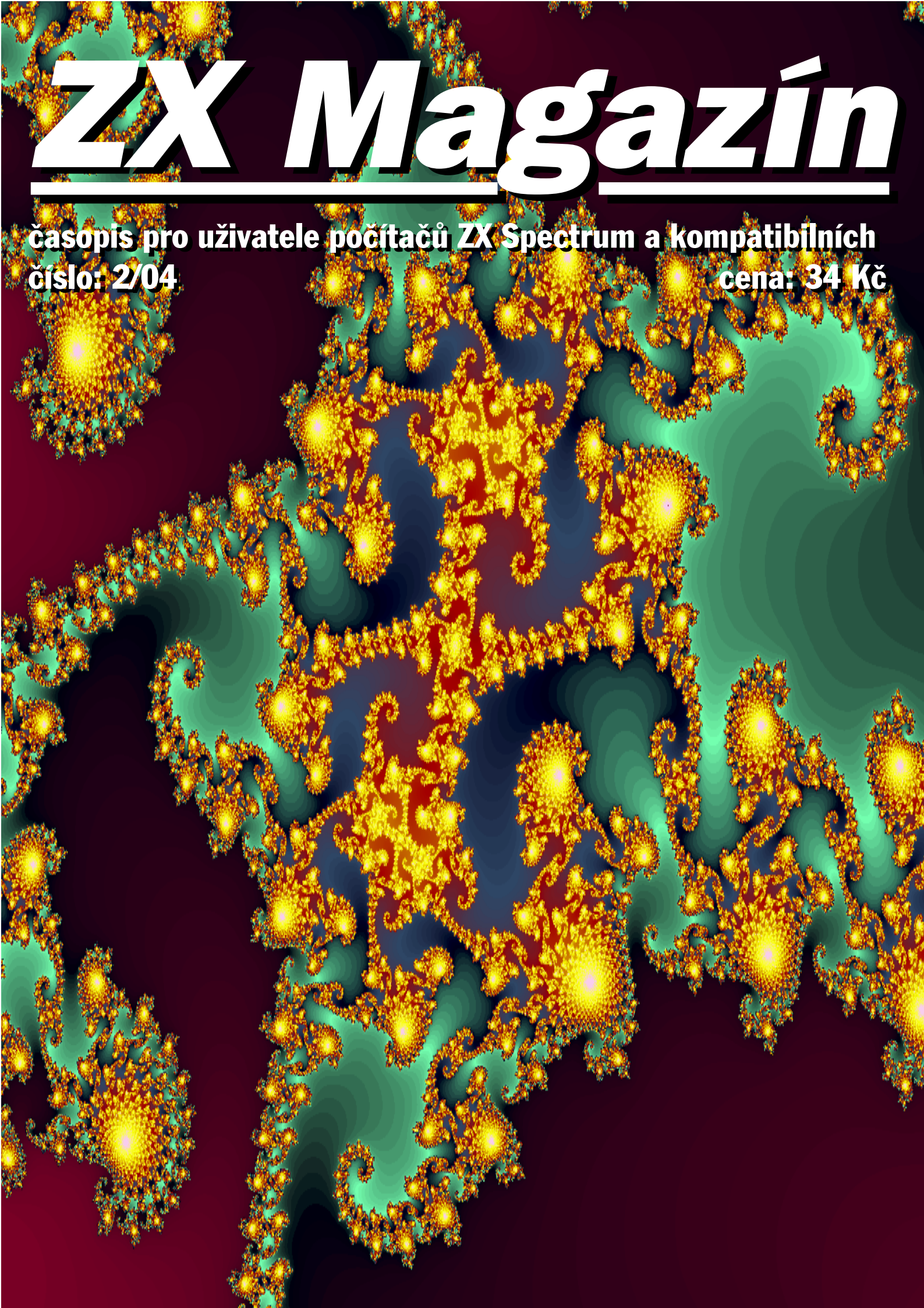


ZX Magazín

časopis pro uživatele počítačů ZX Spectrum a kompatibilních

číslo: 2/04

cena: 34 Kč



Začátek

05 jsme tu !

Překvapení? Já doufám, že ne, a že už jste nové číslo netrpělivě očekávali. Doba temna skončila, teď se bude vydávat jak na běžícím páse. Náš cíl je 4 čísla za rok! Osobně sice nevím, zda se to vždy podaří, ale minimálně 3 čísla do roka vyjít musí i kdyby trakaře padaly.

Tentokrát jsem dostal úvodník na starost já. A jak vidím, k dispozici mám hned celý sloupec, takže si to musím pořádně užít (není totiž jisté, jestli mě ti dva po přečtení těchhle řádků nechají ještě někdy něco napsat).

Předně mám radost, že ZXM zase hezky vychází, že je zadarmo, na internetu, a především že už nehrozí ostuda s nevráceným předplatným. Bohužel ale minulost je tu s námi pořád, a tak zrovna s čistým štítem nepokračujeme. Nedávno se ozval čtenář až z Německa, který před lety za ZXM osobně zaplatil bývalému vydavateli, ten ale v zápětí skončil a samozřejmě časopis nikde, peníze nikde, ani omluva nikde. A zdaleka není jediný, kdo takhle dopadl. My s tím ale bohužel nic nenaděláme. Je jen na samotném poškozeneš, aby si svou pohledávku vymohl zpět (btw. až půjdete, dejte mi vědět, posílám své ústřížky složenek a půjdu na tu „práteleckou“ návštěvu s vámi; a těch vydavatelů obejdeme rovnou více).

Trošku je nám líto, že vy, čtenáři, jste stále hodně pasivní. Dejte najevo svou chválu nebo naopak nespokojenost! Projevte se! Velice bychom přivítali vaše náměty (tzn. o čem byste chtěli, abychom napsali), článkem z vaší strany samozřejmě také nepohrdneme. Nevíte-li co napsat, vězte, že mne osobně by například zajímaly rozhovory s někým zajímavým ze scény. Speccy akcí do roka je plno, stačí tam jen někoho odchytnout (příp. ještě opít) a vyzpovídat. Nebo rovnou přes internet, tam po chvilce hledání narazíte na hotové legendy jako Franta Fuka, George K. atd.

Jistě jste si už stačili všimnout, že fungujeme díky internetu. Aragorn dává v těchto dnech našemu webu zcela nový (profil) image. Určitě tedy nakoukněte (<http://xzm.speccy.cz>) a napište nám i něco hezkého (nebo klidně i škaredého) do CHATu.

Velice mě poslední dobou těší situace na mnohobitech. Bývalí spectristé už jsou tak zhnuseni, že oprašují své osmibitové miláčky a vrací se zpět k nám. Tohle si opravdu nemyslím, už několik takových lidí mne kontaktovalo! Je vidět, že spectristé moc dobře vědí, že to jde i jinak než přes mega/giga/tera a programování typu RandomClicking (viz intro, milovníci Dementů budou opět nadšeni).

Věřím, že nové číslo ZXM se Vám bude líbit. Tentokrát je perfektně vyvážené a na své si jistě přijde úplně každý. Možná jsme to jen trošku přehnali s tou velikostí PDFka. Osm mega je přeci jen osm mega, ale Vy už to nějak tím drátem určitě protáhnete. Ty obrazy prostě stojí za to!

Keep smiling, keep coding, keep Speccy & ZXM!

MTs

Obsah

Vodní blábol

místo blábolu úvodního, podvodního, závodního 2

Matematické rutiny

vždyť programovat je o hubu 3

Forever five

jak to vypadá, když si ataristi, commodoristi i spectristi hrají hezky spolu 5

SAPI-1 emulátor na ZX Spectru

i na Spectru nám běhají emulátory 8

MDOS File Commander

název mluví za vše, ne? 10

Silvestrovská pecka a Mluvicí balík

jak to spolu může souviset? 12

Abe's mission

...a zase bude co pařit 13

Bit

co jste si mohli přecíst v legendárním Bitu 16

Intro

...správný spectrista čte ZXM odzadu 19

V příštím čísle najdete:

Jako vždycky Intro a bezva obrázek na obálce.

Recenzi, návod, mapu a cheat. To vše a snad i více o dungeonu Mezi věžemi.

Jak připojit ZX Spectrum 128 na RGB přes SCART, stručně, jasně, přehledně pro machry i pro začátečníky.

Co jste nikdy nechtěli vědět o hardware ZX Spectra a nenapadlo vás se zeptat.

Cesta do Ruska a Návrat do Ruska. Žádné cestopisy, ale dozvíte se, jak pokořit ruské Dizzy hry.

Další emulátor pro Spectrum. Protože na Spectru toho jde emulovat spousta. Vy jste to nevěděli?

Pochopitelně tam může být i článek, který sami napíšete a nám pošlete. Tož pište!



ZX Magazin – časopis pro uživatele počítačů ZX Spectrum a komp.

Vydavatel a šéfredaktor: Jaroslav Směták

Redakční rada: Jaroslav Směták, Martin Tobola, Jiří Doležal

Sazba: volně podle Lubomíra Bláhy

Grafická úprava: Jan Hanousek, Lubomír Bláha

Příprava obálky: volně podle Lubomíra Bláhy

Tisk předloh: Jaroslav Směták

Adresa redakce: Jaroslav Směták, Velká Lhota 111, 756 27

<http://xzm.speccy.cz>

Neprochází jazykovou korekturou. Za obsah příspěvku a jeho původnost ručí autor. Inzerce přijímá redakce. Za její obsah ručí inzerent. Cena inzerce dle dohody. Distribuce formou předplatného a soukromými prodejci.

Vychází nepravidelně. Doporučená cena: **34 Kč**

©2004 ZX Magazin, Zbyněk Vanžura

Jakékoli reprodukce a přetisk materiálů z tohoto časopisu jsou možné pouze s **písemným** svolením vydavatele.

Matematické rutiny

...pokračování z minulého čísla

+GAMA

Integer násobení

Klasické 8-bit x 8-bit unsigned

Input: H, E; L = 0, D = 0

Output: HL

```
sla h ;první iterace
jr nc,$+3
ld l,e
addhl,hl ;7x
jr nc,$+3
addhl,de
```

Klasické 16-bit x 8-bit unsigned

Input: A, DE; HL = 0, C = 0

Output: A:HL

```
adda,a ;první iterace
jr nc,$+4
ld h,d
ld l,e
addhl,hl ;7x
rla
jr nc,$+4
addhl,de
adca,c
```

Klasické 16-bit x 16-bit unsigned

Input: DE, BC; HL = 0

Output: DE:HL

```
sla e ;první iterace
rl d
jr nc,$+4
ld h,b
ld l,c
addhl,hl ;15x
rl e
rl d
jr nc,$+6
addhl,bc
jr nc,$+3
inc de
```

Násobení s tabulkou mocnin 8-bit x 8-bit signed

Input: B, C (oboje -128..127)

Output: HL

Rutina používá jeden z výrazů $2ab = (a + b)^2 - a^2 - b^2$ nebo $2ab = a^2 + b^2 - (a - b)^2$ v závislosti na přetečení $(a + b)$. Mocniny dvou se hledají v tabulce dlouhé 512 bajtů a zaručené na násobek 256. Obsahuje prvky $SqrTab[x] = x^2$. Pokud bereme jeden z operandů jako zlomek $-1..1$ násobený 128, 2ab provádí shift výsledku do registru H. To je užitečné hlavně pro $x \cdot \sin(y)$. Jinak musíme posouvat HL doprava (dělit 2). Místo toho můžeme vydělit dvěma jednotlivé prvky tabulky, což ale sníží přesnost.

Mul8x8ld h,SqrTab/256

```
ld l,b
ld a,b
ld e,(hl)
```

```
inc h
ld d,(hl) ; DE = a^2
ld l,c
ld b,(hl)
deh
ld c,(hl) ; BC = b^2
adda,l ; zkus (a + b)
jp pe,Plus ; jump if no overflow
subl
subl
ld l,a
ld a,(hl)
inc h
ld h,(hl)
ld l,a ; HL = (a - b)^2
ex de,hl
addhl,bc
sbch,de ; HL = a^2 + b^2 - (a - b)^2
; sra h ; odREMuj pro real výsledek
; rr l
ret
Plus ld l,a
ld a,(hl)
inc h
ld h,(hl)
ld l,a ; HL = (a + b)^2
or a
sbch,de
or a
sbch,de ; HL = (a + b)^2 - a^2 - b^2
; sra h ; odREMuj pro real výsledek
; rr l
ret
```

Generátor tabulky mocnin je založen na tom, že rozdíl mezi konsektivními mocninami (0, 1, 4, 9, 16, 25, ...) je sekvence lichých čísel (1, 3, 5, 7, 9, ...).

```
ld hl,SqrTab;násobek 256
ld b,l
ld c,l ;v BC lichá čísla
ld d,l
ld e,l ;v DE mocniny
SqrGen ld(hl),e
inc h
ld (hl),d ; x^2
ld a,l
neg
ld l,a
ld (hl),d
deh
ld (hl),e ; -x^2
ex de,hl
inc c
addhl,bc ;další liché číslo
inc c
ex de,hl
cpl ; jednobajtová náhrada NEG, DEC A
ld l,a
rla
jr c,SqrGen
```

Násobení s tabulkou mocnin 6-bit x 6-bit signed

Přesněji, tato rutina násobí dvojici čísel x a y , pokud $(x + y) \leq 127$ a $(x - y) \geq -128$. Jsou-li x a y 6-bit hodnoty, je toto pravidlo vždy dodrženo.

