

ZX Magazín

časopis pro uživatele počítačů ZX Spectrum a kompatibilních
číslo: 1-2/00

cena: 48 Kč

Spectrácké textovky bojují o titul té nejlepší!

Nevíte, jak sbalit holku? Tak to raději zabalte!

Blíže o packování opět UVNITŘ!

HiSoft C útočí!



Vodní blábol

(a nedáme si pokoj!)

Ahoj Spectristé,

ZXM opět vyšel jako maxičíslo, nebude se si nic nalhávat, je to ekonomicky výhodnější, protože se zaplatí pouze jedno poštovné. To, že ZXM opět vyšel po dlouhém čase je proto, že Tritol, který ZXM připravoval změnil zaměstnavatele (viz INTRO minulého ZXM) a já ho nebyl schopen dohnat k tomu, aby dodal grafické podklady, druhým zpožďujícím faktorem bylo to, že jste smečka ignorantů a přišlo zoufale málo obnovených předplatných, no a za třetí jste ještě jednou ignoranti a lenoši, neboť pořád slibujete, že pošlete hromadu článků a já zatím nic nedostal. Ale co, možná příště. Zatím si s +Gamovými a mými příspěvky vystačíme.

Čau u dalšího čísla. Užijte si léto, přidejte na obě Specy-sleziny (info uvnitř) a hlavně:

NENECHTE SPECY CHCÍPNOUT!!!! Mat^NC

Tritol odváděl skvělou práci, myslím, že to, co jsem se sazbou ZXM provedl, je proti Tritolově stylu ubohá chatrč.

Článečky od sebe uvidíte rovněž jen tehdy, napíšete-li je a pošlete-li je redakci. Takže nevidíte-li tu žádný na téma, které by zrovna vás interestovalo, chutě sedněte a počněte ho psát, ať stihne vyjít dříve jak za pět let :)

A nezapomeňte předplácet, magazín bez předplatitelů zemře jednou doopravdy...

+GAMA

KLUB 602

MARTINSKÁ 5, PRAHA 1,

TEL.: 02/24213287

Literatura a software pro ZX5, ZX Magazín, hry, A-Mouse, diskety, Public Domain.

Nabídka software: Calculus, Apollon, Tolstoj, Ultra-BT, KUD 1 a 3, Artiller, SQ-Tracker, DevastAce II, V-Base, Source I a II, Public 1-23.

Hry: Inferno, Fuxoft uvádí, Hydrofool, Kuix Snax Dizzy, Skool Daze, Trapdoor I a II, Dizzy collection, Cartoon collection, Superstar Seymour, kolekce 4MOST: Fun, Cute, Toppers, Thrillers, Megaheroes, Worldsports.

O ceník a aktuální informace si napište na adresu redakce ZXM.

Obsah

Vodní blábol

místo blábolu úvodního, podvodního, závodního 2

Sherlock

horký kandidát do soutěže o nejlepší textovku 3

Konec PackMakerů v Čechách?

packovací software opět pod lupou 7

Zilogovy packery

další temná historie 11

HiSoft C 1.1

moderní programovací jazyk a jak na něj 12

George K.'s Tolkien

vy ho nemáte? přečtěte si tento návod 22

O grafice

ještě o ní nevíte všechno 23

O.P.A.T.

napište si svůj vlastní OPAT 25

FOTO reportáž

Zlíncon'99 jak ho znáte i neznáte (však ho ještě poznáte) 27

NoWInky

aneb co pěkného nejspíš uvidí jen majitelé BETASHITů 31

NoWInky plus!

tentokrát jsme k vám velmi štědrí 32

Wild Chars

něco pro programátory OPATu 32

Towdie

kterak pokořiti gamesu 33

Intro plus!

neříkal jsem, že jsme štědrí? 34

Intro

...tady samozřejmě nesmí chybět 35

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

Vydavatel a šéfredaktor: Matěj Kryndler

Redakční rada: Lubomír Bláha, Jaroslav Merta, Pavel Říha, Jiří Doležal

Sazba: volně podle Lubomíra Bláhy

Grafická úprava: Jan Hanousek, Lubomír Bláha

Příprava obálky: volně podle Lubomíra Bláhy

Tisk předloh: Epson EPL 4300

Adresa redakce: Matěj Kryndler, Lotyšská 8/645, 16000 Praha 6

Vychází ve spolupráci s Computer Clubem 602.

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.

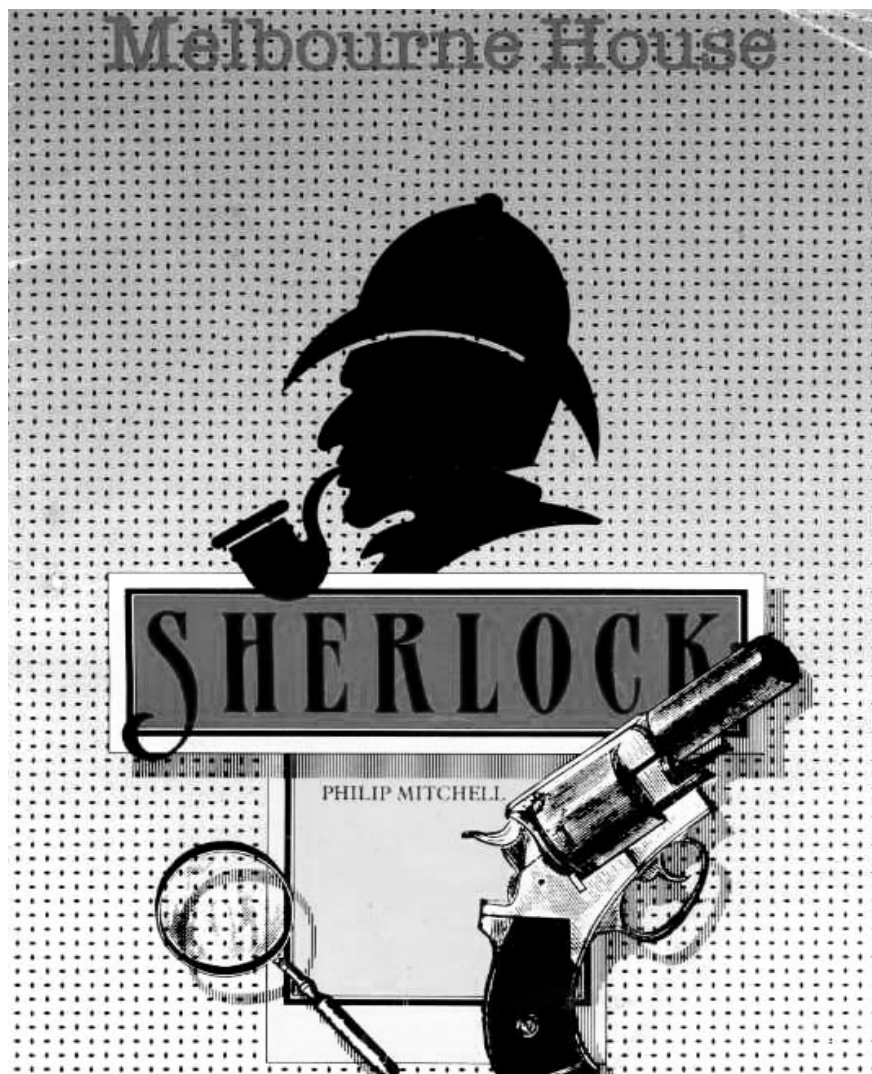
Vychází nepravidelně. Doporučená cena: **48 Kč**

©2000 ZX Magazín, Matěj Kryndler

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

Sherlock

+GAMA



Úvodní blábol

Textovek pro Spectrum není málo. Málo není ani těch skutečně dobrých. I v češtině vyšlo několik skutečných perel jako Indiana Jones III a Podraz III od Fuky, Fidlerův Belegost či Heroes od George K (promiňte, pokud jsem zrovna nějakou VAŠI textovku opomenul). Mezi těmi anglickými se pak vyskytly takové skvosty, že nebylo možno jinak než přeložit je do češtiny (pokud to nějak šlo). Výsledkem toho pak je to, že třeba The Neverending Story na internetu nenajdete jiné, než počeštěnou verzi.

Která textovka je ze všech úplně nejlepší, to je vskutku těžké určit. Kandidátů je mnoho, třeba Worm in paradise od Levelu 9 (a s ním i celá Silicon Trilogy, jejíž je součástí), popisovaný už v YS 00/97 a ZXM 4/98-1/99, Tolkienovská trilogie The Hobbit (i o něm se psalo v ZXM na pokračování), Lord Of The Rings a Shadows Of Mordor,

nebo i to Neverending Story. Já mám ale jasno. Nejlepší Spectráckou textovkou všech dob je Sherlock.

Up. Nějak jsem vynechal textovky ruské. Divíte se? Při těch jejich bukvicích se to nehraje nijak dobře. Přesto musím ale dvě superbomby zmínit. Pochopitelně se do 48ky nevlezou a zabírají valný kus BETASHiTové diskety. Tou první je „Zvjózdnoje následije“, Star Inheritance, která se rusům strašně líbí, i když už to není novinka. Má pěknou úpravu, program bych ovšem označil za poněkud jednodušší, ovládá se to pomocí menu a text je doplňován grafikou na celé třetině obrazovky. Bůhví proč ji ti východní maniaci označují za pokračování Elite. Druhá z těchto azbukovek (to, co na vás tyto hry valí, se nedá nazvat rozumným textem) je „Zerkalo“, Mirror, program nemá tak odlinknutý, texty zobrazuje v úplně miniaturním okně, taky se ovládá pomocí menu a grafiku má přímo fantasticky nakreslenou

(opravdu jsou to ty lepší z obrázků, které na Spectru můžete potkat), tentokrát přes celou polovinu obrazovky. Tvrdí se však o ní, že ji kdosi dokázal dohrát za dva dny bez návodu (pravděpodobně génius, já se v obou těchto hrách nedostal ani z první místnosti). Nebylo by možná zcela od věci, kdyby někdo do ZXM přispěl o ruských textovkách trochu podrobněji.

Věnujmež se však nyní Sherlockovi.

Na programu i vlastním dobrodružství pracoval patnáct měsíců tým několika lidí vedených Philipem Mitchellem, autorem Tolkienovek (The Hobbit, Lord of The Rings, Shadows of Mordor), a když říkám pracoval, tak tím myslím, že se asi skutečně neblákali. Jenom v povelech můžete použít asi 800 různých slov, provádět 53 různých akcí, hra je pochopitelně reálnodobá a využívá osvědčené systémy ANIMTALK a ANIMACTION, musíte Sherlockovi zajistit dostatek času na spaní, při cestování musíte

normálně platit, ve hře se skutečně objevuje asi půlmezo textů (v tomhle se Univerzum nemýlí!) plus nějaké ty obrázky, autoři pamatovali i na výstup přes tiskárnu (ještě vědět jakou, ale asi se tím myslí ZX Printer, nevím, nezkoumal jsem to). Prostředí odpovídá do jisté míry skutečné podobě příslušných míst, dobrodružství je strhující, plné intrik a nebezpečí, zločinci jsou vskutku zákešní a přežití vyžaduje bystrý úsudek a rychlou akci. Sherlock je prostě asi nejsostifikovanější textovka, vůbec pro Spectrum kdy vyvinutá.

Průměrnou dobu řešení této textovky udává výrobce jako „několik měsíců“. Dobrou chuť.

Ovládání (aspoň krátce)

Po nahrání (nebo poté, co jste umřeli) čeká hra na klávesu. Pokud stisknete ‚N‘, nebudou se zobrazovat obrázky (někdy to zdržuje). Tuhle feature mají zabudovanou všechny Mitchellovy hry (vyzkoušejte třeba u Hobbita). Normálně se obrázky zobrazují jen tehdy, stojíte-li v místnosti poprvé, pak už ne a i popis místnosti se pak vypisuje jen zkráceně, obrázek znovu s plným popisem získáte jedině při použití povelu LOOK.

Při inputu lze k mazání použít CS+O (delete) nebo SS+O ke smazání celé řádky.

Co se gramatiky týče, věta MUSÍ obsahovat sloveso. U slovesa mohou být i příslovce - není pravda, že jsou jen pro okrasu, jak psal Univerzum. EXAMINE a CLOSELY EXAMINE jsou dva různé povel, které nedělají totéž (obyčejné EXAMINE například neodhalí některé důležité detaily, ale CLOSELY EXAMINE trvá mnohem déle). Totéž platí i pro všechna určení způsobu, jako QUICKLY a podobně.

