

WORLD
CRAFT

ART

ATARI

klub Praha



ROČNÍK II.

Vydává 487. ZO Svazarmu —

ATARI KLUB v Praze 4.

Šéfredaktor a vedoucí redakční rady
JUDr. Jan Hlaváček.

Zástupce šéfredaktora ing. Stanislav
Borský.

Obálku navrhl RNDr. J. Tamchyna.

Adresa redakce:

487. ZO Svazarmu - ATARI KLUB Praha
REDAKCE

poštovní příhrádka 51

100 00 Praha 10

Řídí redakční rada: V. Bílek, ing. J. Biskup, RNDr. J. Bok, CSc., Ing. S. Borský, ing. V. Friedrich, ing. O. Hanuš, RNDr. L. Hejna, CSc., Z. Lazar, prom. fyz., CSc., Ing. M. Vávřda.

Olisk povolen se souhlasem redakce při zachování autorských práv a uvedením pramene. Rukopisy nevyžádané redakcí se nevracejí. Za původnost a věcnou správnost ručí autor.

Vychází šestkrát ročně. Neprodejné. Členům klubu distribuováno zdarma. Nepravidelné přílohy na objednávku jsou kompenzovány zvláštním klubovým příspěvkem.

Rozsah čísla 58 stran. Neprošlo jazykovou úpravou.

Tiskne PORS, závod 001, reprografický provoz.

Do tisku předáno ve V/1988

Vydávání schváleno OV Svazarmu

Praha 4 a OŠK ONV Praha 4.

Evidenční číslo ÚVTEI 86 042.

© ATARI KLUB Praha, 1988

OBSAH

Ze života Svazarmu	
Ocenění naší práce	2
VII. obvodní konference Svazarmu v Praze 4	2
ATARI KLUB v Kolvě	4
V Budějovicích soutěžili	4
II. celostátní setkání zástupců ATARI KLUBŮ	5

Zprávy výboru	
První výsledky ankety čtenářů ZAK	6
Tajemství programování — výzva zkušeným programátorům	8
Informace o obsahu Zpravodaje AK Bratislava — 1/88	8
Úprava magnetofonu členům	8
Schůzky sekce ATARI ST	9
Informace o obsahu Zpravodaje AK Humenné — 1/88	9
Kazety mimopražským	9

Pro začátečníky	
Nebojte se přerušení	10
Kopírování programů uložených na disketě	16

Tipy — triky	
Studený start počítače (potřetí)	21
Neučesané poznámky k programu DESIGN MASTER	23
Jednoduchý RAMDISK pro 800 XL	26
Vracíme se k blokům IOCB	28

Otázky a odpovědi	
Co je to programovací jazyk C	29

Uživatelské programy	
FILE MASTER — návod na použití programu	31
STAP — program pro tisk popisky kazety	34
COPY.TSD — návod ke kopírování programu	36
TRANSCOPY — program pro kopírování programu ve formátu Turbo 2000	42
GRAPH II — návod na použití souboru programů	~

Technické novinky	
Nový mikroprocesor pro ATARI XL/XE?	54
Disketová jednotka XF551 a DOS	54

Litárna	
Programy pro ATARI rozhlasem	56
Tiskárna ATARI 1029 a její	57
Vážená redakce,	57



ze ŽIVOTA svazarmu

OCENĚNÍ NAŠÍ PRÁCE

Dne 28. 4. 1988 převzal předseda naší ZO s. Václav Dostál na slavnostním zasedání OV Svazarmu Prahy 4 diplom s titulem **VZORNÁ ZÁKLADNÍ ORGANIZACE**, který nám byl udělen rozhodnutím OV Svazarmu ze dne 18. 2. 1988.

Při předávání vyznamenání byl zdůrazněn velký přínos na popularizaci výpočetní techniky v rámci naší relativně mladé ZO, ale i na veřejnosti při mnoha akcích, které jsme pořádali nebo se jich alespoň aktivně zúčastnili. Udělené vyznamenání je nejen oceněním práce všech našich aktivních členů, ale zároveň i zavazuje do budoucna.

VII. OBVODNÍ KONFERENCE SVAZU PRO SPOLUPRÁCI S ARMÁDOU V PRAZE 4

Dne 28. 5. 1988 se uskutečnila VII. konference OV SVAZARMU. Hlavním bodem programu byla Zpráva OV Svazarmu v Praze 4 o plnění usnesení VI. období konference Svazarmu v Praze 4 z roku 1985 doplněná po XVII. sjezdu KSČ a po 6. společném zasedání ÚV a RÚV Svazarmu. Přednesl ji předseda OV s. Bello Jamříška.

V obsáhlé zprávě konstatoval, jakých úspěchů dosáhly v uplynulém období ZO Svazarmu působící v největším pražském obvodu — v obvodu, který je vlastně pátým největším „městem“ v ČSSR. Bylo však též řečeno, že 90 ZO a v nich téměř 13 000 členů nemá k dispozici odpovídající materiálně-technické zázemí. Ostré kritice byla podrobena mnoho let odkládaná investiční výstavba, zejm. pak výstavba nového objektu OV Svazarmu u Krčského lesa, kde by našlo svůj stánek vedení OV a několik ZO.

Úkoly obvodní organizace Svazarmu v Praze 4 do roku 1990 byly delegátům předloženy v pracovní brožuře. V desetistránkovém dokumentu jsou úkoly rozděleny do 5 hlavních oblastí:

- zkvalitňování úrovně a účinnosti politickovýchovné práce, která tvoří páteř branné výchovy, výcviku i zájmové branné činnosti,
- plnění úkolů pro ozbrojenou sílu v přípravě branců,
- zájmová branná činnost, kde nadále trvají aktuální úkoly zabezpečující její masový rozvoj a branně výchovný obsah,
- finanční a materiálově technické,
- řízení, výstavba organizace a její vnitřní život.

V rámci třetího hlavního úkolu jsou pak rozpracovány úkoly pro jednotlivé odbornosti. Výpočetní technika ještě není považována za samostatnou odbornost. Je zahrnuta pod odbornost elektroniku. Pokud jde o konkrétní úkoly (které se týkají výpočetní techniky), citujeme:

„— Vytvořit podmínky pro využití audiovizuální, akustické, počítačové a další progresivní techniky pro zabezpečení výcviku branců a politicko-výchovnou práci. Zaměřit se na tvorbu programů pro tyto oblasti a realizovat je formou nabídky výcvikovým střediskům branců a základním organizacím. Účinně spolupracovat s politicko výchovnou komisí a sekci branné přípravy.

— Dále pokračovat v řešení prostorových podmínek pro činnost Kabinetu elektroniky. V obsahu činnosti kabinetu kromě počítačové techniky věnovat pozornost i dalším oblastem této činnosti.

— Při zabezpečení zkvalitnění materiálně technické základny odbornosti se zaměřit především na tvorbu zdrojů vlastními prostředky.“

V brožurce předložené delegátům konference byla publikována i zajímavá tabulka, obsahující „Návrh úprav plánu rozvoje členské základny a výstavby ZO Svazarmu v Praze 4 do roku 1990“. Vyplývá z něj mj., že k 31. 12. 1987 v rámci obvodu Prahy 4 bylo 90 ZO s 12 970 členy a že velký podíl na nárůstu členů v rámci OV Svazarmu má v uplynulém období zejm. naše ZO (ačkoliv právě toto jsme v přednesené zprávě neslyšeli).

Konference se vyznačovala velice bohatou a plodnou diskusí. Většina diskutujících jen stručně představovala úspěchy v práci jednotlivých ZO, zato však věcně a s podloženými argumenty kritizovala mnohdy léta neodstraňované nedostatky (prostorové podmínky, materiálně technické zajištění, neskutečný byrokratismus atd.). Zazněla i taková slova, že z oficiálního názvu naší společenské organizace se díky nevšimavému přístupu kompetentních orgánů FMNO vytratilo slovo SPOLUPRÁCE. Bohužel ale netaktickým postupem při vedení diskuse byla řada zajímavých diskusních příspěvků předčasně ukončena.

Konference přijala usnesení, které v plném znění otiskujeme:

U s n e s e n í

VII. obvodní konference SVAZARMU v Praze 4 konané dne 28. května 1988

VII. obvodní konference Svazarmu v Praze 4

A. s c h v a l u j e :

1. Zprávu OV Svazarmu v Praze 4 o splnění usnesení VI. obvodní konference Svazarmu v Praze 4 doplněného po XVII. sjezdu KSČ a 6. společném zasedání ÚV a RÚV Svazarmu.
2. Úkoly obvodní organizace Svazarmu v Praze 4 do roku 1990.
3. Zprávu kontrolní a revizní komise Svazarmu v Praze 4.

4. Úpravu plánu rozvoje členské základny a výstavby základních organizací Svazarmu v Praze 4 do roku 1990.
5. Volbu nového OV Svazarmu v Praze 4.
6. Volbu nové kontrolní a revizní komise Svazarmu v Praze 4.
7. Volbu delegátů na městskou konferenci Svazarmu v Praze.
8. Socialistický závazek obvodní organizace Svazarmu v Praze 4 k VIII. sjezdu Svazu pro spolupráci s armádou.

B. u k l á d á :

1. OV Svazarmu v Praze 4
 - soustavně vytvářet podmínky pro splnění úkolů obvodní organizace Svazarmu v Praze 4 do roku 1990. (Termín: průběžně)
2. Předsednictvu OV Svazarmu v Praze 4
 - zpracovat připomínky z diskuse k vyšším orgánům a předat je k řešení. (Termín: 20. 6. 1988)
 - odpovědět na dotazy a připomínky vznesené k OV Svazarmu v Praze 4. (Termín: do 30. 8. 1988)
 - zabezpečit proškolení nově zvoleného funkcionářského aktivu. (Termín: do 30. 9. 1988)
3. Radám odbornosti, sekcím a komisím
 - zpracovat závěry obvodní konference do plánu činnosti. (Termín: do 30. 9. a dále průběžně)
4. Základním organizacím Svazarmu v Praze 4 a delegátům obvodní konference
 - projednat závěry konference na členských schůzích, přijmout vlastní opatření ke splnění závěrů konference

Václav Dostál

ATARI KLUB V KOTVĚ

Ve dnech 25. až 27. května 1988 se uskutečnila v největším obchodním domě v ČSSR PRIOR - KOTVA prodejní propagační akce pod názvem „Centrum vědy a techniky“. ATARI KLUB Praha byl požádán o odborné zajištění této akce.

Po dobu tří dnů odpoledne až do večera probíhalo předvádění dvou mikropočítačů ATARI 800 XE, jednoho 130 XE a jednoho 520 ST. Předváděcí centrum ve 3. patře OD KOTVA bylo ve stálém obležení diváků, jimž soudruzi Pilný, Tvrdek a další poskytovali potřebné informace a především ukazovali, co ATARI umí.

japi

V Budějovicích soutěžili

Jak naši redakci informoval ing. Jiří Novák, kroužek našeho ATARI KLUBU pracující v Českých Budějovicích uspořádal v době jarních prázdnin veřejnou soutěž pro školní děti. Přes 60 dětí soutěžilo za klávesnicemi počítačů ATARI. V jaké disciplíně? V hraní si, tedy v nejdokonalejším ovládnutí herních programů. I. ročník soutěže „O cenu ředitele ZŠ Vltava“ se líbil. Jak dětem, tak organizátorům a vedení školy.

red

II. celostátní setkání zástupců ATARI KLUBŮ

Ve dnech 3. a 4. června 1988 pořádal náš ATARI KLUB pod záštitou ČÚV Svazarmu celostátní setkání ataristů. Bylo pozváno celkem 42 zástupců z 22 dosud známých ATARI KLUBŮ, pracujících jak pod hlavičkou Svazarmu, tak i jiných společenských organizací. Setkání se uskutečnilo v účelovém zařízení ČÚV Svazarmu v Mnichovicích—Božkově u Prahy. Pozvání akceptovalo celkem 36 zástupců z 18 klubů z celé republiky, reprezentujících přibližně 7000 ataristů.

Neoficiální jednání začala probíhat okamžitě po příjezdu jednotlivých účastníků již v páteční podvečer a pokračovala prakticky po celou noc (jak se stalo tradicí na podobných setkáních). Vzhledem ke značnému množství kvalitních příspěvků připravených do diskuse a z toho vyplývajícího časového stresu se pracovní strávená noc ukázala jako nezbytnost. Je to poučení pro pořadatele dalších setkání - akce je nutné organizovat alespoň celý víkend.

Oficiální jednání zahájil v sobotu ráno předseda výboru naší ZO soudruh Václav Dostál. Hned poté následoval jeden z nejdůležitějších bodů jednání, a to organizační zajištění příštího setkání. Bylo dohodnuto, že se uskuteční v říjnu tohoto roku v Brně.

Z hardwarové oblasti byly nejzajímavější příspěvky seznamující s novou disketovou jednotkou ATARI XF 551, s Minigrafem Aritma a možností jeho připojení k ATARI a s přípravou výroby zásuvného modulu (cartridge) s programy systému SUPER TURBO v našem AK. Zvláště zajímavá byla i informace o systému zrychleného přenosu dat označeném jako TURBO D, vyvinutém v AK Tlmače. Zaujala též informace o ukončení vývoje souřadnicového zapisovače v AK Brno a o přípravě jeho výroby údajně za přijatelnou cenu, pravděpodobně ještě v letošním roce.

Jako nejzajímavější příspěvek celého setkání byla většinou zúčastněných oceněna informace AK Tlmače o přípravě a zkouškách telefonního modemu. Nejdůležitějším úkolem vyplývajícím z informace je zajištění povolení provozu od kompetentních orgánů ministerstva dopravy a spojů. Tohoto úkolu se v zájmu pomoci kolegům v Tlmačích ujal náš klub.

Ze softwarových přednášek bylo nejzajímavějším seznámení s C—jazykem a se systémem SUPER TURBO, včetně tzv. zařízení "T.", umožňujícího po zabudování do programu ukládat data v systému Turbo. V současné době je tato úprava provedena již u mnoha uživatelských programů a ty jsou k dispozici na půjčovaných referenčních kazetách.

Dále byly projednány i otázky publikační činnosti. Zástupci klubů se navzájem informovali o edičních plánech.

Pravděpodobně nejvýznamnějším závěrem z II. celostátního setkání ataristů je dohoda o zásadách při výměně původních programů mezi jednotlivými kluby. V bouřlivé diskusi se utkaly dva zásadně protichůdné návrhy. Jedna skupina zastávala názor o naprosto bezplatné výměně programů, druhá skupina argumentovala nutností finančně ocenit práci nejzdatnějších programátorů, z čehož ale vyplývá nutnost realizovat rozšiřování programů mezi kluby za úplatu. Na závěr se většina klubů přiklonila k variantě druhé, kterou prosazoval zejména náš a brněnský AK.

Vznikl návrh na konání pravidelných burz programů za účasti odpovědných a dostatečně fundovaných zástupců ze všech AK, kteří by kolektivně posoudili předvedené programy, na místě se dohodli o jejich hodnotě a za takto dohodnutou „cenu“ by je přímo na burze získali pro své kluby. Další šíření programu pak již bude záležet na jednotlivých AK. (V rámci našeho klubu se předpokládá, že šíření většiny takto získaných pro-

gramů mezi našimi členy by mohlo být uskutečňováno za více méně symbolickou cenu).

NA ZÁKLADĚ VÝŠE UVEDENÉ DOHODY ŽÁDÁME VŠECHNY NAŠE ČLENY, KTERÍ SE AKTIVNĚ ZABÝVAJÍ PROGRAMOVÁNÍM A DOMNÍVAJÍ SE, ŽE VYTVOŘILI NĚJAKÝ ZAJÍMAVÝ PROGRAM VHODNÝ PRO ŠIROKÉ UPLATNĚNÍ, ABY SE PŘIHLÁSILI BUĎTO OSOBNĚ U JEDNATELE VÝBORU PŘI PRAVIDELNÝCH ČTVRTEČNÍCH SCHŮZKÁCH NEBO PÍSEMNĚ NA ADRESU REDAKCE ZPRAVODAJE ATARI KLUBU PRAHA.

Setkání zástupců ATARI KLUBU z celé republiky probíhalo až do pozdních odpoledních hodin a jak prohlásil jeden ze zúčastněných, mělo pouze jednu závažnou vadu: trvalo JEN 24 hodin.

Závěrem této stručné zprávy bychom chtěli za všechny zúčastněné poděkovat zaměstnancům ČÚV Svazarmu v Mnichovicích za ochotu, se kterou plnili všechny naše požadavky a tím napomohli k vytvoření výborných podmínek pro důstojný průběh jednání.

Ing. Stanislav Borský



První výsledky ankety členů ZAK

Jak jsme již informovali v ZAK č. 1/88, naše anketa byla připravena s cílem získat náměty a připomínky důležité pro orientaci a další zaměření činnosti redakce a redakční rady ZAK, ale též pro získání podkladů pro zaměření činnosti ATARI KLUBU Praha. Ukazuje se však, že některé ze statisticky zpracovaných údajů by mohly být mimořádně zajímavé například pro obchodní organizace

zajišťující dovoz této výpočetní techniky k nám. Proto jsme výsledky části ankety poskytli vedoucím pracovníkům GR PZO TUZEX a GR Obchodu průmyslovým zbožím.

Ankety se zúčastnilo 266 členů (valná většina neanonymně), což považujeme za dostatečně reprezentativní vzorek pro stanovení obecnějších závěrů (13,3 % z počtu ATARI KLUBU Praha). V řadě anketních lístků bylo slovní doplnění odpovědí, u 22 lístků byl přiložen dopis. Všechny dopisy jsme pečlivě přečetli. Na ty nejzávažnější jsme odpověděli individuálně přímo respondentovi. Ostatní slovní odpovědi byly - pokud to šlo - rozříděny, zobecněny a zahrnuty do statistiky. Některé z připomínek, zejména k formální stránce časopisu, ale i k zajišťování služeb členům AK, již byly realizovány (obsah každého čísla, obsah ročníku, půjčování kazet s programy atd.). I když slovně vyjádřené náměty, připomínky i kritika byly velice obtížně vyhodnotitelné, získali jsme z nich cenné informace, přičemž k řadě z nich se lze vracet.

Na některé myšlenky (lépe řečeno prosby) obsažené v dopisech nelze nereagovat, i když pomoc není snadná. Fandové hladově po informacích chtějí kupovat literaturu, popisy či manuály programů a projevují silný zájem o získání softwaru a nákup dalších periférií. Jsme si vědomi toho, že řeči a sliby kom-

petentních osobností nejsou k ničemu, když technika, programy ani know-how za rozumné množství Kčs prostě není. Není ostatně ani za TK. Věříme, že každý soudný člověk snad pochopí, že tento nepříznivý stav není schopna změnit hrstka nadšených amatérů z ATARI KLUBU, kteří ve svém volném čase, dobrovolně a nezištně vymýšlejí SW, HW doplňky, zajišťují služby kolegům, vytvářejí časopis a další publikační materiály atd. Tato práce může jen částečně tišit tu velikou bolest, za kterou lze považovat absenci výpočetní techniky. . .

Vyhodnocení odpovědí na otázky ankety je typicky počítačová úloha, i když některé z odpovědí se obtížně formalizují. Byl vytvořen program pro ATARI XL/XE, umožňující zavádění dat do souboru "per partes". To dovoluje přidávat data do jednotlivých záznamů postupně podle zpracovávaných úloh. Protože každá statistika vyžaduje speciální subrutinu, omezili jsme se pro začátek na zpracování těch nejdůležitějších odpovědí: na otázky technického vybavení účastníku ankety, jejich plánů a potřeb týkajících se dalšího rozšíření vlastní konfigurace, dále na vztah k programování, názor na nejužitečnější rubriku, nejlepší články v ZAK a názory na to, co v ZAK čtenářům chybí.

Jaké jsou tedy první výsledky?

1. Majitelé počítače:

— ATARI 800 XL (XE)	86,84 %
— ATARI 130 XE	12,78 %
— jiný	0,38 %

2. Majitelé periferií:

— magnetofon	25,56 %
— magnetofon (s TURBO 2000)	67,67 %
— disketová jednotka	11,65 %
— tiskárna ATARI 1029	7,90 %
— 1 křížový ovladač	34,21 %
— 2 a více křížových ovladačů	54,89 %
— světelné pero	7,90 %
— jiná zařízení	1—3 %

3. Respondenti mají zájem své vybavení doplnit o:

— tiskárnu	55,26 %
— disketovou jednotku	44,74 %

— světelné pero	10,52 %
— úpravu TURBO 2000	6,77 %
— křížový ovladač	4,51 %
— ostatní periferie	do 4 %

4. Vztah k programování:

— aktivně programuje	97,37 %
— průměrný podíl programování	41,66 %
— užívá BASIC	84,96 %
— užívá TURBO BASIC aj. než BASIC	12,41 %

5. Nejžádanější rubrika ZAK:

— TIPY a TRIKY	33,83 %
— Uživatelské programy	32,33 %
— Koutek techniky	17,67 %
— Pro začátečníky	6,01 %
— ostatní celkem	10,61 %

6. Co v ZAK chybí, co si čtenáři přejí?:

- (seřazeno podle největšího počtu odpovědí)
- mnohem více informací pro začátečníky,
 - jak programovat - rady, finty, krátké rutiny
 - popisy uživatelských programů
 - zkrácené popisy herních programů
 - manuály programovacích jazyků
 - možnost inzerce v časopisu.

7. Nejlepší články publikované v ZAK ročníku I (1987):

- manuál TURBO BASIC (3/87)
- TURBO 2000 (příloha II/87)
- ATARI 800 XL s pamětí 320 kB (6/87)
- RAMDISK pro ATARI 800 XL (4/87)
- "PEEK & POKE" . . . důležité adresy systému . . . (příloha I/87)
- ovladače (bez možnosti přesněji specifikovat článek, neboť respondenti neuvádějí)
- Světelné pero (4/87)

☆ ☆ ☆

K vyhodnocení dalších informací z ankety se budeme vracet. Samozřejmě, že se budeme snažit orientovat činnost tak, aby to odpovídalo vyjádřeným potřebám. Je možné, že zveřejněné výsledky ankety vyprovokují další členy k diskusi.

Děkujeme všem, kteří anketě věnovali svůj čas.

— Iar —

TAJEMSTVÍ PROGRAMOVÁNÍ — VÝZVA ZKUŠENÝM PROGRAMÁTORŮM

Redakce ZAK má snahu (podle přání mnoha začínajících uživatelů výpočetní techniky ATARI XL/XE) otiskovat (i na pokračování) seriály věnované **TAJEMSTVÍ PROGRAMOVÁNÍ**.

Nejde nám jen o prosté popsání programovacího jazyku a jeho příkazů a stručné vysvětlení k čemu jsou (tak jak je běžné v dostupných příručkách), ale především o podrobné vysvětlení jednotlivých příkazů s ukázkami praktických návodů a příkladů, se zdůvodněním proč tak a ne onak. Jedná se nám prostě o předání znalosti a zkušenosti.