Základem je rovnice $4xy = (x + y)^2 - (x - y)^2$ a použití stejné tabulky, jako u rutiny minulé, s tím, že všechny její prvky jsou děleny 4. Pokud zůstane tabulka bez úpravy, bude rutina provádět dělení s pevnou desetinnou čárkou. Pokud je jeden operand zlomek v rozsahu $(-1, 1)$ násobený 64, celočíselná část výsledku se posune do registru H. Tabulka musí být zarovnána na násobek 256.

Input: B, C

Output: BC

Mul6x6ld h,SqrTab/256

```
ld d,h
ld a,b
adda,c
ld l,a
ld a,b
subc
ld e,a
ld a,(de)
sub(hl)
ld c,a
inc h
inc d
ld a,(de)
sbca,(hl)
ld b,a
```

Celočíselné dělení

Dělení je hnušná operace i s hardwareovou podporou, ani od dobře napsaného kódu nesmíte čekat rychlostní rekordy. Ale...

Pokud je dělitel mocnina dvou, použijte rotace.

Pokud dělíte konstantou, použijte tabulku.

Je-li to možné, rozdělte operaci tak, aby se prováděla s mocninami dvou. $X / 10 = (X + X / 2 + X / 8 - X / 64 + ...) / 16$.

Pokud dělíte čísla dobře limitovaná (třeba šestibitová), použijte tabulku. Dělenec a dělitel mohou představovat vyšší a nižší bajt adresy výsledku.

Někdy lze násobit převrácenou hodnotou (s pevnou desetinnou čárkou), ovšem se zatížením chybou.

Někdy (u malých čísel) může být nejrychlejší metodou opakované odčítání, které je obecně naopak nejpomalejší.

SLIA je jen částečně dokumentovanou instrukcí Shift Left Inverted Arithmetic (kódy 30h..37h s prefixem CBh, DDh CBh nebo EDh CBh), známá i jako SLL (Shift Left Logical). Posouvá registr doleva, nastavuje nejnižší bit na 1. Většina rutin nepoužívá

Programování

registru B, což umožňuje tvořit DJNZ smyčky, je-li potřeba šetřit místem v paměti a rychlost není tak podstatná.

Klasické 8-bit / 8-bit unsigned

Input: D / E, A = 0

Output: D, A = mod

```
sla d      ;8x
rla
cp e
jr c,$+4
sub e
inc d
```

Slabinou tohoto algoritmu je komplementace carry. Pokud odečteme dělitel od části zbytku, carry se nastaví na 0 a je potřeba přidat k výsledku 1. Jde se tomu vyhnout zanedbáním carry a komplementovat až celý výsledek (s čímž roste režie).

Input: D / E, A = 0, carry = 0

Output: A, E = mod

```
rl d      ;8x
rla
sub e
jr nc,$+3
adda,e
ld e,a ;ulož zbytek
ld a,d ;komplement
cpl
```

Záleží-li na každém cyklu, je tu ztrátový algorytmus. Rutina se dělí na takřka identické dráhy kódu (jsou zapsané vedle sebe).

Input: D / E, A = 0

Output: D, A = mod

```
sla d
rla
cp e
jr c,C1
NC0 sube
sla d C1 sla d
rla rla
cp e cp e
jr c,C2 jr nc,NC1
```

```
NC1 sube
sla d C2 sla d
rla rla
cp e cp e
jr c,C3 jr nc,NC2
```

```
NC2 sube
sla d C3 sla d
rla rla
cp e cp e
jr c,C4 jr nc,NC3
```

```
NC3 sube
sla d C4 sla d
rla rla
cp e cp e
jr c,C5 jr nc,NC4
```

```
NC4 sube
sla d C5 sla d
rla rla
cp e cp e
jr c,C6 jr nc,NC5
```

```
NC5 sube
sla d C6 sla d
rla rla
cp e cp e
jr c,C7 jr nc,NC6
```

```
NC6 sube
sla d C7 sla d
```

Klasické 16-bit / 8-bit unsigned

Input: HL / C, A = 0

Output: HL, A = mod

```
addhl,hl ;16x
rla
cp c
jr c,$+4
subc
inc l
```

Klasické 16-bit / 16-bit unsigned

Input: A:C / DE, HL = 0

Output: A:C, HL = mod

```
sla c ;16x
rla
adchl,hl
sbchl,de
jr nc,$+4
addhl,de
decc
```

Klasické 24-bit / 8-bit unsigned

Input: E:HL / D, A = 0

Output: E:HL, A = mod

```
addhl,hl ;24x
rl e
rla
cp
jr c,$+4
subd
inc l
```

Klasické 24-bit / 16-bit unsigned

Input: A:BC / DE, HL = 0

Output: A:BC, HL = mod

```
sla c ;24x
rl b
rla
adchl,hl
sbchl,de
jr nc,$+4
addhl,de
decc
```

Klasické 32-bit / 8-bit unsigned

Input: DE:HL / C, A = 0

Output: DE:HL, A = mod

```
addhl,hl ;32x
rl e
rl d
rla
cp c
jr c,$+4
subc
inc l
```

Generátory náhodných čísel

Osmibitové náhodné číslo

Základní rovnice tohoto jednoduchého generátoru je $x[i + 1] = (5 * x[i] + 1) \bmod 256$. Její jediná výhoda je jednoduchost a malé nároky na paměť. Díky povaze tohoto generátoru lze jen část vyšších bitů považovat za náhodné.

Input: -

Output: A = pseudonáhodné číslo s opakováním 256

Rand8ld a,Seed

; obvykle 0

```
ld b,a
adda,a
adda,a
adda,b
inc a ;nebo ADD A,7
ld (Rand8+1),a
ret
```

Šestnáctibitové náhodné číslo

Opět podobná metoda, ovšem s lepšími výsledky. Jde o optimalizovanou rutinu jedné staré spectrácké hry.

Input: -

Output: HL = pseudonáhodné číslo s opakováním 65536

Rand16 ldde,Seed

; obvykle 0

```
ld a,d
ld h,e
ld l,253
or a
sbchl,de
sbca,0
sbchl,de
ld d,0
sbca,d
ld e,a
sbchl,de
jr nc,Rand
inc hl
Rand ld (Rand16+1),hl
ret
```

Konverze čísel na řetězce

16-bit integer na ASCII (decimálně)

Input: HL = číslo, DE = adresa řetězce

Output: řetězec na (DE)

Num2Dec ld bc,-10000

```
call Num1
ld bc,-1000
call Num1
ld bc,-100
call Num1
ld c,-10
call Num1
ld c,b
Num1 ld a,'0'-1
Num2inc a
addhl,bc
jr c,Num2
sbchl,bc
ld (de),a
inc de
ret
```

16-bit integer na ASCII (hexa)

Input: HL = číslo, DE = adresa řetězce

Output: řetězec na (DE)

Num2Hex ld a,h

```
call Num1
ld a,h
call Num2
ld a,l
call Num1
ld a,l
jr Num2
```

```
Num1rra
rra
rra
rra
Num2or F0h
daa;hnusná finta
adda,A0h
adca,40h
ld (de),a
inc de
ret
```

```
rld
ret
```

Některým osobám se nelíbí instrukce DAA (nekompatibilita s původní DAA na 8080). Pokud i vy patříte k jejím nepřátelům, zkuste si následující:

```
DECHEXLd a,h
call DECHEXA
ld a,l
DECHEXA push af
rrca
rrca
rrca
rrca
call DH0
pop af
DH0 and %00001111
add a,"0"
cp 10+"0"
jr c,DH1
add a,"A"-"0"-10
DH1 ld (de),a
inc de
ret
```

Lze volat DECHEXL (HL->ASCII HEX, out=4 bytes), DECHEXA (A->ASCII HEX, out=2 bytes), DH0 (A_low->ASCII HEX, out=1 byte).

Výpis paměti (hexa)

Jde o typickou často prováděnou úlohu, kterou lze optimalizovat. Rutina vezme bajt z (HL) a vytiskne.

Input: HL = adresa dat

Output: výpis

Makro PRINT_CHAR nahrad'te rutinou pro tisk znaku, nezapomeňte zachovat HL.

```
xor a
rld
call Nibble
...
```

Nibblepush af

```
daa
adda,F0h
adca,40h
```

```
PRINT_CHAR ; znak je v A
popaf
```

Cyclic Redundancy Check (CRC)

Tato 16bit CCITT-CRC rutina počítá CRC s použitím polynomu 1021h. Jiný běžný CRC-16 používá polynom A001h a začíná hodnotou 0 (čiže je možné mylně interpretovat sadu nul jako platná data).

Input: DE = adresa dat, C = délka dat

Output: HL = CRC

Crc16 ld hl,FFFFh

Read ld a,(de)

inc de

xor h

ld h,a

ld b,8

CrcByte addhl,hl

jr nc,Next

ld a,h

xor 10h

ld h,a

ld a,l

xor 21h

ld l,a

Next djnzCrcByte

dec c

jr nz,Read

ret

Jako každý třetí víkend v březnu se i letos, konkrétně 19.-21.3.2004, konal pátý ročník mnohoplatformní osmibitové akce s názvem

forever five

Sweet Factory

A protože je ZX Spectrum osmibitové, neměli tam reportéři **CI5** chybět, aby se pak objevila reportáž v našem oblíbeném časopise ZX Magazin.

Výprava z druhé strany naší malé republiky vyrážela v pátek dopoledne. Já/CI5 v osm ráno z Domažlic, Shrek/MBManiax v devět z Plzně a Tritol/DTP v půl jedenácté z Prahy. Na místo činu jsme dorazili ve čtyři dopoledne a dohromady moje auto urazilo 1000 kilometrů.

Protože jsme přijeli mezi prvními, zabrali jsme si hned žíněnky, abychom měli na čem spát. Museli jsme je pak bránit, někdo se na ně začal sápat, i když jsme na nich měli bagáž.

Shrek měl s sebou celou ZX sestavu, já jen notebook a Tritol jenom Compact Flash karty. Na Slovensko jsem vezl i jedno CD pro Pola, zašel jsem k organizátorům se zeptat, kde je Pol. Odpověď jsem dostal: šel pro svačinu do obchodu. Za chvíli jsme se dozvěděli, že i Aragorn dorazil před námi, tak jsme se zeptali kde je. Odpověď: Šel s Polem pro svačinu. Během dne pak ještě dorazili i jiní Spektristi jako z00m s Tornadem, Logout s TV, LCD a Gasman už tam byli.

Shrek, Tritol a z00m (a jeho +2 s jeho MB02) se snažili přijít na to, proč nefungu-

je Shrekovo CF. Nakonec se ukázalo, že chyba je v MBIDE, z00m má svoje MBIDE upravené. Prostě originální MBIDE od 8BC nejde tak, jak by mělo.

Já jsem se za pomoci Tritola snažil nastavit barevné zvýrazňování syntaxe assembleru Z80 do editorů kwrite a kate. Mimochodem, pokud byste po něm někdo toužil, ozvěte se mi a obratem vám ho pošlu.

V pátek už se asi nedělo nic tak mimořádného, byli jsme na večeri v Kolibě, z00m tvořil hudbu do realtime compa, Tornado kreslil do realtime compa, Mike nám ukázal pěknou animaci pro Speccy na PeCi (o té bude ještě řeč), Logout se vytahoval se svým 486 notebookem.

V sobotu jsme přivítali zbytek spektristů: Factor6, TDM, Omega, POke, Zilog, 8BC, diZZy, Nairam, Baze, BlakeR, Soaz (jen kvůli terminálu od Ziloga). Jestli jsem na někoho zapomněl, nebo někoho přehodil, tak se omlouvám (xorry).

V sobotu dopoledne nás čekalo placení mýtného, tj. průchod dveřmi za 300 slovenských korun, a pak připravené soutěže: hod disketou 5,25 a turnaj v C64 fotbale (vrh harddiskem a upalování Atari se nekonalo, protože jsem je na Forever nemohl dovézt, možná příště - pozn. +GAMA) a večer pak více nebo méně standardní compa.

V hodu disketou nikdo ze známých ne-soutěžil, ale byl jsem u toho, jak to několi-krát odnesla zářivka na stropě, někdo přehodil celý sál a trefil protější stěnu, taky některá hlava přihlížejících diváků. Naštěstí je disketa jen z papíru, takže nic neutrpělo vážnější poškození. Výherce soutěže pak vyhrál balík nových 5,25 disket, kdyby je vyhrál Spektrista, tak by mu asi byly k ničemu, protože my už používáme Compact Flash karty, v nejhorším případě pak harddisky. Až se bude příště házet CF kartou, určitě se přihlásím, abych jich pak vyhrál celý balík.

Soutěž v počítačovém fotbale z našich řad obhajovali Shrek s Logoutem (tedy MBManiax). Tento fotbal z produkce firmy Microprose se hrál na Commodoru C64, což nesl s nechutí hlavně Shrek (a mám ho u toho komouše vyfoceného). (platforma se podepsala i na výsledcích - commodoristé fotbal vyhráli; snad by bylo lépe hrát na něčem neutrálním - pozn. +GAMA)

Někdy před jednou hodinou jsme vyrazili na oběd, protože v jednu hodinu měl dorazit Zilog a ten byl s Logoutem domluven, že se setkáme rovnou v Kolibě. Bylo nás celkem pozeňnané, takže když jsme otevřeli dveře Koliby, nějaký odcházející host si povzdechl: "Zase nějaký zájezd...". Takže jsme celou Kolibu zaplnili. Servírka se od loňska změ-

nila, byla k nám lachná a pár objednávek spletla. Pak se ve dveřích objevil Zilog, řekl: "dobry den" a na stůl položil monitor od terminálu. Za ním ještě dorazili Baze s Blackerem. Zilog si poručil halušky a smažák a pak skoro zapomněl, že si ho objednal. Předtím ještě vytasil digitální foťák a začal nás s ním fotit.

