovaného zařízení. První položkou jednoho bloku popisujícího jedno z nastavení je klíčové slovo, další jsou jeho hodnoty.

```
#!!!!!!!!!!!!!!!!!!!!!!!!!!!! VÝCHOZÍ NASTAVENÍ TISKÁRNY!!!!!!!!!!!!!!!!!!!!
# První řádek je typ nastavení, druhý je hodnota
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Typ zařízení
!Zařízení
#!0 # neznámý typ
#!1 # jehličková tiskárna
#!2 # inkoustová tiskárna
#!3 # laserová tiskárna
```

```
!10 # plotr
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Jednotka použitá dále
# mm
# palce palce
!Jednotka
!mm
#
```

Řádky, které nejsou zakomentovány, inicializují své hodnoty na výchozí hodnoty. Jsou změněny, když se v tiskovém souboru objeví řídicí kód, který upravuje danou hodnotu.

Shrnutí

Aplikace EmuLator byla vytvořena s cílem poskytnout univerzální řešení problému interpretace souborů ve výtisku. Je obsažen v instalační verzi na disketách připojených k popisné části díla. Program obsahuje sadu základních grafických funkcí, které umožňují interpretaci souborů vytištěných jehličkovou tiskárnou IBM ProPrinter a plotrem HPGL.

Tiskový soubor může obsahovat jednoduchá textová data a grafický obrázek. Program analyzuje tiskový soubor a po nalezení řídicího kódu zavolá příslušnou grafickou funkci, prezentující efekt operace na monitoru počítače. Hledání řídicích kódů v souboru zahrnuje porovnávání znaků načtených z proudu tiskového souboru se známými kódy uloženými v textovém souboru ovladače. Po rozpoznání řídicího kódu jsou další datové bajty považovány za argumenty pro volání grafické operace. Nerozpoznaná sekvence znaků je považována za jednoduchý text ASCII zobrazený na obrazovce. Soubory uložené v textovém formátu jsou také prezentovány tímto způsobem.

Program lze snadno upravit pro emulaci jiných typů zařízení úpravou textových souborů v ovladači tiskárny v rámci vestavěných grafických funkcí. Je možné rychle vytvořit další objekty, které provádějí další funkce tiskárny nebo plotru. To však vyžaduje kompletní rekompilaci programu.

V popsané podobě lze program upravit pro čtení libovolného neznámého formátu tiskového souboru, ale i dalších souborů obsahujících jakákoli data, která lze zobrazit na obrazovce monitoru (např. vektorová a bitmapová grafika, textový editor atd.).

Nutno přiznat, že univerzálnost vytvořeného programu nijak pozitivně neovlivnila rychlost analýzy. Kontrolní kódy se hledají počínaje nejdelším slovem ve slovníku. Tato metoda umožňuje sjednotit podporu různých formátů ovládání, tj. je možné číst jak ovládací jazyky v textovém formátu (např. PostScript), tak i řídicí sekvence začínající znakem ESC (např. PCL a ESC/P). U souborů obsahujících převážně znaky ASCII se však rychlost analýzy snižuje s rostoucí délkou řídicího slova. Program je možné upravit nahrazením kontejnerové třídy Borland (od níž je odvozen *emuSlovník*) rychlejší datovou strukturou s efektivnější metodou vyhledávání. Dalším řešením je použití různých metod analýzy souborů pro každý formát. To však vyžaduje, aby programátor napsal pro každý z nich samostatný modul, podobně jako tradiční kompilované ovladače a importní filtry.

Program je možné vylepšit obohacením o mechanismy pro výměnu dat pomocí Schránky, případně i OLE 2. Přidáním vhodných funkcí do tříd odvozených z *GraphicalObject*, může být dokonce možné ovládat a vylepšovat jednotlivé objekty, např. škálování pro bitmapy, libovolné transformace pro vektorové kresby nebo změna typu či velikosti písma.

EmuLator umožňuje tisknout zobrazené soubory po jejich interpretaci. Data lze tisknout na jakékoli tiskárně dostupné v prostředí Windows. Díky hardwarové nezávislosti kontextu zařízení GDI je program schopen provádět implicitní převody libovolného formátu, který je mu znám, způsobem, který mu umožňuje přenášet výsledky do jakéhokoli zařízení dostupného systému Windows.

S přihlédnutím k výše uvedeným úvahám a postřehům funkčním z konstrukce a provozu aplikace

Emulátor se konstruktéři domnívají, že cíl a předpoklady této práce byly plně splněny.