Představujeme si, že bychom začali s BASICEM, TURBO BASICEM a PASCalem. Přijdou-li jiné návrhy, uvítáme je.

Nabídněte redakci ZAK svou spolupráci ve jménu boje za získání druhé gramotnosti v řadách ataristů. Své nabídky s podrobnou osnovou vaší práce zašlete obratem. Budeme nuceni se držet zásady: "kdo dřív přijde, ten dřív mele".

red

INFORMACE O OBSAHU ZPRAVODAJE AK BRATISLAVA — 2/88

V podstatně kvalitnějším technickém provedení a ve velmi hezké obálce vyšlo 2. číslo Zpravodaje 301. ZO Svazarmu — ATARI KLUBU v Bratislavě. Má rozsah 48 stran (text psaný normalizovaně na psacím stroji), výsledný formát A5. Časopis obsahuje:

— Zprávy výboru: Informace o výroční členské schůzi, oznámení o složení výboru AK, plán činnosti, potvrzení stanoviska, že bratislavský AK přijal jako druhý standard přenosu dat systém TURBO 2000. Informace o Dni otevřených dveří dne 28. 2. Informace o doplňkovém vybavení počítače ATARI XL/XE.

— Pro začátečníky: Kopírování programů (prg. TRANSCOPY). Přehled propojovacích kabelů (tabulka) a popis paralelního vstupu/výstupu na počítači.

— Koutek techniky: Schéma propojení počítač — TV konektorem EUROSCART AV. Prodloužení životnosti barvicí pásky pro tiskárny ATARI 1029. Systém TURBO 2000 (informace) Popis obvodu PIA.

— Svět okolo mikropočítačů. Kompilátory a interprety (z Bajtku 8/87). O softwaru počítačů ATARI ST a PC. Kazetový operační systém pro ATARI 800 XL (z Bajtku 5/87).

— Programy pro pobavení a poučení: Úvod do rubriky. ATARI STATISTIC 1. BIO-RHYTHM. EXCELSOR (popisy programů).

— Naš rozhovor: s celním inspektorem o nové právní úpravě celních předpisů v oblasti dovozů výpočetní techniky.

— Oznamy: redakční výzva.

— Slovo na závěr: informace o tom, že budoucí číslo bude monotematicky zaměřené na zvukové a hudební programy a ovládání přes MIDI výstup.

HI

ÚPRAVA MAGNETOFONU ČLENŮM

ATARI KLUB Praha zavádí od 9. 6. 1988 novou službu svým členům:

úpravu magnetofonu XC-12 pro zrychlený přenos dat v systému TURBO 2000 nebo SUPER TURBO (podle přání zákazníka).

Přijem magnetofonů je každý čtvrtek od 18 do 19 hodin v SOUŠSŽ v Ohradní ulici v místnosti č. 101.

Upozorňujeme, že majitel ztrácí úpravou firemní záruku, avšak získává půlroční záruku na správnou funkci obvodu TURBO 2000, resp. SUPER TURBO. Bližší informace podají pracovníci HW skupiny. V každém případě vyžadujte od převjímacího technika potvrzení o odevzdání magnetofonu, jakož i paragon při úhradě služby. Výbor stanovil za tuto službu cenu 150,— Kčs.

výbor

SCHŮZKY SEKCE ATARI ST

Každému je snad již dostatečně známo, že v rámci ATARI KLUBU Praha pracuje relativně samostatně sekce uživatelů počítačů řady ST. Uživatelé ATARI ST se scházejí v budově matematicko fyzikální fakulty UK v Praze 8, V Holešovičkách 2, v 1. patře.

Termíny schůzek, které začínají v 17,30 hodin, budou pro období po prázdninách upřesněny na klubové nástěnce v SOU SSŽ.

Na schůzkách budou pořádány přednášky k užívání počítače ATARI ST, k užívání periférií a jednotlivých programů. Schůzek se mohou účastnit pouze členové 487. ZO Svazarmu - ATARI KLUBU, kteří se prokáží legitimací Svazarmu a klubovou průkazkou.

☆☆☆

V této souvislosti si dovoluujeme upozornit, že sekce ATARI ST připravuje pro své členy zvláštní edici upravených manuálů systému ST, jako např. Mikroprocesor 68000, Uživatelská příručka, GFA BASIC, BIOS, X-BIOS, GEMDOS, GEMAES, CVDI, Linea Handler, Programovací jazyk C aj. Tyto publikace budou vycházet jako zvláštní přílohy ZAK v tzv. "řadě ST". Vzhledem k současným počtům uživatelů ATARI ST nebudou vydávány v takovém nákladu, jako publikace určené pro uživatele ATARI XL/XE. Budou určeny především pro členy sekce ST, dále pak pro další vážné zájemce z řad členů klubu a posléze pro socialistické organizace, které zavedly počítače ATARI ST. Způsob objednávání těchto publikačních materiálů bude obdobný, jako je tomu u základních příloh ZAK (bližší viz ZAK č. 2/88).

ing. Jan Suchánek,
vedoucí sekce ST

INFORMACE O OBSAHU ZPRAVODAJE AK HUMENNÉ — 1/88

ZO — Východ Svazarm Humenné nám oznámila, že založila ATARI KLUB, který sdružuje 65 uživatelů mikropočítačů ATARI XL/XE. Klub pomáhá i menším klubům ve Vranově nad Toplou a v Pavlovicích nad Úhom. AK v Humenném začal vydávat svůj

zpravodaj. První číslo má 24 stran A4 psaných na psacím stroji a obsahuje:

- Úvodní slovo předsednictva AK
- Statut ATARI KLUBU, složení předsednictva klubu
- Programujeme v jazyku LOGO (13 stran)
- Úprava magnetofonu ATARI XC-12 pro přenosovou rychlost 2000 Bd (podle V. Rockaie)
- Ako na to? (informace o TURBO 2000 a zaváděcí TURBO 2500 (R. Polášek))
- Změna v —DL—
- VBI — muzika
- Popis hry F-15 STRIKE EAGLE

Adresa klubu:

ZO Svazarmu

ATARI KLUB — ZO Východ

ul. Hrnčiarska

066 01 Humenné

KAZETY MIMOPRAŽSKÝM

Půjčování kazet mimopražským členům

Od 21. 5. 1988 rozšířila SW skupina půjčování referenčních kazet i mimopražským členům, kteří se zúčastňují sobotních setkání v SOU SSŽ v Ohradní ulici. Je to konkrétní odpověď na požadavky prezentované delegáty na naší I. konferenci (viz ZAK č. 1/88). Není třeba zdůrazňovat, že realizace této služby je založená na dobrovolné práci dalších obětavců. Rádi bychom, aby rozšíření půjčování kazet i na sobotní schůzky přineslo užitek dalším členům. Na druhé straně bychom si přáli, aby se nevyskytlo porušení zásad půjčování, které jsou následující:

1. Členství v ATARI KLUBU Praha.
2. Půjčování se provádí každou lichou sobotu v době od 10 do 11.30 hodin v SOU SSŽ.
3. Kazeta se půjčuje na dobu 14 dní, tj. max. do příští liché soboty.
4. Ostatní podmínky půjčování jsou uvedeny v QUICK REPORTU z 20. 2. 1988.

Poznámka: V době přípravy této informace k tisku (červen 1988) je k dispozici 27 kazet ve třech provedeních.

výbor

Pro začátečnicku

NEBOJTE SE PŘERUŠENÍ

Přerušení je pojem, který vyvolává respekt i u mnohých zkušených uživatelů mikro-počítačů. Snad je to i tím, že přerušovací programy jsou psány vesměs ve strojovém kódu. Ale právě přerušení je mocným nástrojem nejednoho počítače - a Atari není výjimkou. Proto se s jeho přerušovacím systémem seznámíme poněkud podrobněji.

Co je to vlastně přerušení? Ve slovníku výpočetní techniky se můžeme dočíst, že přerušení je „... zastavení provádění posloupnosti instrukcí jinou posloupností nebo návrat k dříve přerušené posloupnosti“. Jednodušeji řečeno, celý proces přerušení si můžeme představit jako tři postupné fáze:

- zastavení běhu původního programu,
- provedení přerušovacího programu (podprogramu),
- návrat do původního programu a jeho pokračování.

Proces přerušení tedy připomíná volání a provádění podprogramů (například v BASICu). Rozdíl mezi podprogramem a přerušením je však v tom, že u podprogramu můžeme přesně určit, kdy bude daný podprogram realizován, zatímco provedení přerušovacího programu závisí na vnějších okolnostech (podmínce přerušení).

Přerušení můžeme přirovnat k následující situaci z běžného života: V průběhu pracovní činnosti zazvoní telefon. Pracovník přeruší právě prováděnou práci, zvedne sluchátko a telefonuje. Po ukončení rozhovoru položí sluchátko zpět na vidlici a vrátí se zpět k původní rozdělané práci. Přirovnáme-li pracovní činnost k hlavnímu programu, pak proces telefonování bude přerušovacím programem a zazvonění telefonu podmínkou přerušení. Je zřejmé, že výskyt této podmínky je zcela nezávislý na našem pracovníkovi a jeho původní činnosti. Všimněme si také, že ani nelze určit, kolikrát během činnosti telefon zazvoní. Obdobně nelze předem určit, kolikrát během programu dojde k přerušení, tj. k splnění podmínky přerušení.

Přerušení se využívá například ke komunikaci počítače (procesoru) s okolím, třeba ke čtení znaků z klávesnice. Pokud by procesor pravidelně během programu testoval, zda není stisknuta nějaká klávesa, zbytečně by se snižovala rychlost provádění programu. Přerušovací systém umožňuje procesoru, aby se plně věnoval své práci (běhu programu). Pokud dojde ke stisknutí klávesy, vyšle klávesnice nebo její řídící obvod signál do procesoru, který vyvolá přerušení a teprve nyní dojde k testu klávesnice.

U mikropočítačů Atari je přerušovací systém plně využíván. Tři úrovně přerušení využívá nejen operační systém pro komunikaci se vstupními a výstupními zařízeními počítače, včetně grafického procesoru ANTIC, ale může je použít i uživatel počítače ve svých programech. K tomu je třeba povšechná znalost programování ve strojovém kódu a rovněž znalost jednotlivých typů přerušení a práce s nimi. Následující informativní přehled se snaží uživateli přiblížit nejčastěji využívaná programová přerušení.

DISPLAY LIST INTERRUPT - obrazové přerušení

Obrazové přerušení známé pod zkratkou **DLI** je nejčastěji využívaným programovým přerušením. Obrazový mikroprocesor ANTIC umožňuje modifikovat všechny své instrukce tak, aby po provedení příslušné instrukce (zobrazení řádku) došlo k vyvolání přerušovacího programu.

K uložení přerušovacího programu lze s výhodou použít šestou stránku paměti (adresy 1536 až 1791), která je uživateli přístupná v BASICu i při programování ve strojovém kódu. Vytvoříme proto přerušovací program, který změní hodnotu druhého barvového registru (pozadí). Všimněte si, že podprogram začíná uložením návratové adresy a použitých registrů (v našem případě stačí uložit pouze střadač — registr A):

```
8      PHP
72     PHA
173 198 2   LDA COLOR2
141 10 212 STA WSYNC
24      CLC
105 2      ADC #2
141 198 2   STA COLOR2
141 24 208 STA COLPF2
104      PLA
40      PLP
64      RTI
```

Příslušný program v BASICu včetně inicializace vypadá takto:

```
10 GRAPHICS 0:FOR I=1536 TO 1555:READ A:POKE I,A:NEXT I
20 DATA
8,72,173,198,2,141,10,212,24,105,2,141,198,2,141,24,208,104,40,64
30 POKE 512,0:POKE 513,6:POKE 710,194
40 DL=PEEK(560)+256*PEEK(561):FOR I=7 TO 28 STEP 3:POKE DL+I,130:NEXT
I
50 POKE 54236,192
```

Na řádcích 10 a 20 je uložen výše uvedený strojový přerušovací program na počátek 6. stránky paměti. Řádek 30 zapisuje počáteční adresu programu do registru VDSLST na adresách 512 (nižší byte) a 513 (vyšší byte). Na řádku 40 je modifikován program obrazového procesoru ANTIC (Display List). Přičteme-li ke kódu instrukce 128, dojde po provedení této instrukce k přerušení. Příkaz na řádku 50 povoluje obrazové přerušení.

VERTICAL BLANK INTERRUPT — svislé obrazové přerušení

Svislé obrazové přerušení (**VBI**) je vyvoláno vždy po dokončení jednoho obrazového snímku. Neboť obrazovka se obnovuje každou 1/50 sekundy, je toto přerušení vyvoláváno s frekvencí 50 Hz. Přerušovací podprogram se provádí v době zpětného běhu paprsku z pravého dolního do levého horního rohu obrazovky a může mít délku až 2 kB. Přerušovací program pro VBI nekončí instrukcí RTI, ale skokem na adresu XITVBV (58466), tj. JMP XITVBV.

Vytvoříme například přerušovací program, který bude měnit barvu rámečku obrazovky s frekvencí 1 sekundy. Změna se tedy provede vždy po 50 obrazových snímcích:

```

198 206      DEC 206
165 206      LDA 206
208 16        BNE END
173 200 2     LDA COLOR4
24           CLC
105 2         ADC #2
141 200 2     STA COLOR4
141 26 208    STA COLBAK
169 50        LDA #50
133 206       STA 206
76 98 228     END JMP XITVBV

```

Tentýž program v jazyku BASIC:

```

10 GRAPHICS 0:FOR I=1536 TO 1560:READ A:POKE I,A:NEXT I
20 DATA
136,206,165,206,208,16,173,200,2,24,105,2,141,200,2,141,26,208,169,50,
33,206,79,98,228
30 POKE 206,50
40 POKE 549,0:POKE 549,5
50 POKE 54296,54

```

Na řádku 30 nastavujeme výchozí barvu rámečku. Příkazy řádku 40 provádějí vlastní inicializaci přerušení. Do registru VVBLKD na adresách 548 a 549 přitom ukládáme počáteční adresu přerušovacího programu. Řádek 50 povoluje přerušení VBI.

Při spuštění programu může dojít ke zhroucení celého systému. Tato situace vzniká v případě, že zpětný zatemňovací impuls vyjde mezi oba příkazy na řádku 40. V registru VVBLKD je v této chvíli chybná hodnota, která způsobí zablokování další činnosti počítače.

Tuto řídkou, ale nepříjemnou vlastnost umožňuje odstranit jednoduchá strojová rutina:

```

104          PLA
1600         LDY #0
1626         LDX #6
1697         LDA #7
76 92 228    JMP SETVBV

```

Podprogram operačního systému na adrese SETVBV nastavuje automaticky správnou počáteční hodnotu adresy přerušení VBI. Předchozí program tedy upravíme tak, že řádky 40 a 50 nahradíme následujícími:

```

40 FOR I=1561 TO 1570:READ A:POKE I,A:NEXT I
50 DATA 104,160,0,162,5,169,7,76,92,228
60 I=USR(1561):POKE 54296,54

```

Přerušení klávesou BREAK

Jak již bylo uvedeno, pomocí přerušení je zajišťováno i čtení znaků z klávesnice počítače. Mezi všemi klávesami má zvláštní místo klávesa BREAK. Při jejím stisknutí je spuštěn speciální přerušovací program, který například v jazyce BASIC provede zastavení výpočtu a výpis hlášky:

```
STOPPED AT LINE xxxxx
```

Tento přerušovací program začíná na adrese BRKIRQ, která je u modelů XL/XE rovna

49298. Na adresu ukazuje vektor BRKKY uložený na adresách 566 a 567, čili hodnota: PEEK(566)+PEEK(567) * 256

Chování počítače při stisknutí klávesy BREAK můžeme změnit tak, že vytvoříme vlastní přerušovací program a vektor BRKKY nasměrujeme na jeho začátek. Zkusíme si jednoduchý příklad, který při stisknutí klávesy BREAK vypíše v levém horním rohu obrazovky text ATARI 800 XL. Vlastní program v BASICu se tím však nepřeruší.

```
1600 LDY # 0
185146 L1 LDA TXT,Y
14588 STA (88),Y
200 INY
19212 CPY # 12
208246 BNE L1
104 PLA
64 RTI
TXT.BYTE 33, 52, 33, 50, 41, 0
.BYTE 24, 16, 16, 56, 44, 0
```

Posloupnost hodnot v závěru programu obsahuje vnitřní kódy jednotlivých znaků zobrazovaného textu (tj. 33=A, 52=T atd.).

Jako v předchozích případech uložíme tento program do volné 6. stránky paměti pomocí příkazů jazyka BASIC:

```
10 FOR I=1530 TO 1551:READ A:POKE I,A:NEXT I
20 DATA 160,0,185,14,6,145,88,200,192,12,208,246,104,64
30 DATA 33,52,33,50,41,0,24,16,16,56,44,0
40 POKE 555,0:POKE 567,5
```

Data na řádce 20 obsahují vlastní přerušovací program, data na řádce 30 tvoří zobrazený text. Řádek 40 obsahuje příkazy, které mění hodnotu registru BRKKY tak, aby ukazovala na počátek přerušovacího programu.

Pokud nechceme zrušit původní funkci programu BRKIRQ, ale chceme ji pouze rozšířit, změníme příkaz RTI na konci našeho přerušovacího programu na příkaz JMP BRKIRQ, tj. na skok do původního přerušovacího programu. Upravený inicializační program v jazyce BASIC bude mít tvar:

```
10 FOR I=1535 TO 1562:READ A:POKE I,A:NEXT I
20 DATA 160,0,185,15,6,145,88,200,192,12,208,246,76,146,192
30 DATA 33,52,33,50,41,0,24,16,16,56,44,0
40 POKE 555,0:POKE 567,5
```

Po provedení tohoto programu a stisknutí klávesy BREAK dojde k zobrazení nápisu ATARI 800 XL i k přerušení běžícího programu jako v obvyklém režimu.

Přerušení pomocí časovačů

Počítače Atari obsahují pět časovačů, tj. čítačů, které snižují svou hodnotu o jednotku každou 1/50 sekundy. Pokud hodnota některého čítače dosáhne nuly, provede se speciální akce. U časovačů 3, 4 a 5 se při dosažení nuly nastaví hodnota speciální buňky v paměti CDTMF na jednotku. Tuto situaci můžeme ve vlastním programu testovat. Časovače 1 a 2 spustí při dosažení nulové hodnoty přerušovací program uložený na adrese CDTMA. Počáteční hodnotu čítačů nastavujeme v dvoubytovém registru CDTMV, tato hodnota může být nejvýše 65535, což odpovídá časové prodlevě přibližně 22 minut.

Adresy registrů, s kterými pracují uvedené časovače, jsou souhrnně uvedeny v následující tabulce:

	CDTMV	CDTMA	CDTMF
1	536, 537	550, 551	
2	538, 539	552, 553	
3	540, 541		554
4	542, 543		556
5	544, 545		558

To znamená, že například adresa přerušovacího podprogramu časovače 1, CDTMA1, je dána hodnotou: $\text{PEEK}(550) + \text{PEEK}(551) \times 256$

Časovač 1 je standardně využíván operačním systémem počítače. Proto se jeho používání bez důkladné znalosti časování systému nedoporučuje. Pro účely časového přerušení je proto nejvhodnější časovač 2. Uvedeme si nyní jeho využití v úloze, kterou jsme řešili pomocí svistého obrazového přerušení VBI, tj. přepínání barvy rámečku obrazovky s frekvencí 1 sekundy. Díky využití časovače se přerušovací program značně zjednoduší:

```

173 200 2    LDA COLOR4
24           CLC
105 2        ADC # 2
141 200 2    STA COLOR4
141 26 208   STA COLBAK
169 50       LDA # 50
141 26 2     STA CDTMV2
96           RTS

```

V závěru strojového programu obnovujeme původní hodnotu časovače, a tím docílujeme jeho opětovného spuštění. Všimněte si ještě, že přerušovací program pro časovač končí instrukcí RTS, a ne RTI jako ostatní přerušení.

Uvedenou rutinu můžeme opět uložit do šesté stránky paměti jako předchozí přerušovací programy. Příslušný program v BASICu má tvar:

```

10 GRAPHICS 0:FOR I=1536 TO 1553:READ A:POKE I,A:NEXT I
20 DATA 173,200,2,24,105,2,141,200,2,141,26,208,169,50,141,26,2,96
30 POKE 538,50:POKE 539,0
40 POKE 552,0:POKE 553,5:POKE 54286,64

```

Na řádku 30 nastavujeme počáteční hodnotu čítače, na řádku 40 adresu přerušovacího podprogramu. Změnou podtržených hodnot můžeme jednoduše změnit frekvenci přepínání barev. Například hodnota 25 zajistí změnu barev s frekvencí 0,5 sekundy, hodnota 150 s frekvencí 3 sekundy, a podobně.

Ukládací podprogram SETVBV

Podprogram operačního systému SETVBV na adrese 58460 slouží k nastavení adres některých přerušovacích programů. Tuto rutinu jsme již využili k ukládání počáteční adresy podprogramu přerušení VBI, kde odstranila možné zhroucení systému při přepisu buněk 548 a 549 (VVBKLD).

Tutéž rutinu můžeme využít se stejným úspěchem i k nastavení dalších registrů, jejichž změna „po bytech“ pomocí příkazů POKE by mohla za určitých okolností vést k chybné

činnosti, případně zhroucení. Tato situace může nastat i při změně hodnoty čítačů 1 až 5.

Před spuštěním podprogramu SETVBV musíme uložit výchozí hodnoty do jednotlivých registrů procesoru:

X — horní byte adresy (ukládáné hodnoty)

Y — dolní byte adresy (ukládáné hodnoty)

A — číslo časovače (7 = VBI)

Například k nastavení výchozí hodnoty 50 pro časovač 2 z předchozího příkladu lze užít krátkou rutinu:

```
104      PLA
160 50    LDY # 50
162 0     LDX # 0
169 2     LDA # 2
76 92 228 JMP SETVBV
```

Výše uvedený program v BASICu tedy upravíme změnou nebo přidáním následujících řádků:

```
30 FOR I=1554 TO 1563:READ A:POKE I,A:NEXT I
40 DATA 104,160,50,162,0,169,2,76,92,228
50 I=USR(1554)
30 POKE 552,0:POKE 553,6:POKE 54286,64
```

Několik slov závěrem

Uvedené čtyři druhy přerušení tvoří pouze část přerušovacího systému počítačů Atari, jde však o přerušení uživatelsky nejvíce využívaná. Přerušovací systém jako součást operačního systému tvoří u modelů XL/XE skoro polovinu celého operačního systému (samozřejmě včetně rutin pro vstup a výstup). Prvních 42 buněk druhé stránky paměti obsahuje přerušovací vektory, pomocí kterých komunikuje systém s programy, popřípadě uživatelem.