Po obědě, tedy spíše už k večeru měly být hlavní soutěže, ale uzávěrka příspěvků se stále posouvala (to kritizují i ataristé a commodoristé - pozn. +GAMA). Nakonec se začalo s promítáním až v sedm večer a trvalo až do jedné hodiny v noci. Marně jsem se snažil upozornit kolemsedící, že mám hlad a že jídlo je mi přednější než democompo. Nikdo se ke mně nepřidal, takže všichni zůstali za trest bez večere (i já).

Před vlastním promítáním nás pořadatelé vyhnali ven, aby zkontrolovali, kdo platil, a rozdali hlasovací listy.

Promítání bylo dlouhé, někdy nudné a někdy velice zábavné. Nebudu vykládat, co všechno jsme viděli, jen připomenu zajímavé momenty.

Jako první bylo realtime compo, ve kterém z našich řad uspěl z00m.

V music compu byla velice zajímavá skladba od Factora6, který s ní dokázal roztloukat celý sál. Ze všech příspěvků v music compu bylo Spektrálních nejvíce, celých dvacet, úplně zaplnily celých "votesheet". Z ataráckých a commodoráckých skladeb mě

nezaujala ani jedna skladba, na rozdíl od loňského roku, kde byla alespoň jedna. (Commodoristi velmi kritizovali aparaturu, kterou letos připravovali spectristé. Aby nebylo tolik slyšet AYpsilonkové pískání, byly zdůrazněny basy, takže skladby na SIDu, který basuje už sám o sobě, zněly velmi trapně - pozn. +GAMA)

V demech pak soutěžili MBManiix s velice odvážnými texty proti Microsoftu, Billu Gatesovi, Commodore a Atari. POke poznamenal: "Všichni to víme, ale konečně to někdo řekl nahlas". Z ataráckých dem se nám všem líbilo demo Basix (autor Raster/CPU), které sice bylo efektů prosté, ale nápad byl velice originální.

Hlasování skončilo v jednu v noci a někteří šli už na kutě.

Jak nám v sobotu ukazoval Mike (nebo to byl Pol?) tu animaci, tak se Tritol rozhodl udělat ji přímo na Speccy, protože data animace byla od LCD a byla jen samé SCREEN\$, které se LOADovaly do videoramky. V emulátorech vypadá hezky, ale na DEMFIRu byla nepoužitelná, vzhledem k tomu, že soubor s TAPkou obsahoval 4738 souborů. Tritol všechna tato data pospojoval do jediného souboru (33 MB) a pak jsme k tomu museli ještě napsat player. Nejprve se objevil problém, jak do holého divIDE nahrát DEMFIR, když nemáme kazeták, jen PeC. Ukázalo se, že RealSpec ani Hypraloader pod DOSem nedokážou bez chyby vypískat TAPku do

zvukovky (a co Zilogovo TAP2LPT, viz ZXM 1/04, strana 3? - pozn. +GAMA), nakonec se objevil Pol a předvedl nám, že "Linux umí všechno". Ve finále jsme s Tritolem psali player na MandrakeLinuxu, kompilovali pomocí ASLu a vyrobili ISO image pro DEMFIR a rovnou naprali na CF kartu. Prostě s Linuxem na věčné časy a nikdy jinak. Celou animaci jsme pak pustili na bigscreenu a proto, co ji neviděli, je celá barevně, ale bez zvuku. Data dodal LCD a už tuším, jak to udělal. Vzal filmový klip, převedl ho do nekomprimovaného AVI formátu 256*192 pixelů příkazem 'mencoder klip.avi -nosound -ovc rawrgb -vop scale=256:192 -o filmzx.avi' a pak svým programem BMP2SCR jej "vygraboval" do poctivých barevných spektrálních obrázků.

V neděli bylo jen vyhlášení vítězů a předávání cen, ale jak už jsem dodal, diskety jsou dneska dobré akorát tak na házení (a možná ještě do PeCe), ale moderní ZXS by mělo už používat CF karty.

Cestou zpátky jsme se stavěli v Bučovicích na oběd. Řeknu vám, až někdy pojedete autem cestou z Brna do Uherského Hradiště (třeba na Zlincon nebo na Shucon), rozhodně se tam zastavte a najděte restauraci "Katolický dům". Opravdu stojí za to a já nebo Shrek si tu příležitost nenecháme ujít. A až ji najdete, sedněte si ke stolu úplně vlevo až u okna.

Tak zase někdy příště.



celkový pohled

pravá party-sestava 128+2 s MB-02+



Atari! yeah!

známe své kořeny, i v hospodě...





naši borci se číní, aneb C64 vs ZX



máma má mísu, Tritol má emulátor



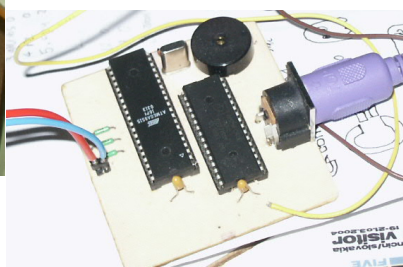
na Foreveru vyšel nový ZXM

Sinclairova přenosná televize, vhodná na party



UNIXový terminál - radost všem dětem!

přenosné PMD-85, vhodné na party



spectristé... (?)



SAPI - 1

emulátor na ZX Spectru



Se socialistickým "mikropočítačem" SAPI-1 jsme se na stránkách ZX Magazínu již setkali (ZXM 4/01, strana 18). Připomeňme si, že je to

velká bedna, v základní konfiguraci se čtyřmi kB ROM a procesorem 8080. Pokud není SAPI vybaveno vzácnou grafickou kartou, dává na obrazovce textový mód 40x24 znaků s dvěma atributy.

Spectrum určitě znáte a emulátory Spectra na nějaké Amize, Applu, nebo přinejhorším aspoň na PeCi znáte určitě taky. Co ale mají se SAPI společného?

Emulátor

Tvůrci 80kB Spectra (pro potřeby CP/M) se jistě nevzhlíželi v československém průmyslovém počítačle, ale potřebovali cross-assembler pro jednočipy řady 8048. Pro Spectrum nemohli žádný sehnat a SAPI zřejmě používat nechtěli (*zabere spoustu prostoru - mohu potvrdit, měli jsme to doma*). Existovala sice možnost program pro Spectrum nějak přeprogramovat, což by ovšem vyžadovalo spoustu práce, a podobně pak postupovat s každým dalším programem, který by bylo potřeba převést, existovala ovšem i možnost využít nové možnosti Spectra přistránkovat si RAM místo původní ROMky a napsat emulátor, na kterém by všechny další programy ze SAPI bez problémů chodily.

Program byl sice původně určen pro Spectrum 80 kB (v módu "cracker"), po drobné úpravě by ale chodil i na jiných úpravách nebo se zařízeními, která zvětšují paměť a umožňují nastránkovat RAM místo ROM.

Emuluje SAPI-1 s operačním systémem verze 4.0 (t.j. Monitor + MIKOS), a to tak věrně, že i při případné chybě se zhroutí stejným způsobem jako SAPI. Jediným rozdílem je nahrávání - SAPI (stejně jako Ondra nebo PMD-85) používá fázovou modulaci, vyžadující hardwarovou podporu. Spectrum proto používá záznam logicky identický se záznamem SAPI, ale fyzicky je použit normální záznam Spectra. Pro přenos dat ze SAPI-1 do tohoto formátu ovšem existuje program EmuSAPI, který je také určen pro Spectrum s rozšířenou pamětí.

Emulátor stačí nahrát, sám si otestuje paměť, rozmístí se a spustí, ohlásí se systémem SAPI a můžete pracovat.

Ondra

Zmínka o tomto počítači s emulátorem příliš nesouvisí:

Původní verze Ondry měla v ROM systém převzatý ze SAPI, ovšem šlo o MIKOS verze 5. Systém Ondry vznikl vlastně jen úpravou a překompilováním ROMky verze 4, díky odlišnému hardware byly ale změny tak velké, že přenositelné jsou snad jen datové soubory, spustitelné programy ne (už jen proto, že běhají v úplně jiné části paměti). Zatímco SAPI má třeba VideoRAM v oblasti #3800-#4000, Ondra má grafickou VideoRAM v posledních deseti kilech. Tam ale na SAPI běhají strojčkové programy (standardní aplikace se spouští od #4400, zatímco na Ondrovi od #1000). I služby BIOSu jsou jinde.

Nutno podotknout, že šlo o dost pitomý nápad. Ondra byl počítač určený pro děti a poněkud krkolomné a nesrozumitelné (třeba důsledné používání hexadecimální soustavy) ovládání MIKOSu je poněkud silné kafe. Nehledě na to, že Ondra má několik specifik, co se týče zobrazování. Emulace textového režimu na grafice nebyla napsaná zrovna šťastně - scroll o jeden řádek trvá snad několik sekund, při časově náročných operacích se zobrazování vypíná a uživatel neví, co se vlastně na obrazovce v tu dobu děje (to je zvlášť "užitečné" při nahrávání z kazety).

Nová ROM verze 87, napsaná Vitem Libovickým, je sice z programátorského hlediska podobný "skvost", má ale rychlý scroll a hlavně při nahrávání z kazety aspoň něco zobrazuje (i když poblikává).

A její ovládání je narozdíl od MIKOSu dětsky jednoduché, stačí zadat jméno programu, který se bude nahrávat. Jméno uvozené znakem '#' znamená, že se bude nahrávat ze sítě místo z kazety, prázdný název znamená první nalezený soubor (a samotná '#' první soubor, který přijde po síti).

Takže ještě jednou pro jistotu - programy ze SAPI na Ondru jsou normálně nepřenositelné (musíte upravovat zdroják a kompilo-

vat na jinou adresu).

Organizace SAPI-1

Chcete-li vědět, jak emulátor funguje, hodí se znalost struktury SAPI.

Ve čtyřech kilech EPROM je umístěn

Mapa paměti SAPI-1

FFFF	
	pracovní oblast aplikací
6000	
	kód aplikací
4400	
	volné pro uživatele nebo aplikaci
4300	
	buffery MIKOSu
4000	
	VideoRAM na kartě AND-1
3800	
	volné místo EPROM
3400	
	porty a přerušení
2400	
	RAM na procesorové kartě JPR-1
2000	
	volné místo EPROM
1000	
	MIKOS a Monitor+
0800	
	Monitor a BIOS
0000	

BIOS a Monitor (zajišťuje základní práci s pamětí) a Mikos s Monitorem+ (Monitor+ odchyťává příkazy pro MIKOS, MIKOS je "mikro kazetový operační systém" - odvozený z operačního systému PhobOS).

Ačkoliv Eda Smutný ve svém Kursu číslkové techniky (Součástková základna mikropočítačů IBM PC XT, 5. díl, 602. ZO Sva-zarmu) důsledně nedoporučuje mapování portů do paměťového prostoru procesoru Z80, sám se tohoto nešvaru v dopouští nejen v SAPI, které bylo původně vybaveno procesorem 8080, ale i v Ondrovi, který nic jiného než Z80 nikdy neměl (tuto "podivnost" ale používají i jiní, je tak třeba napa-pován diskový řadič u řady MSX) a u SAPI si pro něj vyhradil poměrně velkou oblast.

Oblast RAM začíná v podstatě Video-RAMkou (#3800), následují buffery systému. Aplikace se ukládají a startují od adresy #4400,

Monitor

V okamžiku, kdy na vás z emuátoru vy-bafne tečka (prompt MIKOSu), budete ho muset umět ovládat. Nejprve se seznámíme s Monitorem.

Monitor se ovládá jednopísmennými pří-kazy, po zadání některých očekává i zadání parametrů (vždy hexadecimálně).

Mapa paměti Ondry (pro srovnání)

FFFF	VideoRAM
D700	volno pro uživatele
D640	český font
D360	malá písmena
D280	velká písmena
D000	zásobník a buffery MIKOSu
CE7F	
	pracovní oblast aplikací
	kód aplikací
1000	volno pro uživatele
0D00	Monitor a MIKOS
0000	

Existují ovšem některé klávesové povely, které fungují při takřka všech činnostech počítače. SHIFT+C přeruší provádění progra-mu a vrátí se do Basicu nebo assembleru (je-li v paměti). SHIFT+W přeruší provádění programu a vrátí se do Monitoru. SHIFT+P připojí nebo odpojí výstup na tiskárnu. SHIFT+Q vytvoří na obrazovce "okno", kam budou směřovat všechny výpisy a scrolls, za-tímco zbytek obrazovky (třeba nějaký důleži-tý výpis) zůstane zachován. Po této kombi-naci se zadává jednou klávesou velikost okna (počet řádků od horního kraje obrazov-ky) - nula je zrušení okna, klávesy 1-9 jsou pro 1-9 řádek.

Povelů není mnoho (však je taky EPROMka dost malá), ale prostředí je uživa-telsky nepřívětivé, nepřehledné, a proto vy-padá složitě.