Pořádek slov ve větě je poměrně volný, akorát předmět musí být až za příslušným slovesem. Klidně můžete napsat COUNT THE MONEY QUICKLY, COUNT QUICKLY THE MONEY a dokonce i QUICKLY COUNT THE MONEY, ale ne THE MONEY COUNT QUICKLY nebo (trochu logičtěji znějící) WITH STICK CAREFULLY OPEN THE WINDOW. V tomto případě se předměty (WINDOW a WITH STICK) nesmí dostat před OPEN (WITH STICK zde totiž nezastupuje příslovec určení, jak by se mohlo zdát). Dalším pravidlem je, že přídavná jména musí být vždy před příslušným podstatným jménem, takže TAKE BOTTLE GREEN je blbost. Lze zadat i souvětí s více povely (slovesy), oddělenými spojkami ‚AND‘, čárkami, nebo i několik vět za sebou, každou oddělenou tečkou. Počítač je ale zpracovává postupně a mezi dvěma příkazy se provádí akce ostatních postav.

Používání zobecňujících slov typu EVERYTHING, ALL nebo EXCEPT je úplnou samozřejmostí.

Podstatné jméno nesmí stát na místě adjektiva, například DRINK FOAMING

BEER je dobře, ale DRINK BEER IN BOTTLE je blbost, protože BOTTLE je název samostatného předmětu.

Systém ANIMTALK umožňuje konverzovat s ostatními postavami, získávat od nich informace a dokonce je i částečně ovládat. Některé je dokonce užitečně nejprve pozdravit! K provádění povelů slouží klasický příkaz SAY TO WATSON „GIVE ME THE LAMP“. Pokud někoho oslovíte, nemusíte už pak zadávat ani příkaz SAY, například v sekvenci SAY TO LESTRADE „HELLO“ [enter] „WATSON IS INNOCENT“ [enter]. Ano, postavám je možno (a dokonce nutno) sdělovat svoje dojmy a poznatky jako třeba „BROWN KILLED SMITH“ nebo „MAJOR FOULKES HAS THE PISTOL“ či dokonce „THE GUN IS GREEN“. Chcete-li informace nejen rozdávat, ale i získat, použijte povel SAY TO ... „TELL ME ABOUT ...“, je to jediný způsob, jak ve hře provádět výslechy, získávat alibi nebo se dozvědět adresy.

Příkazy je možné zadávat co nejkratší, ovšem po celých slovech, kromě několika nejzákladnějších příkazů (směry, LOOK, INVENTORY a podobně). Napíšete třeba OPEN [enter], Spectrum se vás zeptá OPEN WHAT?, odpovíte DOOR [enter] a tak dále. Směry jsou klasicky jednopísmenné W, S, N, E, U, D, i když se dají vypisovat celým slovem. Pokud nebylo ještě vloženo žádné písmeno, lze směr zadat i jako stisk kurzorové šipky, hned se provede a není ani třeba mačkat enter (je to zvláštní, textovku, kde se směry zadávaly kurzorovou šipkou, jsem kdysi dávno taky napsal, ale nikdo se na to tehdy netvářil jako na kdovíjak dobrý nápad, tak nevím, co si o tom myslet - pozn. +GAMA). Jsou pochopitelně povoleny i šikmé směry NE, NS, SE, SW, taky vypsatelné celým slovem.

Zvláštním způsobem je zadání přímo názvu lokace, kam chcete vstoupit, takže třeba místo GO NORTH zadáte GO TO LODGINGS nebo ENTER LODGINGS, taky lze použít GO THROUGH THE DOOR, pochopitelně lze použít i otevřené okno (a nepro následujte mne upozorněními, že LODGINGS není na severu). Zvláštní je, že průhlednými nebo otevřenými spojeními s vedlejšími místnostmi lze do nich nahlížet pomocí povelu LOOK THROUGH. Že se k pohybu dá použít i příkaz FOLLOW snad ani nemusím dodávat, hodí se, když jdete za někým klikatou cestou, nemusíte pak sledovat směry, kudy přesně jde, ale zadáte jen jeden FOLLOW a pak opakování posledního příkazu.

Pro cestování drožkou je možno tuto kdykoliv přivolat, zatímco vlaky jezdí podle pevného jízdního řádu a ten se vyplatí znát. Drožkáři nezapomeňte říct, kam má jet, jinak vás obšťastní stupidními kidy. Pěšky prý Holmes po Londýně moc nechovil, proto se ve ani hře bez nájemných vozidel neobejde, hlavním viníkem ale bude spíš paměť Spectra 48.

A protože za všechno se platí, uvědomte si, že v tehdejší Anglii se používaly jako měnová jednotka libry, šilinky a pence, v

programu se značí libry librou, šilinky a pence d. Takže devatenáct liber se napíše jako L19. Deset šilinků se napíše buď jako 10s nebo jako 10/-. Šilinky a pence se napíšu jako 10/9, samotné pence jako 9d. Pokud neudáte žádnou cifru, dostanete většinou odpovídající sumu nazpátek, takže to by snad mělo být v pořádku.

Čas se dá psát všelijak, například: 10 (myslí se hodin), 11 PM, 9:30 AM, 10'45 PM.

Dalším zapracováním reálného světa je střídání dne a noci - v noci bývá tma a lecos není vidět. Zapracován je i spánek - Holmes sice není nijak náročný, ale ostatní spánek potřebují, navíc je tu jedna záludnost - většina úmrtí se přihodí ve spánku.

Jednotlivé postavy jsou navíc nezávislé a při každém spuštění hry se mohou chovat trochu jinak, občas se stává, že někdo vyvede takovou blbost, že hru už nejde dohrát a musíte to zkoušet znovu.

Je nutno počítat i s váhou a tvrdostí předmětů - lahvi dveře nerozbijete, ale lahev o dveře rozbijete. Pokud něco neunesete Holmes, unese to třeba někdo silnější.

Používat lze skoro všechny předměty, i ty, které zrovna nemáte u sebe, ale leží někde okolo (pokud je nemá v kapse někdo jiný, to pak nejde), například přijdete ke kameni a zadáte THROW STONE AT WINDOW i bez sbírání.

Zvláštní povel ‚@‘ nepotřebuje ani odenutí. Zadává se zásadně do prázdné řádky a znamená opakování posledního povelu, nemusíte pak jak blbci nic vyfukávat.

O LOOK a LOOK THROUGH už byla řeč, lze je zkrátit jako ‚L‘ nebo L THROUGH.

INVENTORY lze zkrátit jako ‚I‘.

U EXAMINE pozor, je to něco jiného než CLOSELY EXAMINE (už jsem to sem psal).

PAUSE hru zamrzne a čeká na klávesu.

SAVE sejí na kazetu (nebo MBčko, nebo tape emulátor pro D40...).

LOAD nahrává.

QUIT nespustí hru od začátku (není tolik paměti, aby si pamatoval výchozí stav), tudíž se taky snaží LOADnout nějakou pozici (na originál kazetě byl kód hry doplněn sejí vlou pozicí odpovídající startu hry).

PRINT zapne výstup i na tiskárnu, svůj postup si tak můžete archivovat. Blíž jsem to nezkoumal, asi je myšlen ZX Printer.

NOPRINT vypne tiskárenský výstup.

Některé povely:

Směry: DOWN, D, EAST, E, NORTH, N, NORTHEAST, NE, NORTHWEST, NW, SOUTH, S, SOUTHEAST, SE, SOUTHWEST, SW, UP, WEST, W.

Akce: ATTACK, CARRY, CLIMB, CLOSE, COUNT, DRIVE, DROP, ENTER, EXAMINE, FOLLOW, COME, GET, GIVE, GO, HAIL, INVENTORY, I, LOCK, LOOK, L, OPEN, PAY, PUT, SIT, TAKE, TALK, TELL, WAIT, WEAR.

Určení: CAREFULLY, CLOSELY, FAIRLY, FURIOUSLY, QUICKLY, SILENTLY, SWIFTLY, SUDDENLY, IMMEDIATELY, MADLY, STEADI-

LY, WARMLY, SOFTLY, QUIETLY.

Předložky: ABOUT, FROM, IN, INTO, OF, OFF, OUT, THROUGH, TO, WITH.

Řídící: LOAD, SAVE, NOPRINT, PAUSE, PRINT, QUIT.

Že to nejsou zdaleka všechny možné povely je snad jasné. Když jsem třeba nevěděl, že hra používá slovo HAIL, potřeboval jsem přivolat drožku a jen naslepo jsem zkusil CALL CAB. Zabrало to. A tak to bude i se synonymy k jiným příkazům.

Jinak, pokud chcete vědět, jakou že to šílenou kompresí dosáhli toho půlmega textů, tak je to jednoduché. Používají stále stejné věty, ve kterých se vyskytují stále stejné slovní obraty nebo slova, případně nepatrně proházená. Sloh je sice potom trochu jednotvárný a ubírá to hře na pestrosti, ale aspoň se vleze víc textů. Něco za něco.

Pokud se vám na tom obrázku něco nelíbí, tak úplně stejný, ale z Commodorácké verze, byl na zadní straně manuálu, já za to nemůžu. Tenhle jsem ovšem vytípal ze Spectra (snad tu někde je).

A teď k hraní

Není v silách průměrného čecha podobnou hru pařit, proto jsem rád, že mi přispěchali na pomoc James Elliot, Philip Ian, Philip Kendall a Jeremy Smith.

To, co tu najdete, je trochu zpřeházené - například se vyptáváte na věci a osoby, o jejichž existenci se máte dozvědět až za chvíli. Sherlock je ale reálnimová textovka a je potřeba všechno uspořádat tak, aby se to dalo nějak rozumně stihnout.

Podávat dokonalý návod s přesností na příkaz není zas tak úplně možné, systém AnimAction totiž někdy provádí celkem náhodné akce, pokud možno při každém hraní jinak, a stát se tudíž může všechno. Nezapomeňte proto pružně reagovat na vzniklou situaci (ale existují i situace, kdy jediným řešením je příkaz QUIT).

Navíc se ve hře vyskytují některé překlepy, asi vás zaujme SINGED NOTE, není to ovšem nic zpívaného, ale překlep v názvu předmětu, nezapomeňte ho tedy zadávat taky s tímto překlepem.

Hrajmež

(Nejdřív vyzkoušíme převleky)

OPEN PLAIN DOOR, N, WEAR CHINA MANS DISGUISE, TAKE OFF CHINA MANS DISGUISE, WEAR OLD MANS DISGUISE, TAKE OFF OLD MANS DISGUISE (povšimněte si, že nám oba převleky zůstaly v kapse).

S, SAY TO WATSON „HELLO“ (to aby nás Watson neignoroval, může se to stát), WAIT (dokud Watson nedočte k vraždám), TAKE LAMP, SAY TO WATSON „DROP CHRONICLE“, „FOLLOW ME“, OPEN MY DOOR, W, D, OPEN FRONT DOOR, S (a jsme venku).

HAIL CAB, SAY TO CABBIE „HELLO“, CLIMB INTO CAB, SAY TO CABBIE „GO TO KINGS CROSS ROAD“, WAIT (v 8:45 AM bychom tam měli dorazit), GIVE MONEY TO CABBIE (měl by vrátit L5/4s), CLIMB OUT

OF CAB, NE, NE, NE, NE, SAY TO LESTRADE „HELLO“, WAIT (kolem 9:16 AM přijede vlak), WAIT (až Lestrade vlez do vlaku), CLIMB INTO TRAIN, WAIT UNTIL 10:00, WAIT (tím jsme si zkrátali neustálé opakování příkazu WAIT, asi v 10:25 by měl vlak zastavit).

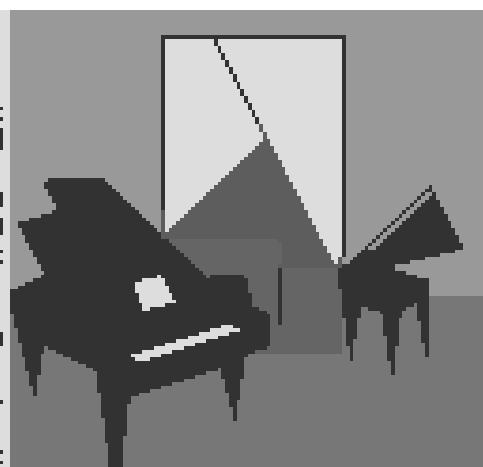
CLIMB OUT OF TRAIN, FOLLOW LESTRADE, SAY TO STRAKER „HELLO“ (jinak nás místní policajti nepustí na důležité lokace), S, W, SAY TO POLICEMAN „HELLO“, S, SW, S, SW, E, CLOSELY EXAMINE CORPSE, CLOSELY EXAMINE NOTE, CLOSELY EXAMINE BRIDGE, W, N, N, NE, S, OPEN OAK DOOR, S, OPEN HEAVY DOOR, E, WAIT (v 11:45 by mohl přijít Lestrade, čekáme, dokud si s lidmi nepromluví o alibi), SAY TO DAPHNE „HELLO“, „TELL ME ABOUT PATRICIA FENDER“ (P.F. z lístku u mostu, tady je to trochu zpřeházené, ale času není moc a tak

WAIT (Lestrade bude z majora tahat alibi)

LOOK (jestli tu je pořád Watson), SAY TO WATSON „FOLLOW ME“, OPEN SOLID DOOR, N, OPEN SITTINGROOM DOOR, E, OPEN FRONT DOOR, N, N, W, S, S, OPEN BACK GATE, E, OPEN BACK DOOR, N, OPEN PLAIN DOOR, W, OPEN SITTINGROOM DOOR, W, U, CLOSELY EXAMINE BED, CLOSELY EXAMINE WINDOW, CLOSELY EXAMINE GRAMOPHONE, CLOSELY EXAMINE PIANO (pokud jsme dávali pozor, už do případu trochu vidíme), D, OPEN FRONT DOOR, W, N, N, N, N, WAIT (ve 2:50 přijde Lestrade), SAY TO LESTRADE „MAJOR IS INNOCENT“ (teď jsme dali najevo, že nám to opravdu jasné je, a nejen tohle, Lestrade ovšem žádá důkazy, tím nám končí zahřívací kolo, kdy jsme se s případem seznámili, a nastupuje makačka hlavou).