Kromě uvedených druhů přerušení může uživatel využívat i přerušení pro práci se sériovým kanálem, přerušení programovatelného obvodu POKEY (klávesnice, zvuk, ovládače) a tzv. break interrupt, který je způsoben instrukcí strojového kódu BRK. I tento výčet je neúplný. Tato a další přerušení jsou však vázána na znalost příslušné části operačního systému, činnosti sériového kanálu, obvodu POKEY, apod. Tyto informace však přesahují rámec našeho seznámení s přerušovacím systémem počítačů Atari. I uvedené čtyři základní (a pro uživatele nejjednodušší) typy přerušení podstatně rozšiřují možnosti uživatele a programátora (zejména v jazyce BASIC). Záleží tedy především na něm, jakým způsobem a k jakému účelu nabytých vlastností využije. Barevné změny jsou samozřejmě pouze zlomkem možností, které přerušení umožňuje.

(Podle seriálu Nie boj sie przerwań!, Bajtek, ročník 1987, upravil -fis-)

KOPÍROVÁNÍ PROGRAMŮ ULOŽENÝCH NA DISKETĚ

ing. Roman Cupal, Ústí nad Labem

V ZAK 1 a 2/87 bylo popsáno několik kopírovacích programů a postupů pro kopírování souborů uložených na kazetách. Jelikož roste počet vlastníků disketových jednotek, považujeme za účelné seznámit nové uživatele této výkonné periferie s kopírováním souborů na tomto zařízení.

Existují tři základní možnosti kopírování souborů uložených na disketě:

1. kopírování programů v BASICu,
2. kopírování jednotlivých souborů a celých disket pomocí funkcí DOSu,
3. kopírování jednotlivých souborů a celých disket pomocí kopírovacích programů.

Pro úplné začátečníky je třeba poznamenat, že počítač může spolupracovat s disketovou jednotkou a se soubory na disketě tehdy, je-li v počítači zaveden FMS (File Management System, tj. systém správy souborů) některého z diskových operačních systémů (obvykle soubor DOS.SYS). Toho se dosáhne vložením diskety obsahující soubor DOS.SYS do disketové jednotky a zapnutím počítače.

1. Kopírování programů v BASICU

První způsob kopírování vyžaduje, aby byl počítač aktivován v režimu se zavedeným ATARI BASICem (na obrazovce je zobrazen nápis READY). Předpokládáme, že uživatel již zná přechod z DOS do BASICu (např. prostřednictvím funkce "B" v DOS 2.5).

Programy vytvořené v BASICu lze kopírovat jejich načtením do paměti počítače (např. z kazety nebo jiné diskety) a následným zapsáním na disketu. Jedná se o stejný proces, jako při práci s kazetovým magnetofonem, avšak s tou změnou, že namísto příkazů CLOAD a CSAVE se použije pro čtení a zápis programů v BASICu obecný tvar

LOAD"Dn: jméno. extender"

SAVE"Dn: jméno. extender"

kde *n* je číslo disketové jednotky, pro kterou příkaz platí. To v tom případě, kdy jsou používány dvě, tři nebo čtyři disketové jednotky. Jestliže je připojena (užívána) jen jedna disketová jednotka, může být číslo "1" v předepsaném tvaru zápisu vynecháno.

Psát se nemusí ani uvozovky na konci výrazu. Jméno programu může mít max. 8 znaků, extender max. 3.

Programy vytvořené v BASICu lze číst a zapisovat i dalšími obecně známými způsoby (např. příkazy LIST-ENTER).

2. Kopírování programů funkcemi DOSu

Většina diskových operačních systémů (dále jen DOS) obsahuje v souboru DUP.SYS funkce pro kopírování jednotlivých souborů i celých disket.

DUP.SYS se načte do paměti počítače buď příkazem DOS (je-li počítač v režimu se zavedeným BASICem) nebo zapnutím počítače se současným stiskem OPTION (odpojení BASICu je pro kopírování vhodnější). Dále se postupuje podle manuálu příslušného DOSu.

Například u DOS 2.0 a DOS 2.5 je pro kopírování souboru určena volba "C" a "O", volba "J" pro kopírování celých disket. U příkazově orientovaných DOSů (např. DOS XL) jsou tyto

funkce vnějšími příkazy a je třeba je načíst z diskety (např. COPY.COM, DUPDSK.COM apod.).

Kopírovací funkce DOSu zpravidla neumějí přenášet soubory mezi disketovou jednotkou a magnetofonem. Pro tento účel je nutno použít kopírovacích programů.

3. Kopírování programů kopírovacími programy

Všechny dříve uvedené kopírovací funkce je možno realizovat kopírovacími programy, které obvykle přinášejí:

- větší komfort obsluhy,
- větší rychlost procesu kopírování,
- větší kapacitu bufferu (paměti),
- možnost přenosu mezi různými periferiemi,
- možnost konverze programu při přenosu,
- přenos mezi různými verzemi DOSu.

Kopírovací programy můžeme rozdělit na tři základní druhy:

- a) sektorové (sector copier),
- b) souborové (file copier),
- c) speciální (konverzní).

a) Sektorové kopírovací programy

Čtou a zapisují obsah diskety sekvenčně sektor po sektoru bez ohledu na pořadí souborů na disketě. Provedou tedy duplikát diskety bez ohledu na verzi DOSu nebo duplikát AUTOBOOT diskety. Většina sektorových kopírovacích programů neumí číst diskety s vadnými sektory. Pokud disketa vadné sektory obsahuje, je třeba použít např. programy **DKOP V.3.0** nebo **DISK DOCTOR II ULTIMATE DUPLICATOR**. Uvedené programy však neumějí číst diskety se zvětšenou hustotou (1040 sektorů).

Poněvadž sektorové kopírovací programy využívají pro čtení a zápis sektorů rutiny obsažené v OS počítače, nemusí být v počítači přítomen FMS (soubor DOS.SYS). To znamená, že je lze kromě volby "L" DOSu načíst a spustit též některým diskovým zavaděčem binárních souborů (např. MICRODOS) nebo dokonce z kazety binárním zavaděčem typu BL/C (kromě programu US SECTOR COPIER 4).

Sektorové kopírovací programy umožňují obvykle tyto volby:

- volbu čísla zdrojové a cílové disketové jednotky,
- zápis sektorů s nebo bez verifikace (porovnání zkopírovaných dat s originálem),
- volbu formátování cílové diskety dle zdrojové,
- volbu hustoty formátování,
- počet kopií,
- zapisovat či nezapisovat prázdné sektory,
- možnost kopírování zvoleného rozsahu sektorů na jinou disketu nebo na jiné místo na téže disketě.

Dále popíšeme postup práce se zřejmě nejoblíbenějším **programem pro duplikaci celých disket**:

US SECTOR COPIER 4 od E. Reusse

Po načtení volbou "L" DOSu nebo některým binárním zavaděčem se po krátkém úvodním textu objeví hlavní menu.

Menu sděluje čísla připojených disketových jednotek:

Source = zdrojová

Destination = cílová

a kapacitu bufferu pro kopírování (**Memory**):

56 kB pro 800 XL/XE

120 kB pro 130 XE,

ale příp. i 256 kB, jestliže už někdo vybavil svůj počítač pamětí vyšší.

Nápis "**Setup**" vyzývá k nastavení průběhu kopírování.

Tlačítkem **OPTION** volíme při dvou disketových jednotkách směr kopírování (Exchange drives = výměna disketové jednotky).

Tlačítkem **SELECT** volíme formátování cílové diskety. Buď chceme ("YES") nebo nechceme ("NO") disketu naformátovat. Disketa bude naformátována na hustotu totožnou se zdrojovou disketou. Při kopírování s formátováním nebudou zapisovány prázdné sektory. Při kopírování bez formátování budou zapsány všechny sektory.

Tlačítkem **START** zahajujeme kopírování. Hlášení „**Source disk**“ nás vyzve k vložení zdrojové diskety (při jedné disketové jednotce) a po opětovném stisknutí **START** začíná čtení. Ohlásí se hustota zdrojové diskety:

SINGLE 720 sect. po 128 byte

MEDIUM 1040 sect. po 128 byte

DOUBLE 720 sect. po 256 byte

a průběh čtení sektorů můžeme sledovat jak číselně, tak graficky.

Po naplnění bufferu nás hlášení „**Destination disk**“ vyzve k vložení cílové diskety. Stiskem **START** pokračujeme v kopírování. Jestliže jsme v úvodu předvolili požadavek formátování (**SELECT . . . Format: YES**), provede se nejdříve formátování (s oznámením **FORMATING**) s následným kopírováním první části diskety. Toto se opakuje několikrát (dle hustoty diskety a dle kapacity paměti počítače), dokud není zkopírována celá disketa.

Upozornění: Nepřerušujte průběh kopírování předčasně stiskem **RESET**, i když víte, že zbytek disku je prázdný. Např. u rozšířené hustoty (**MEDIUM**) nedojde k zápisu druhé části VTOC (tabulky obsahu diskety) do sektoru 1024.

Vzhledem k zápisu sektorů bez verifikace může dojít k chybnému duplikátu.

b) Souborové kopírovací programy

Tyto programy nám umožňují kopírovat soubory uložené na disketě některou verzí DOSu jako logické celky označené svým jménem. Kopírovat je tedy možno na stejné disketě, z diskety na disketu či na jiné zařízení.

Jsou programy, které **kopírují soubory po jednom**, a při jedné disketové jednotce je třeba neustále měnit zdrojovou a cílovou disketu. Jsou ale také programy, které **kopírují zvolený seznam souborů najednou**, rozdělený do částí dle kapacity bufferu.

Jako zástupce první skupiny si popíšeme **program na kopírování jednotlivých souborů mezi jakýmkoli periferiemi**:

FILE COPIER 1.45 od Martina Koehlinga (1984)

Po načtení a spuštění volbou „L“ DOSu se objeví **vstupní menu**:

Enter INPUT filespec

or **"*n"** for directory of drive **n**

or **"D"** for DOS

Nejspíš si nejdříve vypíšeme adresář zdrojové diskety stiskem **"*"** a zvolíme soubor ke kopírování. Poté zadáme ÚPLNOU specifikaci zvoleného souboru, tj. **"Dn: jméno.extender"** a soubor

zadej specifikaci souboru pro VSTUP

nebo **"*n"** pro adresář disketové jednotky číslo n (pro adresář disketové jednotky 1 stačí zadat **"*"** a RETURN

nebo **"D"** pro návrat do DOSu.

se načte do bufferu. Pro disketovou jednotku č. 1 je možno číslo "1" vynechat. Wild—cards jsou povoleny.

Načte-li se soubor bez chyb, vypíše se délka souboru v bufferu a max. velikost bufferu:

...bytes in buffer /Mem: ... /

Kapacita bufferu:

DOS 2.5 — BASIC vypnut ... 38962 byte

DOS 2.5 — BASIC zapnut ... 30770 byte

z kazety — BASIC vypnut ... 44594 byte

z kazety — BASIC zapnut ... 36402 byte

Nyní se objeví výstupní menu:

Enter OUTPUT filespec

zadej specifikaci souboru pro VÝSTUP - vypíše obsah bufferu na zadané zařízení;

or "⌘" to CLEAR buffer

nebo "⌘" pro výmaz bufferu - vymaže se buffer a objeví se vstupní menu;

or "V" to VERIFY

"V" pro porovnání souboru, který je právě v bufferu, se souborem na periférii, jejíž označení a jméno souboru je nutno po stisku "V" a "RETURN" dále zadat. Totožnost souborů bude porovnávána byte po byte;

or "F" to FORMAT

nebo "F" pro formátování disku - na žádost o specifikaci zařízení zadat Dn:

or "@" to RUN program in buffer spuštění programu v bufferu.

Pozor: Program FILE COPIER 1.45 nespouští binární soubory! Spouští pouze programy zkopírované z autoboot kazety nebo autoboot diskety do DOS souboru. Program se zeptá, odkud byl boot program zaveden:

Boot Tape stiskni T

Boot Disk stiskni D

a podle odpovědi naplní příslušné registry adresami, přesune obsah bufferu na správné adresy a spustí program.

Další volby - adresář a DOS - jsou stejné jako při vstupu.

Poznámky: Tento kopírovací program umožňuje přenášet soubory mezi všemi zařízeními, jejichž obslužné programy jsou obsaženy v paměti počítače (buď v ROM nebo zavedeny do RAM) a jejichž názvy a adresy obslužných programů jsou zapsány v tabulce HATABS. Tedy běžně C:, E:, S:, P:, K: ; při nahrání DOSu - D:; při nahrání KOS - F:; při nahrání RS 232 - R:

POZOR - pokud chceme pracovat s disketovou jednotkou, musí být File Copier zaveden volbou "L" DOSu (nelze z MICRODOSu).

Jsou uživatelé, kteří spouštějí programy, které mají pracovat se soubory na disketě (jako SpeedScript, SpeedCalc, Čapek atd.) binárním zavaděčem (MICRODOS apod.). To je chybné!!! MICRODOS je pouze zavaděč binárních a boot souborů a neobsahuje FMS. Proto pokus o čtení či zápis na disketu např. SpeedScriptem skončí chybou 130 - neznámé zařízení. Tyto programy spouštějte vždy volbou "L" DOSu nebo jako AUTORUN.SYS soubory.

Kopírovacím programem, který kopíruje sadu souborů najednou, je např.:

FILE - COPY od Norberta Hegemanna (1986)

Tento program přenáší najednou celou předvolenou sadu souborů ze zdrojové na cílovou disketu. Počet záměn disket při použití jedné disketové jednotky je určen pouze celkovým součtem délek všech zvolených programů a velikostí bufferu.

Slovníček

Source Drive

= zdrojová disketová jednotka

Destin Drive	= cílová disketová jednotka
Change Source Drive	= změň zdrojovou disketovou jednotku
Change Destin Drive	= změň cílovou disketovou jednotku
Direktory from Source Drive	= adresář zdrojové disketové jednotky
Set/Clear Copyflag	= nastav/zruš označení ke kopírování
Select Filename	= zvol jméno souboru
Begin the Copy	= začátek kopírování
End of Copy	= konec kopírování
Restart a Program	= opakovaný start programu
Insert Source/Destin Disk and Press Start	= vlož zdrojovou/cílovou disketu a stiskni START

Postup při použití programu FILE — COPY

- Program File Copy načíst do paměti počítače volbou "L" DOSu.
- Tlačítka OPTION a SELECT navolit čísla disketových jednotek.
- Vložit zdrojovou disketu a stisknout START.
- Ve vypsaném adresáři listovat klávesou SELECT a klávesou OPTION vybírat soubory ke kopírování. Jméno označeného souboru je zobrazeno inverzními znaky.
- Po výběru stisknout START; proces kopírování začíná.
- Zaměňovat diskety v disketové jednotce dle pokynů programu.
- Přenos dat probíhá potichu. Soubory jsou přenášeny v tom pořadí, v jakém jsou zapsány v adresáři. Nakonec se všechny soubory na cílové disketě uzamknou.

c) Speciální kopírovací programy

Za speciální kopírovací programy lze považovat programy, které při přenosu současně mění formu uložení dat nebo programů na záznamových médiích.

Jsou to například:

- programy pro kopírování a konverzi programů mezi různými nekompatibilními verzemi DOSů. např. 2 → 3; 3 → 2; 2 → 4; 3 → 4;
- programy ke změně disketových sektorů nebo kazetových bloků se 128 byte dat na diskové sektory se 125 byte dat a zpět;
- programy, které při kopírování mění formu programu — konverze mezi formami: boot kazeta, boot disketa, binární soubor, soubor spustitelný File Copierem 1.45, Turbo 2000.

Při přenosu mezi boot disketou, boot kazetou a souborem File Copieru nedochází k žádným problémům, neboť tyto soubory jsou naprosto totožnými soubory dat. Liší se pouze záznamovým médiem (disketa, kazeta) a počtem datových byte v sektoru.

Při přenosu (binární soubor) → (boot disketa, boot kazeta, File Copier) je obvykle změněna jen hlavička několik byte na začátku souboru. Problém je, že některé kopírovací programy neumějí zpracovat binární soubory složené z více segmentů, s několika inicializačními adresami atp.

Nejnebezpečnější je úprava na binární soubor. Většina konverzních programů tento problém řeší rozsáhlejší úpravou. Vzniklý binární soubor se načítá vždy od stále stejné adresy za DOSem, pak se přesouvá na správné místo a spouští. To znamená, že k upravovanému programu je do binárního souboru přidána "move" rutina a soubor se tím prodlouží o více než 256 byte. V tom je skryto nebezpečí: při opakované konverzi — boot soubor → binární soubor a zpět — narůstá program pokaždé o čtvrt kilobyte.

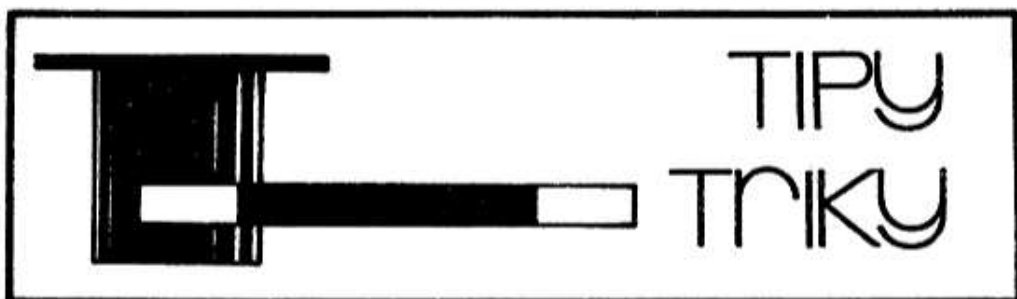
Proto je vždy efektnější provádět konverze manuálně některým monitorem při znalosti strojového kódu a formátu dat na disketě a kazetě.

Poznámka: Někteří uživatelé se ptají, proč se většinou nepovede konverze autoboot diskety do binárního souboru [programem BINARY, SUPER-COPY apod.] nebo do Multibootu.

Z kazet známe programy, které jsou složeny z více částí. Většinou první částí je boot program, který se automaticky zavede a spustí. Další části jsou vlastně datové soubory uložené sekvenčně na pásce obsahující strojový kód dalších rutin, obrazová data, textová data atp. Běžící boot program tyto části nahraje na jemu známé adresy a spustí další program.

Takto je uložen program též na autoboot disketě s tím rozdílem, že v boot programu [který začíná na sektoru 1] musí být určeno, na kterých sektorech začínají další části, jak jsou dlouhé, kde má pak program pokračovat.

Toto nemůže konverzní kopírovací program z hlavičky zjistit a převede tedy do binárního souboru pouze první boot program [mnohdy velmi krátký] a zbytek nechá být. Takovéto disky se dají upravit jen manuálně překopírováním sektorů do více souborů nebo sloučením do jednoho souboru a úpravou V/V operací v programu.



STUDENÝ START POČÍTAČE (potřeti)

ing. Petr Jandík, Praha

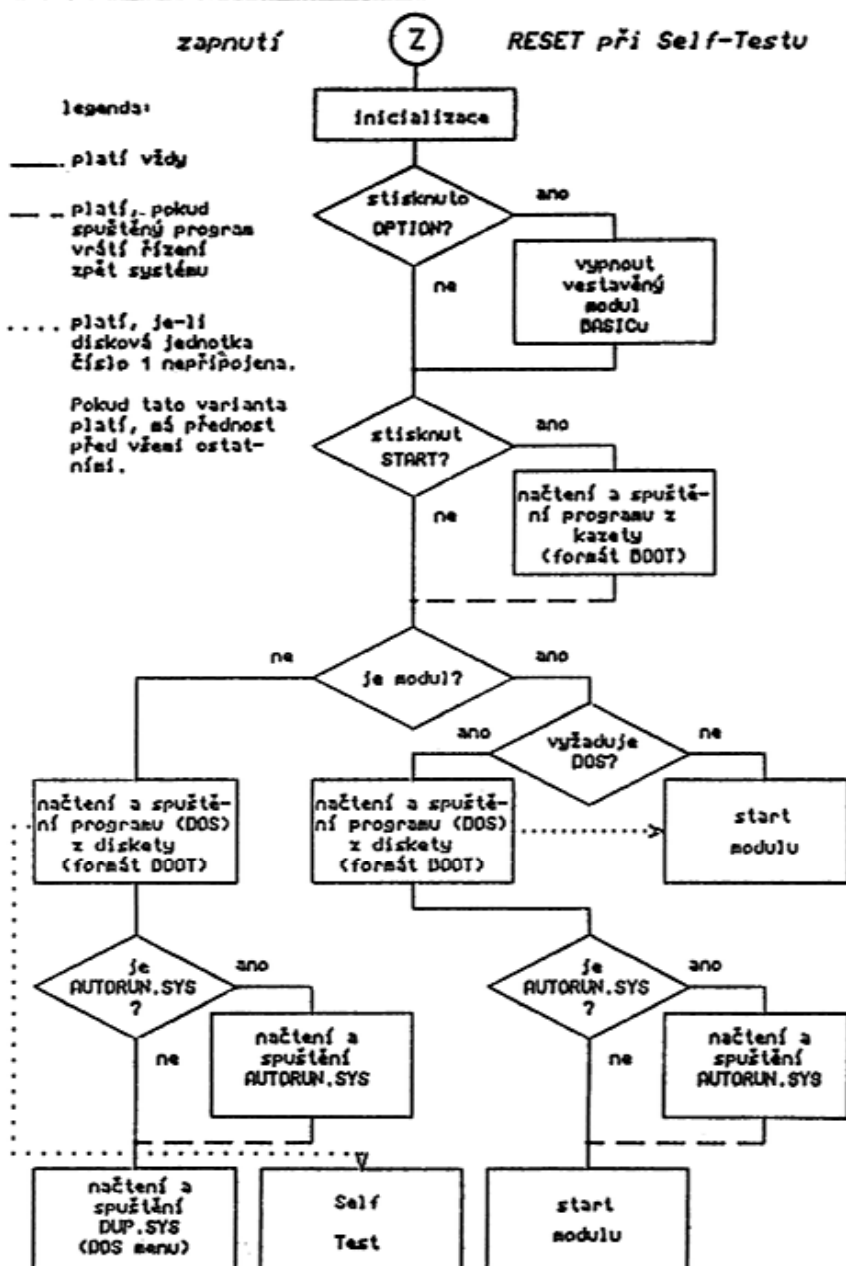
V dokumentaci, která je dodávána k počítači, není dostatečně vysvětlen význam stisknutí kláves OPTION a START při zapnutí počítače. Tyto klávesy patří k základním ovládacím prvkům, které řídí tzv. studený start.

Studeným startem se nazývá základní inicializační pochod, který probíhá po zapnutí počítače. Kromě toho se tento pochod spustí po stisku klávesy RESET při Self Testu, nebo pokud byl závažným způsobem porušen obsah datové základny systému, nebo byl změněn stav zásuvného modulu (zasunut, vyndán, nebo změněn), nebo pokud byl nastavením proměnných WARMST a COLDST studený start vynucen programově. Tento pochod je velmi důležitý, neboť nám umožňuje zavádět do paměti základní programové vybavení, kterým může být hra ve strojovém kódu, obslužný program nového zařízení, DOS a podobně. Podrobný popis dějů při studeném startu by obsáhl několik stránek, omezím se proto pouze na děje podstatné pro uživatele. Zjednodušený průběh je znázorněn na obrázku.