.A ASSIGN přiřazuje vstupní a výstupní periférii (C-CONSOLE terminál, R-READER kazeta nebo děrná páska pro čtení, P-PUNCH kazeta nebo děrná páska pro zápis, L-LIST tiskárna) fyzické zařízení (T-konzole s tiskárnou, C-konzole, B-IN kazeta, OUT tis-kárna, P-kazeta, L-tiskárna)

.C CLEAR studený start, nastaví všechna data a podmínky na defaultní hodnoty, vy-prázdni souborové buffery.

.D DISPLAY zobrazí paměťový výpis. Oče-kává dvě adresy - odkud a kam.

.F FILL naplní blok paměti konstantou (očekává na vstupu odkud, kam a čím plnit).

.G GOTO skočí na zadanou adresu (me-zerou lze potvrdit původní adresu, nebo za-dat novou, zadává se bez mezery ve formátu Gxxxx, odešle se mezerou nebo enterem). Lze připsat i další dvě adresy, které budou sloužit jako breakpointy.

.M MOVE překopírování bloku paměti (za-dávají se tři adresy: začátek bloku, konec bloku, kam).

.Q QUERY zobrazí stav přiřazení jednotli-vých periférií (jména zařízení zobrazuje tak, jak jsou uvedena u povelu ASSIGN).

.S SUBSTITUTE editace obsahu paměti. Vyžádá si adresu, pak je možné buď posu-nout se na další bajt stiskem mezery, nebo zapsat novou hodnotu, po znaku apostrof lze vložit znak. Ukončí se enterem.

.X EXAMINE zobrazí všechny registry (po stisku enteru) nebo umožní měnit jejich hodnoty (po postupném odmačkávání po-mocí mezery).

Povely rozšířeného Monitoru+ jsou ná-sledující:

.J JUMP byl-li z kazety nahrán program, spustí se.

.I INPUT zruší příznak otevření vstupního souboru (jméno zařízení se zadává jako u ASSIGN).

.O OUTPUT zruší příznak otevření výstup-ního souboru (jméno zařízení se zadává jako u ASSIGN).

.R READ čte ze souboru Intel-hex formát.

.W WRITE zapisuje do souboru ve formá-tu Intel-hex. Zápis poslední části souboru se musí ukončit povel EOF.

.E EOF uzavře Intel-hex soubor, zadání startovací adresy (zapisuje se na konec souboru).

.N NULL děrování blanku (120 prázdných znaků), směřuje do souboru.

Další tři povely spustí příslušný program, byl-li nahrán do paměti, podobně jako to dělá JUMP.

.P PEDIT spustí editor PEDIT.

.T TRACER spustí debugger TRACER.

.B BASIC spustí Basic-EX.

Poslední povel není vlastně povel, ale prefix ukazující, že bude následovat povel MIKOSu pro kazetu.

.K KAZETA očekává jeden z příkazů pro MIKOS.

MIKOS

Tak tady jsou příkazy MIKOSu.

K_A APPEND otevření výstupu bez vyhle-dávání volného místa (rovnou zapíše hlavič-ku).

K_B BASIC čte záznam ve formátu Mik-ro-BASICu. Jako parametr se zadává adresa, na kterou se mají data uložit. U tohoto pove-lu se na chvilku zastavíme.

U první sestavy SAPI-1 (ZPS 1, základní počítačový systém 1) obsahovala 4 kB ROM kromě základního Monitoru i celý interpret integer Basicu, kazetová rutina je psaná co nejúsporněji a používá jiný formát, než poz-dější verze (třeba délka jména souboru je 64 znaků).

Druhá verze (ZPS 2) už byla vybavena ROM verze 4 s Monitorem a MIKOSem. Jako interpret Basicu se začal používat Ba-sic-EX s desetinnou čárkou, nahrávaný z ka-zety, mění se formát souborů (sektor 256 bajtů, délka jména je nadále 11 znaků). Pro import dat ve starém záznamu (např. výsle-dky měření, ukládané na pásku) byl zaveden právě povel K_B.

Aby v tom byl ještě větší zmatek, ROM 87 pro Ondru zavádí další formát (sektor 1024 bajtů, jméno 12 znaků s třípísmen-nou příponou).

K_C CLOSE uzavření výstupního soubor-u.

K_D DIRECTORY výpis adresáře (projíždí celou kazetu a vypisuje jména souborů).

K_F FIND otevření vstupního souboru.

K_I INIT inicializace kazety (nahraje úvodní hlavičku se jménem kazety).

K_L LOAD načte binární soubor.

K_M MOTION zapne motor, je možné přetáčet kazetu.

K_O OPEN otevření výstupního souboru.

K_S SAVE uložení binárního souboru, jako parametry se zadávají první a poslední bajt bloku.

Všimněte si, že jméno se nezadává jako parametr, MIKOS se na něj ptá v dialogo-vém režimu (který má zjednodušovat jinak poměrně složitou práci s kazetou) odděleně. Lze použít omezenou hvězdičkovou konven-ci - při čtení hvězdička znamená libovolný soubor.

Během vyhledávání a čtení souboru se zobrazují názvy souborů, čísla čtených sek-torů a počet sektorů v souboru. Znak ':' indi-kuje hledání souboru, znak '*' signalizuje čtení, znak '#' indikuje chybu při čtení.

MDOS File Commander

Sweet Factory

V letošním vydání ZX Magazínu (chtěl tím snad někdo říct, že ZXM vychází jednou ročně? - pozn. +GAMA) se podíváme na zoubek programu MFC, který napsal člen týmu agentů CI5 the amateurs, **MTs**.

Tento program je určen, jak již z názvu vyplývá, pro systémy D40/80/Kompakt. (Z názvu to sice nevyplývá, ale hodí se i majitelům Divide)

V samotném nezkráceném názvu se honosí přívlastkem "Commander", takže je od něj očekáváno, že hravě zvládne základní práci se soubory, jako je kopírování, přejmenování a mazání (mazat jako smahnout, ne užpinit).

Tuto práci a ještě spousty jiných opravdu zvládne, podíváme se jak.

MFC najdete na internetu, konkrétně na adrese <http://mfc.web.wo.cz>. Kromě programu samotného by vám neměly uniknout také jeho utility, spořiče obrazu, konvertory pro Windows a hlavně návod. A s ním taky začneme.

Vzhledem k tomu, že samotný MFC prošel třemi stádii vývoje (vývojová nultá, testovací jednička a finální dvojka), návod je rozdělen na dva kusy a je ve formátu PDF. Takže až dočtete ten tlustší, tak si přečtete změny, které jsou ve verzi 2 a informace si v hlavě musíte nějak dát dohromady.

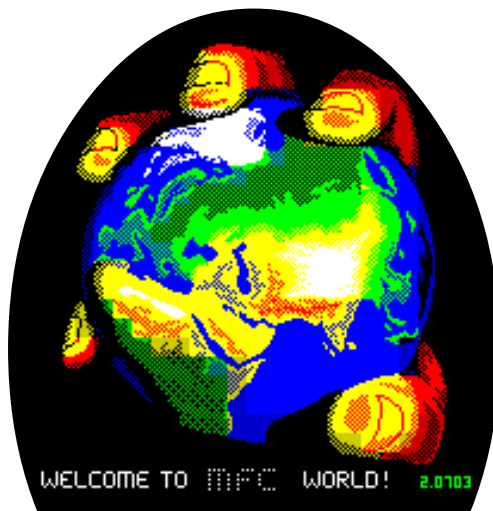
Přimlouval bych se za to, aby návod k MFC byl pohromadě, sám jsem se ve dvou manuálech pořádně neorientoval. (I já připojuji hlas - pozn. +GAMA)

Jinak je ale manuál zpracován velmi důkladně, až přepečlivě. Z něj se dozvíte nejen ovládání a popis funkcí programu, ale i podrobné informace, jak si napsat utility, informace o disketové jednotce a o disketě, popisy formátů nových typů souborů .D_0 a .MFC.

Pokud si budete chtít tento manuál vytisknout, narazíte na menší problém. Manuál je totiž vytvořen tak, aby se dal pohodlně číst na obrazovce, ale není připraven na tisk na papíry formátu A4.

Jediné, co příručce chybí k naprosté dokonalosti, je stručný seznam "hot keys". Takový seznam je sice obsažen v textu obsahu, ale ten je na dvě stránky, pohodlně by se vešel na jednu a ten by si uživatel pak vytisknul a dal někam, aby byl při ruce.

Takže máte nastudováno? Vrhne se totiž na samotný MFC.



V tomto textu budu popisovat verzi aktuální k lednu 2004 a její číslo je 2.0703.

Po spuštění (LOAD *"MFC") vás uvítá celkem pěkný obrázek, který si MTs "vypůjčil" z galerie soutěžních obrázků z party *Chaos Construction 2001*. Pokud tady stisknete mezerník (na Didakticích SPACE), můžete si v MFC jednorázově (tedy do resetu) nastavit některé funkce. MFC lze sice

písmenu logické jednotky, se daná disketa načte. Může to být jak disketa z D40/80, tak z PC (u PC diskety se předpokládá, že bude naformátovaná na DD kapacitu, tj. 720K).

Průběh čtení systémových sektorů vidíte v dolním řádku. Pro uživatele asi zbytečná informace, pro profíka asi jen do té doby, než některý sektor nelze přečíst, pak se dobře pozná, který to je.

Po načtení diskety probíhá **ShakeDir**, takže pokud jste měli úmyslně vytvořené mezery v adresáři tak, aby po zkopírování souboru byl adresář v pořadí, jaké vám vyhovuje, tak jim dejte sbohem. Stejně tak je to se smazanými soubory. Pokud si během práce nějaký soubor omylem smažete a uložíte adresář na disketu, tak už není žádná šance ho obnovit. Pokud ho smažete basicovským příkazem ERASE, tak nějakou nádeji máte.



spouštěť už i s vaším vlastním nastavením, neboť je k dispozici program v basicu, kterým si MFC napoukujete podle libosti, ale vhodnější by byla funkce zajišťující uložení nastavení pro příští spuštění.

Až nastavení dokončíte (nebo přeskočíte), MFC vás vyzve k výběru mechaniky do levého okna. Pokud jste majiteli MDOS3, diVIDE-MDOSu nebo MDOSu 2.1, pak jistě uvítáte možnost si vybrat ze 4 mechanik, pokud máte stále originální MDOS, pak písma C a D jsou nepřístupná. Tuto funkci doplnil MTs do MFC na moji žádost. Protože jsem zrovna pracoval na MDOSu 3 a on na MFC, dohodli jsme se na akceptaci mechanik C a D.

Po udeření do klávesy, která odpovídá

Po setřesení adresáře přichází další na commander neobvyklá funkce - **ScanDisk**.

ScanDisk má za úkol najít a pokud možno opravit všechny chyby ve FAT tabulce diskety. Takovou chybou mohou třeba být sektory bez souboru nebo překřížená FAT. Pokud najde nějakou chybu, zablokuje MFC zápis na disketu a pípne.

Tuto ochranu zápisu může uživatel vypnout, pokud si myslí, že to souborům na disketě neublíží. V seznamu souborů bude takto automaticky opravený soubor označen vykřičníkem.

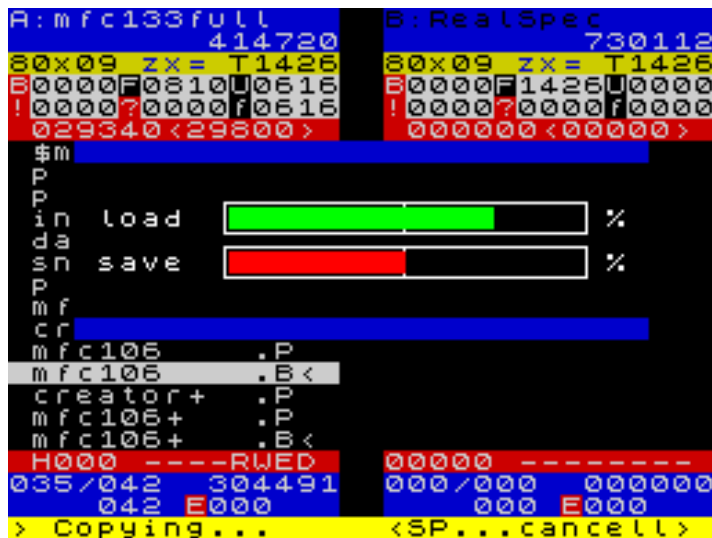
Nemusíte se ničeho obávat, ScanDisk netrvá nikterak dlouho, na rozdíl od Windows.

Sám si myslím, že funkce kontroly disku by měla spíše náležet DiskDoctorům, ale na druhou stranu se dají takto i z mírně poškozené diskety bez problémů zkopírovat životně důležité soubory (například zdrojové texty).

Po setřesení adresáře a kontrole disku se nám konečně objeví seznam souborů a nejen ten.

Uvidíte celkem komplexní informace o disketě a o souborech: jméno diskety, volné místo, formát diskety, vadné, volné, použité, zkřížené, ztracené a skutečně použité sektory.

Někde uprostřed pak seznam souborů,



do kterého se vejde 14 souborů a pak musíte stránkovat.

Nad i pod seznamem se zobrazují informace o souboru, na který ukazuje kurzor – nahoře je délka souboru a startovní adresa, dole pak méně důležité informace: body flag pro bezhlavičkové typy souborů z MASTERcopy (u souborů typu Basic pak délka programu bez proměnných) a atributy. Poslední dva řádky pak ukazují počet označených souborů, počet souborů na disketě, velikost označených souborů, počet obsazených položek v adresáři diskety a počet souborů smazaných ScanDiskem.

