

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

Fakulta strojní

Katedra technické kybernetiky Školní rok: 1991/92

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Tomáše CEE
obor 23-40-8 ASŘ výrobních procesů ve strojírenství

Vedoucí katedry Vám ve smyslu zákona č. 172/1990 Sb. o vysokých školách určuje tuto diplomovou práci:

Název tématu:

Programová podpora práce konstrukčního oddělení

Zásady pro vypracování:

- 1) Seznámení se s problematikou, výběr činností vhodných pro počítačové zpracování
- 2) Algoritmizace vybrané činnosti a návrh vhodného prostředí kompatibilního s používanými aplikacemi v k.p. Elektropraga
- 3) Realizace návrhu pro k.p. Elektropraga

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Ostř. technická fakulta
260 02 LIBEREC

KTK / ASŘ - S

✓ 201 / R - S

Rozsah grafických prací:

Rozsah průvodní zprávy: 40 - 50 stran

Seznam odborné literatury:

- /1/ Urban, J.: Počítačová grafika v projektování. Skripta ČVUT, Praha 1984
- /2/ Sborníky: Konstruování s využitím počítače systému CAD. ČSVTS Bratislava 1987
- /3/ Drs, L.: Počítačová grafika. Skripta, ČVUT Praha 1985
- /4/ Enderle, G.: Computer Graphics programming. Springer Verlag, Berlin 1984
- /5/ Dokumentace k PC aplikacím Autocad, Pascal atd.

Vedoucí diplomové práce: RNDr. Klára Císařová

Konzultant: Ing. Jindřich Doležal, Elektropraga
Jablonec n.N.

Zadání diplomové práce: 31.10.1991

Termín odevzdání diplomové práce: 29.5.1992

L.S.

Vedoucí katedry

Doc. Ing. Vladimír Věchet, CSc.

Děkan

Prof. Ing. Jaroslav Exner, CSc.

V Liberci

dne 22.10. 1991

Vysoká škola strojní a textilní v Liberci

Fakulta strojní

Obor 23-40-8

Automatizované systémy řízení výrobních procesů
ve strojírenství

Katedra technické kybernetiky

Programová podpora práce konstrukčního oddělení

Tomáš Ceé

KTK ASŘ FS - 170 78

Vedoucí práce : RNDr. Klára Císařová - VŠST Liberec

Rozsah práce a příloh

počet stran : 44

počet obrázků : 3

počet příloh : 4

UNIVERZITNÍ KNIHOVNA
TECHNICKÉ UNIVERZITY V LIBERCI



3146076881

datum : 29.5.1992

Místopřísežně prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury.

v Liberci dne 29.5.1992 Tomáš Ceé



Tímto chci poděkovat své vedoucí diplomové práce
RNDr. Kláře Císařové za pomoc při jejím vypracování a za cen-
né připomínky týkající se obsahu diplomové práce.

Obsah :	Strana :
1. Úvod	6
2. Grafické systémy	8
2.1. Základní vlastnosti a funkce	8
2.2. Praktické aplikace	10
3. Řešení práce	16
3.1. Důvody zadání	16
3.2. Popis problému a jeho teoretické řešení	17
4. Programový systém pro výpočet rozměrů odlévacích forem	22
4.1. Základní informace	22
4.2. Práce s programem	23
4.3. Seznam chybových hlášení programu	30
4.4. Programové řešení	31
5. Závěr	40
6. Seznam použité literatury	43
7. Seznam příloh	44

1. Úvod

V posledních letech zaznamenal rozvoj výpočetní techniky zásah prakticky do všech oborů lidské činnosti. Výpočetní technika pomáhá téměř všude : od jednoduchých kancelářských aplikací, přes účetnictví až po složité řídicí a expertní systémy. V největší míře se ovšem počítače prosazují v těch odvětvích a oborech lidské činnosti, v nichž se vyskytuje přemíra rutinních, často se opakujících operací a nebo je potřebná předběžná znalost řešení, které nelze v praxi jednoduše uskutečnit buď z důvodů rizika nebo vysokých nákladů na zkoušky. V posledně jmenovaných případech se nejvíce používá tzv. *počítačové simulace*, která se zabývá právě řešením a modelováním skutečných stavů a jevů pomocí počítače.

Největšího rozmachu však asi ale doznala oblast AIP (*Automatizace inženýrských prací*), která se zabývá zaváděním a využíváním výpočetní techniky v "inženýrské praxi". V této oblasti jsou asi nejpoužívanějšími tzv. CAD (*Computer Aided Design*) a další CA. systémy (*CAE - Computer Aided Engineering* , *CAM - Computer Aided Manufacturing* , *CIM - Computer Integrated Manufacturing*). Za své rozšíření vděčí tyto systémy zejména neustále klesajícím cenám hardware. To umožnilo konstrukci systémů profesionální výkonnosti na úrovni počítačů třídy PC, které ještě donedávna požadavkům těchto systémů plně nevyhovovaly. Tyto nově vzniklé systémy lze označit jako *systémy střední výkonnosti* (na rozdíl od systémů budovaných s řídicími minipočítači např. typu VAX apod.). Jednoduché grafické systémy pronikly i do úrovně *home computer*. Kromě výše uvedených jsou ještě používány systémy vysoké výkonnosti. Jádro těchto systémů tvoří *grafická stanice* (*graphics*

workstation), která je vybavena speciálním grafickým procesorem, displejem s vysokou rozlišovací schopností a komunikuje s nadřazeným počítačem (VAX, CRAY apod.).

2. Grafické systémy

2.1. Základní vlastnosti a funkce

Zpravidla všechny systémy a programy pro AIP patří do oblasti *interaktivní počítačové grafiky*, která poskytuje velmi názorné a ve většině inženýrských aplikací jediné možné vyjádření konkrétního problému, které je i přirozenější, než komunikace v alfanumerickém režimu.

Na nejvyšší úrovni systému je zpravidla některý vyšší programovací jazyk, sloužící k realizaci uživatelských programů. Může být kompilační (FORTRAN-77, Pascal, C) nebo interpretační (Lisp, APL). S tímto jazykem jsou svázány jazyk pro práci s datovou bází a jazyk grafického programování. Jazyk grafického programování produkuje již příkazy pro *display procesor* (*procesor řídící vlastní grafický výstup*). Celý grafický systém může být ovšem přímo napsán ve svém programovacím jazyce vysoké úrovně (Lisp, C apod.).

Základní struktura pro tvorbu software je zachycena v různých normách. Většina systémů vychází z mezinárodní normy *ISO-GKS* (*Graphics Kernel System*). Zde jsou definovány základní struktury a požadavky zpracování grafické informace. Z této normy vyplývá, že stanice by měla umožňovat tyto základní operace :

- segmenace obrazu (segments, blocks, arrays)
- výběr, manipulace, zvětšování částí zobrazení (windowing, zooming)
- vyplňování rovinných obrazců, šrafování (fill, hatch)
- transformace obrazců v rovině (rotate, move, mirror)

- přidání, mazání, duplikace, rozdělení obrazců
- zpracování 3D scény (projekce, transformace objektů, řešení viditelnosti scény)

Z toho jasně vyplývá nutnost zavedení určitých struktur, které jasně a přehledně definují vlastnosti grafických objektů :

Metasoubor (Metafile, drawing file) - soubor, ve kterém jsou ukládány informace o zpracování obrazu. Soubor je v podstatě tvořen instrukcemi pro grafický procesor (resp. pre-procesor). Regenerace obrazu probíhá snímáním metasouboru resp. jeho části podle užitého zobrazení.