Po zapnutí nebo stisku klávesy RESET za podmínek pro studený start se nejprve provede základní inicializace operačního systému. Při tom se nastaví obsah potřebných proměnných datové základny OS, vytvoří se tabulka obslužných programů periférií atd. Tyto akce nás jako uživatele nemusí zajímat. Potom se testuje, zda je stisknuta klávesa OPTION. Pokud ano, vypne se vestavěný modul BASiCu, který má z hlediska programového stejnou povahu, jako zásuvný modul (cartridge). Dále se testuje, zda bylo stisknuto tlačítko START. Je-li tomu tak, provede OS pokus načíst z kazety strojový program ve formátu BOOT. Pokud

legenda:

— platí vždy

— — platí, pokud
spuštěný program
vrátí řízení
zpět systému.... platí, je-li
disková jednotka
číslo 1 nepřipojena.Pokud tato varianta
platí, má přednost
před všemi ostat-
ními.

Průběh studeného startu počítače

nastane chyba, čtení je po výpisu **BOOT ERROR** přerušeno a počítač se nachází v nedefinovaném stavu, ze kterého se lze dostat jen opakováním studeného startu klávesou **RESET**. Někdy se také objeví **Self-Test**.

Takto zaváděné programové vybavení může být různé povahy. Může to být hra nebo obslužný program pro připojení tiskárny přes konektory ovladačů, zaváděč **Turbo 2000** a podobně.

Pokud končí zavedený program instrukcí **RTS** a předá tak řízení zpět systému, pokračuje se po čárkovaném spoji. Nyní se operační systém podívá, zda je aktivován modul. Může jím být vestavěný **BASIC** nebo externí modul, zasunutý do příslušného otvoru. Není-li modul zjištěn, pokusí se systém zavést základní program z diskety č. 1. Základním programem bývá obvykle **DOS**, ale může to být i zaváděč hry, nebo něco jiného. V našem schématu předpokládáme, že jde o **DOS**. Není-li v tomto okamžiku disketa č. 1 aktivní, tj. zapnutá a schopná komunikace s počítačem, nic se nezavede a studený start končí **Self Testem**. Pokud se **DOS** úspěšně zavedl, dívá se, zda je na disketě soubor s názvem **AUTORUN.SYS** a v kladném případě jej zavede a spustí. Z toho vyplývá, že názvem **AUTORUN.SYS** můžeme pojmenovat jen strojový program ve formátu **DOSu**. Když takový program na disketě není, nebo když nastartovaný program obsahuje povel pro přechod do **DOSu**, načte se a spustí **DUP.SYS**, který realizuje **DOS** menu.

Pokud je modul zjištěn, pokračuje se podle hodnoty v **CARTFG**, která určuje mimo jiné, zda se bude k modulu zavádět nějaká disková podpora. Pokud ne (například u her), modul se nashoduje. Pokud ano, načte se z diskety **DOS** (nebo to, co na ní je ve formátu **BOOT**), a jde-li o **DOS**, a je-li na disketě soubor **AUTORUN.SYS**, načte se a spustí. Pokud tento program vrátí systému řízení, nashoduje se modul. Není-li disketová jednotka č. 1 v okamžiku čtení aktivní, nashoduje se modul bez diskové podpory. Proto nelze v takovém případě s disketou pracovat a pokus o práci s ní končí chybou 130.

NEUČESANÉ POZNÁMKY K PROGRAMU DESIGN MASTER

Imrich Martišek, Bratislava

Tento příspěvek jsme velice rádi přivítali. Zveřejníme i další nápady, související s aplikací tohoto velice užitečného programu. Rovněž bychom rádi přinesli základní návod k užívání programu DESIGN MASTER.

Při tisku obrazů vytvořených pomocí programu **DESIGN MASTER** na tiskárně **ATARI 1029** z volby "**HARDCOPY**", což je součástí programu, nastane jeden problém: Protože tiskárna **ATARI 1029** „chybí osmá jehla“*), znamená to, že každý osmý řádek nebude vytisknutý.

Kolegové z klubu **ENOSOFT** Prievidza dali k dispozici program s názvem "**Hardcopy**" (autorem je L. Kontra), který se načítá do počítače povelom **RUN"C:**". Tento program zajišťuje tisk kompletního obrázku.

Výpis obdobného programu, který načítá obraz uložený na disketě ve standardním tvaru a provede jeho tisk na tiskárně **ATARI 1029** je uveden v listingu č. 1.

Celý proces je možné použít i naopak. Když máme obraz vytvořený v **GR.8** a chceme jej doplnit grafikou, kterou umožňují funkce **DESIGN MASTERu** (např. šrafování, zrcadlové obrazy, různé velké texty apod.), uložíme jej na disketu programem, jehož výpis je uveden v listingu č. 2. Vyvoláme jej v programu **DESIGN MASTER**, podle potřeby jej doplníme a vytiskneme jej výše uvedeným způsobem.

*) O jehlách u tiskárny **ATARI 1029** viz redakční omluva na jiném místě **ZAK**.



```

1 REM *****
2 REM *Program na nactani obrazu z *
3 REM *diskety a vytisteni na A1029 *
4 REM *I.Martisek-Bratislava 6.2.88 *
5 REM *tel. 408304 *
6 REM *****

```

LISTING č. 1

```

9 GRAPHICS 0:POSITION 3,6:? "Program na nactani obrazu z diskety ":POSITION 3,7
?:? "a jeho tisk na tiskarne ATARI"
9 DIM A$(20),B$(20)
10 POSITION 0,8:FOR X=0 TO 39:? "._":NEXT X
20 POSITION 5,15:? "* Vloz disketu s obrazem":POSITION 5,16:? "* Zapni tiskarnu"
21 POSITION 5,17:? "* Az se nahraje DOS, stiskni START"
22 IF PEEK(53279)<>6 THEN 22
35 GRAPHICS 0:POSITION 6,12:? "Zadej nazev obrazu"
36 INPUT B$
44 A$="D":A$(LEN(A$)+1)=B$
27000 GRAPHICS 8+16:COLOR 1
27100 CLOSE #3:OPEN #3,4,0,A$
27110 DIM Q$(30):Q$(1,28)="OhhhBhEhDhIhH Vd"
27120 DL=(PEEK(560)+PEEK(561)*256)+4
27130 MS=PEEK(DL)+PEEK(DL+1)*256
27140 DIR=Z7

```

```

27150 X=USR(ADR(QE$),7,MS,7679)
27160 CLOSE #3
28000 FOR Q=1 TO 300:NEXT Q:GRAPHICS 40:? CHR$(125):? " Jestli chces HARDCOPY, t
ak stiskni START "
28010 IF PEEK(53279)<>6 THEN 28010
28020 ? CHR$(125):? "moment..."
29000 TRAP 31000:S=0:KANAL=1:CLOSE #KANAL:OPEN #KANAL,8,0,"P:"
29001 READ X:S=S+X
29003 POKE A,X
29004 NEXT A
29005 GRAPHICS 56
29009 ? #KANAL;CHR$(27);CHR$(27);CHR$(57)
29010 X=USR(1536,KANAL*16):CLOSE #KANAL
29015 GOTO 28000
30000 DATA 104,104,104,141,255,6,169
30001 DATA 0,133,203,160,0,162,3
30002 DATA 189,2516,32,206,6,202
30003 DATA 16,247,162,7,169,0,157
30004 DATA 0,4,202,16,250,133,204
30005 DATA 169,7,166,203,224,189,208
30006 DATA 1,74,133,207,169,0,133
30007 DATA 205,133,206,165,204,32,169
30008 DATA 6,165,203,32,169,6,24
30009 DATA 165,88,101,205,133,205,165
30010 DATA 89,101,206,133,206,177,205
30011 DATA 162,7,42,72,144,28,138
30012 DATA 72,56,169,6,229,04,170
30013 DATA 169,0,56,42,202,16,252
30014 DATA 133,208,104,170,24,165,208
30015 DATA 125,0,4,157,0,4,104
30016 DATA 202,16,220,230,204,165,204
30017 DATA 197,207,144,168,162,7,189
30018 DATA 0,4,32,206,6,202,16
30019 DATA 247,200,192,40,144,140,165
30020 DATA 207,201,3,240,15,24,169
30021 DATA 7,101,203,133,203,169,155
30022 DATA 32,206,6,76,10,6,169
30023 DATA 1,133,212,169,0,133,213
30024 DATA 96,133,208,138,72,169,0
30025 DATA 133,209,166,208,240,11,24
30026 DATA 105,40,144,2,230,209,202
30027 DATA 76,179,6,24,101,205,133
30028 DATA 205,165,206,101,209,133,206
30029 DATA 104,170,96,133,208,138,72
30030 DATA 152,72,174,255,6,169,11
30031 DATA 157,66,3,169,0,157,72
30032 DATA 3,157,73,3,165,208,32
30033 DATA 86,228,132,208,104,168,104
30034 DATA 170,165,208,16,7,104,104
30035 DATA 165,208,76,162,6,96,64
30036 DATA 1,65,27
31000 GRAPHICS 40:? ? " Neni pripojena tiskarna ":FOR Q=1 TO 400:NEXT Q:? CH
R$(125):GOTO 28000

```

Využíváním části obrazové paměti můžeme vkládat do obrazu část nebo části jiných obrazů. To umožňuje kombinovat obrazy nebo jejich části vytvořené jinými programovými prostředky. Ukázkou tohoto postupu jsou mapky ČSSR.

```

1 REM *****
2 REM *Nacítání obrazu z disku      *
3 REM *I. Martisek-Bratislava      *
4 REM *****

```

LISTING č. 2

```

23000 GRAPHICS 40:? "Zadej nazev obrazu":DIM A$(15),B$(15):INPUT B$
29010 A$="D":A$(LEN(A$)+1)=B$
28100 CLOSE #3:OPEN #3,8,0,A$
28110 DIM QE(30):QE$(1,28)="OhhhBhEhDhIhH Vd"
29120 DL=(PEEK(560)+PEEK(561)*256)+4
28130 MS=PEEK(DL)+PEEK(DL+1)*256
28150 X=USR(ADR(QE$),11,MS,7679)
28160 CLOSE #3:GRAPHICS 40:? "ulozeni na disk OK"

```

Jak upozornil recenzent, načtení obrázku z diskety i kazety lze i modifikovaným programem uvedeným zde v listingu č. 2.

Takový program vyžaduje změnit řádek 28100 na

28100 CLOSE #3:OPEN #3,4,0,A\$

a řádek 28150 změnit na

28150 X=USR (ADR(QE\$),7,MS,7679).

JEDNODUCHÝ RAMDISK PRO 800 XL

Jan Viček, CSc., Praha

V [1] byl uveden program, který využívá části paměti RAM jako periferie označené R:. Lze používat všechny běžné příkazy vstupu a výstupu, nejen SAVE a LOAD, jak je uvedeno. Bohužel práci ztěžují drobné chyby (stiskneme-li během přenosu některou klávesu včetně BREAK, může dojít k zamknutí systému a ztrátě zapsaných souborů, dále lze sice vytvořit 10 souborů, ale z desátého nelze číst), nedostatečné ošetření chybových stavů (IOCB otevřený pouze pro čtení není chráněn proti zápisu, chyba uživatele při otevírání IOCB způsobí často poškození vytvořených souborů a jejich adres, což obvykle zjistíme pozdě) a nepříjemná omezení (zákaz RESET, chceme-li vytvořit 11. soubor, musíme zničit všech 10 předchozích, potíže s grafickými módy).

Proto předkládám nový program, který má tyto vlastnosti:

- je kratší a umožňuje využití o 1,5 KB více RAM,
- pracuje ve všech grafických módech,
- RESET nezkaží ani program, ani vytvořené soubory kromě toho, který je právě otevřen pro zápis; po RESET je však nutno znovu definovat periferii R: pomocí ? USR (1549),
- lze pracovat s 9 soubory označenými R1: až R9:, R: má stejný význam jako R1:; lze vytvořit i soubor s číslem menším než naposled vytvořený, přičemž vytvořením souboru Ri: neztřácíme soubory s číslem menším než i, kdežto soubory s číslem větším než i se stanou nepřístupné pro čtení (viz pozn. 2).

K dispozici jsou následující skupiny příkazů:

- a) O. # k,4,0,"Ri:" otevře IOCB č. k pro čtení ze souboru,
O. # k,8,0,"Ri:" otevře IOCB č. k pro zápis do souboru. Číst lze z kteréhokoliv vytvořeného souboru (přístupného), při zápisu může být číslo souboru nejvýše o 1 větší než číslo naposled vytvořeného souboru.
Současné nesmí být otevřen více než jeden IOCB pro periferii typu R:. S takto otevřeným IOCB lze běžně pracovat pomocí příkazů ?, PR., I., GET, PUT a nakonec je uzavřít CL. k nebo END.

Je-li IOCB č. k otevřen pro Ri: pak po **ST.#k,A** je v A okamžitý počet přístupných vytvořených souborů; stejný účinek má příkaz **A=PEEK(1548)**, na rozdíl od předchozího příkazu však zde nemusíme žádný IOCB otevírat.

- b) **L."Ri:"** nebo **L."Ri:", č. řádku** nebo **L."Ri:",od,do** vytvoří soubor, který můžeme číst **E."Ri:"** (tak lze program snadno rozdělit na více samostatných částí).
- c) **S."Ri:"** vytvoří soubor s přeloženým programem, který lze číst **LO."Ri"** nebo, chceme-li program současně odstartovat, **RUN"Ri:"**. Pozor! Takto vytvořený soubor **nesmíme číst E."Ri:"** — může dojít ke krachu, který už nelze pomocí **RESET** napravit!
- d) **X. č. příkazu**, další parametry stejné jako u **OPEN** — činnost dle čísla příkazu.

Po načtení programu dáme **G.32700** (nebo **RUN**, je-li to jediný program v paměti), případně předem upravíme řádek 32703. Řádek 32733 slouží k tisku 2-bytových adres začátků přístupných souborů; poslední je o 1 zvětšená adresa konce posledního souboru.

Program nejen zvětšuje použitelnou RAM o 22 KB — je vhodný např. pro studium struktury souborů, při současném kopírování více programů v **BASICu**, při verifikaci nahrávky na kazetu pouhým načtením (pozor však na výpadek proudu, to je ale bolest všech verifikací), při rozdělení výstupu na dva — na obrazovku a do souboru Ri:, při delším výpočtu pro ukládání výstupu, např. nechceme-li stále sledovat obrazovku, při zpracování rozsáhlých programů po částech atd.

```
32700 REM *** RAM-DISK 22K - VERSE PRO 800XL BEZ DISKET * VLCEK,1988 ***
32701 RESTORE 32710:S=0:FOR Q=1 TO 238:READ W:S=S+W:POKE Q+1535,W:NEXT Q
32702 IF S<>26919 THEN ? "CHYBA DAT,KONTROL.SUMA SE LISI O ";S-26919:END
32703 Q=USR(1548):LIST "R:";NEW
32710 DATA 46,6,99,6,131,6,152,6,226,6,29,6,0,104,162,223,189,59,2,240
32711 DATA 10,201,82,240,5,232,232,232,208,242,96,157,60,2,133,69,169,82
32712 DATA 157,59,2,169,6,157,61,2,96,165,69,240,3,160,161,96,174,12,6
32713 DATA 165,42,201,4,240,5,232,201,8,208,6,228,33,176,3,160,160,96
32714 DATA 201,8,165,33,144,3,141,12,6,10,170,189,234,6,133,67,189,235,6
32715 DATA 133,68,165,46,208,27,165,69,240,25,165,42,201,4,240,15,173,12
32716 DATA 6,10,170,165,67,157,236,6,165,68,157,237,6,169,0,133,69,160,1
32717 DATA 96,165,33,10,170,165,67,221,236,6,208,37,165,68,221,237,6,208
32718 DATA 30,160,136,96,166,69,188,74,3,192,8,240,3,160,135,96,120,160
32719 DATA 0,140,14,212,162,254,142,1,211,145,67,208,13,120,160,0,140,14
32720 DATA 212,162,254,142,1,211,177,67,202,142,1,211,136,140,14,212,88
32721 DATA 230,67,208,179,230,68,165,68,224,208,240,7,224,0,208,167,160
32722 DATA 162,96,162,216,134,68,48,158,172,12,6,96,0,0,0,0,0,160
32723 FOR Q=0 TO PEEK(1548):? PEEK(1772+Q+Q),PEEK(1773+Q+Q):NEXT Q:END
```

Poznámky:

1. Program je určen pro **ATARI 800 XL** bez disketové jednotky, protože používá paměti na adresách 67, 68, 69 vyhrazených pro **FMS** (jiné paměti stránky 0 nepoužívá). Při případném užití s disketovou jednotkou stačí všechna čísla 67, 68, 69 ve výpisu nahradit jinými vhodnými, např. 203, 204, 205, přičemž paměti 67, 68 nutno nahradit dvojicí po sobě jdoucích paměti.
2. Při zápisu do souboru Ri: jsou pro čtení znepřístupněny všechny soubory s číslem vyšším než i. To je obecně nutné, ale v odůvodněných případech je lze snadno zpřístupnit opravou registru 1548, kde je uložen okamžitý počet přístupných souborů; při opravě nesmí být otevřen žádný IOCB pro periférii typu Ri: Toho lze např. využít při vzájemné záměně R1: za R2: (použijeme pomocný soubor, např. R3:) apod. Je však nutno mít přehled o tom, které soubory jsou částečně zkažené a nelze je tudíž normálně používat.

Literatura:

- [1] Strnad, P.: **RAMDISK pro ATARI 800 XL**, Zpravodaj AK, 1987, 4, s. 24

VRACÍME SE K BLOKŮM IOCB

V době, kdy již byl v tisku příspěvek „Bloky IOCB a co s nimi“ — autor JABS [viz ZAK 1/88, s.22—26], dostali jsme ještě jeden příspěvek na toto téma od ing. Františka Komína. Původní příspěvek proto doplňujeme o nové informace, pro zjednodušení formou oprav a doplňků cit. článku.

1. Tabulka uprostřed stránky 23 lze doplnit následovně:

Operace	ICCMD	Operace	ICCMD
.	(3)	.	(17)
.	(5)	.	(18)
.	(7)	.	(32)
.	(9)	.	(33)
.	(11)	.	(35)
.	(12)	.	(36)
.	(13)	.	(37)
SPECIAL	(14)	.	(254)

2. Na str. 24 na konci bodu 4 lze doplnit:

bit 7 = 1 kontinuální čtení/zápis (CSAVE)
 = 0 blokové čtení/zápis (LIST)

3. Na str. 24 na konci bodu 5 lze doplnit poznámku:

(standardní option handleru, použití závisí na tiskárně)

4. Na str. 25 lze v textu pod čarou ve druhém řádku v levém sloupci za slovo POKE vložit „z BASICu“.

5. Na str. 26 před odkaz na použitou literaturu, kterou lze doplnit o

— Reschke, J., Wiethoff, A.: Das ATARI Profibuch

— Operating System, User's Manual,

je vhodné zařadit poznámku převzatou z příspěvku ing. Františka Komína:

Poznámka:

Pro psaní grafických a jiných znaků ve výpisech programů se používá následující konvence (viz totéž číslo Bajtku):

Inverzní znaky — píší se podržené.

Grafické znaky, psané současným stisknutím CONTROL a znaku — píší se v hranatých závorkách. Ostatní znaky:

text	klávesy	text	klávesy
[CLEAR]	ESC SHIFT <	[GORA]	ESC CTRL -
[DOL]	ESC CTRL =	[LEWO]	ESC CTRL +
[PRAVO]	ESC CTRL *	[WSTECZ]	ESC DELETE
[USUN]	ESC CTRL DELETE	[WSTAW]	ESC CTRL INSERT
[DEL]	ESC SHIFT DELETE	[INS]	ESC SHIFT INSERT
[TAB]	ESC TAB	[CTRL TAB]	ESC CTRL TAB
[SET TAB]	ESC SHIFT TAB	[2]	ESC CTRL 2



red

OTÁZKY ODPOVĚDI



Na otázku

co je to programovací jazyk C nám poskytl odpověď Lubomír Pospíšil, odborník z nepovolanejších, kterému je náš ATARI KLUB zavázán mnohým.

Děkujeme.

Nebude řeč o barevných "cečkách", která sbíraly před časem naše děti, nýbrž o podobně módní vlně, kterou jsou dnes kompilátory C—jazyka. I my vám dnes nabízíme výměnu: bejziky za jedno tvavomodré céčko, tedy přesněji za "Deap Blue C", neboli kompilátor C—jazyka, či lepší "Lightspeed C" od firmy Clearstar Softechnologies. Tato jakoby výměna by vám měla dodat chut' začít programovat v jazyce C.

Co nám jazyk C nabízí a proč vlastně vznikl?

C-jazyk vytvořil D. Ritchie v Bellových Laboratořích v r. 1972. Každý nový jazyk vznikl s jistým cílem. Jazyk C je určen pro profesionální programátorskou práci při vývoji komerčního softwaru. Na první pohled čtenáře patrně vyděsí pohled na demonstrační výpis programu v jazyce C, neboť instrukce jen málo připomínají to, co jsme dosud používali ve starém dobrém BASICu. C—jazyk je však považován za jeden z moderních jazyků, který zahrnuje v sobě všechny požadavky kladené na jazyk dnešní teorii počítačů.

Vede programátora ke strukturovanému a modulárnímu programování, jehož výsledkem je srozumitelný kód snadno přenosný mezi různými typy počítačů, které ani nemusí pracovat se stejnou CPU. Uvádí se, že dnes většina kompilátorů a interpretačních jazyků je psána v C—jazyce právě kvůli snadné přenosnosti na budoucí verze či modely počítačů.

Jaké vybavení je třeba vlastnit, abychom mohli činit první pokusy ponořit se u své klávesnice do tohoto "profesionálního" řečiště?

Jsou to:

- Atari XL, XE alespoň s 48 kB,
- disketovou jednotku,
- C-kompilátor "Deep Blue C" nebo "Lightspeed C" pro Atari,
- editor textu pro psaní textu programu určeného ke kompilaci,
- linker kompilovaných programů.

Editor - kompilátor - linker jsou zpravidla zahrnuty do souboru prodávaného vcelku. Připomeňme, že před časem u nás šířený Deep Blue C měl na disketě vadný file linkeru CLINK a nedal se proto použít. Vhodný editor je např. Atari Program - Text Editor u nás šířený pod názvem MEDIT. Editor totiž nesmí automaticky přidávat čísla řádek nebo kontrolní znaky, které by při kompilaci působily chyby.