WAIT (ve 3:15 PM přijede vlak), CLIMB

```
You are in a large
lavishly decorated
bedroom, a broad
window overlooks a
small lane leading
north. Below there
is the stairway.
To the east there
is the window. You
can see Watson,
basils bed, basils
piano, a
gramophone and the
window.
```



```
Mon 08:00am
> .
> .
> .
> LOOK.
> +
```

hle je to jednodušší všechno stihnout)

SAY TO BASIL „HELLO“, „TELL ME ABOUT YOUR ADDRESS“, OPEN DINNINGROOM DOOR, S, OPEN PLAIN DOOR, W, SAY TO COOK „HELLO“, „TELL ME ABOUT BASIL“, SAY TO GARDENER „HELLO“, „TELL ME ABOUT MAJOR“ (s ním se taky seznámíme až za chvíli, ale co, prohazovat se s tím nebudu)

OPEN PLAIN DOOR, E, OPEN DINNINGROOM DOOR, N, OPEN HEAVY DOOR, W, S, OPEN WOODEN DOOR, W, EXAMINE DESK, OPEN DRAWER, CLOSELY EXAMINE DRAWER, OPEN FALSE BOTTOM, LOOK, EXAMINE BOOK, EXAMINE NOTE, E, N, OPEN LARGE DOOR, N, N, N, E, E, S, OPEN FRONT DOOR, S, OPEN SITTINGROOM DOOR, W, OPEN SOLID DOOR, S, CLOSELY EXAMINE BODY, CLOSELY EXAMINE BOOKCASE, LOOK, W, CLOSELY EXAMINE CLOTHES, SAY TO WATSON „GET CLOTHES“ (budou se hodit, je na nich jmenovka), E, N, WAIT (v 1:30 by měl přijít Lestrade s majorem), FOLLOW LESTRADE (do knihovny),

INTO TRAIN, WAIT UNTIL 4:30, CLIMB OUT OF TRAIN, SW, SW, SW, SW, SW, WAIT (Lestrade odjede do Parliament street, tam je Scotland Yard, pokud to nevíte), HAIL CAB, CLIMB INTO CAB, SAY TO WATSON „CLIMB OUT OF THE CAB“, SAY TO CABBIE „GO TO SIDMOUTH STREET“, WAIT (celá tahle následující šaškárna s majorem se snad dá zkrátit tak, že rovnou přijedeme k opiovému doupěti v Slater street a tam si v převleku za číňana na majora počkáme)

CLIMB OUT OF CAB, LOOK, OPEN OAK DOOR, NW, WAIT, WEAR CHINA MANS DISGUISE, SE, HAIL CAB, CLIMB INTO CAB, WAIT UNTIL 10:30 (drožkář furt bude otravovat se svými kecy, možná to jde nějak obejít), důležité je čekání přesně v 10:28 stopnout ENTERem, WAIT (dokud v 11:02 nepřileze major, nevleze do domu, nevyleze zas ven a neporučí si drožku do Slater street, mělo by být nejpozději 11:25), teď se nechme dovézt do Slater street, CLIMB OUT OF CAB (když tam před 11:51 dorazíme), N (doufám, že máme s sebou lampu),

Návod

TAKE OFF CHINA MANS DISGUISE, S, SAY TO LESTRADE „MAJOR VISITED THE DEN“.

A tady se objevuje problém - Lestrade odpoví, že se zítra vrátí do Leather Head, tuhle větu musí stihnout říct ještě v pondělí, pokud už začalo úterý, na celý den Lestrade zaleze a uvidíme ho až někdy ve středu. Někdy se taky stane, že Lestrade majora zastřelí. A nezapomeňte drožkářům platit, jinak se ty jejich výhrůžky skoro ani nedají poslouchat (číst).

Pokud ještě nepočlo úterý, pokračujmež. HAIL CAB, CLIMB INTO CAB, DROP CHINA MANS DISGUISE, SAY TO CABBIE „GO TO KINGS CROSS ROAD“, WAIT, SAY TO WATSON „FOLLOW ME“ (pokud vlez do drožky; jinak mu to řekneme až venku),



CLIMB OUT OF CAB, NE, NE, NE, NE, WAIT UNTIL 1:45 AM, CLIMB INTO TRAIN, WAIT UNTIL 3:00, CLIMB OUT OF TRAIN, S, S, S, S, OPEN BACK GATE, E, OPEN BACK DOOR, N, OPEN PLAIN DOOR, W, S, OPEN SAFE, LOOK, EXAMINE LETTERS, CLOSE SAFE, N, OPEN SITTINGROOM DOOR, W, OPEN FRONT DOOR, W, N, W, S, SW, S, SW, E, WAIT UNTIL 9:20, SAY TO LESTRADE „MRS BROWN KILLED HERSELF“, Lestrade se zase ptá po důkazech, takže SAY TO LESTRADE „EXAMINE STREAM“, W, N, N, NE, N, E, N, N, N, N, WAIT UNTIL 10:45, CLIMB INTO TRAIN, SAY TO LESTRADE „TRICIA BLACKMAILED MRS BROWN“ (ten nechá pavec se zas ptá po důkazech), WAIT UNTIL 12:00, CLIMB OUT OF TRAIN, SW, SW, SW, SW, SW, WAIT (Lestrade odjede drožkou, není dobré, když jsou na scéně drožky dvě, dochází pak ke zmatenostem), HAIL CAB, CLIMB INTO CAB, SAY TO CABBIE „GO TO PORTMAN STREET“, WAIT, CLIMB OUT OF CAB, LOOK, OPEN FRONT DOOR, S, OPEN HEAVY DOOR, S, OPEN SAFE, LOOK, READ NOTE, EXAMINE FOLDER, SAY TO WATSON „GIVE ME CLOTHES“, SAY TO TRICIA „HELLO“, „TELL ME ABOUT BLOODSTAINED CLOTHES“, LOOK, SAY TO MRS JONES „HELLO“, „FOLLOW ME“ (ten look není úplně od věci, tohle je totiž Jonesová a mrtvo-

la patří Fenderové, tím LOOKem si snad nastavíme správné užívání jmen, pokud bude Jonesová remcat, tak povel SAY TO MRS JONES „FOLLOW ME“ zopakujeme)

N (pozor, aby nás v domě nezamkla), HAIL CAB, CLIMB INTO CAB, SAY TO MRS JONES „CLIMB INTO CAB“, LOOK (jestli je Jonesová i Watson v drožce), SAY TO CABBIE „GO TO PARLIAMENT STREET“ (asi to bude potřeba taky zopakovat), WAIT, CLIMB OUT OF CAB, SAY TO MRS JONES „CLIMB OUT OF CAB“ (zas opakovat, jinak uteče), SAY TO MRS JONES „FOLLOW ME“, E, SAY TO LESTRADE „MRS JONES KILLED TRICIA“, GIVE CLOTHES TO LESTRADE, SAY TO LESTRADE „EXAMINE CLOTHES“, SAY TO MRS JONES „TELL LESTRADE ABOUT

THE CLOTHES“, SAY TO LESTRADE „MRS JONES KILLED TRICIA“, W, HAIL CAB, CLIMB INTO CAB, SAY TO CABBIE „GO TO CAMDEN STREET“, WAIT, CLIMB OUT OF CAB, WEAR OLD MANS DISGUISE, E, OPEN WINDOW, WAIT UNTIL 10:00 PM (někdo otevře okno), NE (pořád potřebujeme lampu), NW, NE, OPEN HEAVY DOOR, SE, TAKE NOTE (neprozkoumávat), NW, LOOK, OPEN PLAIN DOOR, NE, OPEN BACK DOOR, NE, CLOSELY EXAMINE PILE, LOOK, TAKE NOTE (neprozkoumávat), SW, SW, SW, SE, SW, W, TAKE OFF OLD MANS DISGUISE, EXAMINE TORN NOTE (mělo by tam být VAKS USH KNWH UL UNMH YWNQH. UNMH QNTH FTK RTLVBK GHNUS SNK KUNTUHG USH YLWAQH BLKABP NTLEBG - snad jsem se nepřepsal)

EXAMINE RIPPED NOTE (A BLV SNIH USH YWNBK. XLET YTAQH AK NQQHYUNR-WH. WHU FH MBLV VSHTH XLE)

(Kód pro tyto dopisy je A=I, B=N, E=U, F=M, G=D, H=E, I=V, K=S, L=O, M=K, N=A, P=G, R=B, S=H, T=R, U=T, V=W, W=L, X=Y, Y=P a jejich samotný text je WISH THE SALE TO TAKE PLACE. TAKE CARE MRS BROWN DEATH HAS STARTED THE POLICE NOSING a I NOW HAVE THE PLANS. YOUR PRICE IS ACCEPTABLE. LET ME KNOW WHERE YOU - tady to není překlep, je to

tím, že kousek dopisu schválně chybí).

HAIL CAB, CLIMB INTO CAB, SAY TO CABBIE „GO TO PARLIAMENT STREET“, WAIT, CLIMB OUT OF CAB, E, WAIT UNTIL 7:00, SAY TO LESTRADE „HELLO“, „BASIL HAS THE PLANS“ (zas je to třeba opakovat, dokud se Lestrade nezeptá na místo prodeje), W, HAIL CAB, CLIMB INTO CAB, SAY TO CABBIE „GO TO CAMDEN STREET“, WAIT, CLIMB OUT OF CAB (kolik nás už ty drožky vlastně stály?), LOOK (tuhle vyjžděku do Scotland Yardu si lze možná odpustit a oznámit Lestardovi až potom všechno naráz), WAIT UNTIL 9:50, E, LOOK THROUGH WINDOW, NE, TAKE NOTE, SW, I, EXAMINE SINGED NOTE (DSWYP GSUV CUVK-SUP CSUX GSIC PPYA GPI UVK KS ZXCYVK ILK KS WXSPU UVK ZRD PPYL Y.V.L, podle kódu A=M, C=R, D=B, G=D, I=O, K=T, L=W, O=P, P=L, R=U, S=A, U=E, W=S, X=N, Y=I, Z=Y se rozluští jako BASIL DAEH REHTAEL RAEN DAOR LLIM DLO TA YTRIHT OWT TA SNALP EHT YUB LLIW I H.W., pozpátku si to už přečtete sami).

E, HAIL CAB, CLIMB INTO CAB, SAY TO CABBIE „GO TO PARLIAMENT STREET“, WAIT, CLIMB OUT OF CAB, E, SAY TO LESTRADE „HELLO“, „THE SALE LOCATION IS THE OLD MILL ROAD“ (opakovat, dokud neodejde na západ), W, FOLLOW LESTRADE (do drožky), WAIT, CLIMB OUT OF CAB, NE, NE, NE, WAIT UNTIL 12:15, CLIMB INTO TRAIN, WAIT UNTIL 1:30, CLIMB OUT OF TRAIN (je nezbytně nutné vystoupit dříve než Lestrade), S, S, CLIMB INTO CAR (dříve než Lestrade, Straker a policajt, jinak v autě už nebude místo), WAIT UNTIL 4:30 (ENTERem zastaví ve 4:28), WAIT (ve 4:46 Lestrade vystoupí), CLIMB OUT OF CAR (pokud nevidíme Basila s agentem jít na sever, je něco špatně)

N, N, N, (Lestrade musí okomentovat, co vidí), SAY TO LESTRADE „FOLLOW ME“, S, S, S, CLIMB INTO POLICE CAR (nesmí být víc jak 5:05 PM), SAY TO POLICE „FURIOUSLY GO TO KINGS CROSS ROAD“ (zas to asi bude chtít párkrát zopakovat), WAIT UNTIL 6:00 (dorazit musíme PŘED 6:20 PM), CLIMB OUT OF CAR (agent by měl odjždět drožkou na Buckingham Palace road), NE, NE, CLIMB INTO TRAIN, WAIT (jedeme podzemím, na Paddington station nevystupujeme), WAIT (přijedeme na Victoria station, můžeme použít LOOK ke shlédnutí přestřelky, a to je konec).

A teď zkuste říct, že to není dokonalá textovka...