Grafický prvek (Graphic primitive, display element)

- libovolná část obrazu, která je realizována v grafickém procesoru na hardware úrovni (např. bod, úsečka - vektor, alfanumerické znaky, kružnice, části křivek, části ploch - pro 3D aplikace).

Přirozeně ne každá stanice má tyto prvky řešeny na hardware úrovni.

Grafický objekt (object, display group, segment) - je tvořen množinou prvků, které mají stejné vizuální vlastnosti - atributy zobrazení, např. lomená čára, krychle. Mohou být v systému identifikovány jedním jménem. Grafický objekt je základním prvkem metasouboru, se kterým lze provádět různé operace.

Symbol - identifikovaný objekt v metasouboru (resp. programový segment) pro generování některé množiny grafických prvků. Atributy definující status a vizuální vlastnosti nejsou definovány. Volání symbolu způsobí vykreslení kopie symbolu s atributy volajícího objektu.

Atributy - každý grafický prvek (objekt) může mít řadu atributů, které definují jeho vizuální vlastnosti, např. barvu, souřadnicovou transformaci, velikost textu, orientaci, typ textu nebo čáry, intenzitu apod.

Vrstvy, hladiny, roviny obrazu (Layers) - samotný metasoubor může být rozdělen na podsoubory, které obsahují v podstatě nezávislé obrazy. Podle specifikace zobrazení pracujeme v dané hladině a na displayi jsou zobrazeny jen definované hladiny.

2.2. Praktické aplikace

Na začátku této kapitoly si musíme uvědomit, že většina systémů, budovaných na bázi počítačů třídy PC patří k systémům CAD střední výkonnosti. Je možné na nich řešit úlohy středního rozsahu a obtížnosti. Výkonné grafické stanice jsou dnes budovány na mikropočítačích s 32-bitovou architekturou a pracují s nadřazeným minipočítačem (např. typu VAX) nebo s velkým sálovým počítačem (*mainframe*). Pro řešení větších projektů je třeba zvolit vhodné technické a programové vyba-

vení, které ve třídě osobních počítačů nemusíme nalézt.

Na druhé straně poskytují počítače třídy PC uživateli možnost využití většiny grafických funkcí ve smyslu normy GKS při podstatně nižších pořizovacích nákladech jak za hardware tak i za software.

Systemy, pracující s PC zpravidla předpokládají provozování na pracovišti, které je vybaveno :

- počítačem IBM PC-XT a vyšším (případně kompatibilním)
- pevným diskem o minimální kapacitě 10 MB
- grafickou kartou s alespoň střední, rozlišovací schopností (Hercules, EGA, VGA, SVGA , ATT) s možností připojení grafického monitoru (dual monitor) s vysokou rozlišovací schopností
- digitalizačním tabletem nebo myší
- tiskárnou, plotterem

Jako příkladu bych uvedl některé z těchto systémů s jejich stručným popisem :

Verse CAD Advanced

implementace : IBM PC (a kompatibilní), HP Serie 2000,

Olivetti M24

System je řízen pomocí nabídkových menu funkcí. Volba příslušné funkce se provádí z klávesnice, digitalizačního tabletu nebo pomocí myši. V knihovně jsou implementovány základní grafické prvky : úsečky, obdélníky, kružnice atd. Tuto knihovnu je ovšem možno rozšiřovat o uživatelské grafické prvky. System umožňuje zpracovávat pro jeden obraz až 250

hladin. V nejnižší verzi obsahuje všechny základní funkce pro práci s objekty a zobrazeními. Rozšiřující modul *Versapro-3D* je určen pro zpracování 3D scén. Mezi základními grafickými prvky má implementovány např. kvadratické nadplochy aj. Umožňuje řešení viditelnosti scény a neviditelné hrany nejsou vykreslovány.

Personal Architekt

implementace : IBM PC (a kompatibilní)

System sestává ze dvou modulů (Design-modul a Drafting-modul). Je určen, jak již název napovídá pro stavebnictví a architekturu, ale lze ho použít i v jiných oblastech, neboť obsahuje všechny základní funkce CAD systémů. Práce je orientována na 3D prostor. Nejprve se pomocí Design-modulu vytvoří prostorový model (lze ho zadávat pomocí XYZ- souřadnic jednotlivých bodů hran a ploch nebo digitalizací např. půdorysu, bokorysu nebo obecného řezu z tabletu). Po zadání modelu lze provádět všechny běžné operace : měnit bod pohledu, projekce (axonometrie, perspektiva), zpracovávat jednotlivé řezy, dopracovávat detaily apod. Každý obraz je možné zpracovávat až v 256 hladinách. Knihovna grafických prvků obsahuje řadu symbolů z oblasti stavebnictví, což značně urychluje práci. K dispozici jsou také různé kvadratické nadplochy, prostorová interpolace aj.

Drafting-modul slouží k vlastnímu zobrazování. Umožňuje vykreslování různých typů čar, šrafování, interpolaci, mazání, přesouvání, zrcadlení, kopírování a další operace s objekty.

Cadvance

Implementace : IBM PC (a kompatibilní)

System obsahuje základní grafické prvky. Umožňuje šrafování, windowing, mazání objektů a jejich částí, posuny, rotace, kopírování a další základní operace s objekty. Lze zpracovávat až 127 hladin. Obsahuje vlastní databázový podsystém pro vytváření a uchovávání uživatelských grafických prvků a objektů. Jako rozšíření je nabízen 3D modul, pracující s prostorovými objekty.

Modelmaker-300

Implementace : IBM PC (a kompatibilní)

System umožňuje vytváření 3D modelů pomocí digitalizačního tabletu, zpracování různých druhů projekcí těchto modelů a řešení viditelnosti. Objekty lze zadat jako pevná (neprůhledná) tělesa s různě barevnými stěnami. Pro zpracovávanou scénu může být modelováno různé osvětlení objektů. Použití *hidden surface* algoritmů umožňuje řešit viditelnost i pro scény s pevnými tělesy (na rozdíl od běžných *hidden lines* algoritmů, které řeší pouze viditelnost hran). Součástí systému je např. výpočet nadploch (prostorová interpolace - "záplatování"), výpočet těžišť, objemů apod.

Cadstar

Implementace : IBM PC (a kompatibilní)

System je určen ke konstrukci 3D objektů, které je možné zadávat pomocí základních 2D pohledů. Uživatel si může definovat vlastní pohledy v perspektivě se zadaným osvětlením.

Umožňuje rovněž výpočet interpolací, vzdáleností, délek, těžišť, momentů apod.

MaxCAD

Implementace : Apple MacIntosh, MacIntosh Plus

Je to hierarchický objektově orientovaný 3D systém. Konstrukce jsou ohraničeny pouze velikostí paměti počítače. V knihovně mohou být uloženy již zkonstruované objekty (např. dům, nábytek). S objekty lze provádět všechny základní 3D operace. Je možné měnit bod pohledu, úhel pohledu, zorný úhel atd. Rozšiřující modul umožňuje tzv. *solid modelling* - tzn. zobrazování těles s různě barevnými povrchy a s různým osvětlením.

AutoCAD

Implementace : IBM PC

Systém je určen pro řešení řady problémů z oblasti CAD. Uživatelé nabízí všechny základní funkce, ale mimo to umožňuje tvorbu aplikačních uživatelských systémů s pomocí dalších programových prostředků. To umožňuje vytvářet integrované systémy pro řešení úloh z různých oblastí, které mohou být využívány i odborníky s minimálními znalostmi systému AutoCAD. Je sestaven z několika modulů, *Advanced Drafting Extensions ADE-1,2,3* a *AutoLISP*. AutoLisp je interaktivní programovací jazyk zabudovaný do systému AutoCAD, který umožňuje např. vytváření externích příkazů systému nebo přístup k prvkům databáze grafických prvků.