Pokusme si nyní předvést, jak probíhá vytvoření krátkého programu, který by pro demonstraci na obrazovce napsal jednoduchý text. Opište pečlivě následující program za použití textového procesoru a uchovejte jej na disketě pod jménem TEST.C, jako tzv. zdrojový program. Uživatelé "Lightspeed C" mají na disketě k dispozici file CEDITOR.COM, který poslouží k tomuto účelu. Výpis programu neuzivá čísla řádek a mezery nebo volné řádky slouží jen pro lepší čitelnost bez významu pro kompilátor:

```
main()
($
printf("Toto je první program v C-jazyce");
$)
```

V další fázi zrušte v paměti editor a zaveďte z diskety kompilátor, který je vždy jako

file CC.COM. Ten se vás již sám zeptá na jméno filu, který má být zkompilován. Nejde-li ke zjištění chyby syntaxe, tak kompilátor na disketě vytvoří nový file TEST.CCC a ten se označuje jako "object file". Object file není ještě možno spustit, protože k němu musí být doplněny standardní funkce jazyka a to procesem nazývaným "linkování". V našem prvním programu se bude linkovat funkce printf() určená pro výstup textu. V jiných programech to budou třeba funkce pro sčítání, sinus apod.

Linkování provedeme pro Deep Blue C tím způsobem, že pomocí editoru MEDIT napíšeme file TEST.LNK:

```
TEST
AIO
PRINTF
DBC.OBJ
```

který bude v procesu linkování dávat pokyny "Linkeru" CLINK.COM, jaké programové moduly mají být spojeny v jeden funkční celek.

Lightspeed C linkuje automaticky po skončení kompilace a volí potřebné standardní programové moduly uvedené výše. Je to input-output (AIO), run-time modul (DBC.OBJ) apod. Musíme zdůraznit, že všechny potřebné fily je třeba mít na stejném disku a využití RAMdisku práci velmi urychluje.

Vyložíme si nyní program:

Každý program v C—jazyce bude rozdělen na řadu funkcí, přičemž jedna z nich bude Hlavní (). Je pro ni vyhrazeno standardně jméno

```
main()
které nelze měnit. V našem programu jsou
závorky prázdné, protože zde funkce nepřijímá ani nevrací žádné parametry. Vlastní
programové instrukce takto zadané funkce
jsou ohraničeny svorkovými závorkami.
Protože na klávesnici Atari nejsou svorkové
závorky k dispozici, implementace C—jazyka
pro Atari zavedla místo nich kombinaci
($ $)
```

V našem programu funkce main() obsahuje jedinou instrukci pro tisk textu, a to

```
printf("...text...");
```

Vidíme, že v případě funkce printf() již pře-

dáváme jako parametr řetězec textu. Název funkce značí "print formatted". Lze vidět jistou podobnost s jazykem Pascal, kde rovněž příkazy končí středníkem a funkce se ohraničují slovy BEGIN a END.

Čtenáři doporučuji program zkusit i přes zdánlivou banalitu, neboť poskytuje možnost celý proces edice, kompilace a linkování bezchybně nacvičit.

Funkce printf dovoluje přerušovat text pomocí "backslash" a jednoho písmene. Zkuste modifikovat první program následujícími verzemi:

```
printf("Toto je první\n program v C-jazyce");
printf("f Toto je\n první\n program v C-jazyce");
printf("f Toto\ndje\nd první\nd program v C-jazyce\n");
```

Povšimněte si přitom pozice kurzoru po skončení. Podrobnosti o kontrolních "escape sequence" najdete v dokumentaci, která se pro obě verze C-jazyka trochu odlišuje.

Light- Deep
speed: Blue:

\n	\n	nový řádek
\f	\f	mazání obrazovky
\b	\b	zpětný krok
deset. oktal.		
kod	kod	ASCII charakter
\t	\t	tabulátor
\d	-	kurzor dolů
\u	-	kurzor nahoru
\l	-	kurzor vlevo
\r	-	kurzor vpravo
\"	\"	uvozovky aj.

Zkuste nyní v jazyce C napsat opět krátký program pro sčítání dvou čísel typu integer (celočíslné hodnoty v mezích $2^{15} = 32\,767$). Opište následující program a postupujte stejně jako v prvním případě:

```
main()
($
int n1, n2, vysl;
printf("Zadej dva scitance:");
scanf("%d %d", &n1, &n2);
vysl = n1 + n2;
printf("\nSoucet cisel %d plus");
printf(" %d jest roven %\n", n1, n2, vysl);
$)
```

Příklad obsahuje povinnou deklaraci typu používaných proměnných. Zde na 3. řádku typ integer pro n1, n2 a výsledek vysl. Řádek 6 je sčítání a přiřazení, které je zřejmé. Nový je typ další standardní funkce scanf() pro vstup dat všech typů. Tato funkce obsahuje jakožto první parametr string udávající formát přijímaných dat. Následuje odpovídající množství pointeru proměnných. Pozor, nestačí jen jméno proměnné, musí předcházet znak &, který indikuje, že se jedná o pointer paměti.

Zbytek příkladu již neobsahuje nic nového.

Nedočkávat čtenářům budiž předem prozrazeno, že jazyk C sestává z malého množství klíčových slov (kolem 30) a je na programátorovi, aby si pomocí nich definoval své funkce. K oběma verzím C—kompilátoru firmy však dodávají řadu hotových funkcí pro grafiku, zvukové generátory atd. Tyto funkce pak lze s výhodou volat i z jiných jazyků (Action, Assembler).



FILE MASTER — NÁVOD NA POUŽITÍ PROGRAMU

(c) ANTIC, Publishing, by Jason Worley;
překlad a úprava Miroslav Ondříšek

Kdo má rád pořádek ve svém softwarovém bohatství, ten jistě ocení program FILE MASTER, jehož autorem je Jason Worley a ke kterému má copyright ANTIC Publishing. Pravda, program má cenu pro toho, kdo má k dispozici kromě počítače i disketovou jednotku a příp. i tiskárnu. Program totiž slouží pro výpis adresářů disket a jejich třídění v pěti zvolených kritériích (podle abecedy, doplňujících znaků [extenderů] apod.).

Původní program byl vytvořen v ATARI BASICu. Autor tohoto příspěvku vedle překladu návodu přepracoval program do TURBO BASICu XL 1.5, což několikrát zrychlilo práci s ním. Upravený program je k dispozici v klubu. Z praktického hlediska uživatele je nutno nejdříve načíst program TURBO BASIC XL 1.5. a k němu pak upravený FILE MASTER, pokud si uživatel na disketě nevytvoří tzv. AUTORUN.

☆ ☆ ☆

Po načtení programu do paměti počítače se zobrazí úvodní hlavička, po ní informace o volných bytech v paměti a nakonec hlavní menu.

Hlavní menu:

- | | |
|---------------------------|---|
| Add Records | - záznamy |
| Display Indexes On Screen | - zobrazení indexů (indexových částí názvů programů) na obrazovce |
| Print Indexes In Memory | - tisk indexů uložených (načtených v paměti) |
| Sort Indexes In Memory | - třídění indexů v paměti |
| Save Indexes On Disk | - zapsání všech indexů z paměti na disketu |

Load Indexes From Disk	- načtení indexů z diskety
Delete Index Disk File	- vymazání souborů se specifikovaným doplňujícím znakem (extenderem)
Delete Index From Memory	- vymazání obsahu diskety specifikovaného čísla z paměti
Clear Memory	- vymazání obsahu celé paměti
Exit Program	- návrat do programovacího jazyku
Display Current Directory	- zobrazení adresáře na aktuální disketové jednotce
Change Drive Number	- změna čísla aktuální disketové jednotky
Records in memory . . .	- oznámení o počtu záznamů v paměti.

Pomocí kurzorových šipek (není nutný současný stisk klávesy CTRL) se vybírá potřebná funkce (je zobrazena inverzně). Funkci zvolíme klávesou RETURN .

1. Add Records

Program s uživatelem komunikuje formou dotazů.

Nejdříve se objeví dotaz, zda chceme vytisknout štítek s adresářem vložené diskety. Odpovídáme "Y" (ano) nebo "N" (ne) a potvrdíme stisknutím RETURN .

Počítač oznámí, že je připraven tisk a vyžaduje další stisknutí RETURN .

Následuje dotaz, zda budeme vkládat označení typu programu. Pokud chceme označení typu vkládat, napíšeme "Y" a stiskneme RETURN . Objeví se dovolený seznam zkratk typů (např. G—Game, D—Database, M—Music atd.).

Dále na dotaz zadáme číslo diskety, ze které budeme číst adresář (záleží na uživateli, jak si diskety očísluje) nebo slovo "END" pro návrat do hlavního menu.

Program nyní vypisuje názvy souborů a vyžaduje vložení jedné z povolených zkratk označení typu programu (soubory DOS.SYS a DUP.SYS jsou ignorovány).

Pokud jsme tuto funkci odmítli, zobrazí se jen adresář a tento je načten do paměti.

Toto se opakuje s různými disketami (které postupně vkládáme do disketové jednotky) tak dlouho, dokud nezadáme slovo "END" místo čísla diskety nebo dokud je v paměti místo.

2. Display Indexes On Screen

Po zvolení této funkce se vypíše tabulka všech názvů souborů, které byly načteny z disket y následující formě:

1. sloupec — číslo diskety
2. sloupec — název programu (až 8 znaková indexová část)
3. sloupec — doplňující část (extender) - 3 znaky
4. sloupec — velikost souboru (počet sektorů)
5. sloupec — typ programu.

Typ programu se samozřejmě vypisuje (celým slovem) jen tehdy, když jsme jej zadali v režimu Add Records.

Pokud je výpis delší, než se vejde na jednu obrazovku, zastaví se a pokračuje až po stisknutí klávesy RETURN . Konec výpisu je signalizován nápisem "End of File". Stiskem klávesy RETURN se vrátíme zpět do hlavního menu.

3. Print Indexes In Memory

Tento příkaz způsobí vypsání všech souborů uložených v paměti na tiskárnu, pokud je připojena.

4. Sort Indexes In Memory

Tímto příkazem si v paměti můžeme soubory seřadit podle pěti různých hledisek (abecedně, resp. podle délky souborů):

- Extension — třídění podle doplňující části (extenderu)
- Program Type — třídění podle zadaných označení typu programu
- Disk Number — třídění podle zadaných čísel disket
- Alphabetically — třídění podle názvů programů (indexovaných částí)
- File Length — třídění podle délky souborů.

Typ třídění se vybírá kurzorem stejně jako v hlavním menu. Po výběru se zobrazí přibližný čas, po který bude třídění trvat. Poté lze vypsat obsah paměti funkcí „Display Indexes On Screen“ nebo „Print Indexes In Memory“.

5. Save Indexes On Disk

Celý obsah paměti se zapiše na disketu. Po dotazu "Enter 3-Digit Extension For Filename" zadejte maximálně 3 znaky, které program použije jako extender v názvu souboru, který se jmenuje PROGINDX. Zadáte-li tedy např. AAA, bude váš program zapsán pod názvem PROGINDX.AAA.

Po skončení zapisování se zobrazí hlášení označující kolik záznamů se zapsalo a kolik se — z jakéhokoli důvodu — nezapsalo.

6. Load Indexes From Disk

Je to inverzní funkce k funkci předešlé — k obsahu paměti se přičte obsah souboru, který specifikujete.

7. Delete Index Disk File

Tato funkce smaže soubor PROGINDX s vámi specifikovanou doplňující částí (extenderem), pokud na dotaz odpovíte "Y". Po volbě je nutné nejdříve uvést označení extenderu a potom na výzvu napsat "Y".

(Poznámka: činnost funkcí 6. a 7. si nejlépe zkontrolujete prostřednictvím funkce 1.)

8. Delete Index From Memory

Tímto způsobem můžete z obsahu paměti vymazat obsah diskety, jejíž číslo specifikujete. Při výpisu však tyto záznamy nezmizí, jsou jen označeny hvězdičkou. Teprve když soubor "uschováte" na disketu, takto označené záznamy se nehrají.

9. Clear Memory

Tato funkce smaže obsah celé paměti a provede inicializaci programu.

10. Exit Program

Touto funkcí opustíte program a vrátíte se do programovacího jazyku, ve kterém jste program spustili.

11. Display Current Directory

Zobrazí adresář na aktuální disketové jednotce.

12. Change Drive Number

Umožňuje změnit číslo aktuální disketové jednotky.

STAP — PROGRAM PRO TISK POPISKY KAZETY

OLVIS

Vzpomínáte si? V úplně prvním čísle ZAK (1/87, s. 23—24) jsme nesměle navrhli normalizovaný způsob zápisu údajů na popisný štítek kazety. Slibili jsme tehdy, že čtenáře seznámíme s konečným návrhem podle připomínek, které jsme očekávali. Přišly jenom dvě. Navržená forma zápisu se ale vžila alespoň u té pečlivější části členů klubu.

Jednodušší způsob evidence programů na popiskách kazet vyžaduje systém TURBO 2000, resp. SUPER TURBO, kde se vžilo pouze uvedení názvu programu a jeho začátek [podle otáček počítačů magnetofonu]. Opět jen ti pečlivější si poznamenávají ještě nejvhodnější Loader. Popiska pro "turbo" kazetu však vyžaduje hodně řádků. To nespĺňuje jediná firemní popiska kazet, které se u nás vyskytují.

Již před časem jsme požádali naše spolupracovníky v tiskárně o výrobu vhodných popisů pro "turbo" kazety, ale zatím jsme se nedočkali. Pro ty, kteří mají přístup k počítačové tiskárně přinášíme program STAP, vytvořený autorem původního normalizačního návrhu zápisu na popisky kazet. Program vhodně zapadá do miniseriálu "evidenčních programů", které zveřejňujeme v tomto a příštím čísle ZAK.

Program STAP je vytvořen v ATARI BASICu. Po jeho načtení do paměti počítače a spuštění pomocí RUN je na zobrazenou výzvu možné napsat „označení uživatele“. Musí se vejít do max. 35 znaků. Následuje RETURN. Obrazovka je pak připravena pro vkládání údajů prvního programu, a to ve standardní (normalizované) formě:

- | | |
|------------------------------|----------------------|
| 1. název programu | (text max. 20 znaků) |
| 2. začátek programu na pásce | (číslo max. 3 znaky) |
| 3. způsob načítání (LOAD)+ | (údaj max. 4 znaky) |
| 4. způsob kopírování (F)- | (údaj max. 1 znak) |
| 5. počet křížových ovladačů | (údaj max. 1 znak) |

Je dlužno dodat, že program pracuje interaktivně, tzn. že ihned po zadání všech údajů k jednomu programu vytiskne celý řádek. Řádek odpovídá šíři popisů kazety. Údaje tištěné v řádcích pak respektují výšku kazety (přehyb). Znamená to, že před zahájením práce je nutné připojit a zapnout tiskárnu a vložit do ní papír. Práce s programem je snadná a srozumitelná.

Konec vkládání (a tisku) údajů o programech na kazetě se ukončí volbou "N" po výzvě „Konec popisů?“ a stiskem RETURN.

ATARI®

Klub Praha

```

90 REM *** STAP 1.1 ***
92 REM *** OLVIS 88/03 ***
200 DIM R$(40),U$(30),N$(20),Z$(3),L$(4),F$(1),J$(1),X$(40),Q$(40),I$(45)
215 ? CHR$(125):POKE 752,1:POKE 16,64:POKE 53774,64:R=4:L=4
220 DL=PEEK(560)+256*PEEK(561):POKE DL+3,70:POKE DL+6,6
225 GOSUB 450
230 POSITION 4,5:? "OZNACENI UZIVATELE(max.35 mist)":POSITION 34,8:? "#":POSITIO
N 4,8:INPUT I$:U$=I$
240 IF LEN(I$)>30 THEN GOSUB 550:? CHR$(125):GOTO 225
260 R$(1)="":R$(40)="":R$(2)=R$:X$(1)=" ":X$(40)=" ":X$(2)=X$:Q$=X$
270 N$(1)="":N$(20)="":N$(2)=N$:Z$=N$:L$=N$:F$=L$:J$=L$
290 LPRINT R$:X$(1)="":X$(3)=U$:X$(40)="":LPRINT X$:X$=Q$
310 X$(1)="":X$(3)="C": ZAC LOAD*F*J":X$(40)="":LPRINT X$:L
PRINT R$
315 ? CHR$(125):GOSUB 450
320 X$(3,7)="NAZEV":POSITION 3,1:? X$(3):POSITION 1,2:? R$(3)
325 IF R>4 THEN 332
330 POSITION 0,3:? N$:POSITION 24,3:? Z$:POSITION 30,3:? L$:POSITION 35,3:? F$:P
OSITION 37,3:? J$
332 POSITION 0,R:INPUT N$:IF N$(1)=" " THEN GOSUB 500:GOTO 332
334 POSITION 23,R:INPUT Z$:POSITION 29,R:INPUT L$:POSITION 34,R:INPUT F$:POSITIO
N 36,R:INPUT J$
360 X$=Q$:X$(1)="":X$(3)=N$:X$(24)=Z$:X$(30)=L$:X$(35)=F$:X$(37)=J$:X$(40)="":
LPRINT X$:R=R+1:L=L+1
365 IF R=21 THEN R=4:? CHR$(125):GOSUB 450
370 N$(1)=" ":GOTO 332
400 IF L>16 THEN 440
420 FOR X=L TO 15:X$=Q$:X$(1)="":X$(40)="":LPRINT X$:NEXT X
421 LPRINT R$:LPRINT X$:X$(8)=U$:X$(40)="":LPRINT X$:LPRINT R$
422 POSITION 10,22:? "DALSI POPISKA? A/N":GOSUB 550
424 OPEN #2,4,0,"K":GET #2,X:CLOSE #2
426 IF X=32 OR X=65 THEN R=4:L=4:GOTO 260
428 IF X=78 THEN GRAPHICS 0:END
429 GOTO 422
440 LPRINT R$:GOTO 422
450 POSITION 0,0:? "_____STAP 1.1_____** popiska pasky **"
460 RETURN
500 POSITION 10,22:? "KONEC POPISKY? A/N":GOSUB 550
510 OPEN #2,4,0,"K":GET #2,X:CLOSE #2
520 IF X=32 OR X=65 THEN 400
530 IF X=78 THEN POSITION 10,22:? " ":RETURN
540 GOTO 510
550 SOUND 0,50,10,9:FOR W=1 TO 50:NEXT W:SOUND 0,0,0,0:RETURN

```

COPY . TSD - návod ke kopírovacímu programu

Petr Vacek, Jan Pilný - Praha

1. Úvodní poznámky

Kopírovací program COPY . TSD (**Turbo-Standard-Disketa**) umožňuje kopírovat programy ve strojovém kódu celkem šesti směry:

Turbo — Turbo
Turbo — Standard
Turbo — Disketa
Disketa — Turbo
Disketa — Standard
Disketa — Disketa

To z něj dělá účinného pomocníka každého uživatele, který používá systém TURBO 2000 (dále jen "Turbo")¹⁾ souběžně se standardním systémem, případně s disketovou jednotkou.

Pomocí volby z nabídkových menu lze vytvářet libovolný počet kopií na různá zařízení. Většina kopírovacích programů je umístěna na nejnižším možném místě v paměti RAM (0480 hex.). Kopírovaný program se pak ukládá do paměti těsně za konec kopírovacího programu. Z toho vyplývají dvě omezení. Kopírovaný program nelze spustit, protože se při kopírování nahraje na jiné místo, než na které je ve skutečnosti adresován. Dále je omezena kapacita paměti, protože část paměti je zabrána samotným kopírovacím programem. Následkem toho nelze kopírovat některé extrémně dlouhé programy. Snaha ušetřit co nejvíce paměti vede ke zkracování kopírovacího programu na co nejmenší délku na úkor grafického provedení, slovních hlášení a celkové dokonalosti a univerzálnosti programu.

Kopírovací program COPY . TSD je koncipován na jiném základě. Využívá paměti „ukryté“ pod operačním systémem. Ta je běžně přístupná po odpojení paměti ROM s operačním systémem. Operační systém je však potřebný pro standardní přístup na periferie (Standard—kazeta, Disketa). Program COPY . TSD je tedy umístěn v paměti RAM místo operačního systému a pro volání systému používá volací rutiny umístěné zpravidla v zásobníku.

Jeho umístění místo operačního systému odstraňuje výše uvedené nedostatky, tedy umožňuje kopírovat programy umístěné v rozmezí adres 0480 až BFFF (hex.) tj. programy o délce do 47 kB. Programy je tedy možno spustit přímo z kopírovacího programu COPY.TSD. Tento způsob kopírování navíc umožňuje kopírovat složené vicedílné soubory z diskety do Turbo či Standardu.

Po nahrání kopírovacího programu do počítače se na obrazovce objeví úvodní hlavička, kterou lze stiskem tlačítka RESET zrušit a tím okamžitě spustit kopírovací program.

V horní části obrazovky se vypíše název programu a jména autorů.

Pod ním je první dialogový řádek. Na něm se zobrazují nabídkové menu při volbě funkce uživatelem, některá hlášení o prováděných funkcích a údaje (názvy) vkládané uživatelem.

Dále následují tři informační řádky. V prvním se zobrazuje kazetový název kopírovacího programu, v druhém pak disketový název. Tyto názvy nejsou většinou shodné, protože disketový název je omezen známým formátem. V třetím informačním řádku se zobrazuje délka kopírovacího programu v bytech v desítkové soustavě.

Ve spodní části obrazovky je druhý dialogový řádek, ve kterém se zobrazují některá chybová hlášení nebo výpis obsahu adresáře diskety. Zobrazování v tomto řádku je provedeno jemným posunem textu vlevo (finescroll).

1) Richter, J.: Systém zrychleného přenosu dat TURBO 2000. Zpravodaj ATARI KLUBU Praha, 1987, příloha II/87.

2. Nahrávání programů ve formátu Turbo nebo z diskety do počítače

Po spuštění programu COPY.TSD se v prvním dialogovém řádku objeví vstupní menu "NAHRÁVANI — 1)TURBO 2)DISK D=DIRECTORY".

Po volbě probíhá nulování celé paměti. To se projeví malým zdržením mezi stiskem tlačítka a začátkem vykonávání zvolené funkce.

2.1. Nahrávání programů v Turbo

Volba nahrávání programu ve formátu Turbo do paměti počítače se provede stisknutím klávesy 1. Po spuštění magnetofonu (PLAY) obrazovka ztmavne (zmizí obraz) až do nalezení některé hlavičky programu uloženého na pásce. Magnetofon se zastaví, v prvním dialogovém řádku se zobrazí název nalezeného programu.

Nyní máme dvě možnosti:

- a) Pokud chceme tento program kopírovat, stiskneme tlačítko RETURN. Od toho okamžiku se rozběhne magnetofon a program je nahráván do počítače.
- b) Pokud stiskneme jiné tlačítko než RETURN, magnetofon se spustí, ale program není nahráván. Kopírovací program hledá hlavičku dalšího programu uloženého na pásce.