Název: Sherlock
Typ: textovka
Autor: P. Mitchel
Producent: Melbourne House
Hodnocení:

90 %

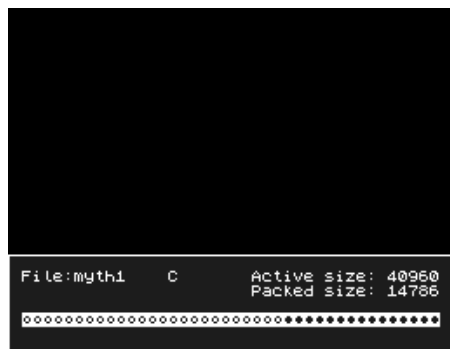
Konec PackMakerů v Čechách? Ne!

(pokud si nekoupíte BETASHiT)

+GAMA

Pořád se to omílá kolem dokola. Který packer používat a který ne. Přitom na výběr skutečně nemáme. Co s tím? Řešení, jak už jste se mohli dočíst, přináší mistr Stupidius čili Zilog. Pokud se to k vám ale ještě nedonesly novinky, tak se mistr Zilog nechal slyšet, že vymyslel nový, zcela geniální způsob pakování a nemůže jinak, než ho do svého packeru taky zahrnout. Z tohoto důvodu se dokončení PackLandu odkládá přinejmenším na dobu po letošním DoxyConu, bude-li se pod záštitou mrtvol z E.S.A. vůbec konat. A co budeme dělat do té doby?

Pakovat. Pakovat tím, co je. Ale stejně teď zkuste přivřít oči a představit si, jak sedíte za volantem BETASHiT, motory mechanik řvou na plné obrátky a váš ZX Profi (Pentagon, Sprinter, ...) právě natahuje nejslavnější ruský packer, pokud možno toho nejzvučnějšího jména...



Tohle je například MS Pack 01.96 v akci. A vůbec. Připomeňme si je radši všechny.

Archivery:

Ty jsou jenom dva, oba BETASHiTové, pokud nevíte, o co gou, obraťte se na ZXM 2/99. Při testu jsem je nechal spakovat několik souborů, z toho některé už pakované něčím jiným (v praxi to tak většinou taky bývá). Výsledek?

LZ – Compressor vytvořil archiv dlouhý 127 945 bajtů. Přesněji řečeno, protože BETASHiT neumí víc jak asi 64 kilovou velikost souborů, byl tento archiv rozdělen do dvou souborů. Výsledek je to o něco horší

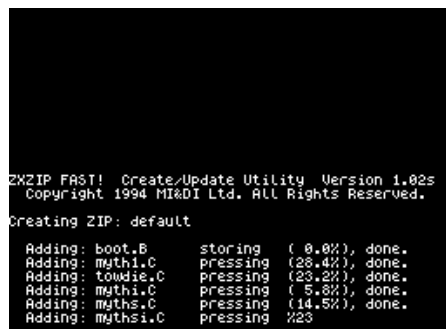
než u ZX ZIPu, asi o 2,7 %, program však vyniká snadným ovládáním a příjemným rozhraním. Koneckončů, jeden obrázek už v ZXM byl, tady ho vidíte přímo v akci. Asi si toho nevšimnete, ale ten font je opravdu doporučení...

Name	Size	Packed	ratio
boot	033200	033200	100%
myth1	40960	14786	67.1%
towdie	40960	14786	67.1%
mythi	350720	14786	100%
myths	350720	14786	100%

ZX – ZIP vytvořil archiv o délce 124 513 bajtů. Pochopitelně taky ve dvou souborech. Ani tady však není ovládání nijak krkolomné, jak jsem zjistil, kompatibilita s PeCoidním ZIPem je nulová, ale existuje program ZX-ZIP pro MS-DOS a ten vám dokáže archivy ze spectráckého ZX-ZIPu rozbalit (jako Hobety). Nějaký přílišný význam to ale asi nemá, nenapadá mne, k čemu by to mohlo být dobré. I obrázky ZX-ZIPu už v ZXM byly, vzpomeňte si.

Na druhou stranu, aby toho nebylo málo a zmatek byl větší, existuje i PKUN-ZIP pro TR-DOS, který pro změnu pracuje s archivy z PeCoidního ZIPu...

Víc se snad k archiverům ani dodat nedá.



Screen packery:

Tady už to bude zajímavější. K čemu screen packery slouží, není třeba vysvětlovat. Pomineme-li nesmyslné programky,

z nichž jeden byl dokonce dodáván v jednom balíku s Artistem II, kralovala u nás tuzemská série Pressorů. Ta má dokonce jeden primát – Pressor 6 je vybaven možností packu jen části obrázku. Pravda, i některé ruské packery umožňují vypnout barvy nebo pakovat jen vybrané třetiny, ale tak, jako to má Pressor 6, to nemá už nikdo. Pokud ovšem plánujete nějaký super účinný screen packer spáchat, prosím snažně, zabudujte tam ten výběr okénka taky. Další příjemná věc je neustálé zobrazení obrázku jako pozadí, takže uživatel má pořád perfektní přehled o tom, jaký obrázek že to právě je v paměti. Rusové musí používat funkci VIEW SCREEN.

Jaké obrázky jsem to pakoval? Jeden obrázek jsem vzal z Towdieho, a protože jsem si nebyl jist, zda jsou to data dostatečně reprezentativní, přidal jsem i obrázek ze hry Navy Seals jako trochu jedno-dušší a méně náročný na pakování, a jako nejjednodušší jsem vzal obrázek ze hry Popeye.

Takže až budu jako výsledky testu uvádět zmatená čísla, tak to první je Towdie, to druhé Navy Seals a to třetí Popeye.

Screen Compressor 1.00 je jeden z těch hodně mizerných ruských, packuje hůře než české Pressory. Omega Hackers Group si sice dali záležet na ovládání, ale i tam bych jim vytkl to, že tak mrňavou šipčkou se na mrňavá tlačítka špatně strefuje.

A jak pakoval? Výsledky jsou tyto: 5457, 3523, 2311.



Powerfull Code Decreaser 6.2i není přímo screenpacker, ale nabízí volbu packu screenu. Towdieho zvládl lépe než Pressory, ale hlavně Popeye už byl horší. Nicméně je od něj hezké, že tuto netradiční volbu obsahuje.

Packování

POWERFULL CODE DECREASER V6.2
improved by VEKA/CPU 23.09.97.

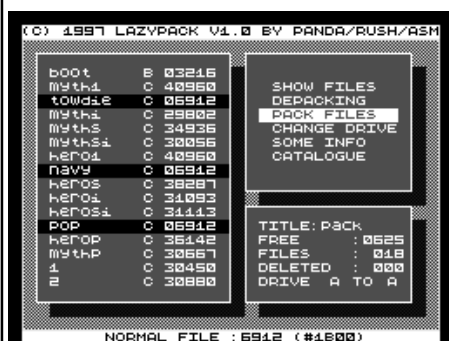


A jak packoval? 5122, 3492, 2343.

Andrews Screen Crasher je zvláštní tím, že nabízí tejpové operace. Packuje velice slušně, výsledky jsou 4769, 3171 a 1999.



LAZYPACK 1.0 pochází od lidí z mezinárodní skupiny RUSH, jeho autor se krom jiného podílel třeba na tvorbě Crime Santa Claus. Tento screenpacker používá pří Huffmanovo kódování, tím se liší od ostatních, a packuje opravdu perfektně, projeví se hlavně na jinak špatně packovatelných screenech. 4550, 2996, 1908.

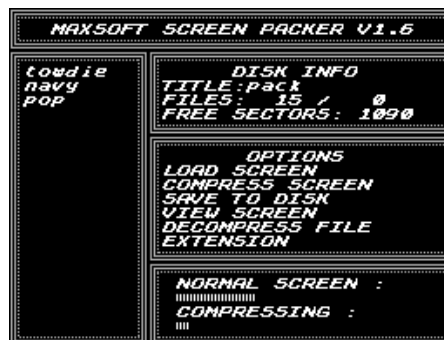


Maxsoft Screen Packer 1.6 je program, který spojuje genialitu s totální idocií. Zpracovaný je velice slušně, obsahuje dokonce screensaver, což je věc na Spectru jinak takřka nevidaná. Nabízí tři možnosti packování, LIN, RTD a BUF, a ke každé z nich výběr ze tří depackerů – FAST, SLOW nebo žádný. FAST je rychlý a nepoužívá zásobník, SLOW je pomalý a zásobník používá. Kupodivu SLOW má asi i delší rutinu. Jinak BUF a RTD používají stejnou metodu, jenom BUF vyžaduje za obrazkem 6912 bajtů volného místa pro buffer, zatímco se při depacku projde a přepíše případná tam uložená data. A jaké byly výsledky?

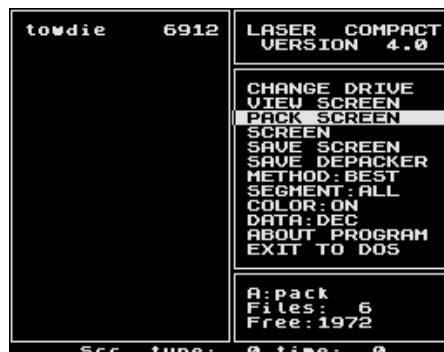
FAST.LIN 5177, 3478, 2286.

FAST.RTD 4652, 3036, 1947.
FAST.BUF 4593, 2977, 1888.
SLOW.LIN 5204, 3505, 2313.
SLOW.RTD 4670, 3054, 1965.
SLOW.BUF 4623, 3007, 1918.

Nejlepší výsledky tedy dával FAST.BUF, bez bufferu by to byl FAST.RTD. Co mu ale musím vytknout je to, že uživatel sám musí nastavit, kterou metodu má použít, místo toho, aby program sám vyzkoušel, která dopadne nejlépe. Přece jenom, přepínat tolik voleb, to je o nervy...

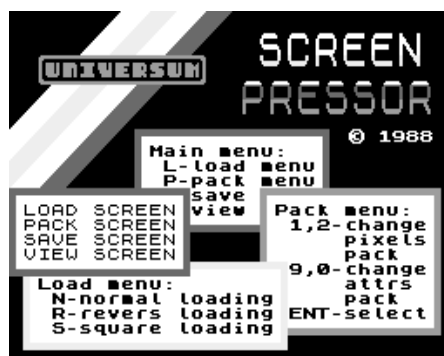


Laser Compact 4.0 napsal autor packerů Hrum a Hrust. A to by mohlo stačit. Výsledky (délka je pochopitelně včetně rutiny): 4537, 2948, 1770. Bez rutiny by to bylo ještě méně (program umožňuje sejvovat jen data a použít pro více obrázků jednu rutinu, je to další šetření místem v paměti). Ve srovnání se všemi ostatními dostupnými je prostě zázračný.



Mr Pack 2 není sice míněn jako screen packer, ale když už tu byl ten Powerfull Code Decreaser 6.2i, tak proč ne i Mr Pack? Protože dopadl bezkonkurenčně nejhůř: —, 3928, 2825.

Pressor 1 je Universumův packer, psaný v Basicu, výsledný kód není nejspíš ani



relukovatelný. Nic jiného neumí.

A jak Pressor 1 packoval? 5326, 3494, 2298.

Pressor 2 – 5 packují všechny stejně, tak jsem je sfouknul do jedné položky. Pořadí je to Basic, od verze 3 je doplněn i o Loadmaker, do verze 2 a 3 je zabudován kompletní Wlezeley epsilon (tedy Wlezeley 5), verze 5 už má jen editor screenů, stejný jako ve Wlezeley 7. Pressor 4 se mi ale do pracek nedostal. Pro zájemce o tyto programy, Pressor 1 nahrává obrázek na adresu 51050, ostatní na adresu 51000. No a k těm výsledkům:



Pressor 2: 5329, 3487, 2291.

Pressor 3: 5329, 3487, 2291.

Pressor 5: 5329, 3487, 2291.

Pressor 6 je už pure assembler, no vždyť ho určitě všichni používáte, tak ho tu nebudu rozmazávat, jako jediný umožňuje volbu okna k zapakování, což mi u ruských superděl chybí, bohužel je tak trochu D40 only.



A jak packuje? Zas trochu jinak než starší verze. 5375, 3437, 2232.

No a tím jsme snad screen packery jednou provždy vyčerpal.

Top Ten:

Jako zázračný se ukázal Laser Compact 4.0, skvělý je i Lazypack, někdy, to podle vašich požadavků, se vytáhne i Maxsoft Screen Packer. ASC také není špatný, dál za ním by se umístil Powerfull Code Decreaser 6.2i a Universumův Pressor 6, pak ostatní Pressory, pak zaostává Screen Compressor 1.0 a nakonec Mr Pack 2.

A můžeme jít na poslední téma.