Podporuje také tvorbu uživatelských aplikačních systémů využívajících např. různé matematické modely. Implementací

vyššího programovacího jazyka do systému AutoCAD je možné spojit grafické programové vybavení s řešením úloh z různých technických oblastí v integrované systémy umožňující ovládnutí i odborníkům, kteří se profesionálně nezabývají výpočetní technikou.

3. Řešení práce

3.1. Důvody zadání

V konstrukci strojírenských zařízení se dost často vyskytnou úkony a operace, které by bylo možno zautomatizovat pomocí výpočetní techniky, avšak jejich řešení za pomoci standardních aplikačních systémů je nevyhovující a někdy také neuskutečnitelné. Toto byl také hlavní důvod k zadání této práce.

Konstrukční oddělení podniku Elektro Praga Jablonec n/N potřebovalo zefektivnit svou práci a vybralo několik činností, které by bylo možné pomocí počítače velmi jednoduše a rychle řešit. Jednou z těchto operací je i výpočet velikosti odlévacích forem v závislosti na smrštění materiálu, použitého pro odlévání. Při pokusech řešit tuto úlohu pomocí systému AutoCAD nastaly však problémy se zadáváním tolerancí, jejich uchováváním a případnou změnou, s korigováním výsledku výpočtu atd. Proto bylo rozhodnuto, že celý programový produkt bude napsán v nějakém vyšším programovacím jazyce. Já jsem si zvolil Turbo Pascal v6.0, protože mám v jeho používání již nějakou praxi a také tento jazyk, podle mého mínění, obsahuje jedny z nejlepších ladících prostředků, které jsou při tvorbě složitějších programů nutností.

3.2. Popis problému a jeho teoretické řešení

Použité veličiny a jejich zkratky :

TR - typ rozměru [díra, hřídel, rozteč]

RV - rozměr výlisku [mm]

HTV - horní tolerance výlisku [mm]

DTV - dolní tolerance výlisku [mm]

HMRV - horní mezní rozměr výlisku [mm]

DMRV - dolní mezní rozměr výlisku [mm]

RF - rozměr formy [mm]

HTF - horní tolerance formy [mm]

DTF - dolní tolerance formy [mm]

HMRF - horní mezní rozměr formy [mm]

DMRF - dolní mezní rozměr formy [mm]

DT - dílenská tolerance [*HTF*, *DTF*]

S - smrštění [%]

PKS - pomocná konstanta smrštění

Každý materiál pro odlévání je kromě svých dalších vlastností také charakterizován koeficientem smrštění, který udává v procentech míru smrštění materiálu při chladnutí. Celý princip výpočtu pak v podstatě spočívá ve vynásobení rozměru vylisku tzv. pomocnou konstantou smrštění (*PKS*), a upravení vypočteného rozměru do tolerančního pole formy podle typu rozměru :

- u typu hřídele se snažíme rozměr vylisku s dílenskou tolerancí formy (*DT*) posunout až těsně nad dolní mezní rozměr vylisku (*DMRV*).

- u díry se posouvá rozměr vylisku s dílenskou tolerancí formy (*DT*) těsně pod horní mezní rozměr vylisku (*HMRV*).

- u typu rozteč zůstává rozměr vylisku s dílenskou tolerancí formy (*DT*) na původní vypočtené hodnotě (tj. zpravidla ve středu tolerančního pole vylisku.

Počáteční zadání hodnot probíhá u všech typů stejně, zadání horní a dolní tolerance vylisku (*HTV*, *DTV*) se uskutečňuje v mezních úchylnkách nebo jako netolerovaný rozměr. Dílenská tolerance (*DT*) se zadává u typů díry a hřídele buď v normalizovaných tolerančních polích nebo v mezních úchylnkách a u typu rozteč jen v mezních úchylnkách. Hodnota smrštění se zadává ve čtyřech možných variantách :

- smrštění typu 121 (0.1%)
- smrštění typu 126 (0.5%)
- smrštění typu AG (0.15%)
- volná volba smrštění (zadání libovolné hodnoty v %)

Řešení probíhá dále následovně :

- vypočítá se pomocná konstanta smrštění

$$PKS = 1 + (S / 100)$$

- zjistí se příslušný typ rozměru, protože pro každý typ probíhá další postup trochu jinak :

a) Díra na výlisku (Hřídél ve formě)

- vypočteme horní mezní rozměr výlisku a počáteční rozměr formy

$$HMRV = RV + HTV$$

$$RF = HMRV \cdot PKS$$

- zaokrouhlíme rozměr formy (RF) podle rozměru výlisku (RV) takto : je-li RV menší nebo roven 10mm , pak zaokrouhlujeme na pět setin milimetru, jinak na jednu desetinu milimetru. Obojí směrem dolů.
- provede se kontrola, zda RF / PKS je menší, než $HMRV$. Pokud není, je rozměr formy zmenšen při rozměru výlisku (RV) menším nebo rovném 10mm o 0.05mm, při větším je to o 0.1mm a provádí se opětovná kontrola rozměru.
- vypočte se horní mezní rozměr formy

$$HMRF = RF + HTF$$

- zkontroluje se, zda $HMRF / PKS$ je menší, než $HMRV$. Není-li tomu tak, je rozměr formy zmenšen stejně, jako v předchozím kroku a znovu se provede kontrola.

b) Hřídél na výlisku (Díra ve formě)

- vypočítá se dolní mezní rozměr výlisku a počáteční

rozměr formy

$$DMRV = RV + DTV$$

$$RF = DMRV \cdot PKS$$

- zaokrouhlíme rozměr stejným způsobem, jaku u díry, ale nahoru.
- provedeme kontrolu, zda RF / PKS je větší, než dolní mezní rozměr vylisku ($DMRV$). Pokud není zvětšime rozměr formy (RF) stejným způsobem, jaký byl popsán u díry a opakujeme tuto kontrolu.
- vypočteme dolní mezní rozměr formy

$$DMRF = RF + DTF$$

- zkontrolujeme, zda $DMRF / PKS$ je větší, než dolní mezní rozměr vylisku ($DMRV$). Pokud není, zvýšíme rozměr formy (RF) výše popsaným způsobem a znovu kontrolujeme.

c) Rozteč na vylisku

- přímo vypočteme rozměr formy

$$RF = RV \cdot PKS$$

Výsledným údajem všech výše popsaných výpočtů je potom rozměr formy (RF) tolerovaný dílenskou tolerancí (DT).

Zde se ukázalo jako nutné dát konstruktérovi možnost rozhodnutí, zda vypočtený rozměr formy vyhovuje jeho představám a charakteru použití rozměru (požadavky na nejrůznější uložení apod.), protože ne vždy odpovídají vypočtené hodnoty jeho požadavkům.

Z popisu řešení je dále zřejmé, že tento výpočet můžeme aplikovat jen u rozměrů, které mají charakter díry, hřídele

nebo rozteče a nelze jej tedy využít u komplikovaných výlisků a forem, kde typy rozměrů i jejich tolerování nejsou zdaleka jednoznačné. Přesto se ale v praxi ukázalo, že tento produkt usnadní zdlouhavou rutinní práci při přepočítávání rozměrů pro formy.

4. Programový systém pro výpočet rozměrů odlévacích forem

4.1. Základní informace

Tento interaktivní programový systém byl vytvořen na personální počítači IBM PC-XT/12 s grafickou kartou Hercules a dále pak odladěn na IBM PC-AT 286/16 s grafickými kartami Hercules a SVGA. Jak již bylo předem řečeno, program je napsán v jazyce Turbo Pascal v6.0 pod operačním systémem DR-DOS v5.0.