Přiznám se bez mučení, že mně takový výpis připadá až příliš pestrý a přehnaný, co se informací týká. Příkladně bych se k funkci, kdy se podrobné detaily o disketě objeví až po stisku definované klávesy (třeba I – info).

Například řádka s formátem a sektory by být nemusela, obě řádky s informacemi o souboru by mohly být dole a informace o tom, co ScanDisk našel, by mohla být v samostatném okně.

Celý výpis a vůbec celý MFC tiskne na obrazovku pomocí RST 16, takže všechna písmena jsou 8x8 (fujky... 42 znaků na řádek rulez - pozn. +GAMA), ale zato v plných Speccy barvách.

Takže už víme, jak MFC vypadá v základním zobrazení (tím chci říct při zobrazení souborů, ne že by měl mít nějaké rozšířené zobrazení), ale ještě netušíme, co MFC vlastně dokáže a v čem je jeho síla.

Od souborového manažeru, jak jsme si řekli, očekáváme, aby uměl základní práci se soubory – kopírování, přejmenování, smazání.

MFC umí samozřejmě více.

Kromě absolutně nezbytných funkcí, jako je načtení nové diskety, označování souborů po jednom nebo naráz, hravě zvládá následující práce se soubory:

- přesunutí souboru na jinou pozici v adresáři
- označování souborů dle zadané souborové masky
- přejmenování diskety

užitečná věc, pokud netušíte, proč by někdo chtěl měnit pozici souboru na disketě, tak vězte, že třeba já mám radši soubory seřazené tak, aby šly za sebou jako na magnetofonové pásce. Ulehčuje to práci s označováním souborů při kopírování.

Označování souborů podle masky je taky malinko jiné, věnujte více pozornosti této kapitole v manuálu. Pro nemajitele MFC řeknu jen, že maska se trochu liší od klasické hvězdičkové konvence a hvězdička není třeba zadávat.

Při přejmenování nebo změně přípony můžete zadat i netisknutelné znaky. Může se hodit například při přejmenování zdrojových souborů pro SQ-Tracker, který jako první písmeno má právě takový netisknutelný znak. Netisknutelné znaky můžete samozřejmě zadat při jakémkoliv vstupu textových hodnot z klávesnice.

Mimochodem právě přejmenování je velice dobře udělané, nemusíte zadávat celý text, buď stačí upravit ten původní, nebo prostě jen začít psát nový.

Manuál píše, že je možné spustit soubory typu P (Program) nebo B (Bytes), ale dají se pochopitelně spustit i SNAPSHOTy. U typu Bytes se MFC dotáže na nastavení RAMTOPu, adresy, kam se má soubor nahrát, a pak adresy, odkud se má spustit.

Adresa je vždy předvyplněná a vypočítává se z hlavičky souboru. Pokud u spouštěcí adresy odešlete prázdný řádek nebo nulu, program se do paměti jen nahraje.

Z PC disket umí MFC spouštět soubory typu .SNA (pouze 48k) a .MFC. Formát .MFC je nový, definoval ho sám MTs a pomocí konverzní utility pro Windows si můžete převést do formátu

- změnu přípony souboru
- změnu startovní adresy (pouze MDOS)
- změnu atributů souboru
- spuštění souboru přímo z diskety (harddisku, CD, CF... - pozn. +GAMA).

O některých funkcích si řekneme něco více.

Tak například přesunutí na jinou pozici v adresáři je velice

.MFC i soubory .SNA a .Z80.

(MFC Windows konvertory lze díky projektu Wine (<http://www.003.portalis.it/115> nebo www.winehq.org) spouštět na Linuxu! Sám jsem MFC snap Converter a MFC d_0 Converter na Mandrake 9.1 s Wine v21.01.2004 testoval, oba fungují bezchybně. Instalační .exe vytvořený ve WinRAR_3.20 Wine pustil, instalace sice nebyla ukončena tak, jak měla (místo spuštění lrun.exe to vypsallo zcela nesmyslnou žádost o přepsání souboru, nezbylo než storovat), nicméně soubory rozbaleny byly a to zcela v pořádku. Balíček Wine je však každým dnem lepší a lepší, proto věřím, že Vám už proběhne vše hladce. - pozn. MTs).

A dostáváme se k tomu nejlepšímu, ke kopírování a společně s ním k export/import systému.

Kopírování je možné z libovolného typu diskety opět na libovolný typ. Tím myslím, že si můžete kopírovat soubory na disketách mezi PC a ZX a v libovolných kombinacích.

Chápete jeho sílu?

Vezmu třeba kopii spektrácké obrazovky, pomocí MFC zkopíruji na PC disketu a změním příponu na SCR a je to.

Obráceně samozřejmě taky. Pokud znáte program BMP2SCR od LCD, tak si bitmapový obrázek pomocí něj převedete na SCR Spektra a pak opět za pomoci MFC zkopírujete na ZX disketu a máte SCREEN\$ jak vyšitý a navíc si ho můžete pomocí utility v MFC rovnou nechat zobrazit.

Export a import přichází opět s novým typem souboru. Normálně používáme TAPky, EI Manager používá soubory .000 a MFC má .D_0. Nový formát byl podle MTs nutný z důvodu uchování všech diskových informací o souboru a možnosti exportovat i delší soubory než 65535 bytes. Pomocí dalšího konverzního programu od MTs pro Windows můžete ale z .D_0 vytvořit .000.

Na kopírování a exportu/importu je také neobvyklé to, že kopírované soubory MFC "očisťuje" o nepodstatné byty v posledním sektoru. Co to vlastně znamená? Dám příklad. Uložte si jednořádkový program v Basicu, ten zabírá 48 bytes, ale na disketu se uloží celých 512 paměti a tím vlastně i baj-





ty, které nechcete. MFC je při kopírování prostě odstraní (resp. vynuluje) pro lepší orientaci při editaci diskety.

Další neobvyklou věcí u kopírování (u exportu/importu už ne) je ta vlastnost, že pokud se při čtení narazí na vadný sektor (chyba "Sector not found"), MFC jej automaticky zkusí znovu a pokud se ani napodruhé nepovede, nahradí jej samými invertovanými otázkami.

Navíc přiřadí souboru nulovou délku, takže jej ScanDisk pak označí jako vadný.

Tato vlastnost je vhodná, pokud potřebujete z vadné diskety dostat alespoň nějaká data.

V setupu lze nastavit i chování pro "CRC error" příp. celé toto "ignorování" chyb vypnout a donutit tak MFC chybové hlášení vypisovat a kopírování předčasně ukončit.

Důležitá je taky práce s celou disketou, jako je formátování, rychlé formátování, přejmenování diskety.

Naformátovat můžete disketu i na nestandardní formát, v takovém případě bude vytvořena disketa s formátem MDOS.

Pokud naformátujete na formát 80*9*2, máte na konci formátování možnost si z takové diskety nechat vyrobit PC disketu.

Diskety formátované i na disketové jednotce s MDOS1 budou bez problémů čitelné i na PC (*normálně totiž nejsou*).

Rychlé formátování provede výmaz systémových sektorů diskety a v případě že jsou ve FAT označeny vadné sektory, projede pro jistotu všechny sektory znovu, jestli nějaké vadné nepříbily.

Přejmenování diskety zcela jistě použijete po naformátování, protože jinak budete mít disketu zcela bezejmennou.

Funkcí přímo specifických pro MFC je několik, podívejme se na ně pod lupou.

První z nich je ruční zápis systémových sektorů diskety BOOT, FAT, DIR. Jedna ze základních myšlenek MFC je šetrnost k disketám, takže se snaží zapisovat a číst co nejméně. V praxi to vypadá asi takto: přejmenujete soubor (nebo několik), ale disketa se vůbec neroztočí. Zatím se nic nezapsalo, pokud byste si Speccy resetovali, soubory

Další speccyickou funkcí je deaktivace ochrany poškozené diskety. To už jsem naskousl v úvodu, takže jen stručně. Pokud ScanDisk najde nějakou chybu na disketě, zablokuje zápis na disketu. Pokud si ale myslíte, že je zápisu zapotřebí, odblokujte ochranu.

Jedna ze zvláštních funkcí je xor souboru. Najedete kurzorem na soubor, pustíte xor a dozvíte se CRC souboru. Můžete si takto ověřovat, jestli jsou soubory shodné, nebo jestli se zkopírovaly správně. Nevýhodou je, že CRC se spočítá jen pro jeden soubor, takže pokud si budete chtít takto zkontrolovat souborů několik, musíte po jednom.

Poslední a nejlepší funkcí jsou utility a šetříče (screensaver). To dělá z MFC něco víc než jen ubohý souborový manažer.

Pomocí utilit může zvládnout více funkcí, než jaké mu dal do vínku autor. V této době existují pro MFC čtyři utility: **edit** (editace a zobrazení souboru, příp. diskety), **tap** (převod z/do TAP souboru), **load** (převod z kasetofonu na disk) a **sftp** (speccyFTP přenos mezi dvěma ZXS kabelem na 8255).

Spořičů televizí je taky několik, způsob zavedení do MFC je stejný.

Utility mají samozřejmě i nějaká omezení a pravidla. Utilita musí mít v názvu dva konkrétní znaky, je možné ji nahrát pouze ze ZX diskety, má omezenou velikost, nemožňuje pracovat s více disketami nebo mechanikami, zápis do souboru je možný jen na MDOS disketě.

Naproti tomu je velká výhoda v tom, že o rutiny čtení a zápisu do souboru se stará MFC, utilita se věnuje pouze obsahu souboru.

Povídání o tom, jak si napsat vlastní utilitu, a všechny popis dostupných služeb a podprogramů, je součástí manuálů.

Tím jsme vyčerpali všechny možnosti programu MFC, pokud jste jej ještě nezkusili, rozhodně si jej stáhněte a vyzkoušejte.

(MFC se určitě bude hodit nejenom majitelům D40/80, ale i majitelům DivIDE, kteří mohou využívat verzi MDOSu 3 nezávislou na přítomnosti hardware slovenské disketové jednotky).

ve skutečnosti přejmenované nebudou.

Sektory adresáře diskety se zapíšou až po úspěšném dokončení nějaké operace, která si vynutila zápis na onu disketu (tedy např. po kopírování nebo po návratu z utility), nebo na rozkaz uživatele klávesovou zkratkou "X". Pokud si na tuto skutečnost nemůžete přivyknout, MFC má v nastavení volbu "changes - save immediately".

Silvestrovská pecka a Mluvící balík

Bulldog



Je tu návod na další textovku, kterou jsem dojel. Program reaguje na první 2 nebo 3 písmena, napsal jsem jen to nejnětější, co musíte vytukat na klávesnici.

V, VE NU, Z, VEN, V, V, V, V, DO, VE MA, VEN, J, DO, V, V, POU NU, VE SI, Z, Z, VEN, J, DO, J, V, D, D, VE SE, N, N, N, V, S, VE PRO, J, Z, D, Z, S, VEN, S, S, Z, POU SE, POU PRO, POU SI, VE KA, POU NU, PO NU, PO KA, PO SI, VE ZU, V, J, J, DO, J, V, N, S, POU ZU, VE LA, J, D, Z, S, VEN, S, S, Z, Z, Z, S, POU MA, POU LA, S, DO, VE PR, VEN, J, J, V, V, V, J, DO, V, V, D, POU PR.

Hra není obtížná ani rozsáhlá. Liší se od ostatních tím, že jste omezeni časem, ale ta hodina, kterou na splnění úkolu máte, je moc. Jde to stihnout během 5 minut.



Další textovka je **MLUVÍČÍ BALÍK**, hra podle stejnojmenné knihy G. Durella.

Program testuje 2, 3, někdy 4 písmena.

J, J, PR HR, VE FO, PR HR, VE KL, PR HR, VE SR, S, Z, VE HRN, V, V, P HRN, PR STR, VYLE STO, U UT, Z, S, P FO, J, ZAT TRU, P TRU, V, J, VE PL, S, Z, PR ROP, VE DY, Z, PR MAR, VE BR, VYCI BR, P UT, S, P BR, Z, P PL, P PL, P DY, J, ROZB BU, P KL, OD TRO, VE TRO, P SR, S, V, J, V, P DY, V, VE HRN, V, V, PR CA, OT KOH, N HRN, J, PR ZN, OTE DV, P KLI, Z, VE OL, OTE KN, V, S, Z, PR TR, U KV, Z, Z, J, PR VOZ, NAO KOL, P OL, S, Z, S, VE LO, J, V, S, V, P KV, P LO, NAS, S, P TRO, P HRN, PI PI, VYS, S, US POU, J, NAS, VYS, Z, J, Z, S, P ROU.

Ani tato textovka mi nezabrala mnoho času. Líbí se mi na ní, že každé místo je doprovázeno malým obrázkem v levém horním rohu. Také plocha, na kterou se vypisuje text je hezky udělaná. Nahoře uprostřed je směrová růžice, která ukazuje, kterým směrem se můžete vydat. Vpravo nahoře je nápis MLUVÍČÍ BALÍK.

Ale co budu žvatlat, radši si hru nahrajte a vyzkoušejte si všechno sami.



Abe's mission

Tornado

Ano. Rusové opět předělali jednu slavnou PSX pařbu i na naše osmibitové miláčky!