Code packery:

Tyto oblíbené programy se jeden od druhého dost liší v možnostech i požadavcích. Po úvaze jsem otestoval i některé jednoduché kombinace packerů, u těch českých to zvyšovalo míru komprese. Kupodivu se ukázalo, že těm ruským to nedělá zrovna dobře. Snažil jsem se sehnat pro test co nejrepresentativnější data, napadl mne první level Mythu, to je přece ukázková reprezentativní hra se spoustou kódu a grafiky. Protože si s těmito daty poradily implobery hravě, zatímco huffmanny se dost zapotily, myslel jsem si, že data nejsou dost reprezentativní, a zkusil jsem Heroes, tam je přece spousta textů packovaných Tolkienem, tedy odrůdou imploze, to bude pro implobery těžší oříšek. Nebyl, v podstatě se potvrdily výsledky z prvního kola. Každým ruským packerem jsem prohnal kromě nativního souboru (vytípnuté od 24576 do konce paměti, délka 40960) i tento pakovaný PK Lite (jen shrink, jen imploze, shrink+imploze) a PackMakerem 2.4 (oba průchody) v uvedeném pořadí.

Za překlepy ve výsledcích se pro jistotu omlouvám předem, je to strašný mrak čísel. Asi je zkusím ještě shrnout do nějaké tabulky, kvůli přehlednosti. Kdyby tu ale nebyla, tak to znamená, že jsem se na to vybodl.

Jo, a zdravím Jiřího Třasáka! Bez něj by tahle blbost nikdy nevznikla, ale díky němu vím, že někdo tyhle ptákoviny čte, což mne povzbudilo k dalšímu psaní.

PK Lite byl podrobně rozebírán v ZXM 2/99, soubory spackoval takto: Myth – shrink 34936, implod 29802, shrink+implod 30056, a navíc jsem přidal i implod+implod 29450. Heroes – S 38287, I 31093, S+I 31113, navíc I+I 29450.

PK Huff zřejmě soutěží o titul nejhoršího packeru století. Myth 32767, Heroes 38360. Na víc mi nezbyly nervy. Tenhle packer si snad dělá srandu.

Allpacker je starý program napůl v Basicu, fixně packuje od 24200 do konce paměti. Přesněji řečeno od 24200 ne, tam je rutina, která v paměti pak zůstává, vlastní packovaný blok musí začínat výš. A jak packoval? Myth 30862, Heroes 36349. Víc jsem nepackoval, data z jiných packerů neprojdou přes první průchod, to platí pro všechny Proximácké packery.

PackMaker 2.2 – jednotlivé verze Packmakeru se liší víceméně v drobnostech, ale packují všechny stejně. Myth 30667, Heroes 36142.

PackMaker 2.3 Myth 30667, Heroes 36142.



PackMaker 2.4 Myth 30667, Heroes 36142.

Mr Pack, to je ten fialový, při packování zobrazuje pruhy v borderu jako PackMaker. Už se ovládá pomocí menu, jako defaultní začátek bere ramtop, parametry se dají zadávat v libovolném pořadí, uživatel vidí zároveň konec i délku – to je komfort. Škoda že příliš ojedinělý, kromě MrPacků a Tolkienů už nikdo tento způsob nepoužil. Myth 30679, Heroes 36154.

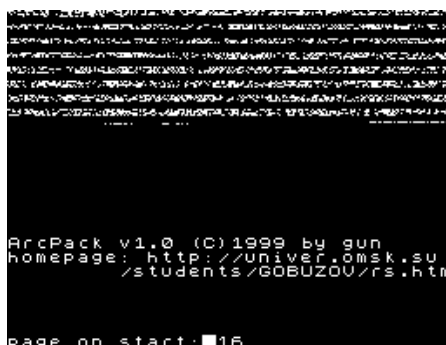
Mr Pack 1 je modrý, taky při pakování pruhuje v borderu. Myth 30679, Heroes 36154.

Mr Pack 2 zobrazuje při packování nápis WAIT. Jinak co do výkonu se jednotlivé Mr Packy od sebe neliší. Zajímalo by mne, proč mají výsledek horší než PackMakery (asi delší rutina, ale proč?). Myth 30679, Heroes 36154.

Ještě jednou zopakuji, ruskými (ukrajinskými, běloruskými,...) packery jsem prohnal soubory Myth a Heroes, každý ve stavu normálním, S, I, S+I z PK Lite a z PackMakera 2.4. Používal jsem zásadně ten mód, který byl označen jako účinnější nebo pomalejší ve víře, že oboje je totéž. Tož podívejme se na ně.

Malá poznámka – délky jsou udávány s rutinou, protože tejpová verze Hrustu 2.1 sejevuje rutinu odděleně, připočítávám u něj k výsledku 256 (to je délka rutiny), i když v praxi to bude ještě víc (musíte přidat nastavení parametrů, při volání z Basicu i přesun depackeru, případně nastavení SP a jiné).

Arc Pack 1.0 je program spartánského vzhledu, ale velkých možností a slušného výkonu. Kromě samozřejmostí, jako je nastavení depacku na různé adresy nebo u nás neznámý posun SPčka, umožňuje i stránkovat ve 128čce a tak při depacku



rozhažovat různé bloky do jednotlivých stránek bez toho, že byste museli psát nějaký program s LDIRy a OUTy. Rusové tvrdí, že packuje lépe než ASC LZPAC. Je to pravda, navíc ve většině případů překoná mírně i KSA LZPAC (Code Cruncher 3.11) a oba LZH packery (jenom u fajlů předpackovaných samotným shrinkem nepatrně zaostává). Zvládne kódy do délky 42240 a je 128 only. Jeho zvláštností je, že se spolu s ním distribuuje i jeho zdroják (nešel mi ale přečíst).

Myth 28704, 29485, 28997, 29610, 30336. Heroes 30402, 30918, 30020, 30337, 34782.

ASC LZPAC vyžaduje Spectrum 128, kromě standardních nastavení autostartu, SPčka a podobně umožňuje do dekomprimovaného bloku zadat až 3 POKE, umí ladovat fajly i z tejpů, packuje fajly maximálně 41000 dlouhé, depacker je asi stobajtový. Běžně packuje o něco hůř než Arc Pack 1.0, v porovnání s KSA LZSS (Code Cruncher 3.11) je sice horší, ale lépe si poradí s fajly předpackovanými implozemi.

Myth 28754, 29622, 29165, 29749, 30451. Heroes 30532, 31048, 30155, 30462, 34905.



DSQ 4.12 je populární packer od tvůrců ze skupiny RUSH, při packování nabízí možnost volby mezi různými velikostmi okna od 0.5 kB do 32 kB! Já jsem při packování použil okno 4 kB velké, kdyby někdo zjistil, že při jiné velikosti okna packuje účinněji, může to být pravda, já to nezkoušel. Rusové tvrdí, že na texty je DSQ lepší než Hrust. Pravda je, že nepackuje špatně, v mém testu byl ale navzdory tomu, že se výkonem Hrumu a Hrustu blíží, nepatrně horší než Powerfull Code Decreaser 6.2i. Nastavování okna bych někdy viděl i jako nevýhodu – musíte zkoušet všech 7 možností, která dopadne nejlépe. Ale stejně – Mat-



Packování

softe, co mu tak předělat diskové operace na nějaký rozumný tuzemský systém?

Myth 26620, 26908, 27535, 28006, —
— Heroes 28052, 28469, 28793, 29101, 35971.

Hrum 3.5 pochází od autora obou Hrustů a Laser Compactu 4.0 (Dima Pjankov, Gorno-Altajsk, Altajskij kraj). Packuje nepatrně hůř než Hrust 1.2, zato zvládne fajly prý až do 49033 bajtů, depacker je dlouhý jen 119 bajtů. Patří k těm nejlepším, packuje jen o pár bajtů hůř než Powerfull Code Decreaser 6.2i, v mém testu rovněž o pár bajtů předčil DSQ. Nejneuvěřitelnější je na něm rychlost packování, je sice pomalejší než Hrust, ale i tak je to dělo jen nepatrně horší než Hrust 1.2, vyšel mi dokonce lepší než tejpový Hrust 2.1!

Myth 25467, 26160, 26669, 27243, —
— Heroes 27975, 28427, 28689, 29030, —.



Hrust 1.2 od toho samého autora je záračný packer záračného výkonu – to ovšem nikoho nepřekvapuje. Tím, co nutí i Rusy padat ze židlí, ze stolu či z podlahy, je jeho rychlost. Udává se prý 1 kB za sekundu. Podle Rusů je v 8 případech z deseti lepší než jejich nejoblíbenější Huffmannovský packer MS-Pack, podle mne je lepší vždycky.

Myth 25044, 25739, 26252, 26801, —
— Heroes 27083, 27536, 27840, 28160, 35430.



Hrust 2.1 jsem v BETASHiTové verzi viděl jen v provedení na hromadné packování. To se dělá tak, že jména fajlů, které chcete packovat, napíšete do nějakého textového souboru, ten sejnete a necháte ho načíst do Hrustu. Ten je pak všechny automaticky spackuje. Prý to má sloužit při

packování materiálů pro časopisy (?), bohužel ho to trochu pro tuto krkolomnost v ovládání znevýhodňuje.

Rozšířila se však v Čechách tejpová verze, taky se neovládá nejlépe a někteří ani nechápou jak, stejně si ale zasloužila zahrnout do testu. Packuje taky rychle, ale asi nemá zapnut nejvyšší výkon – prostě nedosahuje účinnosti starších BETASHiToových verzí. K jejím výsledkům jsem připočítal pouze délku rutiny, v praxi by to bylo ještě o pár bajtů víc. DiHalt mi tvrdil, že nejkratší jsou fajly, které nejdříve spackuje Turbo Imploderem a pak teprve Hrustem 2.1, v mém testu se ovšem prokázal pravý opak, tedy že nejlépe packuje Hrust samotný, a to platí skoro univerzálně pro všechny ruské packery. Dávejte si na to tedy při packování pozor!

Myth 25301, 25930, 26446, 27005, —
— Heroes —, 28226, 28133, 28456, 36082.



KSA LZSS Code Cruncher 3.11 je imploder s dvoukilovým oknem a depackerem 151 bajtů dlouhým. Podle Rusů se hodí hlavně na krátké spritové fajly. Někdy packuje lépe než Arc Packer 1.0 či oba LZH packery, ale špatně tráví soubory předpackované implozí. Na DSQ 4.12 se nechytá. Rychlost má ale ucházející.

Myth 28367, 29358, 29344, 30001, 30437. Heroes 30260, 30633, 30596, 30938, 34807.



LZH Packer 1.4 je poněkud horší než Arc Pack 1.0, předshrinkované soubory ale vezme o něco lépe. Depacker má neuvěřitelnou délku 93 bajtů. Osobně ale k jeho používání nevidím obzvláštní důvod.

Myth 28904, 29625, 29373, 29981, 30405. Heroes 30618, 30874, 30446, 30779, 34859.

LZH Packer 2.4 má zřejmě něco pozměněného v práci s oknem, depacker má dlouhý 91 bajtů, a ačkoliv Rusové tvrdí, že verze 1.4 packuje lépe, mě se to podařilo jen v jednom případě, a sice u předshrinkovaného Mythu, jinak verze 2.4 nad předchozí vítězila. U předshrinkovaných fajlů sice předčil Arc Pack 1.0, u jiných ale taky ne. A na DSQ 4.12 se LZHčka už vůbec nehrabou!

Myth 28727, 29434, 29106, 29685, 30383. Heroes 30450, 30880, 30092, 30405, 34838.



MS Pack je ukrajinský 128 only depacker, prý Huffmann. Zvládne kódy do délky 42240, depacker je dlouhý 196 bajtů, ovládání velmi komfortní. Rusové ho mají strašně v oblíbě, i když v mém testu se přibližně vyrovnal DSQ (u nepředpackovaných fajlů byl ale lepší) a byl o něco horší než Hrum a Hrust. No ale i to ho řadí k těm nejlepším. Dlouhé fajly ale packuje poněkud pomaleji. Pokud je to ale opravdu Huffmann, tak je o dost lepší než ten, který byl použit v PackMakerech a Mr Paccích.

Myth 25812, 26600, 27442, 28038, —
— Heroes 28037, 28556, 29222, 29595, —.



Powerfull Code Decreaser 6.2i pochází opět z dílny nepříliš vyzpytatelného KSA,



původní verze byla takřka nepoužitelná, tuto upravil VEKA. A packuje opravdu skvěle, i když pomalu. Dost pomalu. Příšerně pomalu. Neskutečně pomalu. Navíc umožňuje packovat i screeny, i když mu příliš dobře nejdou. Packuje lépe než DSQ 4.12, předčí dokonce i Hrum 3.5 a blíží se svým výkonem oběma Hrustům! Pokud ale navolíte SLOW mód packování, připravte se na to, že při čekání stihnete shlédnout celovečerní film nebo vychovat potomstvo nebo... Což je po tak fantastickém výkonu celkem zklamání.

Myth 25303, 26010, 26640, 27208, —
—, Heroes 27708, 28153, 28564, 28895,
—.

Top Ten:

Nahlížím do svých tabulek a vidím, že na celé čáře vyhrál starý Hrust 1.2. Powerfull Code Decreaser sice nedosahuje špatných výsledků, ale pakuje tak pomalu... Hrust 2.1 pro tape, rozšířený u nás, své BE-TASHITové sourozence nepředčil. Další jsou se skvělými výsledky Hrum 3.5, velmi komfortní MS Pack a DSQ 4.12. Následuje skupina, která dává podobně vyrovnané výsledky - KSA LZSS, překvapivě ArcPack, LZH 2.4, ASC LZPAC a LZH 1.4.