Jelikož jedním ze základních požadavků byl stejný vnější vzhled jako má AutoCAD, odpadl problém s vymyšlením vhodného prostředí. Veškeré ovládání (výběr z menu, posuv tolerančních polí při korigování výsledků) je orientováno na práci s myší (bylo odskoušeno více typů ovladačů i myší a všude program pracoval korektně), ale bylo pamatováno i na nepřítomnost myši (tento problém je řešen emulací původního kurzoru a jeho ovládaním kurzorovými klávesami).

Programový systém má název TCAD.EXE. Používá ke své činnosti externí textové soubory NETOLER.TCD, TOLER_HR.TCD a TOLER_DI.TCD, které obsahují veškeré tabulkové hodnoty, potřebné pro výpočet (mezí úchytky jednotlivých tolerancí, netolerované rozměry). Jméno pracovního adresáře, ve kterém jsou tyto soubory umístěny, se zadává jako parametr programu při jeho spuštění (Zde je možné s výhodou použít dávkového souboru typu BAT). Nacházejí-li se soubory ve stejném adresáři, jako vlastní EXE soubor, je možno parametr vynechat.

Vzhledem k jednoduchosti obsluhy a s přihlédnutím k tomu, že tento program bude používat jen úzký okruh pracovníků, není, na rozdíl od systému AutoCAD, možnost nápovědy (většina

výběrů a dotazů je vedena ve srozumitelné formě, a v praxi s tím nebyly problémy). Na disku zabírá tento systém 97 kB, což při skutečnosti, že k programu jsou "přilinkovány" ovladače grafických karet Hercules, EGA a VGA, není tak mnoho. Při vlastní práci potřebuje program k dispozici asi 130 kB operační paměti pro uchování jak vlastního kódu programu, tak i pracovních dat.

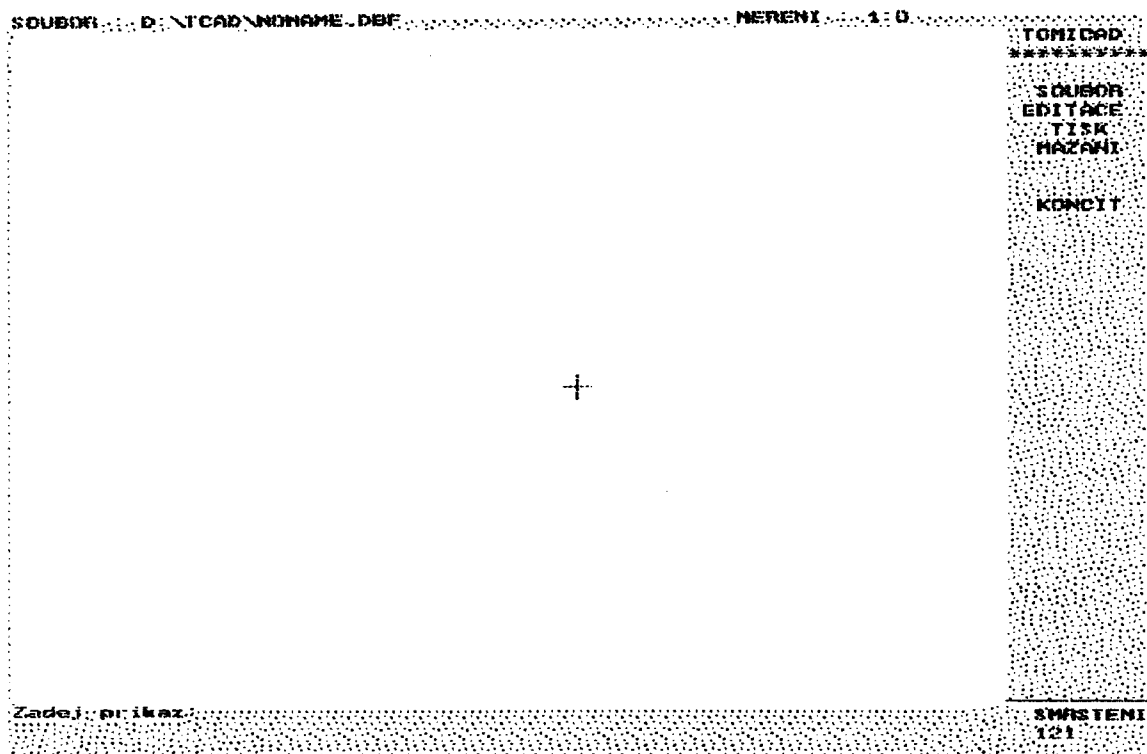
Zdrojový text programu bez pomocných jednotek (unitů) MOUSE.TPU (zajišťuje komunikaci s myší), MANN.TPU (práce s tiskárnou) a LINKBGI.TPU (připojuje k výslednému souboru EXE jednotlivé ovladače BGI a tiskové fonty CHR, přeložené do tvaru OBJ) má velikost 101 kB a obsahuje 2011 řádků zdrojového textu jazyka Turbo Pascal.

4.2. Práce s programem

Jak již bylo výše uvedeno, program zachovává podobné ovládní, jako systém AutoCAD. Z toho vyplývá že všechny příkazy jsou přístupné pomocí myši (ukázáním na příslušnou položku menu a potvrzením) nebo pomocí 3 řádkového příkazového řádku (vypsání příslušné položky menu a potvrzení klávesou *Enter*. Zde se ještě zkušenějšímu uživateli, který již pracuje s tímto systémem delší dobu a zná strukturu jednotlivých menu, nabízí možnost zadávání více za sebou následujících příkazů najednou (např. místo postupného vypsání položek *SOUBOR*, *NAHRAJ* a "jméno souboru" do třech po sobě následujících řádků napíše uživatel jen jeden složený příkaz *SOUBOR NAHRAJ "jméno souboru"*). Při ovládní myši slouží její levé tlačítko pro vybrání položky menu, případné potvrzení hodnoty a pravé

pro výběr nejspodnější položky menu, což bývá zpravidla položka *ZRUŠIT* nebo *KONČIT*. Stejný příkaz generuje i stisk klávesy *Esc* na klávesnici. V případě nepřítomnosti myši je zapnuta emulace 2 tlačítkové myši pomocí kurzorových kláves a tlačítek *Home* a *End* (tato kombinace byla zvolena namísto tradičních kláves *Enter* a *Space* proto, že většina programů kromě komunikace s myší ještě pracuje s příkazovým řádkem, v němž je těchto kláves zapotřebí).

Rozmístění ovládacích a informačních prvků na obrazovce vypadá takto (viz.obr.1) :



obr.1. Vnější vzhled programu

- v levém horním rohu je zobrazováno jméno a cesta k souboru, s nímž právě pracujeme nebo NONAME.DBF, nebyl-li soubor ještě vytvořen.

- napravo od jména pracovního souboru se nachází informace o tomto souboru ve tvaru MĚŘENÍ ep:cp, kde položka cp udává celkový počet měření (výpočtů) uložených do souboru a ep číslo měření, které je právě editováno.

- v pravé části je oblast menu, jehož první položka podtržená hvězdičkami je název menu a další položky slouží již pro vlastní výběr.

- v pravém dolním rohu je zobrazována informace o právě nastavené hodnotě smrštění, se kterým probíhají veškeré výpočty.

- v dolní části obrazovky se nachází příkazový řádek pro zadávání příkazů a hodnot, potřebných pro výpočet.