Jelikož se mi PC verze Abeho moc líbila, hlavně komunikace mezi hlavním hrdinou, ostatními stvořeními nazývanými Muddokani, a jejich strážci, nedalo mi to a specy verzi jsem prostě musel testnout.

K dispozici jsem měl speciální verzi pro ZX Spectrum 128 s českým a anglickým překladem a podporou D80, o kterou se postarali z00m, Wixet a Sweet. Dotyčné touto cestou chválím za jejich obětavou práci (*i když několik nepřeložených nebo spíše nepřeložitelných textů jsem tam našel*)

Pro ty, kteří Abeho ještě vůbec neznají, napíšu krátký příběh, který vás seznámí o co že to vlastně ve hře jde. Takže začneme:

Jednoho krásného dne Abe, po těžké otrockářské práci v továrně na zpracování masa, zahlédne pootevřené dveře do kanceláře a uvnitř vidí své šéfy, jak debatují o kritickém nedostatku zvířat. Pokles produkce a zisku je pro šéfy nepřijatelný, krize musí být zažehnána. Postarat se o to má nový výrobek - maso s Muddokana, rasy Abeho.

Abe se celý vyděšený dává na útěk. Jenže hluk ho odhalí a v továrně je okamžitě vyhlášen poplach. Abeho úloha běžného zaměstnance (uklízeče) je rázem pryč, nyní je terčem číslo jedna...

A nato se ocitáte ve hře. Vaším úkolem je dostat se z továrny (ne v konzervě, živý) a vzít s sebou i co nejvíce svých přátel.

Abe je velice obratný a umí mnoho věcí, které mu pomáhají při plnění jeho cíle.

Úžasná věc je telepatické ovládání jednoho z nepřátel. Toto vám může velice pomoci v některých zapeklých situacích.

Kromě pohybu všemi základními směry (a telepatie) dokáže Abe také skákat do dálky, slézat anebo vylézat po různých patrech továrny, házet granáty a komunikovat.

Po celou dobu hraní se v některých míst-



nostech ukazuje nápověda a je možno využít i terminály. Ty Vám poradí, jak využít své schopnosti a dostat se tak zase o kousek dál.

Graficky se hra v celku vyvedla, pozadí továrny ale mohlo být bohatší. Bohužel paměť ZX je malá a tak nezbývá než věřit, že autoři dělali, co se dalo.

Abeho jsem hrál na PC v emulátoru Real Spectrum. Občas se mi stalo, že rychlým přechodem do další obrazovky se mi objevi-

lo několik error pruhů. Po vrácení a po klidném přechodu bylo vše v pořádku. Těžko říct, zda je chyba v emulátoru (*velice nepravděpodobné - pozn. MTs*) nebo ve hře...

Hudba se ve hře ozve jen v menu a to jen pro AY, při samotném hraní je ticho po pěšině. Takže tady bych dal autorům velký černý puntík. Že by zase došla paměť?

Hratelnost je perfektní a nemám jí co vytknout.

Hra se mi ale zdá velice krátká a vypadá to, že všechny Muddokany ani zachránit nelze. Alespoň mně se to tedy nepovedlo a to jsem vyzkoušel snad všechno.

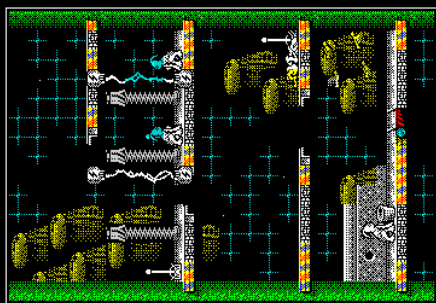
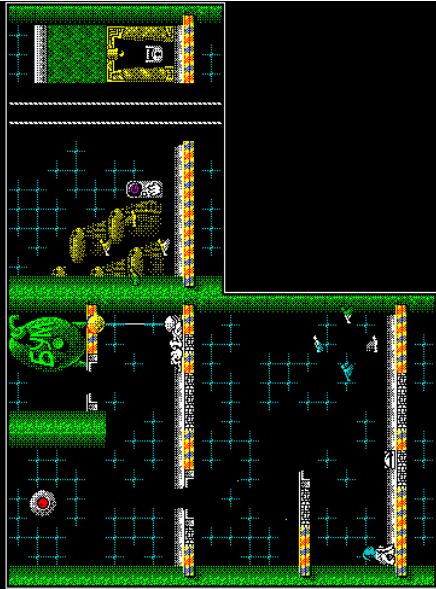
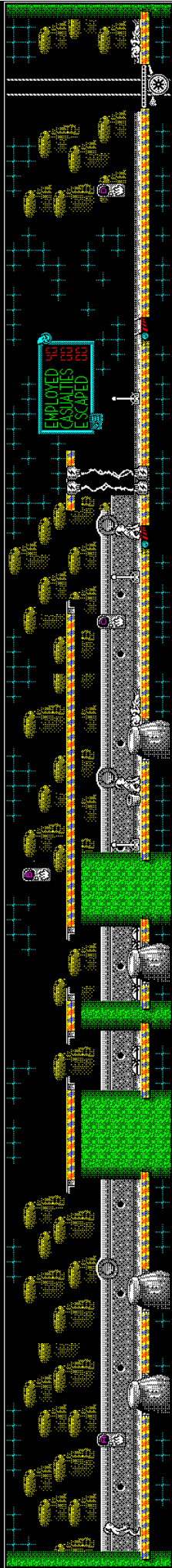
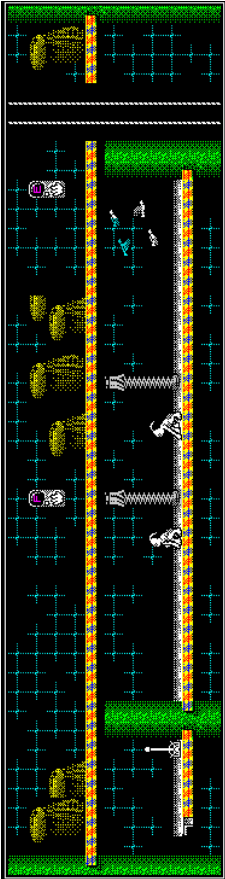
Jestli se +Gama překoná, měli byste někde uvnitř ZXM najít i mapu, kterou spáchal Jim Langmead (velké díky patří taky Pavrovi, protože nás na ni upozornil).

Abe je už defaultně nesmrtelný a pozice se dají v některých místech ukládat. Což je bezesporu velké plus, SAVE pozice se u specy her moc nevidí.

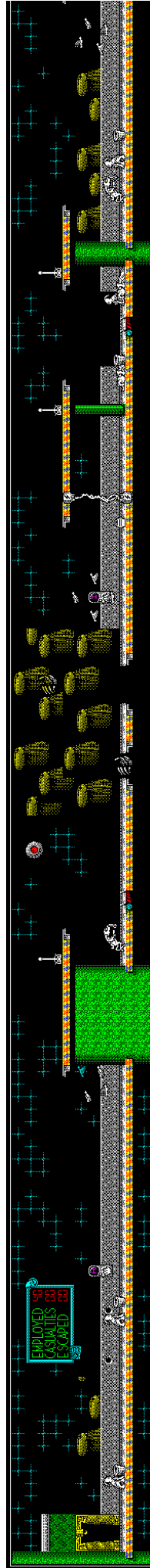
Co dodat? Zpracování hry je velice dobré. Abe je především logická chodička a zuřiví FIREmani si na své rozhodně nepřijdou. Možek je občas opravdu zapotřebí.

Bratři Flaškové (*kteří mimochodem vůbec neznám*) by tuhle hru jistě hráli desítky, ne-li stovky hodin...

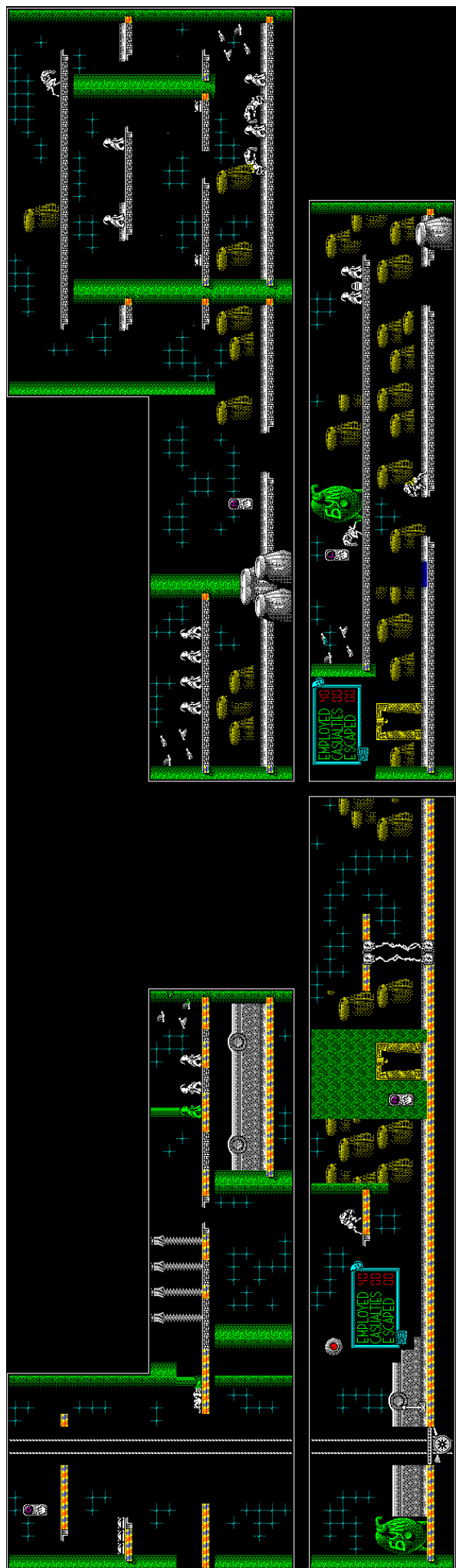
Let's play!



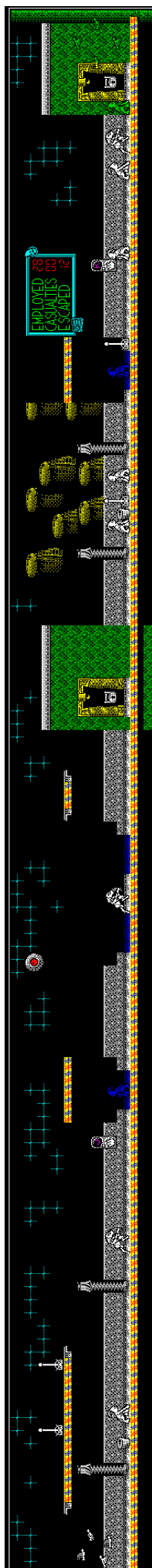
1A



2A



1B



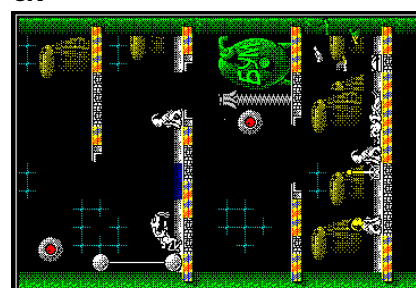
2B

ABE'S ESCAPE MISSION

Protože je mapa Abeho ne- skutečně velká a pro tisk ne- vhodná (drobné bílé details na pozadí černi), zkusili jsme za- chovat její barevnost a rozdělit ji do několika částí, které na sebe navazují zhruba způso- bem, který vidíte na malém plánu.

1A	1B
2A	2B
3A	

3A



BIT

Slovenský časopis Bit, vydávaný firmou Ultrasoft, byl sice zaměřen převážně na hry a nebyl orientován čistě pro Spectrum, majitelé Specter a Didaktiků však byli touto cestou informováni o tehdy velmi kvalitní slovenské softwarové produkci zajišťované touto firmou a jistě tvořili značnou část čtenářů. Zde se můžete přesvědčit o tom, co všechno bylo možno si za ty roky jeho vycházení přečíst.