Z českých dopadl nejlépe PK Lite a Turbo Imploder, nepředčil však žádný ruský. Pak následují Pack Makery, o pár bajtů horší byly MrPacky, prehistorický All Packer a o titul nejhoršího packeru se ucházel PK HUFF.

No a dál už nic. Slyšel jsem sice o packeru TRUSH 3.862, ale nemůžu ho nikde najít, tak má smůlu.

Budiž nám útechou, že Zilog svůj packer snad i někdy dokončí a pak budeme mít ten nejlepší packer na světě (podle Ziloga už teď packuje lépe než Hrust). Trochu mi nejde do hlavy, proč ruské packery nesnáší kombinace s packery jinými, ale nezbyvá než se s tím prostě smířit.

Nyní si nadělme tabulku, zahrnu do ní pro jednoduchost jen ruské packery. Všechno uvedu radši v absolutních číslech a ne v procentech, snad vám to bude k něčemu dobré.

ZILOGovy packery

+GAMA

Doba pokročila a šuška zapracovala, prostě dnes už každý ví, že známý, populární, oblíbený a vůbec skvělý programátor Zilog čili Stupidius připravuje něco super.

O jeho superpackerech šla nejedna fáma, nejprve se na jednom slovenském serveru objevily jakési TrashPackery Alfa, Beta a Gama, což byly nejpřehistoričtější Zilogovy kódy, snad použité v jeho programech (MC PIX). Byly silně platformově závislé (byly psány pro Didaktik M, na jiných strojích jevívaly tendenci ke zhroucení), podivné a nepříliš výkonné. Pro "normálního" uživatele, který nevěděl přesně, o co v tom kódu jde a na jaký druh dat lze použít, v podstatě zcela bez užítku.

Všem brzy došlo, že tohle je jen taková nechtěná předehra, že tohle ještě není ono a že existuje ještě něco mnohem většího a lepšího.

To něco byl packer, který zahrnoval tři metody (viz ZXM 2/99), a sice RLE, implozi a Huffmanna. Byl tak optimalizován, že porážel ruské packery (dokonce i Hrust), protože zatímco Rusové jsou empirici a prasa, prostě zkouší, co se asi tak dá naprogramovat, Zilog šel na to nejprve s matematickým aparátem, už první vzniklý kód tak měl velký potenciál a pak doladováním se to dalo dotáhnout až do extrému.

Těsně před dokončením tohoto packeru Zilog oznámil, že objevil (nebo našel) nový geniální způsob, jak packovat, nezávislý na tom, co již měl hotovo. Práci na TrashPacku zastavil a nabídl zdroják volně k dispozici.

Nový packer, na kterém pracuje, budou v podstatě jen vlastní výkonné rutiny, na user interface není čas a prostor. Výsledek

Zilog nejspíš nabídne lidem, kteří si s tím již poradí a budou schopni přijít na to, jak tyto rutiny použít např. při přípravě dem (a ostatním uživatelům nezbyde než čekat, že někdo z těchto vyvolených nějaké uživatelské prostředí vytvoří a "hotový" packer pusťte mezi běžný lid.

V jakém stavu TrashPack je, co umí a co zajímavého je ve zdrojáku?

Ve zdrojáku je hlavní část packeru, překládá se na adresu 25 000. Šlo by to sice zrelokovat jinač, problém je ten, že k packeru patří ještě jeden kód, ke kterému Zilog již asi zdroják nemá, ten začíná na 30 000 a obsahuje pakované fonty, texty a matematický procesor (protože matematické operace, které packer provádí, hlavně různé testy, by byly v kódu Z80 příliš dlouhé, napsal si Zilog tento interpret zvláštního matematického jazyka). A tady je kámen úrazu - díky tomuto kódu lze pakovat jedině asi od adresy 32 000 do konce paměti, což není přesně to, co uživatel potřebuje, ale přelokovat tento kód už asi nepůjde. Jinak ovšem packer šlape a packuje, a packuje dobře (to jste ještě neviděli).

Zdroják (psáno v MRS) obsahuje kromě návěstí, která jsou většinou asi tříznaková (zkuste z nich pochopit význam), spoustu kódu a zajímavých rutin. Inteligentní LDIR/LDDR, inteligentní swappery částí paměti, superkrátké rutiny, vlastní packovací rutiny (ještě se v nich vyznat, vypadají optimalizované na nečitelnost) a spousta jiných náležitostí, které ani nebudu popisovat, ale jsou taky zajímavé.

Doufejme, že Zilog brzy korunuje svou práci úspěchem a výsledek se dostaví.

Jméno	normal	shrink	implod	shrink+implod	Packmaker	normal	shrink	implod	shrink+implod	Packmaker
Hrust 1.2	25044	25739	26252	26801	0	27083	27536	27840	28160	35430
PCD 6.2i	25303	26010	26640	27208	0	27708	28153	28564	28895	0
Hrust 2.1 Tape	25301	25930	26446	27006	0	0	28226	28133	28456	35082
Hrum 3.5	25467	26160	26669	27243	0	27975	28427	28689	29030	0
MS Pack	25812	26600	27442	28038	0	28037	28556	29222	29595	0
DSQ 4.12	26200	26908	27535	28006	0	28052	28469	28793	29101	35971
KSA LZSS CC 3.11	28367	29358	29344	30001	30437	30260	30633	30596	30938	34807
Arc Pack 1.0	28704	29485	28997	29610	30336	30402	30918	30020	30337	34789
LZH Packer 2.4	28727	29434	29106	29685	30383	30450	30880	30092	30405	34838
ASC LZPAC	28754	29622	29165	29749	30451	30532	31048	30155	30462	34905
LZH Packer 1.4	28904	29625	29373	29981	30405	30618	30874	30446	30779	34859

HiSoft C 1.1

+GAMA

aneb perla z dob, kdy Spectrum bylo ještě mladé

Na úvod o Cčku

Něco o tomto programu už vyšlo v YS 11/99, proto se nezlobte, když se budu v něčem možná opakovat.

Existuje mezi lidmi představa, že spectrácké Cčko musí být nutně neplnohodnotné, že to není ono, že na PeCi je to mnohem lepší. Nemusí to být pravda. Abych vás přesvědčil, trochu ho popíšu. A začnu možná nelogicky právě jeho nedostatky.

Nedostatek číslo jedna - spectristé o Cčko nejevili valný zájem, takže se jeho původce HiSoft přestal starat o další vývoj a ačkoliv Cčko vzniklo už roku 1984, máme od něj pouhé dvě verze: 1.0 a 1.1, které se od sebe navíc liší jen v jakýchsi vychytilých nedostacích.

HiSofti se tak vůbec nedostali k tomu, dotáhnout některé plánované věci do konce, a tak se musíme spokojit s prozatímní situací: typy short, int a long jsou jedno a to samé 16ti bitové číslo, typy float a double nejsou implementovány. A to navzdory tomu, že hned do následující verze se sliboval short 8bitový, int 16bitový a long 32bitový a uvažovalo se i o implementaci plovoucí desetinné čárky. Ze zcela pochopitelných důvodů se proměnné, definované jako registry, zpracovávají jako automatické (prostě Z80 nemá tolik registrů a na funkci to stejně skoro nic nemění), a taková drobnost - místní proměnné se deklarují na začátku funkčního bloku a ne vnořených bloků. Toto poslední opatření má přispět k přehlednosti programu, ale většina dialektů Cčka to nedodržuje, při konverzi z jiných počítačů tedy budete asi muset ve zdrojáku hýbat s deklaracemi proměnných.

Druhá věc, kterou já osobně vidím jako nevýhodu, je přišerný editor, stejný jako editor HiSoft Pascalu, GENSu, inspirovaný řádkovými editory běžnými pod CP/M (podobný editor měl i EditAs, programátoři na tuhle hrůzu byli svého času zvyklí právě z CP/M).

Pochopitelně existuje možnost importu textu z jiného editoru nebo šance pro šikovné programátory nahradit stávající editor lepším.

Někoho by možná děsilo, že Cčko na UNIXu pracuje se soubory sekvenčním způsobem a to Spectrácké to nebude umět (asi nechte Your Spectrum). Umí, původní verze pracovala s kazetou i microdrivem. Sekvenční soubor na micro-

drivu měl ovšem v hlavičce uvedenou délku 0 a soubor na kazetě byl rozsekán na 514 bytové sektory (2 bajty číslo sektoru, 512 bajtů data). Verze, kterou si upravili Rusové pro BETASHiT, používá soubory k nerozeznání od obyčejného bloku CODE. Vzhledem k tomu, že I/O rutina je jen jedna a je používána editorem, kompilátorem i přeloženým programem, nabízí se jednoduchá možnost, jak Cčko zpřístupnit i pro jiné systémy (D40/80).

Byli to rovněž Rusové, kdo se nespokojil s absencí reálných čísel a výsledkem byla krátká a jednoduchá knihovna, kterou tu taky někde najdete. Podobně existuje knihovna pro práci s pamětí Spectra 128 a jiné mnohé krásné. Dokonce i mnohé standardní funkce a konstanty dodávali sami HiSoftáci k Cčku jako knihovnu (tady někde je), například jednoduchou 32bitovou aritmetiku.

Nakonec zajímavost - C pro Spectrum bylo napsáno v Cčku (BDS C od L. Zolmana z BD Software). Po přeložení bylo disassemblováno a ručně v assembleru optimalizováno na krátkost i rychlost, stále ještě v něm zůstává několik původních funkcí BDS Cčka. Je to krásná ukázka toho, že v Cčku je možné napsat skutečně všechno...

No a to by měl být konec plků, teď vás jistě zajímá, jak to Cčko vlastně vypadá z



hlediska programátora. Pokud si ale myslíte, že se přečtením tohoto článku naučíte v Cčku programovat, to jste si radši měli přečíst nějakou pohádku...

Něco o řádkovém editoru

V Cčku se obvykle píše malými písmeny, ENTER dává kódy 13, 10, jako konec řádky se bere kód 10, kód 13 se ignoruje (myslete na to při konverzi textu z jiných

editorů). Symbol shift s klávesou dává kód i těch symbolů, které normálně potřebují extend mód, například složené závorky. <=, <>, >= mají nové kódy 29, 31, 30 a dají se jako <=, != a >= použít normálně v programu. Symbol shift+I dává kód EOF (-1, 255, zobrazuje se jako chr\$ (137)). Protože kód 10 je konec řádku, dává caps shift+6 hodnotu 16.

Program se nemusí psát jen v editoru. Kratší věci můžete naťukat rovnou do kompilátoru, přeložit a spustit (trochu to připomíná okamžité provádění povelu v Basicu), je-li vám to málo, můžete se přepnout do módu přímého provádění a pak se rovnou provede každý napsaný řádek.

Z kompilátoru se do editoru vstupuje stiskem Edit a pak Enter. Při chybě při kompilaci se dá také stisknout Edit a chybná řádka se rovnou vyvedituje.

Řádky jsou sice číslovány, ale jen pro interní potřebu editoru, očíslovávají se při čtení souboru z média. Nejlepší by bylo, kdyby někdo napsal nějaký šikovný full screen editor, kde by se nějakým číslováním a krkolomnými způsoby editace nemusel programátor vůbec zabývat (neříkal jsem už, že je ten editor fakt debilní?).

Povely pro editor se, jako jinde u HiSoftů, zadávají ve formátu C N1,N2,S1,S2 Enter, kde C je příkaz, N čísla a S stringy. Kromě povelu D a N nejsou parametry povinné.

I m,n - vkládání textu od řádku m,

krok n. Píšete a odentrováváte jednotlivé řádky, když vás to přestane bavit, stisknete Edit a Enter.

L m,n - list od řádku m do n (nebo všeho, bez parametrů).

K n - po kolika řádcích má listing čekat se scrollováním.

W m,n - listing textu na tiskárnu, lze ho breaknout.

V - zobrazí status: oddělovač, N1, N2, S1, S2, počáteční a koncovou adresu textu.

S,,d - nastaví oddělovač parametrů pro editor, nesmí to být mezera.

C - návrat do kompilátoru.

B - návrat do basicu, zpátky USR 25200.

D m,n - smaž řádky m až n.

M m,n - řádek m přesuň na novou pozici n.

N m,n - přečísluj řádky s počátkem m a krokem n.

P m,n,s - uloží text od řádku m do n (nebo celý, bez parametrů).

G,,s - na konec případného textu přihraje další, řádky čísluje s krokem 10.

Jméno se zadává ve formátu NAME pro kazetu a 1:NAME pro microdrive (1 je číslo drajvu). V BETASHITové verzi se postupuje podobně, výstup na kazetu je tam ale zcela zrušen. Čísla 1, 2, 3, 4 odpovídají mechanikám A, B, C, D. Při otevření existujícího souboru pro zápis se nejdříve starý obsah smaže (to v tomto Cčku platí při jakékoliv práci se souborem).