- zcela uprostřed se nachází plocha pro zobrazení výsledků měření, které se zobrazují ve dvojí formě. V levém horním rohu pracovní plochy jsou vytištěny vždy kontrolní opisy právě zadaných hodnot a pod nimi barevně odlišen výsledný rozměr formy i s příslušnou tolerancí. V pravé části jsou výsledky zobrazovány v grafické formě : v popsaném měřítku jsou vykreslena toleranční pole původního rozměru vylisku, zadaného z výkresu a toleranční pole rozměru vylisku, který bude odlit do vypočtené formy. Tím má uživatel možnost i vizuální kontroly výsledků a v případě nevhodnosti řešení ho opravit nebo zkorigovat.

Seznam příkazů hlavního menu systému je následující :

- *EDITACE*
- *SOUBOR*
- *TISK*
- *MAZÁNÍ*
- *KONČIT*

EDITACE

Touto položku vstupuje uživatel do hlavní části programu.
Následuje výběr editované položky pomocí menu :

- *ZAČÁTEK* (skok na začátek celé databáze měření, používá se i při zahájení práce s novým, prázdným souborem měření)
- *SKOČ <1..X>* (editace bude prováděna na měření zadaného čísla v rozsahu 1..X)
- *KONEC* (skok na poslední, prázdný záznam celé databáze měření)

Pomocí položek dalšího menu je zvoleno smrštění materiálu, s nímž bude celý výpočet probíhat :

- *121* (smrštění o velikosti 1%)
- *126* (smrštění o velikosti 0.5%)
- *AG* (smrštění o velikosti 0.15%)
- *VOLNÉ* (po zvolení této položky následuje ještě dotaz na hodnotu smrštění v procentech)

Poté má uživatel možnost výběru typu rozměru pro výpočet
(jedná se o rozměr vylisku) :

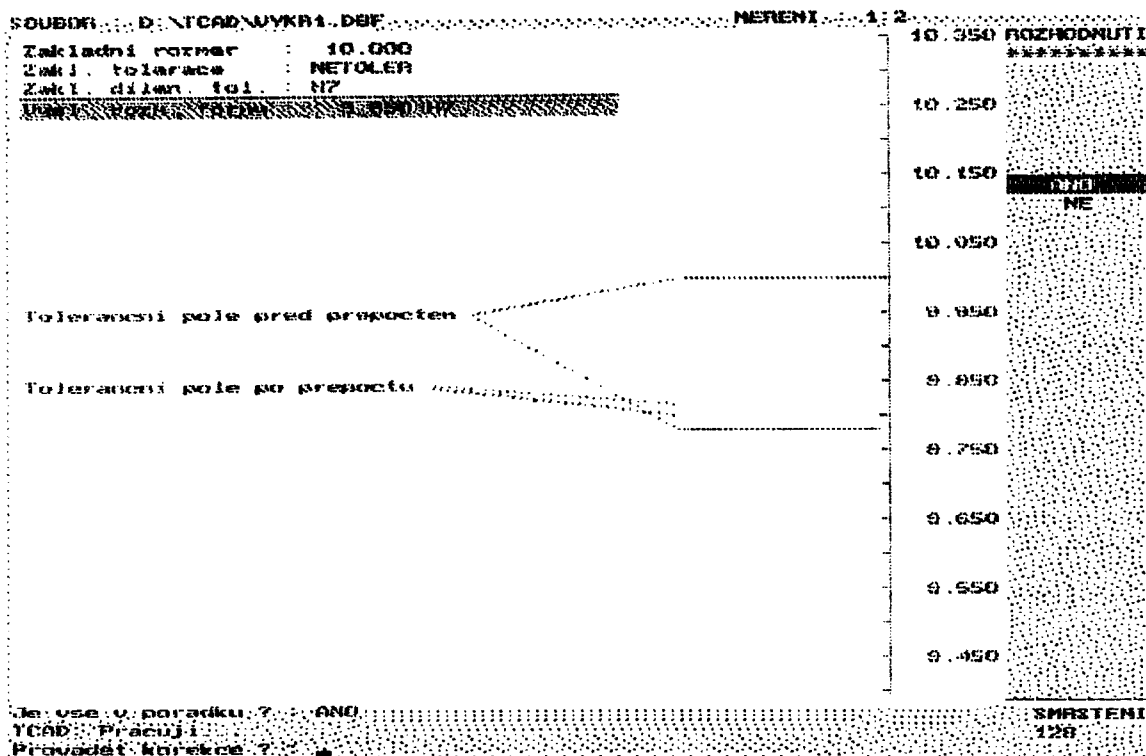
- DÍRA
- HRÍDEL
- ROZTEČ

Následují dotazy na :

- velikost rozměru vylisku
- toleranci rozměru vylisku v mezních úchytkách
nebo vylisek není tolerován
- dílenskou (výrobní) toleranci formy
v mezních úchytkách nebo normalizovaných tole-
rancích

Po potvrzení správnosti zadání (je možnost návratu
do předchozích položek menu a opravy hodnot) proběhne výpo-
čet a jsou zobrazeny výsledky (viz. obr.2). Zde je ponecháno
na uživateli, aby zhodnotil výpočet a buď provedl korekce
(viz. obr.3) (najetím do vypočteného tolerančního pole
(kurzor myši se změnil v šipky), uchopením pomocí levého
tlačítka a tažením zvoleným směrem (nahoru, dolů). Po uvol-
nění tlačítka dojde k opětovnému přepočtení rozměru formy.
Tato korekce neprobíhá spojitě, ale po pěti setinách nebo
jedné desetíně milimetru (podle velikosti rozměru vylis-
kum). Po ukončení korekcí následuje dotaz na provedení dal-
šího výpočtu. Pokud chce uživatel pokračovat ve výpočtu dal-
šího rozměru, odpoví kladně na tento dotaz a začíná editaci
již ne od začátku, ale od typu rozměru (hodnota sarrštění je

pro jeden výkres stejná a jeho případnou změnu může uživatel uskutečnit skokem do přechodního menu stiskem klávesy Esc nebo pravým tlačítkem myši). V případě záporné odpovědi se program navrátí zpět do hlavního menu.