Literatura

- [1] Barkakati N.: *Grafika a animace ve Windows*. Varšava, Intersoftland 1994, (překlad z angličtiny).
- [2] Barteczko K.: *Objektově orientované programování; Praktický úvod do objektově orientovaného programování v C++*. Varšava, LUPUS 1993
- [3] Drozdewicz P.: *Programování pro Windows v C*. Varšavě, Lynx-SFT 1994.
- [4] Faison T.: *Objektově orientované programování Borland C++ 4.5*. Varšava, READ ME Publishing House 1996, (překlad z angličtiny).
- [5] Klein M.: *Průvodce knihovnamí DLL a správou paměti*. Varšava, Intersoftland 1994, (překlad z angličtiny).
- [6] Levine J.: *Programování grafických souborů v C/C++*. Warszawa, Překladatel 1994, (překlad z angličtiny).
- [7] Marciniak A.: *Język PCL*. Poznaň, NAKOM 1992.
- [8] McCoy BC: *Souhrn – domovská stránka často kladených dotazů k tiskárnám*. Usenet, comp.periph.s.printers Seznam často kladených otázek (FAQ), 06.08.1996
- [9] Osiak S.: *PostScript krok za krokem*. Varšava, M&M Publishing Agency 1991.
- [10] Smith NE: *Laserové tiskárny*. Varšava, ZNI MIKOM 1995, (překlad z angličtiny).
- [11] Sowiski R.: *Hvězda plotru*. CADmania č. 5 (11), listopad 1995
- [12] Wacławek R.: *Programová podpora pro laserové tiskárny*. Varšava, HELP Computer Publishing House 1992.
- [13] Wacławek R.: *Okna z kuchyně*. Varšava, HELP Computer Publishing House 1993.
- [14] Zalewski A.: *Programování v jazycích C a C++ pomocí balíku Borland C++*. Poznaň, NAKOM 1995.
- [15] *Jehličková tiskárna D-100MPC*. Bonie, Mera-Bonie Mechanical and Precision Plant 1991.
- [16] *Uživatelská příručka tiskárny LC-20*. Varšava, Intersoftland 1991.

Abstraktní

Tento článek představuje technické aspekty provozu moderních tiskáren a plotrů a také způsoby ovládání těchto zařízení pomocí jazyků pro řízení tisku. Popsaný počítačový program je výsledkem analýzy jednotlivých formátů v řídicích jazycích.

Aplikace EmuLator umožňuje emulovat provoz vybrané tiskárny a plotru a umožňuje grafické zobrazení tiskového souboru na obrazovce počítače. Program byl vyvinut v grafickém prostředí Windows 3.11 a programovací sadě Borland C++ , verze 4.52.

Program EmuLator obsahuje sadu základních grafických funkcí, které umožňují interpretaci souborů vytisknutých z jehličkové tiskárny IBM ProPrinter a plotru HPGL . V rámci těchto funkcí může každý uživatel snadno připravit aplikaci pro analýzu tiskových souborů pro jiná zařízení úpravou textových souborů v ovladači tiskárny.

Podrobná dokumentace a komentáře ve zdrojových souborech programu umožňují rozšíření programu o nové grafické funkce pro další zařízení. Objektově orientované vlastnosti jazyka C++ umožňují rozšířit aplikaci o funkce, které její tvůrci nezamýšleli.

Program EmuLator je spolu se systémem nápovědy dostupný na disketách v instalační verzi.

Obsah

ÚVOD	3
TECHNICKÉ ASPEKTY PROVOZU TISKÁRNY A PLOTRU VE 4	
1. KLASIFIKACE TISKÁREN	4
1.1. Jehličkové tiskárny (nárazové tiskárny)	4
1.2. Tiskárny a psací stroje Daisywheel	5
1.3. Inkoustové tiskárny	5
1.4. Laserové tiskárny a LED tiskárny	6
1.5. Barevné tiskárny	7
1.6. Jiné typy tiskáren	9
2. KLASIFIKACE PLOTU V	10
2.1. Deskové plotry	10
2.2. Perové plotry (bubnové plotry)	10
2.3. Inkoustové plotry	11
3. JAZYKY PRO OVLÁDÁNÍ TISKÁRNY A PLOTU	12
3.1. Rozšířené textové formáty	13
3.2. Jazyky popisu stránky	14
3.3. Další jazyky ovládání tiskárny	16
3.4. Jazyky ovládání plotru	18
PŘEKLADATELSKÉ SYSTÉMY	19
1. PŘEVOD MEZI TYPY SOUBORŮ	19
1.1. Bitmapa na bitmapu	19
1.2. Z vektoru na vektorový formát	19
1.3. Z vektoru na bitmapový formát	20
1.4. Bitmapa do vektorového formátu	20
2. KONVERZNÍ PROGRAMY	20
2.1. Nahržení PostScriptu jinými standardy	20
2.2. Změna jiných standardů na PostScript	20
APLIKACE EMULÁTORU	22
1. POPIS PROGRAMU	22
1.1. Konstrukce	23
1.2. Rozšíření	30
SHRNUTÍ	35
LITERATURA	37

SHRNUTÍ	38
OBSAH	39