F m,n,f,s - v rozsahu m až n hleděj řetězec f. Při nalezení přepne do editačního módu, kde krom jiného umožňuje řetězec nahradit řetězcem s, případně hledat další výskyt.

E n - editovat řádek n. Editace ale není normální editace - i při ní se používají povel:

šipka vpravo a vlevo - posun kurzoru vpravo a vlevo.

ENTER - konec editace, ponechání změn.

Q - konec editace, zrušení všech změn.

R - obnovení řádku tak, jak byl před editací.

L - výpis celého řádku (normálně část za kurzorem není vidět).

K - smazání znaku za kurzorem (normálně není vidět).

Z - smazání do konce řádku.

F - hleděj další řetězec definovaný povel editoru F.

S - nahraď řetězec řetězcem, oba definované povel editoru F.

X - připsování dalšího textu na konec řádku (přejde do módu Insert).

I - Insert, kurzor se mění na ,*‘, až do stisku Enter vkládá znaky.

C - kurzor se mění na ,+‘, až do stisku Enter přepisuje původní znaky novými.

Povely kompilátoru

Kromě nedostatků, které jsem zmínil v úvodu, neliší se HiSoft C 1.1 od definice a je tak kompatibilní s valnou většinou dialektů, nemá zabudovány žádné anachronismy, umí základní preprocesorové povely, podmíněnou kompilaci ovšem nezahrnuje jako takovou, místo toho má příkaz pro prohledávání knihovny.

Uvedu teď povely preprocesoru (můžou se vyskytovat v zdrojovém textu nebo je můžete zadávat rovnou do kompilátoru).

#define identifikátor makro

- definuje symbolická jména nebo konstanty, když ve zdrojáku najde váš nový identifikátor, chová se k němu jako k tomu, co jste zadali jako řetězec (je to takový token, umožňuje přejmenovávat funkce nebo měnit prostředí Cčka k obrazu svému). Například tyto řádky:

```
#define EOF -1
```

```
#define begin {
```

```
#define end ;}
```

#include

- příslušný řádek se nahradí textovým souborem. Bez parametru vkládá do kom-

pilátoru text z editoru. Do svého zdrojového textu můžete zahrnout i jiné soubory těmito povely:

```
#include filename
```

```
#include „filename“
```

```
#include <filename>
```

Navíc je implementováno tzv. prohledávání knihovny, je to náhrada podmíněné kompilace. Text se nekompile celý, ale z definované knihovny se berou jen ty funkce, na které se v předchozím textu odkazovalo. Proto se prohledávání knihovny umisťuje až na konec zdrojáku. **#include** může být jen v hlavním zdrojáku, ne v inkudovaném textu. Prohledávání se zapisuje takto:

```
#include ?filename?
```

#list

potlačit nebo zase povolit listing při kompilaci umožňují příkazy **#list+** a **#list-**.

#direct

při psaní textu přímo do kompilátoru se obvykle více řádků dá zkompileovat. Pokud chcete přímě vykonání každého samostatného řádku, zadejte povel **#direct+**, návrat k původnímu způsobu práce je **#direct-**. Direct mód umožňuje testovat části programu, měnit proměnné, vyvolávat funkce a spouštět smyčky...

#error

tento povel smaže chybová hlášení. Chyby se pak oznamují jen číslem bez slovního popisu, ale uvolní se tím paměť - někdy je tento povel vítanou záchranou.

#translate filename

- zadává se na začátku programu.

Překlad pak vyprodukuje spustitelný samostatný stroják, začínající na adrese 25200. Filename je jméno, pod kterým se má přeložený kód uložit.

Znak EOF se do kompilátoru vkládá stiskem SS+I, po jeho vložení se ukončí kompilace.

O Céčku

HiSoft C rozeznává 6 tříd symbolů: Identifikátory, klíčová slova, konstanty, řetězce, operátory a oddělovače. Poznámky se oddělují znaky ,/*‘ a ,*/‘, nesmí být vnořené, ale mohou obsahovat i konce řádků.

Identifikátor je posloupnost písmen a číslic, začínají písmenem, rozlišují se písmena malá a velká.

Klíčová slova se píší zásadně malými písmeny. Jsou to:

int, extern, else, char, register, for, float, typedef, do, double, static, while, struct, goto, switch, union, return, case, long, sizeof, default, short, break, continue, unsigned, auto, if.

Konstanty jsou integer, long, znakové a řetězce.

Integer konstanta dekadická je -32768

až 32767. Oktálová začíná nulou, hexadecimální začíná ,0x‘ nebo ,0X‘. Končí-li znakem ,l‘ nebo ,L‘, chápe se jako long. Znakové konstanty se píší do apostrofů, třeba ,X‘. Kromě toho existují tyto způsoby zápisu znaků:

```
\n - LF (nový řádek)
```

```
\t - tabulátor
```

```
\b - backspace
```

```
\r - CR (enter)
```

```
\f - FF (nová stránka)
```

```
\\ - zpětné lomítko
```

```
\‘ - apostrof
```

\DDD - DDD je oktálové číslo, udávající kód znaku.

Konstanty řetězcové se uzavírají do uvozovek, kompilátor je ukončuje nulou. Pozor, ,X‘ a ,X“ není totéž! Jedno je char, druhé string („pole charů“).

Konstanty s plovoucí čárkou by se chápaly jako double, ale nejsou implementovány.

Datové typy jsou char, int, float, double, celá čísla jsou ještě short, long a unsigned. HiSoft C short, int a long zpracovává všechny stejně, float a double nejsou implementovány.

Paměťové třídy jsou auto, static, extern, register a typedef..

Odvozené konstrukce jsou array (pole), funkce, pointer (ukazatel na adresu), struct (struktura), union (může obsahovat jakýkoliv z několika objektů).

Lvalue (L-hodnota) je výraz vztažený k objektu. Vysvětlím to na příkladu - máme ukazatel E na hodnotu 12345. *E je potom lvalue, třeba obsah paměti na adrese 12345. Typickou lvalue je třeba identifikátor víc napíšu u výrazů.

Jen krátce o konverzi typů: char na int - hodnota znaku. Pointer a int - lze sčítat i odčítat inty a pointery. Pointer - pointer=int. Int na unsigned. Konverze v aritmetice má jednoduchá pravidla - char nebo short se mění na int, je-li jeden operand unsigned, vše ostatní se mění taky na unsigned, nižší typ se převádí na vyšší, při přiřazení se hodnota vpravo konvertuje na typ hodnoty vlevo. Od nejvyšších k nižším jsou typy řazeny takto: Double, float, long, unsigned, int, short, char.

Výrazy

Unární operátory jsou tyto:

*výraz - pointer, výsledek je lvalue.

&lvalue - výsledek je pointer. Jde použít jen na proměnné a elementy pole.

-výraz - unární mínus.

!výraz - log. negace (1=nulový, 0=nulový).

~výraz - jednotkový komplement.

++lvalue - objekt se inkrementuje před vyhodnocením.

lvalue++ - objekt se inkrementuje po vyhodnocení.

--lvalue - objekt se dekrementuje před vyhodnocením.

lvalue-- - objekt se dekrementuje po vy-

hodnocení.

cast(typ)výraz - konverze na daný typ. V jiných mutacích Cčka se slovo cast vynechává, proto při převodu programu z HiSoft C na tyto mutace musíte v úvodu uvést #define cast'.

sizeof výraz - vrací velikost v bytech.
sizeof(typ) - vrací velikost v bytech.

Multiplikativní operátory jsou tyto:

výraz*výraz - násobení.

výraz/výraz - dělení.

výraz%výraz - zbytek po dělení.

Aditivní operátory:

výraz+výraz - součet; pointer na objekt pole plus něco je pointer na jiný objekt pole.

výraz-výraz - rozdíl, viz sčítání,
pointer-pointer=int (vzdálenost mezi poli).

Shift operátory:

výraz<<výraz - rotuje první výraz doleva o počet bitů daný druhým výrazem.

výraz>>výraz - rotuje první výraz doprava o počet bitů daný druhým výrazem.

Relační operátory:

výraz<výraz

výraz>výraz

výraz<=výraz

výraz>=výraz - všechny dávají 0 při FALSE, 1 při TRUE.

Srovnávací operátory:

výraz==výraz

výraz!=výraz - narozdíl od relačních mají nižší prioritu při vyhodnocování.

Bitové, logické a podmínkové operátory:

výraz&výraz - bitový and.

výraz&&výraz - logický and (1 jsou-li oba nenulové, jinak 0).

výraz~výraz - bitový xor.

výraz\výraz - bitový or.

výraz\\výraz - logický or.

výraz?výraz:výraz - je-li první výraz nenulový, je výsledkem druhý, jinak třetí výraz.

Přiřazovací operátory:

lvalue=výraz

lvalue+=výraz - je to jako lvalue=lvalue+výraz.

lvalue-=výraz

lvalue*=výraz

lvalue/=výraz

lvalue%=výraz

lvalue>>=výraz

lvalue<<=výraz

lvalue&=výraz

lvalue^=výraz

lvalue\=výraz

Primární výrazy jsou:

identifikátor - pole dává pointer na první objekt, ne lvalue; funkce dává pointer na funkci.

konstanta

řetězec

(výraz) - typ a hodnota se bere z výrazu v závorkách, u lvalue jsou zbytečné.

výraz[index]

funkce (parametr,parametr,...)

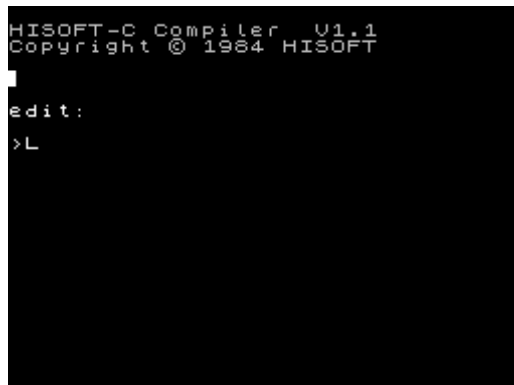
lvalue.identifikátor - lvalue je jméno struktury nebo unionu, výsledek je lvalue ukazující na jméno položky structu či unionu.

pointer>identifikátor - pointer je na struct nebo union, identifikátor jméno jeho člena.

Deklarace

Deklarace upřesňují interpretaci identifikátoru, dávají se před specifikátory typů. Zapisuje se jako:

specifikátor_třidy specifikátor_typu spe-



cifikátory_deklarace

Třidy jsou: auto, static, extern, register, typedef. Specifikátor třídy není povinný, potom se bere jako auto uvnitř funkce a jako extern vně funkce.

auto - proměnné lokální, při výstupu z bloku se uvolňují.

static - lokální, trvají stále a mohou zase do bloku vstupovat.

extern - trvají stále a mohou komunikovat mezi funkcemi. Pokud se povelem pro prohledání knihovny přilepují k programu funkce, které nevracejí int, musí být před voláním těchto funkcí (tedy na začátku programu) definován jejich typ, jinak budou přeloženy jako funkce vracející int.

register - programátor chce, aby pro tuto proměnnou byl vyčleněn přímo registr procesoru. Z80 ale jeho přání vytrvale ignoruje a proměnná se zpracovává jako automatická.

typedef - nevymezuje paměť pro proměnnou, ale definuje později použitelné identifikátory, které se stávají součástí deklarátoru.

Typy jsou: char, short, int, long, unsigned, float, double, typedef jméno, specifikátor struktury nebo unionu. Nezapomeňme, že short, int a long se chovají všechny jako int a float a double nejsou implementovány.

Seznamy deklarátorů se oddělují čárkou. Deklarátory se píší takhle:

identifikátor

(deklarátor)

*deklarátor

deklarátor()

deklarátor[výraz]

Například:

typ *D - pointer na typ

typ D() - funkce vracející typ; pozor, funkce nemohou vracet pole, struktury, uniony nebo jiné funkce, ale mohou

vracet pointery na ně; podobně neexistují pole funkcí, ale mohou být pole pointerů na funkce, třeba

```
int i,*ip,f(),*fip(),(*pfi)();
```

deklaruje int i, pointer ip na int, funkci f vracející int, funkci fip vracející pointer na int a pointer pfi na funkci, která vrací celé číslo. S funkcí lze provádět jen její vyvolání nebo přiřazení její adresy, pokud není jméno funkce v pozici volání, generuje se pointer, třeba v programu

```
int f();
```

```
g(f)
```

se definice funkce g chápe jako

```
g(*fp)();
```

```
int (*fp)();
```

```
{{*fp}();
```

, tedy f jako argument funkce g se chápe jako pointer na funkci f.

typ D[] - pole typů, je povoleno i víc specifikací vedle sebe, nejpraktičtější je to u vícerozměrného pole,

```
static int x[3][2][5];
```

deklaruje pole x jako statické pole intů velikosti 3x2x5 (třírozměrné pole dvourozměrných polí pětirozměrných polí intů), ale třeba také

```
int a[17],*ap[17];
```

deklaruje pole int čísel a pole pointerů na int čísla,

```
int *pa;
```

deklaruje pointer na int, pak

```
pa=&a[0];
```

nastaví pa na pointer elementu pole a[0], tedy pa obsahuje adresu a[0], navíc x=*pa

kopíruje obsah a[0] do x. Protože jméno pole je vlastně pointer na začátek pole (a nikoliv lvalue!), lze pa=&a[a] psát i jako pa=a, a[i] je vlastně *(a+i) a výrazy &a[i] a a+i jsou identické!