Po nahrání programu do počítače se magnetofon zastaví a název nahraného programu se zobrazí v prvním informačním řádku za nápisem "Kazetovy nazev". Ve třetím informačním řádku je zobrazena délka nahraného programu v bytech. V dialogovém řádku je výstupní menu.

2.2. Nahrávání programů z diskety

Stiskneme-li ve vstupním menu klávesu 2, vybereme funkci nahrávání z diskety. V dialogovém řádku se objeví výzva "Nazev:", po které zadáme z klávesnice název programu, který chceme kopírovat a stiskneme RETURN. Název musí splňovat pravidla formátu názvu DOSu. Název je možno zkracovat používáním znaku * (wild card). Následuje stisknutí RETURN.

Pokud není program na disketě nalezen, je vysláno hlášení "Soubor nenalezen" a po stisku libovolné klávesy se vrátíme do vstupního menu.

Pokud není disketová jednotka připojena, není zapnuta nebo není vložen disk, je zobrazeno chybové hlášení "Disketa není připojena".

Pokud byl program se zadaným jménem nalezen, začne okamžitě probíhat jeho nahrávání do počítače.

Po skončení nahrávání je ve druhém informačním řádku s nápisem "Disketovy nazev" zobrazen název nahraného programu, v dalším řádku jeho délka v bytech a v dialogovém řádku výstupní menu.

V tomto místě je třeba se zmínit o mechanismu, jakým jsou soubory kopírovány z diskety. Soubory zapsané na disketě ve formátu DOSu se obecně mohou skládat z libovolného počtu bloků. Pokud kopírujeme z diskety soubor jednoblokový, nesetkáme se s žádným problémem.

U víceblokových souborů je problém kopírování řešen následovně: Jednotlivé bloky souboru se nahrají do paměti na svá místa a za výsledný program je považován jeden blok, jehož počáteční adresa se rovná nejnižší počáteční adrese ze všech dílčích bloků a koncová adresa se rovná nejvyšší koncové adrese ze všech dílčích bloků.

Určitý nedostatek tohoto způsobu kopírování se projeví při kopírování např. dvoublokového souboru, jehož jeden blok je adresován do spodní části RAM (např. 1000 až 1400 hex.) a druhý blok je adresován do horní části RAM (např. B000 až B400 hex.). Po spojení těchto dvou bloků, kde délka každého z nich je 1 kB (celkem je tedy délka souboru 2 kB), v jeden blok s počáteční adresou 1000 a koncovou adresou B400 (hex.) je délka tohoto bloku 41 kB.

Kopírování takového programu pomocí programu COPY.TSD je tedy velice neefektivní

a vyplatí se maximálně pro kopírování do Turbo, kde vzhledem k relativně vysoké nahrávací rychlosti není zvětšení délky tolik na závalu. Samozřejmě nemá smysl takový soubor kopírovat z diskety na disketu kde by zabral mnoho místa nebo do Standardu, kde by se nahrávací doba značně prodloužila.

Nicméně tento způsob kopírování plně vyhoví pro soubory, jejichž bloky nejsou od sebe příliš vzdáleny. Mimo jiné je také umožněno kopírování programů vzniklých překladem assembleru, které bývají v souborech rozdělených na bloky o určité délce, která záleží na typu assembleru. Typickým příkladem je např. assembler MAC 65, který při překladu vytváří soubor po blocích o délce 252 bytů, z nichž každý má svoji hlavičku.

Další komplikací při kopírování z diskety je určení startovací adresy programu. Startování programu je v souborech ve většině DOSů zajištěno tak, že startovací adresa je v souboru zadána jako samostatný blok o délce 2 byty s umístěním do paměti na adresu 02E0 nebo 02E2 (hex.) (inicializační nebo startovací adresa). Víceblokový soubor může být inicializován nebo spouštěn i vícekrát mezi nahráním jednotlivých bloků. V zásadě lze kopírovat jen programy s jedinou startovací adresou (je jich většina), protože kazetová nahrávka, jak ve Standardu tak zejména v Turbo, neumožňuje vícenásobné dílčí spouštění částí programu.

Kopírovací program COPY.TSD hlídá zda soubor startovací adresu vůbec má. Pokud soubor tuto adresu nemá, pak je v prvním dialogovém řádku vypsané hlášení "SOUBOR NEMA STARTOVACI ADRESU". Po stisku klávesy se objeví v témže řádku další hlášení, resp. dotaz: "RET: STARTOVACI ADR. = POCATECNI ADR.". Pokud stiskneme jiné tlačítko než RETURN, pak funkci nahrávání zrušíme, tj. soubor bez startovací adresy nelze kopírovat, a vrátíme se do vstupního menu. Pokud stiskneme po této výzvě tlačítko RETURN, pak — protože soubor nemá startovací adresu — je za startovací adresu vzata počáteční adresa prvního nahraného bloku programu a přejdeme tak do výstupního menu, odkud můžeme program kopírovat. Tato možnost převzetí startovací adresy z adresy počáteční byla zavedena pro kopírování souborů zapsaných v takových DOSech, které tuto variantu připouštějí, tj. soubory se v nich startují automaticky od své počáteční adresy. Tuto vlastnost má například starší DOS.XL.

2.3. Výpis adresáře diskety (Directory)

Stiskem klávesy D ve vstupním menu volíme funkci výpisu adresáře diskety. Podmínkou úspěšného výpisu je připravenost diskové jednotky (připojena, zapnuta, vložena disketa).

Pokud tomu tak není je vypsané ve spodním řádku jemným posunem vlevo chybové hlášení „DISKETA NENÍ PŘIPOJENA !!!“.

Po stisknutí nějaké klávesy se vrátíme zpět do vstupního menu.

Pokud je vše v pořádku, pak se ve spodním řádku vypisují postupně názvy jednotlivých souborů na disketě. Za posledním názvem se ještě zobrazí informace o počtu volných sektorů, která v tomto řádku již zůstane. Výpis adresáře je možno kdykoliv pozastavit přidržetím libovolné klávesy. Po jejím uvolnění výpis opět pokračuje. Po ukončení výpisu následuje návrat do vstupního menu.

3. Zapisování programu do nového formátu

Jestliže byl do paměti počítače úspěšně načten (nahrán) program ve formátu Turbo nebo z diskety, v prvním dialogovém řádku se zobrazí **výstupní menu**:

"1) Start 2) Výstup turbo 3) Standard 4) Disk

Stiskem jedné z kláves 1 až 4 volíme příslušnou funkci.

Kromě těchto čtyř funkcí můžeme navíc stejně jako ve vstupním menu použít klávesou D funkci výpisu adresáře, jak byla popsána v bodě 2. 3.

3.1. Spuštění nahraného programu

Klávesou 1 ve výstupním menu volíme funkci spuštění kopírovaného programu.

Před samotným spuštěním se provede přiklopení operačního systému a povolení přerušení. Program je pak spuštěn od své startovací adresy přes rutinu uloženou v zásobníku. Je třeba podotknout, že ne všechny programy po spuštění z kopírovacího programu bezchybně fungují. Zvláště u her se často můžeme setkat s různými formami blokování, jejichž účelem je zabránit přístupu do samotného programu a ve svém důsledku způsobují, že program touto cestou nelze spustit nebo po spuštění nepracuje tak, jak by měl. Pokud po spuštění funkci 1—Start program nefunguje, neznamená to ještě, že pořízená kopie není funkční.

U některých programů, které nezabezpečují vlastní obsluhu tlačítka RESET (není jich mnoho), je možno se i po jejich spuštění stiskem tlačítka RESET vrátit zpět do kopírovacího programu.

3.2. Výstup programu ve formátu Turbo

Stiskem klávesy 2 ve výstupním menu zvolíme funkci výstupu kopírovaného programu na magnetofon ve formátu Turbo. Ještě před samotným stiskem klávesy 2 musíme připravit magnetofon na nahrávání, tedy vložit kazetu, přetočit pásku na patřičné místo a stisknout tlačítka PLAY a RECORD. Po volbě 2 se zatemní obrazovka a spustí magnetofon. Probíhá nahrávání, které můžeme kontrolovat i poslechem zvuků charakteristických pro Turbo 2000.

Po skončení nahrávání můžeme opět volit další výstup programu.

Programy překopírované do Turbo nahráváme a spouštíme tak, jako všechny programy Turbo, tj. pomocí klasických programových prostředků (Turbo loader) apod.

3.3 Výstup programu ve formátu Standard

Klávesou 3 ve výstupním menu zvolíme funkci výstupu kopírovaného programu ve standardním formátu. Před volbou se musí magnetofon připravit na nahrávání. Po stisku klávesy 3 se zatemní obrazovka, ozve se dvojí charakteristické pípnutí doprovázející nahrávání ve standardním formátu a spustí se magnetofon.

Všechny takto kopírované programy se zapisují na dvě části. První část se skládá vždy ze tří bloků a obsahuje nahrávací program (loader). Druhá část pak obsahuje vlastní kopírovaný program.

Takto vytvořenou kopii ve standardním formátu pak zavádíme do počítače klasickým způsobem, tj. zapnutím počítače se současným stiskem START a OPTION. Nejprve se nahraje úvodní loader (tři bloky), pak se na obrazovce objeví velkými písmeny název programu a pokračuje nahrávání samotného programu. Loader samozřejmě hlídá vznik chyby v záznamu a na chybu reaguje vypsáním hlášení "BOOT ERROR!".

Použití samostatného loaderu pro vytvořenou kopii ve standardním formátu přináší jednak možnost zobrazení názvu programu během jeho nahrávání (což zvyšuje přehledost, neboť jsme ihned informováni, co nahráváme) a umožňuje nahrávání a tedy kopírování programů v celé použitelné oblasti paměti RAM, tedy od adresy 0480 do BFFF (hex.).

3.4. Výstup programu na disketu

Stiskem klávesy 4 ve výstupním menu zvolíme funkci výstupu kopírovaného programu na disketu. Dříve než stiskneme klávesu 4, musíme připravit disketovou jednotku - musí být zapnuta a musí v ní být vložena disketa s dostatečně volnou kapacitou!

Před samotným nahráním programu na disketu dochází k testování názvu programu, zda splňuje předepsaný formát názvu DOSu. Pokud kopírujeme program z kazety, může se snadno stát, že název tomuto formátu neodpovídá, protože formát názvu na kazetě není nijak omezen. Pokud se testováním zjistí, že název programu nemá správný formát, je do

prvního dialogového řádku vysáno hlášení "NAZEV NEVHODNY PRO DOS!". Po stisku libovolné klávesy následuje výzva "ZADEJTE NOVY NAZEV:". Nyní musíme vložit z klávesnice nový název programu tak, aby vyhovoval formátu DOSu. Název ukončíme klávesou RETURN. V ten okamžik se znovu provede testování námi zadaného názvu. Jakmile má název správný formát, je zobrazen na řádku s nápisem "Disketovy nazev" a začne probíhat samotné zapisování na disketu. Zapisování se uskutečňuje standardním způsobem prostřednictvím DOSu 2.5. Při vzniku jakékoliv chyby je vždy zobrazeno příslušné chybové hlášení.

Po úspěšném dokončení zapisování na disketu program COPY.TSD vrátí do výstupního menu, kde můžeme provést novou volbu dalšího výstupu.

Důležité je, jakým způsobem spouštět programy překopírované na disketu. To záleží především na tom, v jakém adresovém prostoru se nachází překopírovaný program. Pokud se program nekryje s DOSem, tedy nachází se **nad adresou 1F00 (hex.)**, pak je možno jej nahrát z DOS menu pomocí funkce L (Binary load). Pokud se program nachází na nižším umístění, jsme nuceni použít jiné programové prostředky²⁾

3.5. Výpis adresáře diskety

I ve výstupním menu lze klávesou D (Directory) vyvolat výpis obsahu adresáře disku, který se provede jemným scrollem do spodního dialogového řádku.

Volání této funkce ve výstupním menu se však doporučuje omezit na minimum, protože při ní dochází k přemazání paměti od 0400 do 0480 (hex.), která je použita jako buffer při čtení adresáře. Přemazání této oblasti by vadilo při kopírování programů umístěných od adresy nižší 0480 (hex.).

4. Návrat do vstupního menu programu COPY.TSD

Návrat do vstupního menu provedeme stiskem klávesy RETURN.

Z ohlasů na praktické používání kopírovacího programu COPY.TSD vyplynulo, že se uživatelům občas stává tato nepříjemná věc: Načte dlouhý program například z kazety. Nyní jej chce nahrát na jinou kazetu. Připraví kazetu a zcela automaticky — snad ze zvyku — stiskne RETURN. Ke svému překvapení zjistí, že program, který již tak dlouho nahrával, je ztracen. Klávesou RETURN se totiž vrátil do vstupního menu, takže mu nezbývá, než program znovu načíst do počítače.

Z těchto důvodů byl kopírovací program COPY.TSD pro nerozvážené uživatele upraven tak, aby bylo možno takovou situaci zachránit. Pokud tedy byla omylem stisknuta klávesa RETURN a zobrazilo se vstupní menu COPY.TSD, můžeme se vrátit zpět do menu výstupního opětovným (druhým) stiskem RETURN.

Pokud však ve vstupním menu stiskneme cokoliv jiného než RETURN, v tu chvíli je již program nenávratně ztracen, protože po volbě ve vstupním menu je celá paměť počítače vynulována.

Jakmile se druhým stiskem RETURN vrátíme zpět do výstupního menu COPY.TSD, můžeme opět zapisovat kopírovaný program ven z počítače.

2) S úspěchem lze použít u nás dost rozšířený a používaný MIKRO-DOS II/D (1984 — Stefan Dorndorf), který umí nahrávat programy od adresy asi 0800 (hex.). Použitelný adresový prostor je však již od adresy 0400 (hex.).

Aby bylo možno kopírovat a nahrávat i programy v tomto umístění, byl vyvinut zvláštní MINIDOS LOADER (1987 — Petr Vacek), který nahrává do celého adresového prostoru paměti. Tento loader je do jisté míry univerzální a lze jej použít pro zavádění takřka všech programů. (Existují výjimky, kdy to nejde, ale není jich mnoho.) Loader se zapisuje na disketu pomocí inicializačního programu. Lze jej zapsat i na disketu, na které již jsou programy, aniž by se tyto poškodily.

5. Důležitá upozornění

V této kapitole jsou uvedeny některé detaily, na které je třeba při používání kopírovacího programu COPY.TSD dávat pozor.

5.1. Nahrávání kopírovacího programu COPY.TSD

Kopírovací program COPY.TSD potřebuje pro svou funkci DOS uložený ve svém adresovém prostoru.

Pokud by program v disketové verzi obsahoval pouze samotný kopírovací program, nebyl by přenositelný na kazetu. Po překopírování takového programu na kazetu by tento po spuštění nepracoval s disketovou jednotkou, protože by postrádal DOS. Proto soubor COPY.TSD na disketě obsahuje kromě samotného kopírovacího programu i DOS. Takový program lze bez problémů uchovávat na kazetě (ať už ve Standardu nebo v Turbo) i na disketě.

Protože při nahrávání kopírovacího programu z diskety přehrává DOS sám sebe, není možno disketovou verzi nahrávat z DOSu funkcí Binary load. Kopírovací program musíme umístit pod MIKRODOS nebo MINIDOS LOADER, odkud jediné může být nahrán a spuštěn. **POZOR! Nikdy nenahrávejte kopírovací program COPY.TSD funkcí Binary load z DOSu. Může dojít ke kolizi dat na disketě a ke znehodnocení jejího obsahu!**

5.2. Programy v BASICu

Programy v BASICu uložené na disketě **není možno tímto programem kopírovat**, protože mají jinou hlavičku než programy ve strojovém kódu.

Basicovské programy zapsané v Turbo jsou ale upraveny tak, že mají formát programu ve strojovém kódu, a proto je možné je tímto kopírovacím programem kopírovat.

5.3. Tlačítko RESET

Stisk tlačítka RESET je v programu COPY.TSD ošetřen tak, že se kopírovací program nemaže, ale spouští se znovu od začátku (od vstupního menu). Zásadně se však **nedoporučuje** tlačítko RESET používat ve chvílích, kdy program provádí nějakou činnost s kazetou ve Standardu nebo s disketou, tj. když nahrává do počítače, zapisuje z počítače nebo když vypisuje adresář. Může tak dojít k porušení některých funkcí kopírovacího programu.

Bez obav můžeme tlačítko RESET použít, když program neprovádí žádnou činnost nebo když pracuje s kazetou v Turbo. Přesto tlačítko RESET používejte jen v krajní nouzi.

5.4. Vicedílné programy v Turbo

Některé extrémně dlouhé programy v Turbo jsou rozděleny na dvě a více částí. Kopírování takovýchto programů je možné pouze opět do Turbo a to postupným kopírováním jednotlivých částí. Takovéto programy většinou není možné spouštět funkcí 1 ve výstupním menu, protože se všechny části programu nevyskytují současně na správných adresových umístěních.

6. Závěr - údaje o programu

Samotný kopírovací program má délku asi 6,5 kByte, s připojeným DOsem pak necelých 13 kByte. Program byl napsán ve strojovém kódu pomocí assembleru MAC 65. Je umístěn v paměti RAM místo operačního systému. Program byl podroben testování a většina programů jím byla úspěšně překopírována libovolným směrem.

Kopírovací program COPY.TSD může (vyjma výše popsanych případů) usnadnit práci uživatele, který souběžně používá systém Standard i Turbo a má k dispozici i disketovou jednotku, neboť obsahuje v jednom funkce, které dosud zastávaly dílčí kopírovací programy.

V závěrečné redakční poznámce citujeme jeden z odstavců, obsažených v rukopisu příspěvku:

"Programové polotovary"

Vývoj každého programu je postupný, a tak než dostane program konečnou podobu, existuje mezi tím více pracovních verzí programu. Někdy se stane, že se omylem nebo i záměrně takovýto polotovar rozšíří mezi uživatele. Není to správné. Proto bychom chtěli vyzvat všechny, aby takovéto zmetky dále nešířili a sami se jich zbavili. A to nejen co se týče tohoto kopírovacího programu, ale všeobecně veškerého softwaru. Upozorňujeme, že správná verze kopírovacího programu má název COPY.TSD".

K citovanému upozornění autorů dodáváme: Naprosto souhlasíme s tím, že šíření "polotovarů", „programových zmetků“ atd. je mimořádně škodlivé především pro prostého uživatele, který nepozná, co mu bylo podvrženo. Ostatně na toto téma jsme na stránkách ZAK již hovořili [viz např. příspěvky J. Richtera, Dr. Tichoty a jiných]. Dodáváme, že kdyby všichni, kteří měli a mají upřímný zájem na absolutním zdokonalení systému TURBO 2000 a výsledné verzi SUPER TURBO ctili vyhlášenou zásadu, že garantem za druhý standard přenosu dat je ATARI KLUB Praha, že každý vývojový prvek je přinejmenším slušné konzultovat s autorem a jeho kolektivem pracovníků a kdyby někteří nechtěli být „papežtější než papež“, bylo by méně zbytečných problémů.

Jedinou prevencí před recidivou podobných případů je:

- nový program nabídnout vedení ATARI KLUBU **současně** s dokumentací, kterou bude možné zveřejnit ve ZAK, přičemž se dohodnout na podmínkách, za kterých lze program dále šířit,
- vyčkat do odborné recenze jak vlastního programu, tak jeho dokumentace,
- nepodléhat těm nejvěrohodnějším příslibům kamarádů: „... že ho opravdu nikomu neposkytnou“.

Jedině za těchto okolností lze zajistit autorský COPYRIGHT a současně ochránit běžného uživatele od zklamání.

red

TRANSCOPY — PROGRAM PRO KOPÍROVÁNÍ PROGRAMŮ VE FORMÁTU TURBO 2000

Karel Bílek, Praha

Program TRANSCOPY umožňuje připojení druhého magnetofonu na konektor pro druhý (zadní) křížový ovladač.

Kopírovaný program ve formátu TURBO 2000 se neukládá do paměti počítače, ale vystupuje na tento konektor. Takto pořízená nahrávka je naprosto rovnocenná původní.¹⁾ Dále je možné provádět selekci programů na zdrojové kazetě, a to zadáním pořadového čísla programu, který nemá být nahrán.