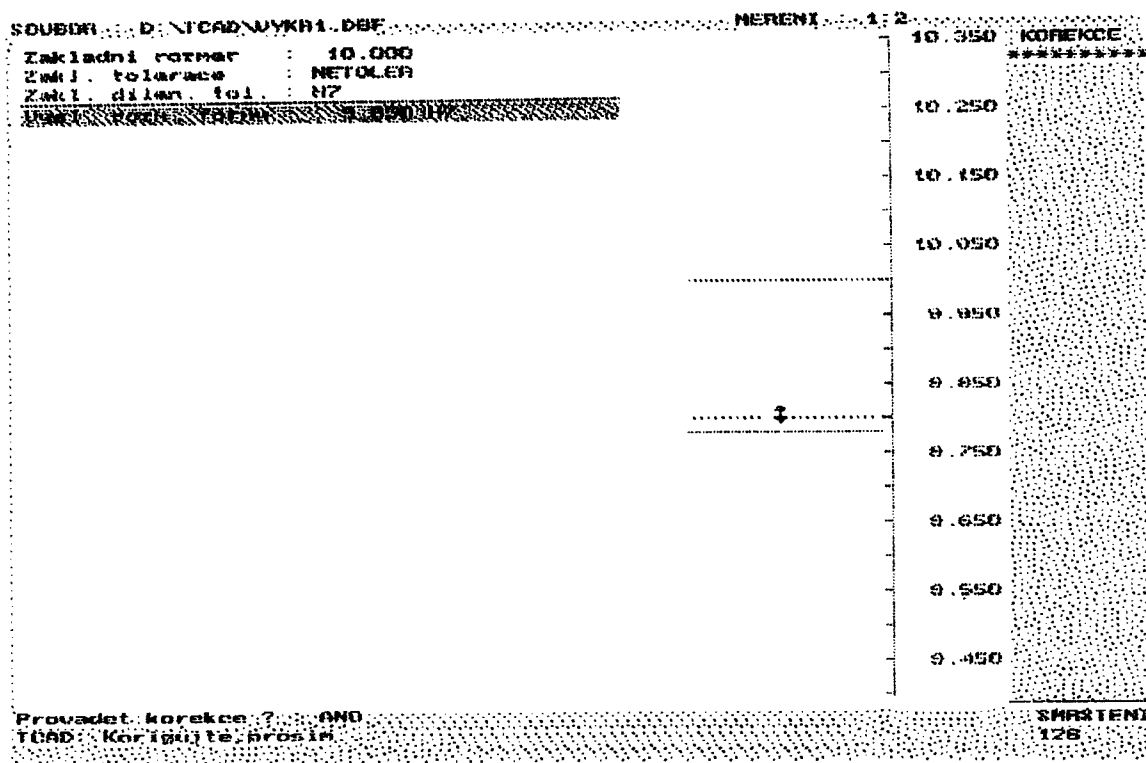
obr.2. Zobrazení výsledků programem

SOUBOR

Umožňuje běžné operace se soubory měření :

- ADRESÁŘ (provádí změnu pracovního adresáře)
- NAHRAJ (nahraje soubor zvoleného jména do systému a umožní jeho prohlédnutí, případně opravu)
- ULOŽ (uchová soubor měření, pokud není

prazdný, do souboru zadaného jména)



obr.3. Korigování výsledků

TISK

Provádí tisk výsledků měření na připojené tiskárně. Poskytuje dvě volby tisku :

- **VŠECHNO** (vytiskne celý soubor výsledků)
- **SKUPINA** (tiskne jenom měření v předem zvoleném rozsahu (části celého souboru))

MAZÁNÍ

Po předchozím dotazu zruší celý soubor měření, který je umístěn v systému.

KONČIT

Provede ukončení činnosti programu a návrat do operačního systému.

Většina činností systému je ošetřena proti chybám, které jsou okamžitě po rozpoznání ohlášeny uživateli ve stavovém řádku.

Program také nepovolí uživateli ukládat na disk soubor s nulovým počtem výpočtů nebo tisk takového souboru.

4.3. Seznam chybových hlášení programu

Chybný příkaz - byl zadán chybný nebo syntakticky špatný příkaz

Chybný parametr - byl zadán špatný parametr (jeho hodnota překračuje povolené meze)

Chybné číslo záznamu - bylo zadáno číslo měření, které je mimo rozsah souboru

Adresář nenalezen - bylo zadáno jméno neexistujícího adresáře

Disk je plný - na disku není postačující volná kapacita pro zápis souboru

Soubor je prázdný - soubor v systému neobsahuje při požadované výstupní operaci (ukládání souboru, tisk na tiskárně) žádnou položku a provedení žádané operace by tudíž nemělo žádný význam

Soubor je plný - byla překročena maximální povolená kapacita systému (příliš velký počet položek)

Chybný soubor - formát souboru neodpovídá standartu nebo je zničený soubor

Tiskárna není OK - při inicializaci tiskárny došlo k chybě (nebyla zapnutá, není papír apod.)

Konec papíru - při tisku výsledků došel papír v tiskárně

Chyba I/O - během tisku došlo k chybě při komunikaci s tiskárnou

4.4. Programové řešení

Program používá ke své práci z prostředků Turbo Pascalu základní knihovnu *SYSTEM* (základní příkazy jazyka), *CRT* (rozšířený ovladač klávesnice a displaye v textovém reži-

mu), *DOS* (umožňuje použití základních služeb *DOSu*) a *GRAPH* (grafická knihovna, podporující nejzákladnější grafické režimy na standartních grafických kartách). Podporuje tři druhy grafických karet : Hercules v režimu 720x348 bodů, EGA a EGAMono v režimu 640x350 bodů a VGA v režimu 640x480 bodů. Tyto tři druhy grafických karet byly zvoleny z důvodu jejich nejčastějšího výskytu v našich poměrech. Kromě těchto základních knihoven jsou v programu ještě používány další jednotky, jejichž popis a funkce v programu následuje :

Jednotka LINKBGI.TPU

Slouží k jednoduchému připojení ovladačů displayů (soubory typu BGI) a tiskových fontů (typ CHR), přeložených programem BINOBJ.EXE do typu OBJ, k výslednému EXE souboru.

Samotná jednotka využívá dalších dvou jednotek BGIDRIV.TPU a BGIFONT.TPU, které byly již dodány s překladačem Turbo Pascalu. Její použití v programu je snadné : ovladače displaye se připojují pomocí procedury *LinkDrivers("seznam")* a tiskové fonty pomocí procedury *LinkFonts("seznam")*, kde *seznam* je řetězec, v němž jsou uvedena jména použitých driverů nebo fontů, oddělených čárkou.

V programu TCAD.PAS použito procedury *LinkDrivers("Herc,EgaVga")*, která zajistí připojení grafických ovladačů karet Hercules, EGA a VGA.

Jednotka MOUSE.TPU

Zajišťuje spolupráci s ovladačem (driverem) myši, komunikujícím přes INT 33H. Při existenci grafických karet CGA, EGA a VGA pouze přebírá parametry a posílá je vlastnímu ovladači.

dači myši, ale jestliže je v počítači nainstalována grafická karta typu Hercules, přebírá kontrolu a zobrazování kurzoru myši, které u původního ovladače nefunguje dobře. Tento problém způsobuje "nezjistitelnost grafického režimu" u této karty, takže po přepnutí do grafického režimu ovladač myši dále počítá a pohybuje s kurzorem v textovém módu, což způsobuje, že myš vrací špatné souřadnice a její kurzor je naprosto nedefinovatelný. Problém chybných souřadnic jsem vyřešil pouhým nastavením myši do rozsahu (MaxX * 8 , MaxY * 8) a při vracení souřadnic jejich zpětným celočíselným dělením osmi. Špatné vykreslování kurzoru jsem vyřešil trvalým "zhasnutím" originálního a vlastním vykreslováním nového. Kurzor je vykreslován vždy, jestliže se změní souřadnice myši. Ke snímání souřadnic myši bylo použito rezidentní procedury navěšené na přerušeni od časovače INT 1CH, které je generováno každých 55 ms. Další vlastností této jednotky je emulace myši při její vlastní fyzické nepřítomnosti. Toto je prováděno stejným způsobem, jako u karty Hercules s tím, že hodnoty souřadnic jsou nastavovány rezidentní procedurou, která je napojena na klávesnicové přerušeni INT 09H, které je generováno při každém stisku klávesy. Do jisté míry je zde vyřešeno i zrychlování kurzoru při delším stisku klávesy stejně, jako je tomu při rychlém pohybu s myší. V grafickém režimu je kurzor vykreslován pomocí procedur *GetImage* a *PutImage*, v textovém je vykreslení kurzoru provedeno přímým zápisem do obrazové paměti počítače (Hercules - B000H:0000H;

CGA, EGA, VGA - B800H:0000H).