Legenda:

R - recenze
N - návod
M - mapa (BM - barevá mapa)
P - preview
+ - plakát
(Nezaručuji, že se podařilo podchytit všechna čísla Bitu)

10/91

SAM Coupé
D40
R 3D pool
R Dragon breed
+R Star dragon
M N Terramex
R Atrtist 2
R Head over heels
M N Great escape
M F19
R Lotus esprit turbo challenge
R Adidas tie break
R The cycles
R Rychlé šípy 1 - záhada hlavolamu
R Multi player soccer manager
R Stunt car racer
rozhovor František Fuka (1)
P Thunder jaws
Basic - program Lupa
assembler Z80

11/91

Amiga 3000
+D
R Madmix 2
R Atomix
+R North & south
R MRS 08E
R Green beret
R Toyota celica GT rally
N Sir Fred

R Helter skelter
R Dick tracy
R Rychlé šípy 2 - Stínadla se bouří
R Loopz
R Sim city
rozhovor František Fuka (2)
P Alien storm
Basic - Picasso
assembler - registry

12/91

R Shadow dancer
R Extreme
R Super Monaco grand prix
+R Prince of Persia
R Cesta bojovníka
R Worm in paradise
R (N) Kayleth
R ZX-7
R Academy
M N Mercenary
R Golden axe
R Welltris
R Eswat
R Gengiskhan
R Magic Johnson's basketball
rozhovor Ján Deák
P Darkman
Basic - užitečné podprogramy
assembler - flagy

1/92

Sinclair story I
R Dizzy III
R R-Type
R Turrican
R Psi chess
R Datalog 2 Turbo
R Tomahawk
M N Spy vs spy 1
M N Spy vs spy 2
R Hexagonia

R Dan Dare III
R Beverly Hills cop
R Cabal
R Battleships
rozhovor Patrik Rak (1)
P Indiana Jones IV
Basic - užitečné podprogramy
assembler - LD

2/92

Sinclair story II
R Night shift
R Midnight resistance
R Hostages
R (N) Chaos
R Mantrik
R Thanatos
M R Saboteur
M R Saboteur 2
R Jet story
R Fighter bomber
R Mutant fortress
R Continental circus
R SWIV
rozhovor Patrik Rak (2)
Basic - efekty v borderu
assembler - LD

3/92

Amiga 500+
cyberspace a virtual reality
A500 HD+ series II
R (N) Lords of chaos
R Too bin
R Puzznic
N Lords of midnight
R Hisoft Basic
R Enduro racer
M R Draconus
M R Starquake
R Strider 2
R Hero quest

R Terminator 2
 R Lone wolf
 R Beach volley
 rozhovor Miroslav Fidler (1)
 P Cisco heat
 Basic - minové pole
 assembler - INC,DEC, LDI, LDIR

4/92

Atari TT 030
 crackování na C64
 Action replay C64
 R Motorcycle 500
 R Turrican 2
 N Copy copy
 R Highway encounter
 M N Batman
 M N Little Puff in Dragonland
 R Hammerboy
 R CJ elephants antics
 R CJ in the USA
 R Sokoban
 R Banger racer
 R Poli Diaz boxeo
 R Lemmings
 +R Stormlord
 P Super space invaders
 rozhovor Miroslav Fidler (2)
 Basic - Life
 assembler - aritmetické a logické opera-

ce

5/92

přehled počítačů
 R Auto cras
 R R.B.I. 2
 R Smash TV
 R ProDos Basic
 R Hysteria
 M N Tintin on the moon
 M N Jet story
 R Logic
 R Los Angeles drugs bust
 R Jackson city
 R Peking
 rozhovor Ondřej Mihula (1)
 P Double dragon III
 Basic - jednoruký bandita
 assembler - skoky, podprogramy, návraty

6/92

Commodore CDTV
 Trojan light pen
 R Dizzy down the rapids
 R Gunboat
 R Dizzy V
 R Lop ears
 R N Art studio (1)
 M (N) Feud
 M N Underwulde
 R Supercars
 R Fireman Sam
 R Merces
 R GP formula 1 simulator
 rozhovor Ondřej Mihula (2)
 +P Prvá akcia
 Basic - Pythagorova věta
 assembler - volání podprogramů

7/92

Amiga 3000 T, UX
 R Seymour at the movie
 N M Universal hero
 R WWF Wrestlemania
 R Turtles 1
 R Exolon
 R Tetris 2
 R F1-Tornado simulator
 R Alien storm
 R N Art studio (2)
 rozhovor Rady Maruša (1)
 P Phantom F4
 Basic - kouzla s písmem
 assembler - zásobník

8/92

Amiga 600
 N M G.I. Hero
 N M Ninja
 R Pipemania
 R Turtles 2
 R F.I.R.E.
 R Pit fighter
 R Fist
 R Super Seymour saves the planet
 +R James Bond Octopussy
 R N Art studio (3)
 R Road runner
 P Astroball
 rozhovor Rady Maruša (2)
 Basic - RUN pro D40
 assembler - posuny

9/92

N M Nosferatu
 R Sky high stuntman
 R Miami chase
 R Bubble Dizzy
 R Moon alert
 +R (N) Crux 92
 R, N Mega Basic (1)
 R Street gang
 P Sherwood
 rozhovor Milan Blažiček (1)
 Basic - Picasso 2
 assembler - rotace

10/92

nová Atari
 Melodik
 +R Phantom F4 1, 2
 R Space crusade
 R Pixy 2
 R Popeye 2
 R Batman
 R Boxing manager 2
 N Mega Basic (2)
 R Delta charge
 P Conman the barbarer
 rozhovor Milan Blažiček (2)
 Basic - Zed'
 assembler - rotace

11/92

Didaktik Kompakt
 N M Rolands rat race
 R Hobgoblin
 R Demon slayer
 R Heartland
 R Ninja rabbit
 N Mega Basic (3)
 P Turbo the tortoise
 +R Notorik
 rozhovor Marek Trefný (1)
 Basic - UDG, krychle
 assembler - operace s bity

12/92

Gameboy
 emulátory PC pro Amigu
 N M Castle master
 R Gazza 2 soccer
 R Heavy metal
 R Chevy chase
 R Renegade
 R Interchange
 R Hotrod
 P Komando 2
 +R Sherwood
 rozhovor Marek Trefný (2)
 Basic - prostorové zobrazení funkcí
 assembler - přesun bloků

1/93

monitory pro Amigu
 N Night raider (1)
 N Neverending story
 M N Rambo III
 R Days of thunder
 R Trapdoor
 R Puzzle
 R Earth shaker
 R Championship run
 P Pick out 2
 rozhovor Slavo Lábský (1)
 Basic - FAT Directory list, Screenshot
 assembler - porovnávání a blokové porov-

návání

2/93

N Night raider (2)
 N M Dizzy VI
 N Archon 2
 N M Robin of the wood
 +R Komando 2
 R Jonny's quest
 N Amaurote
 R Potsworth
 R Buffalo Bill's wild west show
 R Donald Duck's alphabet chase
 P Šach mat
 rozhovor Slavo Lábský (2)
 Basic - hudba z Basicu
 assembler - AND, OR, XOR

3/93

N Night raider (3)
 R Laser squad
 M N Potsworth (1)
 N Dizzy II

R Ivan Ironman Steward super offroad
 R Mountain bike simulator
 N Rychlé šípy 2 - Stínadla se bouří
 N Seymour at the movie
 R Kamikaze
 R Chase H.Q.
 R Kick box vigilante
 R DTP Machine
 R (N) Robocop III
 +P Pedro na ostrove pirátov
 rozhovor Slavo Lábský (2)
 assembler - výměna registrů, akumulátor

4/93

Amiga 1200
 N M Potsworth (2)
 N Cauldron
 N Zorro
 R Grell & Falla
 R Snare
 R Murray mouse supercop
 N Captain Blood (1)
 R 3D Stock cars 2
 R Tetroid
 +R Kliatba noci
 P Pedro v strašidelnom zámku
 rozhovor Slavo Lábský (3)
 Basic - míchání barev
 assembler - interfacing

5/93

N F-16
 N Indiana Jones I, II, III
 N Dizzy IV
 R Match of the day
 R Captain Planet
 R (N) Reckless Rufus
 M Night shade
 M Alien 8
 M Knight lore
 R Operation Wolf
 N Captain Blood (2)
 R International tennis
 R Turbo the tortoise
 assembler - interfacing

6/93

R Bomb Ed
 R Steel eagle
 M N Sleepwalker
 R Beach head
 R Pedro v strašidelnom zámku
 R Badlands
 R 3D Construction kit
 R Last ninja 2, Indiana Jones III, Spy who
 loved me, Strider
 P International kickboxing
 rozhovor Rudolf Priečinský
 Basic - zoom editor
 assembler - přerušení

7/93

Amiga 4000/30
 Dataputer
 tipy a triky pro C64
 R Cybex shell 1.0

N Wonderboy in Monsterland
 R Nigel Mansel world championship
 N Space gun
 R Live and let die
 M James Bond Octopussy
 M Super Robin Hood
 R Uridium
 R Licence to kill
 R Moontorc
 R Klax, EEPRM, APB, Cyberball, Vindica-
 tors

8/93

Sharp MZ 800
 R Street fighter 2
 R Tarzan goes ape
 M CJ's elephant antics (1)
 M CJ in the USA
 R Knight lore
 R Bloodwych
 R Sountracker Ultrasoft
 R Rick Dangerous, Soldier of light, Total
 eclipse
 Basic - Pishkworx
 assembler - úvod (0)

9/93

starší čísla Bitu (1)
 N Dizzy V (1)
 N Dizzy VII
 M CJ's elephant antics (2)
 M Bukapao
 M Viaje al centro della tiera
 R Navy moves
 R Dragon spirit
 R Tuition
 assembler - přesuny (1)

10/93

Amiga CD 32
 N Dizzy V (2)
 M Dizzy VII
 N Gunship
 R Action fighter
 R Rick Dangerous 2
 Basic - den v týdnu, text do kružnice
 assembler - porovnání řetězců (2)

11/93

starší čísla Bitu (2)
 ZXS emulátor
 M Titanic 1
 M Titanic 2
 R Krakout
 N F-19 Stealth fighter
 R Ghostbusters II
 P Pedro v krajině pyramid
 Basic - Space invaders
 assembler - porovnání řetězců (3)

12/93

N Sherwood
 N Pomsta šíleného Ataristy
 BM N Myth
 R Captain Dynamo

R Video pool
 R Baby Mantrik
 assembler - násobení (4)

1/94

N Defender of the crown
 N Slightly magic
 N Academy (1)
 M Pedro na ostrove pirátov
 P Quadrax
 +R Pedro v strašidelnou zámku
 Basic - převod disketa<->kazeta
 assembler - dělení (5)

2/94

N Academy (2)
 BM Dynamite Dan
 M Dan Dare II
 P Towdie
 +R Pedro v krajině pyramid
 Basic - Močál
 assembler - odmocnina, dec2bin (6)

3/94

M N Kliatba noci
 N Shadow of the beast
 BM Dizzy V
 R Gunstar, Merlin, BMX simulator, Eleva-
 tor action
 Basic - Logic
 assembler - hex2bin (7)

4/94

N Pedro na ostrově pirátů
 N Hero quest
 Basic - třetí parametr DRAW, zvětšené
 písmo
 assembler - výpis čísel (8)

5/94

M N Bloodwych
 +R Quadrax
 Basic - animace v Basicu
 assembler - výpis hexa čísel (9)

6/94

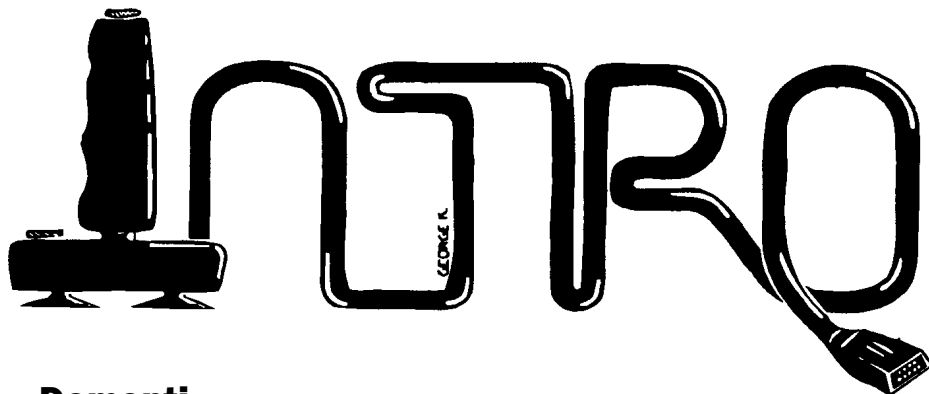
Volkov commander 4.038
 M Yo! Joe!
 +R Towdie
 Basic - hledací pes Rexo
 assembler - komprimace (10)

7/94

R Mortal kombat
 BM Laser squad
 M N Wild west Seymour
 Basic - grafy funkcí
 assembler - dekomprimace (11)

Excalibur 41/95

Otevřený dopis šéfredaktora časopisu
 Bit svým čtenářům



Dementi

Že **není YS jako YS** se mohli přesvědčit Softhouse a Pavero. Ačkoliv bylo o časopise Your Spectrum, který vydávala v letech 1997-2000 Zlínská firma 8 Bit Company, v minulosti x krát informováno (ve všech vycházejících specy časopisech, tedy ZXM a XM), oba bezesporu "praví spektristé" se nyní, kdy jsou čísla YS postupně převáděna do PDF, strašně podivili, že je to česky, a že to s anglickým Your Sinclair ani anglickým Your Spectrum (které taky existovalo) vlastně vůbec nesouvisí.

Naprostu unikátní technologii převodu zdrojového textu z Promethea do .TXT souboru vyvinul Softhouse. Svě know how prozradil nedávno i na Pandoře:

V Prometheovi stránkujeme (CS+3) a každou tokovouto obrazovku s kouskem zdrojového textu si uložíme jako screen\$, nakonec obrázky pomocí cyklu FOR to NEXT a funkce SCREEN\$ postupně načteme. No není to báječné?

My ovšem z časových důvodů raději zůstaneme u MDOsového příkazu OPEN #3, "vystup.B" a Prometheovského PRINTu...

Proč je programování v assembleru

přítomnost: Právě prožíváme první dny roku 2004. To mimochodem znamená, že žijeme v 21. století. Přesto mají pořád někteří lidé pocit, že programovat v assembleru je v něčem dobré či pozitivní. Toto je absolutní omyl a já zkusím popsat proč.

Jediný důvod, který drží tyto lidi u tohoto přesvědčení je mýtus, že programy napsané v assembleru jsou rychlé (no, možná ještě vrozený masochismus). Veškeré ostatní aspekty programování (chybovost, pomalý vývoj, špatná udržovatelnost, neskutečně pomalý tj. drahý vývoj, nepřenositelnost) v assembleru doufám všichni uznávají jako negativní.