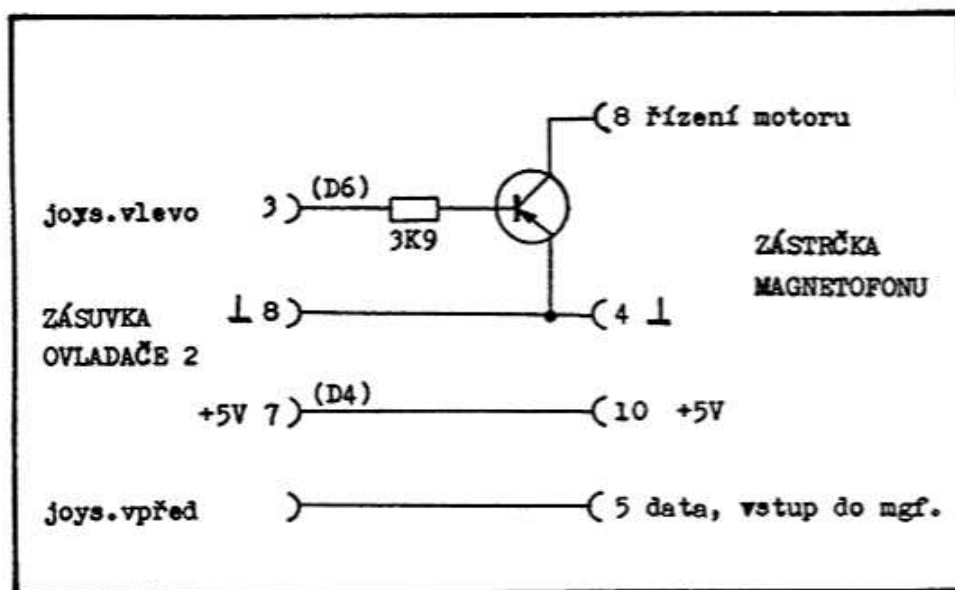
1) Na základě upozornění recenzenta nesouhlasíme s názorem autora. Nahrávka zřejmě bude asi takové kvality, jako při kopírování např. na DOUBLE-magnetofonu. V žádném případě nemůže být rovnocenná nahrávce pořízené klasickým způsobem z paměti počítače. Na toto téma srov. též Višek, O., Vlach, K.: Přehrávání programů bez počítače, Zpravodaj ATARI KLUBU Praha, 1987, 5, s. 40—43.

```

*****
1000 * TRAN COPY *
1010 * program pro kopirovani *
1020 * ve formatu TURBO 2000 *
1030 * Karel Rilek, Praha *
1040 *****
70 IF PEEK(16380)=104 THEN 15
80 POKE 153,0:AD=15330
10 READ A:IF A<>-1 THEN POKE AD,A:AD=AD+1:GOTO 10
15 ? " ":OPEN #1,12,0,"K":POKE 710,0
20 GOSUB 2000:POKE 550,34
30 ? " MAI KOPIROVAT CELOU KAZETU ?":GET #1,KEY:IF CHR$(KEY)="A" THEN 1000
40 TRAP 40: ? CLR: ? " POCET PROGRAMU NA STRANE KAZETY ":INPUT PROG:DIM PROG(PR
50)
60 ? " PORADI PROGRAMU, KTERE MAI KOPIROVAT"
70 ? " VYSTUP = pismeno": ? " ":FOR N=0 TO PROG:PROG(N)=0:NEXT N
80 TRAP 30: ? " PROGRAM CISLO ":INPUT FOR:PROG(FOR)=1: ? " "GOTO
90
90 PROGRAM=PROGRAM+1
100 IF PROG(PROGRAM)=1 THEN POKE 16570,32:GOTO 110
110 POKE 16570,34
120 A=USR(16380): ? " "
130 POKE 54019,11:POKE 54016,0:FOR N=1 TO 1500:NEXT N:A=USR(16448):IF PEEK(203)=
1 THEN POKE 54019,11:POKE 54016,0:GOTO 1500
140 IF PROGRAM<>PROG THEN 80
150 ? " ":POKE 54019,11:POKE 54016,0:GOTO 20
160 POKE 16570,34:A=USR(16380)
170 POKE 54019,11:POKE 54016,0:FOR N=1 TO 1000
180 IF PEEK(764)=23 THEN GOTO 20
190 NEXT N:A=USR(16448):GOTO 1000
200 ? " NOVY START STEJNEHO PROGRAMU A/N":GET #1,KEY:IF CHR$(KEY)="A" THEN P
ROGRAM=PROGRAM-1
210 GOTO 30
220 DL=PEEK(560)+PEEK(561)*256:POKE DL+3,71:POKE DL+6,6:POSITION 3,0: ? "***TRAN
COPY***":POSITION 27,0: ? "KBSOFT"
230 RETURN
3010 DATA 104,169,4,133,51,133,53,169,17,133,52,169,0,133,203,133,208,133,50,32,
31,64,114,233,173
3020 DATA 11,4,133,50,24,109,13,4,133,52,173,12,4,133,51,109,14,4,133,53,169,125
,141,0,4
3030 DATA 159,154,141,11,4,162,0,160,4,169,1,141,240,2,32,66,198,9,6,104,169,1,1
53,203,133,233
3040 DATA 32,94,64,176,7,32,62,198,169,1,133,203,96,133,54,169,48,141,2,211,169,
52,141,3,211
3050 DATA 159,128,133,16,141,14,210,24,160,0,132,48,132,49,140,14,212,140,0,212,
3,203,115,32,4
3060 DATA 65,141,243,169,0,133,46,133,55,160,180,32,255,64,144,236,192,216,144,2
34,230,45,203,241,193

```

3070 DATA 55,160,209,32,4,65,144,219,192,200,176,245,32,4,65,144,7,2,160,199,76,
138,34,40,203,9
3080 DATA 135,64,69,47,208,59,240,16,160,0,165,208,208,4,165,47,145,50,230,50,20
8,2,230,51,130
3090 DATA 209,8,169,1,133,47,32,255,64,144,23,192,207,38,47,160,198,144,243,165,
43,69,47,103,49
3100 DATA 135,50,137,52,165,51,229,53,141,193,169,0,197,49,104,169,192,141,14,21
2,133,16,141,14,210
3110 DATA 159,60,141,2,211,141,3,211,96,32,4,65,144,41,162,4,202,208,253,165,48,
74,37,5,141
3120 DATA 25,203,200,240,24,165,17,240,18,173,15,210,41,16,197,48,240,240,133,48
,9,32,141,0,211
3130 DATA 55,95,139,17,24,95,95,-1



ATARI®

klub Praha

GRAPH IT — NÁVOD NA POUŽITÍ SOUBORU PROGRAMŮ

Pomocí souboru programů GRAPH IT (© ATARI, 1980) lze snadno, přehledně a srozumitelně zpracovat a graficky znázornit nejrůznější ekonomické, statistické a matematické informace. Tímto programem lze také graficky znázornit výsledky získané využitím programů STATISTICS 1.

Použitelné aplikace těchto programů jsou velmi široké. Uplatnění naleznou v nejrůznějších oblastech, například při řízení, výzkumu, strukturální analýze, při průzkumu trhu, ve výuce, ale i při tzv. ekonomice domácnosti apod.



Program umožňuje vytvářet:

1. Sloupcové grafy:

- název grafu max. 20 znaků,
- každý graf max. 32 sloupců se 3 hodnotami, které mohou být zadány v každém sloupci,
- každý sloupec označen max. 3 znaky,
- sloupce zobrazené v max. 3 barvách (např. FACTOR 1: 35,0,25,0, atd.),
- pomocí těchto grafů se mohou zobrazovat kladné i záporné hodnoty.

2. Kruhové grafy:

- v kruhu je možné zobrazit 1 až 12 dílů (výřezů),
- nadpis a další podnadpis mohou mít každý max. 20 znaků,
- když není zadán podnadpis, určí se automaticky,
- každý díl grafu označen max. 3 znaky,
- počítáč určí procentuální poměr jednotlivých dílů,
- části menší jak 1/14 zobrazí společně s označením ETC.

3. Jedno a dvojrozměrné diagramy:

- umožňují zobrazit 1 až 3 funkce,
- rychlost kreslení je volitelná,
- volitelná je i měřka na Y-ové souřadnici.

4. Parametrické trojrozměrné polární diagramy:

- při dvoj- a trojrozměrných diagramech je možné používat všechny funkce definované v programovacím jazyku BASIC,
- AUTO SCALING zabezpečuje sledování průběhu vytváření grafů na obrazovce,
- když je zadána více jak jedna funkce, může být hodnota Y-ové souřadnice odlišná od hodnoty X-ové souřadnice,
- každou funkci lze definovat znovu anebo ji vypustit,
- jakmile se na nejnižším řádku obrazovky ukáže
CHANGE LIMITS RESTART
DO NOT EXIT ENTER LETTER : — ,
je možné:

a) znovu spustit program — stisknutím R a RETURN,

b) změnit hodnoty funkce — stisknutím C a RETURN. Na změnu hodnoty funkce se používá obrazovkový editor.

Pomocí křížového ovladače zapojeného v prvním konektoru lze zjistit souřadnice a sklon libovolného bodu v grafu. Provede se to následujícím způsobem:

- a) po pohybu ovladačem dopředu se objeví kurzor,
- b) pomocí ovladače se kurzor přesune na požadované místo,
- c) po stisknutí tlačítka FIRE se v levém dolním rohu obrazovky zobrazí hodnoty X, Y, Z a sklon.

☆ ☆ ☆

Načtení programu do paměti počítače

Originální soubor programu GRAPH IT tvoří prakticky čtyři samostatné programy vytvořené v ATARI BASICu (na dvou páskách). Samozřejmě, že řada našich uživatelů má programy překopírované různým způsobem. Pokud uživatelé schází označení jednotlivých programů, nejnázřejší je zjistit o který z nich se jedná podle vstupních menu, která v tomto příspěvku podrobně popisujeme.

A. Sloupcové a kruhové grafy

Úvodní menu je následující:

GRAPH IT — COPYRIGHT (c) 1980 ATARI

BAR CHARTS

PIE GRAPHS

EXIT

TYPE FIRST LETTER OF DESIRED

OPTION THEN HIT RETURN

OPTION:

sloupcové grafy

kruhové grafy

konec

napište první písmeno volby

a stiskněte RETURN

Po stisknutí klávesy B lze používat sloupcové grafy, po stisknutí P kruhové grafy.

Aplikaci sloupcového grafu nejlépe ukážeme na příkladu, pro jednoduchost zaměřeného na domácí ekonomiku.

Bude nás zajímat, kolik nás stála strava, ošacení a auto za posledních 5 měsíců.

Pokud jste v úvodním menu nestiskli klávesu B a RETURN, proveďte to. Po RUN se objeví obrazovka.

TITLE 0 to 20 CHARACTERS

nadpis od 1 do 20 znaků

TITLE: —.

Zadejte název grafu, v našem případě ROZPOCET ZA 5 MESICU a stiskněte RETURN. Jestliže uděláte chybu v zápisu, opravte ji použitím klávesy DELETE BACK anebo INSERT a šipka přes CTRL, podobně jako při programování.

Nyní se zobrazí:

GRAPH MAY CONTAIN FROM

1 TO 32 COLUMNS

FROM 1 TO FACTORS MAY BE

GRAPHED TO GETHEN

ENTER NUMBER OF FACTORS:

To se počítač ptá, kolik bude zpracováno faktorů, přičemž oznamuje max. rozsah sloupců (32). Musíte si pamatovat, že faktory mohou být max. tři. Je to vlastně počet dílů každého sloupce vyjádřený na obrazovce v barvách.

V našem příkladu zadejte 3 faktory (strava, ošacení a auto). V zobrazeném menu zadejte 3 a RETURN.

Vycházejme opět z příkladu: Zajímá nás, kolik z nákladů na stravu tvoří položky za maso, mléko a mléčné výrobky, zeleninu a ovoce a na ostatní složky potravin.

Po spuštění programu jste nejdříve vyzváni k zadání názvu grafu.

**THE PIE GRAPH WILL DISPLAY
1 TO 12 SLICES
TITLE 0 TO 20 CHARACTERS
TITLE:**

graf může obsahovat
1 až 12 částí
nadpis na 1 až 20 znaků

Napište **ROZLOZENÍ STRAVY** a stiskněte **RETURN**. Na obrazovce bude dále požadavek na zadání názvu jednotlivých částí budoucího grafu (v našem případě jde o 4 části - maso, mléko, zelenina a ostatní), max. však 3 znaky. Při zadávání bude obrazovka následující:

**LABELS ARE 0 TO 3 CHARACTERS
HIT START TO EXIT**

0 až 3 znaky pro název
stisknutí **START** po ukončení
zadávání názvu dílu

**PIE SLICE 1 LABEL : MAS
PIE SLICE 2 LABEL : MLE
PIE SLICE 3 LABEL : ZEL
PIE SLICE 4 LABEL : OST
PIE SLICE 5 LABEL :**

stiskem **START** se zruší další zadávání dílů a jejich názvů - v daném případě budou čtyři.

Nyní budete vkládat hodnoty pro jednotlivé díly grafu:

**INPUT VALUE FOR
PIE SLICE MAS : 17
PIE SLICE MLE : 36
PIE SLICE ZEL : 39
PIE SLICE OST : 8**

Po každé hodnotě se stiskne **RETURN**.

V dalším kroku je možné zadat podnadpis do max. 20 znaků, který se pak vypíše pod grafem. K jeho zavedení vyzve hlášení:

**SUBTITLE 0 TO 20 CHARACTERS
SUBTITLE:**

Po stisknutí **RETURN** se začne na obrazovce vykreslovat kruhový graf. Po stisknutí libovolné klávesy se objeví otázka **EXIT Y/N**.

C. Jednorozměrné, dvojrozměrné a polární diagramy

Jestliže chceme definovat něco jako funkci, na její zobrazení musíte použít jedno nebo dvourozměrné diagramy. V souřadnicovém systému (podle Descartese) se používají 4 kvadranty. Souřadnicový systém sloupcových diagramů, který je popsán v předcházející kapitole, je vyjádřen jako jednorozměrný v kladném směru. Záporný směr v takovém diagramu není vidět, protože všechny zadané hodnoty jsou kladné. Když v souřadnicovém systému x, y vytvoříme přímku protínající bod 0 a dále body v souřadnicích $x=16, y=16, x=32, y=31$, bude tato přímka protínat ve III. kvadrantu body $x=-16, y=-16, x=-32, y=-32$.

Dvojrozměrné diagramy

Po vytvoření dvojrozměrného diagramu je nutné načíst do paměti počítače druhy z programů **GRAPH IT**. Na obrazovce se objeví:

FROM 1 TO 2 FUNCIONS

zadejte počet používaných funkcí
od 1 do 2

CAN BY PLOTTED

NUMBER OF FUNCTIONS:

Zvolte např. 2 a stisknete RETURN. Další obrazovka má formát.

FUNCTION FORM IS $Y=F(X)$

zadejte tvar první funkce

TYPE IN THE FUNCTION

THEN HIT RETURN

TO ENTER PREVIOUS FUNCTION

HIT RETURN

$Y1=X$

Napište X a stiskněte RETURN.

Vložení funkce a RETURN je potřebné, jestliže chcete změnit nebo uchovat funkci Y1 i po zobrazení grafu. Když je funkce velmi dlouhá nebo při omylu v zadávání, po nakreslení grafu stiskněte R a RETURN. Program se vrátí na Y1 a po stisknutí RETURN uvidíte svoji funkci. Pomocí obrazkového editoru můžete udělat opravu ve funkci. Potom se zobrazí obrazovka pro zadávání 2. funkce:

FUNCTION FORM IS $Y = F(X)$

TYPE IN THE FUNCTION

THEN HIT RETURN

TO ENTER PREVIOUS FUNCTION

HIT RETURN

$Y2 = \cos(X)$

Zadejte například $\cos(X)$ a stiskněte RETURN.

Na další obrazovce se zadává rychlost vykreslování funkce:

PLOTTING SPEEDS

rychlost kreslení

SLOW

S pomalu

MEDIUM

M středně rychle

FAST

F rychle

VERY FAST

V velmi rychle

ENTER PLOT SPEED:—

Vyšší rychlost vykreslování je na úkor detailnosti. Čím větší rychlost, tím menší počet bodů tvoří graf. Křížový ovladač nemůžete používat při vysoké rychlosti (z důvodu správnosti diagramu).

Na další obrazovce je třeba zadat dolní a horní hranici intervalu, ve kterém se budou funkce pohybovat:

START LIMIT MUST BE LESS

dolní hranice musí být menší

THEN END LIMIT

jako horní

STARTING X VALUE : -5

dolní hranice, např. -5

ENDING X VALUE : 5

horní hranice, např. 5

Po zadání každé hranice se musí stisknout RETURN

V zadaném příkladu se zobrazí funkce kosinus podél osy x.

Další obrazovka má tvar:

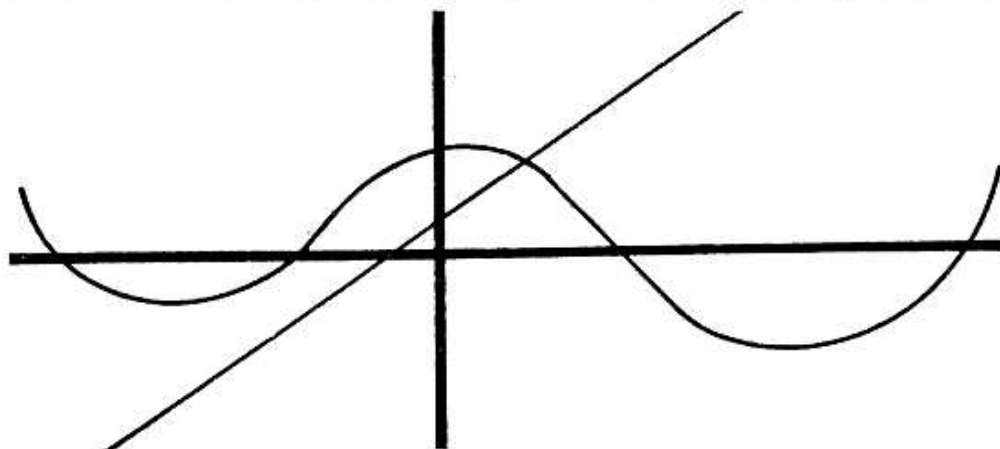
AUTO SCALE THE PLOT (Y*N):-

Po volbě Y a stisknutí RETURN se objeví slovo SCALING.

AUTO SCALE způsobí, že všechny vrcholy funkce budou zobrazené. To je příčina toho, že vlnění okolo osy y je jiné jako vlnění okolo osy x.

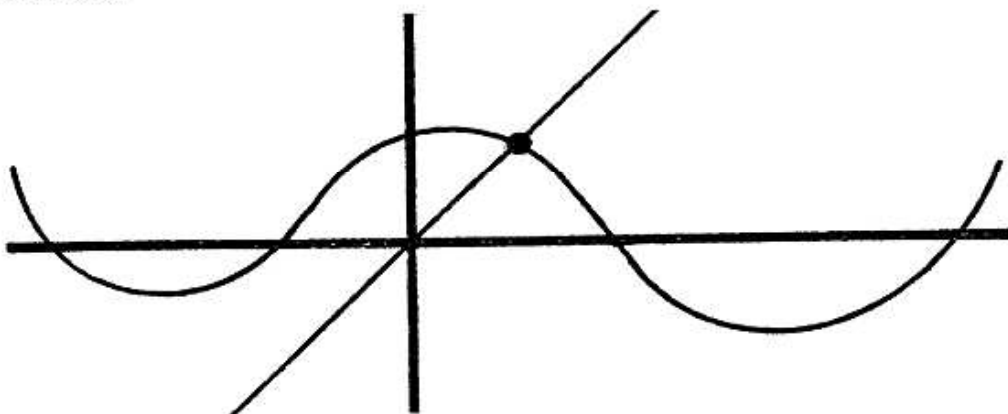
Když budete mít dvě nebo více funkcí, může to zapříčinit vizuální disproporci jiné z funkcí.

Pomocí ovladače v režimu AUTO SCALE můžete nalézt y-ovou hodnotu. Po výběru hodnoty X vybere AUTO SCALE hodnotu Y tak, aby se všechny funkce objevily na obrazovce.



**CHANGE LIMITS RESTART
DO NOT EXIT ENTER LETTER:-**

Změnou limitu funkce anebo opakovaným stisknutím kterékoliv klávesy se program doplní. V dolní části obrazovky se objeví menu pro výběr činnosti. Stisknutím C a RETURN se změní limit funkce. Stisknutím R a RETURN se znovu spustí program dvojrozměrných diagramů. Obrazovka:



X = 0.72327043
Y2 = 0.7496452484
SLP = 0.6614835963

Používání křížového ovladače:

Ovladač se zapojí do zásuvky 1.

Pohybem páky ovladače dopředu se objeví na obrazovce kurzor. Kurzor se posune na místo, ze kterého chcete odečítat hodnoty bodu. Po stisknutí tlačítka FIRE se v levém dolním rohu obrazovky objeví hodnoty pro sklon, X, Y, a Z koordinaci.

Například "procházením se" po křivce kosinu můžete přechít příslušné hodnoty v jednotlivých bodech. Přitom zjistíte, že sklon se pohybuje v intervalu -1 až 1 .

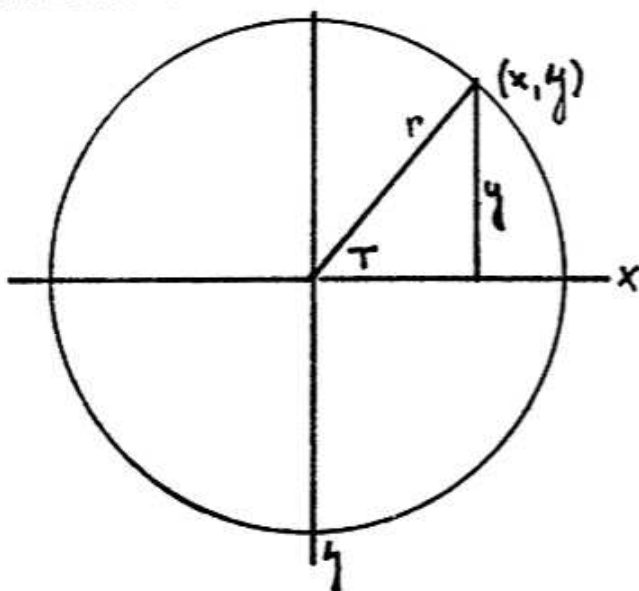
Poznámka: Program pro dvoj- a trojrozměrné diagramy podporuje jen funkce, které se mohou používat v ATARI BASICu. Např. druhá mocnina záporného čísla se nemůže v programu na diagramy používat, protože ji BASIC nezná.

Dvojrozměrné parametrické diagramy (s využíváním polarit)

Program GRAPH IT POLAR PLOTS je možné použít na tvorbu a studium zajímavých geometrických útvarů.

V parametrických polárních diagramech jsou dvě rovnice $X=$ a $Y=$. Slovo parametrický znamená, že polární funkce je opisovaná pomocí dvou měnitelných parametrů X a Y . Polární rovnice definuje úhel theta (T) a vzdálenost r od počátku. Program GRAPH IT používá rovnice pro $X=$ a $Y=$ a konvertuje polární systém na pravouhlý systém z diagramu individuálních vztahů X , Y .

Jestliže chcete definovat funkci v počátku a vzdálenosti z tohoto bodu, musíte vykreslit kruh. To je snadné udělat i s tužkou a s pomocí nitky upevněné ve středu. Nitka reprezentuje konstantní hodnotu, rotace nitky v kruhu reprezentuje nárůst stupňů od 0 do 360. Pomocí trigonometrie to můžeme vyjádřit následovně:



$$\cos(T) = X/r$$

$$X = r \cdot \cos(T)$$

$$\sin(T) =$$

$$Y = r \cdot \sin(T)$$

Jestliže $r=1$ (anebo nějaká konstanta), dostanete kruh.

Po načtení programu GRAPH IT POLAR PLOTS do paměti počítače se objeví první obrazovka, která zadá definování $X=$ ve formě $F(T)$.

FUNCTION FORM IS $X=(T)$
TYPE IN THE FUNCTION
THEN HIT RETURN
TO ENTER PREVIOUS FUNCTION
HIT RETURN
 $X=\cos(T)$

zadejte funkci v tvaru
 $X=F(T)$ a stisknete RETURN

Zadejte např. COS(T) a stiskněte RETURN. Úhel T je potřebné zadat v závorkách. Na další obrazovce musíte zadat funkci Y.

**FUNCTION FORM IS Y=G*T
TYPE IN THE FUNCTION
THEN HIT RETURN
TO ENTER PRECIOUS FUNCTION
HIT RETURN
Y=SIN (T)**

zadejte funkci v tvaru
 $Y=G/T$ a stiskněte RETURN

Napište SIN (T) a stiskněte RETURN.

Objeví se další obrazovka:

**LIMIT WILL BE IN RADIANES
STARTING T VALUE: 0
ENDING T VALUE: 6.3
T INCREMENT: .1**

hodnoty musí být v radiánech
dolní hranice (např. 0)
horní hranice (např. 6.3)
krok růstu (např. 0.1)

Zadejte tedy startovací hodnotu 0 a stiskněte RETURN. Když chcete nakreslit kruh od 0 do 360 stupňů, zadejte koncovou hodnotu 6.3 a RETURN.