Jelikož jednotka obsahuje rezidentní procedury, je nutné použít na začátku programu procedury *InitMouse(EmuKey : boolean)*, která zajistí navěšení procedur

na použitá přerušení a na konci procedury *CloseMouse*, která tyto procedury odstraní. Parametr *EmuKey* u *InitMouse* povoluje nebo zakazuje emulaci myši v případě její fyzické nepřítomnosti. Dále je potřeba při přepnutí karty do grafického režimu (u Pascalu je to procedurou *InitGraph*) zavolat proceduru *InitGraphMouse*, která je použita hlavně kvůli kartě Hercules pro detekci grafického módu a před návratem do textového režimu použít procedury *CloseGraphMouse*, která zruší "grafickou myš". Pomocí procedur *DispMousePointer* a *HideMousePointer* je možno ovládat viditelnost kurzoru myši, v proceduře *QueryMousePos(X,Y : integer; Butt : byte)* vrací jednotka v proměnných X,Y souřadnice myši a v *Butt* stav jejích tlačítek. Tvar kurzoru se nastavuje v textovém režimu procedurou *SetTextMaskMouse* a v grafice pomocí *SetMaskGraphMouse*. Procedura *SetDefMaskMouse* nastavuje původní tvar kurzoru myši. Rozsah pohybu myši a její rychlost se nastavují pomocí *XRangeMouse*, *YRangeMouse* a *SetMouseSpeed*. Procedura *ResetMouse* provádí počáteční inicializaci myši.

Veřejná logická proměnná *ExistMouse* informuje o přítomnosti myši, proměnná *Buttons* udává počet tlačítek myši a *VisMouCur* informuje o viditelnosti kurzoru myši.

Jelikož tato jednotka byla odladěna na počítači řady XT a tam pracovala s uspokojivou rychlostí, na počítačích třídy AT a vyšších je rychlost zobrazování naprosto srovnatelná s původním ovladačem myši.

Jednotka MANN.TPU

Poskytuje veškeré nejzákladnější služby, potřebné pro spolupráci s výstupním zařízením typu tiskárna. Pracuje se stan-

datním přerušením tiskárny 17H. Pomocí funkce *TestPrinter* se provádí počáteční test připojení tiskárny na některém z existujících paralelních portů (LPT1 - LPT4). Není-li tiskárna připojena ani na jednom z nich, vrací funkce hodnotu *false*, pokud připojena je, pak provede její *Reset* a otestuje připravenost. Pokud i tento test proběhne dobře, vrací funkce hodnotu *true*, jinak zkusí znovu provést *Reset* (255 - krát) a poté vrací chybovou hodnotu *false*. Funkce *PrinterReady* testuje připravenost tiskárny přijmout další znak. Je-li tiskárna připravena, vrací chybový kód 0, jinak chybu č.1 (I/O chyba) nebo č.2 (signalizace konce papíru). Pomocí procedury *SetPrinterMode* je nastavován mód a režim tiskárny. Procedura nejprve vyšle řídicí znak 1BH a poté následuje další řídicí znak (sekvence), zadaný již uživatelem jako parametr této procedury. V jednotce jsou předdefinovány konstanty pro základní volbu režimů. Vlastní tisk provádí funkce *PrintChar* (tisk jednotlivých znaků) a *PrintStr* (tisk řetězců). Obě vrací stejné chybové kódy, jako funkce *PrinterReady*. V programu TCAD.PAS je využito funkce *TestPrinter* pro test připojení tiskárny, procedury *SetPrinterMode* pro nastavení zhuštěného typu písma (*Condens*) a funkcí *PrintChar* a *PrintStr* pro vlastní tisk výsledků.

Vlastní program TCAD.PAS by se pro jednoduchost dal popsat několika základními bloky :

Inicializační blok

Provádí vyhledání a načtení pracovních souborů norem

NETOLER.TCD, TOLER_DI.TCD a TOLER_HR.TCD z disku a adresáře, uvedeného jako parametr programu nebo implicitního. V případě chyby při vyhledávání souboru je vypsáno chybové hlášení *Datový soubor XXXXXXXXX.TCD nebyl nalezen*, pokud je některý ze souborů poškozen, vypíše se hlášení *Datový soubor XXXXXXXXX.TCD je neúplný*. Pokud proběhnou tyto operace v pořádku, je ještě otestována přítomná grafická karta a není-li typu EGA, VGA nebo Hercules, je vypsána chyba špatné grafické karty. Podle druhu grafické karty je také provedeno nastavení barev, vykreslena základní obrazovka a inicializována myš. Pak jsou nastaveny globální proměnné programu, jméno pracovního souboru se nastaví na NONAME.DBF, informace o měřeních na tvar 1:0 a je zobrazeno hlavní menu programu. Tato inicializace probíhá vždy je jednou při běhu programu a to ihned po jeho spuštění.

Bloky obsluh jednotlivých menu

Tyto části programu jsou všechny, až na drobné změny, stejné. Nejprve je spuštěna funkce *GetCommLine*, která provádí veškeré čtení vstupních zařízení (klávesnice, myš). Nezjistí-li funkce žádnou událost (stisk klávesy, tlačítka myši, pohyb myši), provádí blikání kurzoru v příkazovém řádku. Po stisku levého tlačítka myši (nachází-li se její kurzor v menu) nebo stisku klávesy *Enter* funkce vrací zvolený příkaz (posloupnost příkazů). Bylo-li stisknuto pravé tlačítko myši nebo klávesa *Esc*, pak funkce vrací nejspodnější příkaz menu (obvykle příkaz *ZRUŠIT* nebo *KONČIT*).

Po načtení příkazu (jejich posloupnosti) je funkcí *SelectCommand* proveden výběr prvního zadaného příkazu z řád-

ku, načteného funkcí *GetComLine*. Současně je v proměnné *EmptyLine* vrácena hodnota *true*, jestliže je příkazový řádek již prázdný a má být následně znovu prováděno jeho čtení funkcí *GetComLine* nebo *false*, obsahuje-li řádek ještě nějaké příkazy (parametry).

Poté následuje provedení příslušných operací podle zvoleného příkazu (jejich sekvence).

Editační blok

Tento blok osahuje vlastně jen několik bloků obsluh pro jednotlivá editační menu (viz. popis menu *EDITACE*). Zpětný návrat na předchozí menu je zajistěn uzavřením bloku obsluhy menu do smyčky typu *repeat - until*, kde výstup z cyklu je podmíněn logickou proměnnou *Exit*, která je nastavována při vygenerování příkazu *ZRUŠIT*.

Blok zobrazování a korekcí výsledků

Je umístěn na výstupu z editačního bloku a slouží ke konečné grafické prezentaci výsledků. Nejprve je proveden výpočet měřítka zobrazení *Scale*, poté je na základě tohoto vykreslena osa *i* s příslušným popisem, pomocí procedury *TolerLines* zobrazeny čáry, představující vypočtené toleranční pole a procedurou *TolerText* je vypsán informační text a vykresleny pomocné ukazovací čáry pro rozlišení jednotlivých tolerancí.

Jsou-li požadovány korekce výsledků, je po najetí myši do nově vypočteného tolerančního pole změněn kurzor a nastaveny hranice myši pro posuv čar, jejichž pohyb je realizován smazáním původních a následným vykreslením čar nových.

Na základě rozdílu původní a nové polohy tolerančních čar a měřítka zobrazování *Scale* je zpětně vypočten nový rozměr formy (přímá úměra), který je také zobrazen.