Programy napsané v assembleru nejsou rychlé. Klasickým argumentem pro použití assembleru je rychlost provádění programu. Panuje totiž všeobecné přesvědčení, že při psaní v assembleru je možné "napasovat" program na konkrétní procesor a tím zabezpečit jeho rychlé vykonání. Toto je mýtus. Dnešní procesory (dnešní znamená ty vyrobené za života myslím velké většiny z nás) totiž při vykonávání posloupnosti instrukcí

```
movl %eax,%ebx
xorl %ebx,%ecx
jnz @loop
```

(toto jsem si náhodně vymyslel, nic v tom nehledejte) vůbec nevykoná co by (neznalý) člověk myslel. No a vzhledem k tristní vzdělanosti je těch neznalých většina. No, přejdeme k věci - proč se vykoná něco jiného. V moderních (< 10 starých řekněme) procesorech (mluvím o CISCU a to např. i

proto, že jinde než na intelu jsem přímé programování v assembleru neviděl) funguje vykonávání instrukcí takto: instrukce (její část určující typ) je vlastně jen adresou v paměti mikrokódu. Takže vlastně když se vykonává "movl %eax,%ebx" tak se vykoná několik tzv. mikroinstrukcí (dokonce existují architektury, kde se mikroinstrukce skládají z tzv. nanoinstrukcí). To by ještě nevadilo, ale pro zrychlení výkonu procesorů se v nich instrukce zpracovávají (neznám přesný termín) zřetězeně (instructions are chopped into small pieces which are fed to pipeline(s) one by one allowing several instructions to be executed in parallel). Tedy vykonání každé mikroinstrukce je rozloženo na několik fází s tím, že když se mikroinstrukce A nachází ve fázi 2, může vstoupit mikroinstrukce B do fáze 1 (ie. začít se vykonávat). To by pořád ještě nevadilo (myslím), co ale vadí je takzvané vykonávání mimo pořadí. Je logické, že aby se mikroinstrukce (resp. instrukce) mohly vykonávat "paralelně" nesmí na sobě záviset - proto se přeskupují. To je také důvodem proč se nevykoná (resp. ne nutně) to co jste napsaly. Proto jsou snahy o zrychlování programů jejich přepisem do assembleru zbytečné. Kód bude přinejlepším stejně rychlý jako ten přeložený z vyššího jazyka. Situace je ale o něco horší, dokonce tak, že si dovoluji napsat: Programy napsané v assembleru jsou pomalé.

I za předpokladu, že člověk při psaní v assembleru použije ty samé algoritmy jako by použil ve vyšším jazyce (dost nereálné) jsou programy psané v assembleru pomalé. Jde totiž o optimalizaci. V principu existují tři fáze optimalizace kterými program při překladu projde. Optimalizace na úrovni vyššího jazyka (ta při psaní v assembleru není). Optimalizace na úrovni assembleru a konečně optimalizace v procesoru (přeskupování instrukcí, prefetch apod.). Optimalizaci v procesoru asi steží ovlivníme a optimalizace na úrovni vyššího jazyka není při psaní přímo v assembleru k dispozici (řekněme, že dokážete provést jednoduché optimalizace - dead code, constant propagation apod. ručně) a tak se podívejme na optimalizaci instrukcí. Normální člověk je schopen myslet záraz na cca 7 věcí (ta reklama tvrdící cosi o mužské neschopnosti tohoto je pičovina). Pokud chceme ručně optimalizovat kód v assembleru musíme si uvědomit následující faktory ovlivňující běh programu:

1) cache fit

je třeba zarovnávat na velikost cache line apod., měl by být zachován princip lokality

2) optimální řazení instrukcí

instrukce by měly používat jiné části procesoru, instrukce by měly zpracovávat na sobě nezávislá data, použití "špinavých triků" (NOP na vhodném místě apod.)

3) optimální výběr instrukcí

CISCové procesory většinou umožňují pro jednu činnost zvolit více možností vykonání - je nutné zvolit optimální (kdo neviděl xorl %eax,%eax místo movl \$0,%eax), instrukce by měla neměla ovlivňovat nic jiného než své operandy (vysvětlím dále)

4) samotný algoritmus (ideálně by měl zůstat funkční), možná ještě něco ale i tak je vidět, že je to na hranici možného. Jen stěží to tedy dokážete lépe než program. Navíc jsou optimalizační kroky, které na úrovni assembleru nemůžete použít. Zde vysvětlím bod 3b. V assembleru není jasná struktura programu, co je vlastně smyslem kódu (to je zjevnější s vyšší úrovní abstrakce - proto jsou vyšší jazyky lepší). V podstatě veškerá data (v registrech o paměti ani nemluvě) jsou tedy brána jako nestálá. Toto slůvko jsem zvolil záměrně. Anglicky se by se to zapsalo "volatile" - nepřipomíná vám to něco? ;) (v jazyku C se takto označuje třída proměnných o jejich obsahu se nesmí nic předpokládat - tj. neprovádí se na nich optimalizace). Všechno to totiž může být modifikováno v podstatě kdykoliv a čímkoliv. Další věcí je "nezjevnost" některých věcí na úrovni assembleru a proto se neoptimalizují.

```
int foo(unsigned short param)
{
    signed short var = param;
    if (var < 0) {
        ...
    } else {
        ...
    }
    return (var & 0xff);
}
```

assembler nezná pojem typu a tak se v něm vykoná podmínka na zápornost proměnné var (toto je zjevný příklad berte to principiálně) a také vykonání logického AND na konci (je zbytečné protože proměnná zůstane nezměněna). A to ještě předpokládám, že existují optimalizující překladače assembleru (v životě jsem žádný neviděl).

Jak vidno programování v assembleru je hloupost. Je škodlivé z mnoha důvodů (špatné programátorské návyky, špatný kód, dlouhá doba vývoje atd.). Zdá se mi, že jediný důvod proč se assembler používá, je pocit některých lidí, že když dokáží přetpět nesmyslnou důnu programování v assembleru tak jsou dobří (podobně jako lidé v armádě, většina manuálně pracujících apod.). Chápu, že u většiny z nich by popření tohoto mýtu znamenalo ztrátu sebeúcty, protože takoví lidé většinou nic (jiného, tj. užitečného) neumějí. ...and their extincing race should be noticed. a last time... (Rudy Ratzinger paraphrasing F. Nietzsche)

Ale to se asi nestane. Hlavně se přestaňte zajímat o assembler (resp. jeho strukturovanou podobu - jazyk C) a zkuste něco smysluplného.

Jsem si vědom významu termínů "assembler", "jazyk symbolických adres", "strojový kód". V tomto textu jsem, ale používal intuitivní významy. Tak si prosím odpustěte komentáře na toto téma. Seru na ně. Stejně jako seru na gramatiku. Jo a vynechte kecý o bootovacích rutinách apod. Zapsání nějakých hodnot někam není programování.

neologism@hysteria.sk

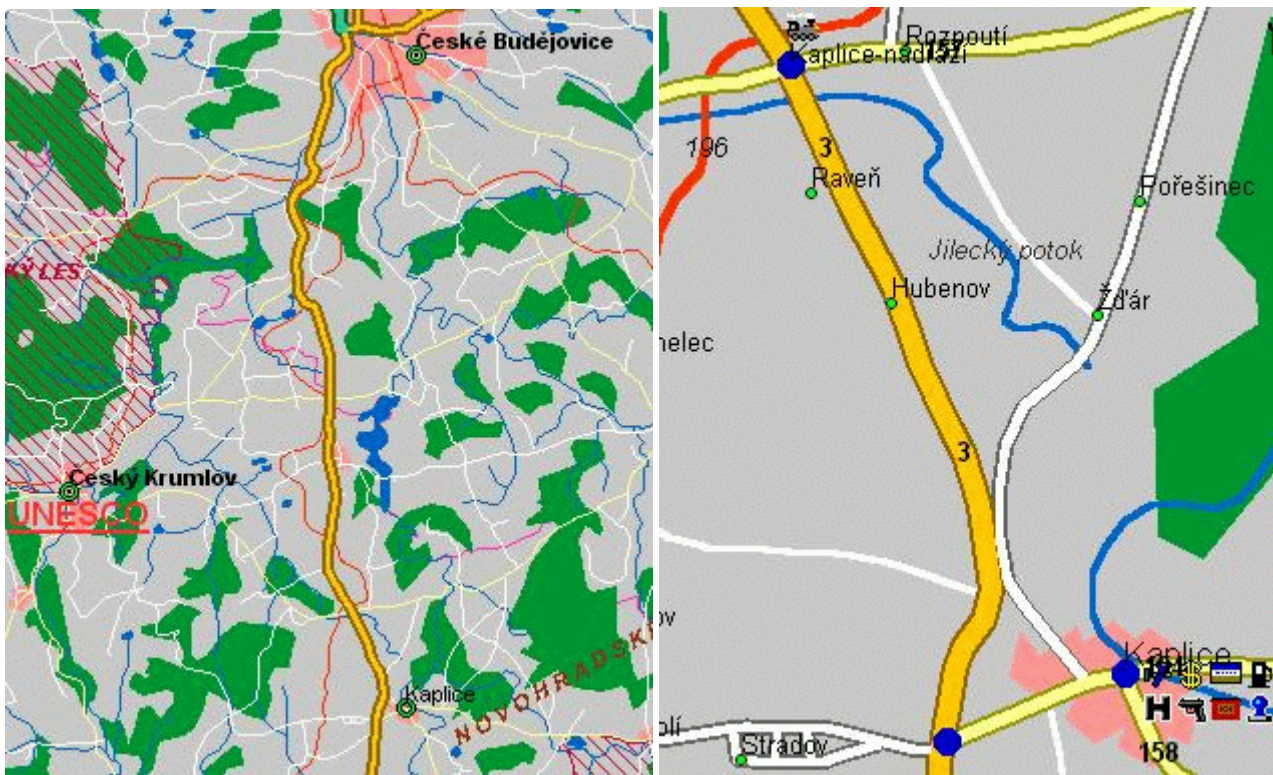
Nenašli jste se dnes zde mezi dementy? Nezoufejte, my to do příště určitě napravíme!

KAPLICON 2004

(30.7.-1.8. 2004)

Kde?

Kaplice. Kaplici najdete v jižních čechách směrem od Českých Budějovic na Dolní Dvořiště. Kousek mapy snad pomůže se zorientovat. V Kaplici se ptejte po **Základní škole**, ve které se bude letos konat. Pokud se dobře pamatují, tak je to tak 300m od náměstí směrem k hlavnímu tahu (E55). Dřívější ročníky byly v Domě dětí a mládeže, který je zrušený. Doporučuji vám přijet **autobusem** nebo **autem** (autobusem se dostanete až do města), vlak stává celých **5** kilometrů od města, autobus zpravidla žádný, ale stopem se do města dostanete (ovšem musíte stopovat sami). Spojení si můžete vyhledat i na netu (www.vlak-bus.cz). Další mapka (ale fakt už poslední) znázorní lépe:



Kdy?

Kaplicon 2004 se koná **30.7.-1.8. 2004** (tj. pátek-neděle o prvním weekendu v srpnu).

K dispozici je asi 8 černobílých **TV Merkur** a pár **EGA monitorů**. Máme taky dostatek **spacáku i karimatek** a abychom věděli, kolik jich máme připravit, připravil jsem pro vás formulář, kde si je můžete rezervovat na <http://zxm.speccy.cz>

Formulář také slouží k tomu, abyste se mohli ohlásit, že přijedete a máme s vámi počítat. Nebo můžete rovnou psát mě na xmagazin@seznam.cz, nebo SMS či brknout na 607 819 905 (Aragorn).

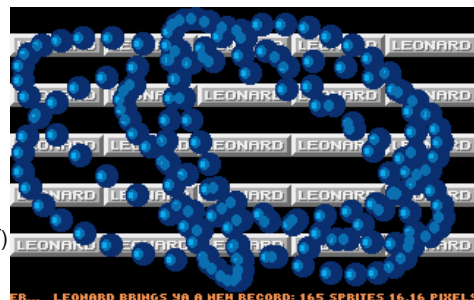
Soutěž?

Letos jsme se jako vydavatelé ZXM rozhodli uspořádat **CodeCompo** s konkrétním zadáním. Máte za úkol vytvořit efekt s co nejvíce pohyblivými bloby 16x16 bodů, na fullscreen pozadí, s hudbou.

Upřesňující podmínky:

- plynulost pohybu
- bloby 16x16
- fullscreen pohyb (to by bylo moc jednoduché, použít jen čtvrtinu plochy obrazovky)
- pozadí (statické nebo pohyblivé) a hudba
- do opakování pohybu aspoň 128 fází (je to těžké SPECIFYcké číslo a animaci to vyloučí)

(v případě dalšího efektu soutěží verze bez efektu, post-party release bude s efektem)



Plánujeme uspořádat i tradiční soutěže **MusicCompo**, **DemoCompo**, **IntroCompo** a **ProgCompo**, zkátka přivezte s sebou svoje nejlepší kousky. Pro úspěch zkusíme i RTC (**RealTimeCompo**), prostě budete kódovat až na místě. Nezapomeňte si vzít taky i svoje Spectrum, abyste nemuseli skuhrat jako Johnny, protože musel pracovat na Didaktiku Gama s D80.

Takže počítáme s vámi. IJK, Sweet Factory a Aragorn