Konverze ze stupňů na radiány není těžká. Provádí se podle vzorce:

$2 \times 3,14159(T) \times \text{radiány} = 360 \text{ stupňů}$. Vynásobením $2 \times 3,14159$ a zaokrouhlením na 1 desetinné místo dostanete 6,3. Když chcete nakreslit jen polokruh, je třeba zadat 3.15. T musí růst krokem 0.1.

Dále se zobrazí otázka: **AUTO SCALE THE PLOT Y/N:**

Jestliže chcete kruh zobrazit na obrazovce, zadejte Y a stiskněte RETURN. Využití AUTO SCALE zabezpečí, že bude vidět zobrazování diagramu. Když se objeví slovo SCALING, program počítá hodnoty pro zobrazení kruhu, potom se kruh objeví.

Křížovým ovladačem můžete zjišťovat hodnoty X, Y a T pro každý bod kruhu.

Znovu spustit program anebo změnit hodnotu se dá stisknutím klávesy na klávesnici. Napište R a stiskněte RETURN, jestliže se chcete vrátit na začátek programu, nebo napište C a stiskněte RETURN, když chcete změnit zadanou hodnotu radiánu.

Příklad na vykreslení květu:

1. definujte $X = \cos(1.5 \times T) \times \cos(T)$ RETURN
2. definujte $Y = \cos(1.5 \times T) \times \sin(T)$ RETURN
3. dolní hranice pro T je 0 RETURN
4. horní hranice pro T je 13 RETURN (pro květ potřebujete 720 stupňů)
5. krok růstu zadejte 0.1 RETURN
6. AUTO SCALE THE PLOT Y/N: Y RETURN

Další příklad:

Když chcete udělat polární funkci zadanou podle vzrůstajícího r jako rovnici o dvou neznámých, X, Y zadejte takto:

$X = r \times \cos(T)$

$Y = r \times \sin(T)$

Zadejte např. $r = 1 - \sin(T)$, pak

$X = r \times \cos(T) = (1 - \sin(T)) \times \cos(T)$

$Y = r \times \sin(T) = (1 - \sin(T)) \times \sin(T)$

Použijete-li nyní pro dolní hranici T = 0 a pro dolní hranici T = 6.3 a krok 0.1, program vám vykreslí srdce.

Chcete nakreslit spirálu? Výše popsané ENDING T VALUE upravte takto:

$$X = (T) * \cos(T)$$

$$Y = (T) * \sin(T)$$

D. Trojrozměrné X, Y, Z grafy

Na plošné televizní obrazovce lze simulovat trojrozměrný prostor. K tomu je třeba načíst program THREE DIMENSIONAL X, Y, Z PLOTS.

První obrazovka má tvar:

FUNCTION TYPE IS Z = F(X, Y)

TYPE IN THE FUNCTION

THEN HIT RETURN

TO ENTER PREVIOUS FUNCTION

HIT RETURN

$$Z = X * X - Y * Y$$

zadejte funkci se dvěma
proměnnými

a stiskněte RETURN

Na další obrazovce je třeba zadat rychlost zobrazování:

PLOTTING SPEED

SLOW

FAST

ENTER SPEED:

rychlost zobrazování

pomalou S

rychle F

Zadejte například S.

Dále je potřebné zadat dolní a horní hranici pro jednotlivé proměnné:

STARTING LIMIT MUST BE

LESS THAN ENDING LIMIT

MINIMUM X VALUE:

MAXIMUM X VALUE:

dolní hranice musí být

menší jak horní

dolní hranice pro X

horní hranice pro X

Zadejte např. — 10 a 10. Po zadání hodnot je vždy třeba stisknout RETURN.

Podobně se zadají i hodnoty pro Y.

Na otázku AUTO SCALE Y/N odpovězte N a stiskněte RETURN. V tomto případě odmítáte AUTO SCALING, protože zobrazené sedlo by se zploštilo. Pro Z se zadají hodnoty jako pro X a Y (začátek —10, konec 10).

Rychlost kreslení se zlepši, když na otázku **REMOVE HIDDEN LINES Y/N:** odpovíme N a stiskneme RETURN.

Když si přejete vidět proces vykreslování, tak na otázku **FASTER NON DISPLAY PLOT Y/N:** odpovíte N a stiskněte RETURN. Je zajímavé vidět proces vykreslování, i když se tím prodlužuje čas kreslení.

Program na obrazovce vypíše přibližný čas kreslení:

INITIALIZING

PLOTTING TIME 5 MINUTES

Po 5 minutách se diagram sedla zobrazí. Jestliže chcete nyní program spustit znovu, zadejte R a RETURN. Jestliže chcete změnit hodnoty, zadejte C a RETURN.

Na kreslení grafu se může používat křížový ovladač. Na trojrozměrné úlohy je bohatá analytická geometrie. Do trojrozměrných úloh může vnést nové znaky sinus. Zkuste nakreslit:

$$Z = 20 * \sin(X * Y)$$

X STARTING: —3

X ENDING: 3

Y STARTING: —3

Y ENDING: 3

Y ENDING: 3

N for AUTO SCALE

Z STARTING: —3

Z ENDING: 3

Je nutné vědět, že velmi komplikované obrazce se vypisují i několik hodin.

(Z originálu přeložil a upravil ing. Ladislav Gál, Bratislava)

TECHNICKÉ novinky

NOVÝ MIKROPROCESOR PRO ATARI XL/XE?

Jedním z posledních členů mikroprocesorové řady 65... je nový šestnáctibitový mikroprocesor 65816. Tento obvod dokáže jednoduchým způsobem emulovat osmibitové procesory 6502 a 6510 při hodinové frekvenci až 4 MHz, což představuje čtyřnásobné zrychlení operačních časů oproti klasickým osmibitovým procesorům. Dalšího zrychlení lze samozřejmě dosáhnout plným využitím šestnáctibitového režimu. Obvod 65816 přitom umožňuje přímo adresovat až 16 MB operační paměti.

Z firem, které ve svých výrobcích používají mikroprocesory řady 65..., využila nový

procesor firma Apple pro svůj inovovaný model Apple IIgs. Příjemné překvapení majitelům osmibitových Commodorů připravila švýcarská firma Swisscomp, která nabízí zásuvný modul Turbo 64 Board s novým procesorem 65816 a přidavnou pamětí RAM o kapacitě 64 kB. Modul pracuje ve třech režimech: 8b/1 MHz, 8b/4 MHz, a 16b. Pro práci v posledním z uvedených režimů slouží další modul téže firmy, který obsahuje přidavnou paměť RAM o kapacitě 1 MB!

Majitelé počítačů Apple a Comodore se již mohou těšit z nového procesoru pro své miláčky. O podobném rozšíření pro počítače Atari se zatím nepíše ani nehovoří. Proto titulky našeho článku končí otazníkem. Doufejme, že jen prozatím...

(Podle článku Turbo 64, Bajtek 1/88 připravil -fis-)

Disketová jednotka XF 551 a TOP DOS

ing. Petr Klinger, Praha

Oblíbené a u nás rozšířené disketové jednotky ATARI 1050 se v současné době již nevyrábějí. Výrobce je nahradil novou disketovou jednotkou XF 551 (je popsána v ZAK 6/87 a 1/88).

Velkou nevýhodou zatím je, že při nákupu jednotky XF 551 obdrží zákazník společně s výrobkem "jen" DOS 2.5 a upozornění, že až na rozdílný volič čísla jednotky je XF 551 ekvivalentem jednotky ATARI 1050.

DOS 2.5 u XF 551 sice bezchybně funguje, ale neumožní nic víc než ATARI 1050. Pomo-

cí nové jednotky nelze navíc zapisovat (bez složité úpravy diskety) na "druhou" stranu diskety. Majitel má v ruce nové zařízení, schopné však poskytnout "staré" služby.

Tato úvaha však není opodstatněná. Druhou stranu diskety nelze bez vystřížení indexové díry [na této druhé straně] pouze formátovat. Máme-li naformátovanou druhou stranu diskety třeba v jednotce 1050, lze na ní pomocí XF 551 zapisovat a z ní číst bez jakékoliv úpravy jednotky. [Týká se to jednostranného používání diskety např. v DOS 2.5.]

Podobně lze mít praktickou výhodu i k dalšímu tvrzení autora příspěvku, spočívající v návrhu skutečně složité úpravy diskety. Autor doporučuje obal diskety opatrně roz-

lepit, v rukavicích vyjmout vlastní disketu, v obalu pomocí vhodného nástroje [např. nože z kancelářské děrovačky] vyrazit dle vertikální osy nový čtecí otvor, disketu vložit zpět a obal zalepit.

Disketu však není třeba rozlepovat. Stačí nůžkami vystříhnout otvor v obálce symetricky podle původního obalu (viz obr.).

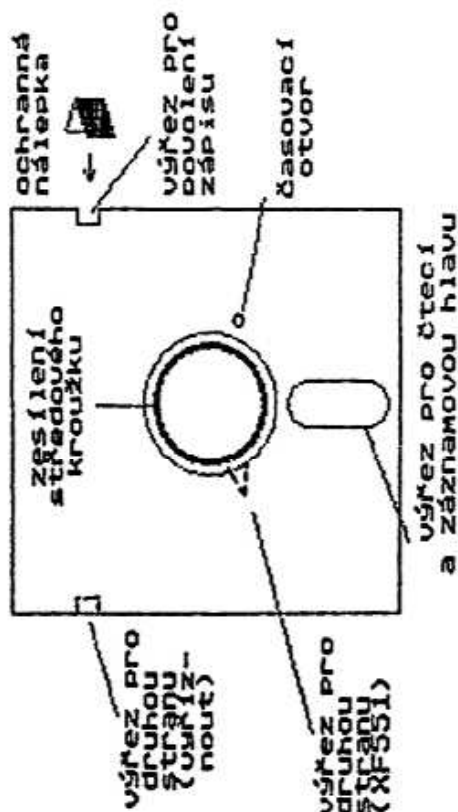
Kromě DOS 2.5 je jednou z variant použití operačního systému TOP DOS, který je v rámci klubu částečně rozšířen. Při jeho nahrání ale máme na obrazovce monitoru konfiguraci systému, která nám označí naši jednotku č. 1 písmenem E, což reprezentuje rozšířenou hustotu jednotky ATARI 1050. Budeme-li se nyní pomocí příkazu S (Set/Status) pokoušet změnit hustotu záznamu na této jednotce příkazem S D1D (nastav na drive 1 double density), systém na tento příkaz nebude reagovat. Stejná situace se opakuje, ať se pokoušíme nastavit jakoukoliv jinou hustotu. Pokusíme-li se naformátovat při takto nastavené hustotě disketu na DD (double density — dvojitá hustota) повеlem I 1Q,A/E, 180, odpoví systém informací "Not Double-Density!". V této situaci se zdá, že naše jednotka nepracuje zcela bez závad.

Pomoci si lze ale i bez hlubší znalosti TOP DOSu. Vyjmeme z disketové jednotky disketu, povel I, Return, 1, Return, Y, Return "docílíme" chybu a příkazem /, Return vyvoláme znovu na obrazovku Status. Nyní je jednotka označena písmenem S, což reprezentuje jednoduchou hustotu. Příkazem S D1x, kde x je S, D nebo Q je možno zvolit jednoduchou (S) nebo dvojitou hustotu (D) na jedné straně nebo oboustrannou dvojitou hustotu (Q), která nám umožní uložit až 360 kB dat. V dané hustotě nám umožní naformátovat disketu (obdobně DOS 2.5) příkazem I. Bez důkladné znalosti TOP DOSu doporučuji ignorovat další doplňující dotazy a známou sekvenci příkazů I - 1 - Y (viz DOS 2.5) neformátovat ve zvolené hustotě disketu. Rozdíl v hustotách DD a QD spočívá v dostupnosti souborů na disketě. Při hustotě DD nahrané na obou stranách disket (2 x 180 kB) je kapacita stejná jako při QD (1 x 360 kB), rozdíl je v max. počtu sou-

borů možných k uložení (120 - 60) a v potřebě výše popsané úpravy diskety s nutností jeho obrácení při DD.

Tím jsou dostupné všechny možné varianty hustot záznamu informace pomocí nové disketové jednotky XF 551. Nahrajeme-li na nově zformátovanou disketu i TOP DOS (příkazem H - obdobně jako u DOSu 2.5), máme při novém nahrávání DOSu z této diskety již jednotku nastavenou v odpovídající hustotě.

(Autorem redakční poznámky a obrázku je ing. Petr Jandík.)



Programy pro ATARI rozhlasem?

Jak nás informoval hlavní koordinátor činnosti ATARI KLUBŮ v Západočeském kraji soudruh ing. Miloš Adam, v poslední době se činnost všech osmi klubů pěkně rozvíjí. Na řadě akcí se účastní i zástupci tisku, rozhlasu a televize. Z podnětu krajské redakce rozhlasu se uskutečnila hodinová beseda o mikropočítačích v ČSSR. Její součástí byl i mimořádně zajímavý pokus: vysílání programů ATARI. Soudruh ing. Adam líčí čtvrtek 24. března 1988 ve studiu krajského vysílání rozhlasu v Plzni následovně:

„Krátkce po 15 hodině se na středních a velmi krátkých vlnách a v rozhlasu po drátě ozval BLEKOTA. Mikropočítač ATARI 800 XL se představil.

U kulatého stolu ve studiu seděli dva moderátoři - Jindra Weidl a Standa Martínek (oba ataristé a po jednom zástupci za ATARI a COMMODORE. Jejich prostřednictvím se mohla veřejnost seznámit nejen s tím, že počítačové kluby existují a fungují, ale že mají i určité výsledky. Mimořádný zájem posluchačů prozrazovaly telefonické dotazy do živého vysílání. Hudební vložky byly zkracovány na minimum (zřejmě k nevelké radosti režiséra pořadu) a přesto se v hodinovém pořadu nedalo odpovědět na všechny dotazy.“

Za VELKÝ POKUS pro organizované ataristy v Západočeském kraji označil ing. Adam první zkušební přenos programů pro ATARI (pravděpodobně i první v ČSSR).

„Po celém kraji byli předem připraveni u rozhlasových přijímačů členové ATARI klubů se svými kazetovými magnetofony. Přímo ze studia (počítače byly zapojeny do rozhlasového zařízení přímo) jsme odvysílali program v BASICu napsaný ve standardním módu 600 Bd. Byl snímán a vyslán zvukový signál po příkazu CSAVE.

Signál byl nejdále zachycen v Domažlicích a Františkových Lázních a to při příjmu na Vkv a RpD. Program se zde podařilo spustit. Na SV se nedařilo nikde.“



Pro pražský ATARI KLUB bylo zvláštní počtou, že Západočeši při svých pokusech nezapomněli na systém TURBO 2000. Je nutné dodat, že právě Západočeši přijali druhý standard přenosu dat jako jedni z prvních a vše se k němu hlásí i nyní. A nejen to.

Ing. Miloš Adam píše, že „pro TURBO 2000 jsme připravili absolutní novinku - dokonalejší verzi karla, kterou vytvořili kolegové v Mariánských Lázních. S turbem jsme ale nepochodili. Zdůrazňuji však prozatím nepochodili. Již v průběhu vysílání jsme si slíbili, že provedeme další pokusy s cílem nalézt uspokojivé řešení.

Až budou dořešeny technické problémy, lze předpokládat, že Čs. rozhlas nalezne prostor a umožní masové šířit takové programy, kde to autorská práva dovolí. Bude se jednat o programy nezištně vytvořené členy klubů zejména pro děti a mládež a pro seriózní využití při práci.

S pracovníky rozhlasu v Plzni uděláme vše pro dořešení vysílání počítačových programů etherem. O výsledcích budeme informovat.“

Děkujeme ing. Miloši Adamovi za zprávu. Ataristům ze Západních Čech přejeme hodně úspěchů při dořešení vysílání programů prostřednictvím rozhlasových vln.

red

Tiskárna ATARI 1029 a jehly

„Tak jste si zase užili pěknou ostudu“, začínal jeden z dopisů, který byl po vydání ZAK č. 1/88 doručen naší redakci. A dopis pokračoval: „To vy chytráci ani nevíte, že tiskárna ATARI 1029 žádné jehly nemá? Jak se tedy může poškodit, když se v ní bude používat kazeta s barvicí páskou upravená podle návodu I. Martiška?“ Děkujeme touto cestou autorovi dopisu za upozornění. Nebyl sám, kdo na nesprávně formulovanou redakční poznámku na str. 67 cit. ZAK reagoval, i když nikdo z dalších oprávněných kritiků nepoužil onen zvláštní tón nadřazené důležitosti.

Pravda. Napsali jsme nesmysl. ATARI 1029 totiž není „pravá“ jehličková tiskárna, protože k tisku používá zvláštního mechanismu, kterému někdo říká „nože“. To nikoho —

a bohužel ani recenzenta — nenapadlo v okamžiku, kdy jsme připravovali článek I. Martiška do tisku.

Na naši omluvu snad jen následující vysvětlení: Snažíme se vždy, když autor ve svém článku navrhuje nějaké řešení, jehož podstatou je zásah do zařízení apod., nějakou formou upozornit, že „každý sám odpovídá za své činy“. Redakce totiž nemůže přebírat odpovědnost za třeba i nesprávné návody a doporučení.

A pokud jde o Martiškovu tiskárnu? Shodou okolností jsme ji od něho koupili pro klub a musíme potvrdit, že pracuje normálně, byť má onu úpravu na prodloužení životnosti barvicí pásky.

Ještě jednou se proto jak čtenářům, tak autorovi příspěvku omlouváme.

Dr. Jan Hlaváček

Vážená redakce,

při zkoušení programu TURBO BASIC XL 2.0 jsem v návodu zveřejněném v ZAK č. 1/88 narazil na nepřesnost. U popisu příkazu SAVE je zde uvedeno, že jestliže prvním znakem jména programu je x, pak se program po LOAD sám spustí. Program ovšem takto pracuje, jestliže prvním znakem je *. Je jasné, že chyba je způsobena spíše nedostatkem v tiskařské technice, která na tento znak zásadně „zapomíná“ a způsobuje problémy při vydávání veškeré technické a matematické literatury.

Chtěl bych ještě poznamenat, že před 18 měsíci jsem četl v jakémsi časopise, že nejslabším článkem systému ATARI je kazetový magnetofon. Díky J. Richterovi a jeho kolegům se však magnetofon stal jedním z nejsilnějších článků tohoto systému. Vyřídte jim prosím můj obdiv a dík.

Vladimír Čech, Praha

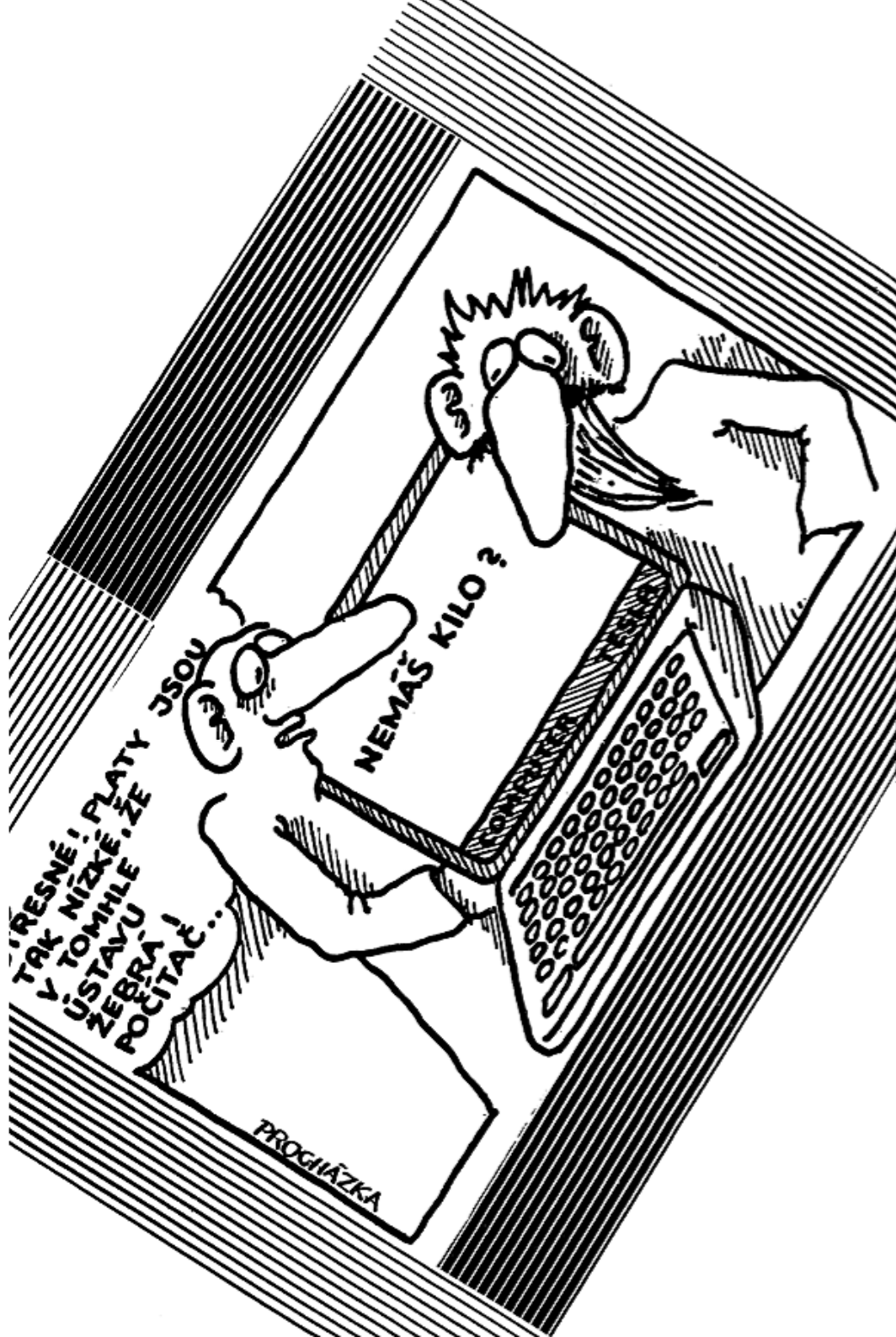
Pokud jde o ten nešťastný křížek a hvězdičku, V. Čech má pravdu v tom, že ani modernější sázecí tiskařské stroje (alespoň ty, na kterých vyrábíme ZAK) nemají v sazbě hvězdičku. Sazečka tudíž píše křížek (x) a redaktor dodělává ručně fixem třetí čárku,

čímž vzniká hvězdička (*). Z toho tedy vyplývá, že za cit. chybu může naše redakce. Omlouváme se.

Pokud jde o chválu na adresu J. Richtera a jeho kamarádů, rádi taková slova otiskujeme.

-red-





RESNÉ! PLATY JSOU
TAK NÍZKÉ, ŽE
V TOMHLE ŽEBŘU
ÚSTAVU
POČITAČ!

NEMÁŠ KILO?

PROGHAZKA

Publikované zo súhlasom - vid' Prohlášení představitelů AK Praha.