Blok souborových operací

Zajišťuje nejzákladnější operace s pracovními soubory (viz. popis menu *SUBOR*). Změna pracovního adresáře je provedena pomocí procedury *ChDir*, která součástí standardní knihovny *DOS.TPU*. Jelikož formát pracovních souborů byl zvolen typu *DBF* (databázový standart), je při zápisu souboru nejprve uložena hlavička *DBF* souboru o délce 225 byte, v níž jsou uloženy informace o počtu záznamů v souboru, jménech a typech jednotlivých položek a verzi použité databáze (v tomto případě *dBASE III+*). Tato hlavička je v programu uložena pro jednoduchost jako dvojice konstant *DBFHeader1* a *DBFHeader2*. Jednotlivé položky záznamu jsou potom převedeny na typ řetězec (*text*) a uloženy na disk. Jednotlivé položky jsou na potřebnou délku doplněny mezerami (*20H*) a konec souboru je označen znakem *SUB* (*1AH*). Čtení souboru probíhá podobně, jako zápis s tím, že po načtení úvodní hlavičky souboru jsou pak postupně čteny jednotlivé záznamy a jejich položky převáděny z typu řetězec (*text*) na číselné hodnoty. U zápisu i u čtení se provádí kontrola na správné ukončení operace (vypnutá kontrola I/O operací a jejich výsledek je testován funkcí *IOResult*).

Blok mazání

V tomto případě se nejedná o doslovné "smazání" souboru,

ale o vyčištění paměti, v níž jsou uchovávány jednotlivé položky výpočtů a opětovné nastavení implicitních hodnot programu (viz. popis inicializačního bloku).

Blok tiskových služeb

Uskutečňuje veškerý výstup programu na výstupní zařízení typu tiskárna. Provede počáteční otestování tiskárny, její nastavení a vlastní tisk podle předem nastavené volby (viz. popis menu *TISK* a popis jednotky *MANN.TPU*).

5. Závěr

V úvodu diplomové práce jsou uvedeny základní vlastnosti grafických systémů a jejich praktické aplikace. Dále je podrobněji popsána problematika zadání vlastního úkolu diplomové práce a její teoretické řešení. Poté již následuje popis vlastního programového systému pro výpočet rozměrů odlévacích forem.

Celý program, jak již bylo předem řečeno, je napsán v jazyce Turbo Pascal a výsledkem je tudíž spustitelný soubor typu EXE. Je tedy možné jej spustit přímo z operačního systému.

Avšak pro jednoduchost práce a předpoklad, že ve většině případů jej bude nutné volat ze systému AutoCAD pro rychlý přepočítání rozměru byla ještě vytvořena v souboru externích příkazů ACAD.PGP položka `TCAD,TCAD,150000,,4` - kde první část položky obsahuje jméno externího příkazu pro vyvolání tohoto programu, druhá vlastní jméno programu, třetí velikost paměti, potřebné pro spuštění programu, čtvrtá obsahuje výzvu po zahájení příkazu (v našem případě je prázdná) a poslední definuje, že po ukončení příkazu bude nastaven zpět původní režim AutoCADu.

Program pracuje v nejvyšších grafických režimech, které povolují příslušné karty :

- Hercules - 720 x 348 bodů

EGA - 640 x 350 bodů

VGA - 640 x 480 bodů.

Existuje ještě možnost použití režimu s vyšším rozlišením a to u grafické karty SVGA (Super VGA). Pro práci s tímto režimem je k dispozici ovladač VGA256.BGI, která umožňuje práci v režimu 800 x 600 bodů. Jelikož však není kompatibilní s každým druhem grafické karty (pracuje jen s typem

TRIDENT), není tento režim ještě podporován (ale není také problémem jej po vyřešení kompatibility ovladače s jinými typy karet v programu použít).

Při tvorbě programu bylo počítáno, kromě ovládání myši, také s řízením kurzoru pomocí klávesnice (emulace myši). To umožňuje použít tento produkt také v případě, že myš není připojena k počítači, na což někteří tvůrci programů tohoto typu často zapomínají.

Jednou z výhod tohoto programu je, že uživatel při práci s ním nepotřebuje popis (help), protože zadávání příkazů je přehledné a jednoduché, takže práci s ním po krátkém seznámení zvládne i člověk bez základních znalostí výpočetní techniky.

Další výhodou je také přehledná prezentace výsledků (což je základní požadavek na všechny grafické systémy) a hlavně možnost jejich korekce (změny) při zvláštních požadavcích na rozměr formy, který nemusí vždy plně vyhovovat potřebám konstruktéra.

Z předchozího popisu je dále zřejmé, že program počítá jen s přesně vymezenými typy rozměrů (díra, hřídel, rozteč) a neumí tedy zpracovávat rozměry komplikovanější. To je však dáno tím, že výpočet těchto je daleko složitější a hlavně není zcela jednoznačný. V praxi podniku Elektro Praga se však ukázalo, že i přes toto je výkonným pomocníkem při práci konstrukčního oddělení, protože velice urychluje práci na výpočtech právě těch jednoduchých rozměrů, kterých je výkresech převážející množství a jejichž výpočty byly v minulosti prováděny ručně s pomocí kalkulačky a byly velmi jednotvárnou a rutinní záležitostí.

A to by měl být hlavní cíl při tvorbě těchto a podobných programových produktů : zjednodušit, případně zcela odstranit automatickou, rutinní práci člověka, jenž se pak může věnovat vlastní tvůrčí práci.

6. Seznam použité literatury

- [1] Urban J., Počítačová grafika v projektování, Praha, ČVUT 1984
- [2] Operační systém MS-DOS, Praha, knihnice ČSVTS-FEL 1987
- [3] Programové vybavení pro AIP, Praha, knihnice ČSVTS-FEL 1987
- [4] Císařová K., Tišer J., Počítačová grafika, Liberec, VŠST 1990
- [5] AutoCAD v10.0 User Manual, AutoDesk
- [6] Turbo Pascal v6.0 Users Guide, Borland
- [7] Turbo Pascal v6.0 Reference Guide, Borland

7. Seznam příloh

Příloha č.1 - Obsah datového souboru NETOLER.TCD

Příloha č.2 - Obsah datového souboru TOLER_DI.TCD

Příloha č.3 - Obsah datového souboru TOLER_HR.TCD

Příloha č.4 - Výpis programu TCAD.PAS

program TCad;

{#R-.S-.I-.V-.(+)}

uses Crt.Graph.Mann.Mouse.LinkBGI;

type TypeLineStr = string[40];

TypeMemoryStorage = record

FPropType : byte;
 FAbbreviation.FDiameter.FFormDiameter : real;
 FToler.FFormToler : string[13];
 FModify : boolean

end;

TypeFileRecord = record

RAbbreviation.RPropName.RDiameter : array [1..7] of byte;
 RToler : array [1..13] of byte;
 RFormDiameter : array [1..7] of byte;
 RFormToler : array [1..13] of byte

end;

TypeNorm1 = record

UpNorm.LowNorm : real;

end;

TypeNorm2 = record

NormDiam.UpNorm.LowNorm : real;

end;

TypeNetolerNorm = record

NormDiam : real;
 Norm : array [1..3] of TypeNorm1;

end;

TypeTolerNorm = record

Param : string[2];
 Norm : array [1..19] of TypeNorm2;

end;

```
const CrossANDMask : TypePattMask = (1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,
                                     1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1);
```

```
CrossXORMask : TypePattMask = (0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
                                0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0);
```

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

Fakulta strojní

Katedra technické kybernetiky Školní rok: 1991/92

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Petra DĚDKA

obor 23-40-8 ASŘ výrobních procesů ve strojírenství

Vedoucí katedry Vám ve smyslu zákona č. 172/1990 Sb. o vysokých školách určuje tuto diplomovou práci:

Název tématu:

Zpracování měření koroze v betonových konstrukcích

Zásady pro vypracování:

- 1) Proveďte analýzu problému
- 2) Navrhněte způsob pořizování dat a jeho realizaci
- 3) Vytvořte programový systém pro grafickou prezentaci výsledků měření

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Ostřední knihovna
STUDENTSKÁ 5
701 00 LIBEREC

KTK / ASŘ - 2

✓ 202 / 92 S