Struktury a uniony:

Struktura je posloupnost jmenovaných členů, z nich každý může být jakéhokoliv typu.

Union může obsahovat jakýkoli jeden nebo více z několika svých deklarovaných členů.

Struktury se nesmí jmenovat stejně jako jiné konstrukce (třeba stejně jako některý int). Struktury (i uniony) nesmí obsahovat sebe samu, ale mohou obsahovat pointer na sebe. Se strukturou se pracuje jen pomocí operátoru & a jejích členů, nemůže být přiřazována ani kopírována jako celek ani předávána funkcím či vracena funkcemi, funkcím lze předávat jen člen struktury nebo pointer na strukturu. Struktury jsou jen external nebo static, ne auto.

HiSoft C v současné verzi neumí bitová pole (fields) a vzhledem k tomu, že HiSoft již Spectrum opustil, nikdy je umět nebude. Lze to obejít bitovým posunem a bitovými operacemi.

Několik příkladů:

```
struct tnode {
```

```
char tword[20];
```

```
int count;
```

```
struct tnode *left;
struct tnode *right;
}s,*sp;
```

Struktura tnode obsahuje pole dvaceti charů, int a dva pointery na sebe sama, s je deklarace téže struktury (při opakování deklarace se nemusí znovu psát dlouhá část definice), *sp je pointer na strukturu tnode. Pak sp->count se vztahuje ke členu count struktury na kterou ukazuje sp, s.left se vztahuje k ukazateli na levý podstrom struktury s, s.right->tword[0] se vztahuje k prvnímu znaku členu tword pravého podstromu s.

Jiný příklad:

```
struct {
    int x;
    int *y;
} *p;
```

++p>x inkrementuje x a ne p, protože implicitně jsou priority brány jako ++(p>x); (++p)>x inkrementuje p před přístupem k x a (p++)>x inkrementuje p po přístupu k x.

Ještě něco jiného:

```
struct key{
    char *hexword;
    int keycount;
}keytab[nkeys];
```

přístup k členům pole je přes pointery, je to pole struktur typu key, každý prvek pole obsahuje pointer na char a int.

Se strukturou i unionem mohou být v podstatě prováděny jen dvě operace: jmenování jednoho z členů pomocí „.“ nebo „->“ a přiřazení adresy operátorem „&“. U „.“ a „->“ je vpravo člen struktury a vlevo její jméno (před „.“ lvalue, před „->“ pointer nebo int).

Inicializace:

Deklarátor může definovat počáteční hodnotu, platí, že inicializátor nebo jejich seznam = výrazu nebo seznamu inicializátorů (to v případě inicializace pole nebo struktury, nevyjmenované členy jsou zaplácny nulami).

Externí definice:

Definují se jako všechno ostatní, přidává se adjektivum extern. Externí proměnné žijí po celou dobu trvání programu, definovat se dají i externí funkce. Zás příklad:

```
int max(a,b,c)
int a,b,c
{int m;
m=(a>b)?a:b;
return((m>c)?m:c);
}
```

int specifikuje typ, max(a,b,c) je deklarátor funkce, int a,b,c je seznam deklarací pro formální parametry, {...} je blok příkazů. Co tato funkce vrátí, je snad jasné - největší číslo ze tří do funkce vstupujících. Všimněte si, že v hlavičce standardní knihovny je funkce max napsána jinak - tam totiž není počet argumentů znám, jejich počet vstupuje jako další argument díky povelu „auto“ mezi deklarátorem funkce „f(...“ a „int arg;{...}“. Doporučuji nahlédnout do hlavičky standardní knihovny, jak přesně je to uděláno, vám do funkce přijde jako jediný parametr lva-

lue délka argumentů v bajtech a vy si zjistíte jak jejich počet (vydělíte si je délkou jednoho argumentu), tak pointer na ně (pomocí &lvalue).

Příkazy Cčka

Výrazový příkaz se píše jako výraz;

a slouží k přiřazení nebo volání funkce. Složený příkaz je sekvence příkazů (složených nebo výrazových) ve složených závorkách.

```
if (výraz) příkaz1;
if (výraz) příkaz1; else příkaz2;
```

je to podmíněný příkaz skoro stejný jako v basicu.

```
while (výraz) příkaz;
testuje před provedením příkazu, při nenulovém výrazu příkaz cyklicky provádí.
```

```
do příkaz while (výraz)
provádí cyklicky příkaz, po provedení testuje a opakuje, dokud je nenulový.
```

```
for (výraz1;výraz2;výraz3) příkaz;
výraz1 je inicializace, výraz2 je test, výraz3 je inkrementace. Kterýkoliv z nich může být vynechán. For je vlastně to samé jako
```

```
výraz1;
while (výraz2){
    příkaz;
    výraz3;
}
```

```
HISOFT-C Compiler V1.1B
1991 TJU B-disk version
```

```
edit:
>L
```

```
switch (výraz) příkaz;
```

příkaz je složený, prefixem „case“ výraz se přiděluje příkazům hodnota, nebo se použije prefix „default:“. Podle výsledku výrazu u switch se předá řízení buď té části příkazu, jejíž case odpovídá, nebo defaultní části příkazu. Po vykonání jedné části se porovnávají case u dalších částí příkazu, nechceme-li to, dá se ze switche vyskočit povelu break;

```
break;
```

ukončí cyklus while, do, for nebo switch.

```
continue;
```

ukončí jeden průchod smyčkou while,

do, for a jde na pokračovací část smyčky.

```
return; nebo return výraz;
vrátí se z funkce, výraz udává vrácenou hodnotu funkce.
```

```
goto identifikátor;
skočí na návěští dané identifikátorem.
Návěští se zapisuje jako „identifikátor;“.
```

```
;
je prázdný příkaz, hodí se, pokud chcete třeba prázdné tělo cyklu while nebo posunout návěští před závorku složeného příkazu.
```

```
inline (k1,k2,k3,...);
vkládá bajty strojového kódu, čísla 0-255 jako jeden byte, větší jako dvoubyte.
Například:
```

```
#define CHAN_OPEN 0x1601
#define Id_a 0x3E
#define call 0xCD
inline (Id_a, 3, call, CHAN_OPEN, 0x22, &c);
otevře kanál 3 a obsah registru HL uloží do proměnné c (0x22 je Id (NN),hl).
```

Standardní knihovna funkcí

Některé často používané přídatné funkce má HiSoft C zabudovány a není je tudíž vůbec třeba dotahovat z knihovny.

Ty ostatní jsou zahrnuty v knihovně stdio.h a stdio.l, přičemž stdio.h je jakási hlavička, která definuje některé konstanty a dvě funkce, u nichž není předem znám počet argumentů (max a min) a zařazuje se povelu „#include stdio.h“ na začátek programu, zatímco stdio.l se zařazuje na konec programu pomocí příkazu pro prohledání knihovny „#include ?stdio.l?“ a jsou z ní použity jen ty funkce, na které se programátor v předchozím textu odvolával.

Přestože je zabudována jen část funkcí, stojí určitě za to seznámit se se všemi.

```
int max(n,...) auto
vrací hodnotu největšího argumentu.
```

```
int min(n,...) auto
vrací hodnotu nejmenšího argumentu.
```

```
int abs(n)
vrací absolutní hodnotu.
```

```
int sign(n)
vrací -1, 0 a 1 pro čísla záporná, nulová a kladná.
```

```
char peek(adresa)
vrací obsah bytu na adrese adresa.
```

```
void poke(adresa,hodnota)
napoukaje nejnížší bajt hodnoty na
```

Programování

adresu `adresa`. Typ `void` obvykle znamená, že funkce nevrací žádnou hodnotu, Spectrum `void` sice neumí, ale tady to vyřeší jeden řádek `#define void int` a pak Ččko sežere i programy psané obvyklým způsobem, `void` je takto definován v `stdio.h`.

`int atoi(s)`
převéde řetězec na číslo, mezery, tabulace a konce řádků ignoruje, umí znaménka.

`void qsort(list, num_items, size, cap_func)`
`char *list;`
`int num_items, size;`
`int (*cap_func)();`

Funkce se sice jmenuje Q-sort, ale to proto, že na UNIXu je quicksort součástí knihoven. Tohle je ve skutečnosti Shellův sort, ale kvůli možnosti používat zdrojáky z UNIXu bylo původní jméno zachováno.

Položek je `num_items`, mají velikost `size`, uspořádány jsou za sebou od pointeru `list`, seznam může být i dvourozměrné pole `char list[num_items][size]`. `Cap_func` je pointer na funkci, kterou se porovnají dvě položky v seznamu (třeba `strcmp` pro řetězce), bere dva pointerové argumenty, volá se `(*cap_func)(x,y)`; a vrátí `-1` když `*x < *y`, `0` když `*x == *y` a `1`, když `*x > *y`.

`char *strcat(base, add)`
`char *base, *add;`
fyzicky připojí kopii řetězce `add` na konec řetězce `base`, pozor, aby se něco nepřepsalo.

`int strcmp(s,t)`
`char *s, *t;`
porovná dva řetězce, při shodě vrátí nulu.

`char *strcpy(dest, source)`
`char *dest, *source;`
fyzicky zkopíruje řetězec `source` do `dest`, pozor, aby se něco nepřepsalo.

`unsigned strlen(s)`
`char *s;`
vrátí délku řetězce.

`int isalnum(c)`
`char c;`
vrátí `1`, jde-li o písmeno nebo číslo, jinak `0`.

`int isalpha(c)`
`char c;`
vrátí `1`, jde-li o písmeno.
Zabudovaná funkce.

`int isascii(c)`
`char c;`
vrátí `1`, jde-li o ASCII znak, přesněji řečeno jestli je menší než `128`...

`int iscntrl(c)`
`char c;`

vrátí `jedna`, jestliže je znak menší než mezera nebo roven `127` (copyright).

`int isdigit(c)`
`char c;`
vrátí `1` pro `'0'` až `'9'`.
Zabudovaná funkce.

`int isower(c)`
`char c;`
vrátí `1` pro malá písmena.
Zabudovaná funkce.

`int isprint(c)`
`char c;`
vrátí `1` u tisknutelných znaků (mezera až `126`).

`int ispunct(c)`
`char c;`
vrátí `1` u těch tisknutelných znaků, které nejsou ani číslo, ani písmeno.

`int isspace(c)`
`char c;`
vrátí `1` pro znaky mezera, nový řádek, tabulátor.
Zabudovaná funkce.

`int isupper(c)`
`char c;`
vrátí `1` pro velká písmena.
Zabudovaná funkce.

`char tolower(c)`
`char c;`
konvertuje velké písmeno na malé, jinak znak nemění.
Zabudovaná funkce.

`char toupper(c)`
`char c;`
konvertuje malé písmeno na velké, jinak znak nemění.
Zabudovaná funkce.

`char *calloc(n, size)`
`unsigned n, size;`
přidělí prostor pro `n` položek, každá velikosti `size`. Vrací pointer na začátek oblasti, jinak nulu (jako `'new'` v Packalu).

`void free(block)`
`char *block;`
zase blok uvolní. Předáváte kopii toho pointeru, který vám vrátila `calloc` (jako `'dispose'` v Packalu).

`char *sbrk(n)`
`unsigned n;`
vymezí `n` bajtovou oblast, kterou pak může používat funkce `calloc` pro heap. Na UNIXu systém sám zahýbe RAMěti tak, aby místo uvolnil, na Spectru je údaj o tom, kde místo uvolnit, přímo ve zdrojovém textu funkce. Ta standardní je napsána tak, že heap vytvoří jako static pole charů, tedy trvalou proměnnou uloženou mezi jinými proměnnými. Chcete-li heap jinde, přepište si laskavě sami.

`void swap(p,q,length)`
`char *p,*q;`
`unsigned length;`
prohodí dvě oblasti určené pointerem `p` a `q` o délce `length`, používá ji třeba `qsort`.
Zabudovaná funkce.

`void move(dest,source,length)`
`char *dest,*source;`
`unsigned length;`
něco jako inteligentní `ldir`, přesouvá oblast od pointeru `source` do pointeru `dest` o délce `length` a pracuje korektně i tehdy, když se obě oblasti překrývají.
Zabudovaná funkce.

`FILE *fopen(name,mode)`
`char *name,*mode;`
pozor, `name` i `mode` jsou pointerem na stringy, ne chary (!), a to navzdory tomu, že `mode` je jednoznakový string! `Mode` může být `'r'` pro čtení a `'w'` pro zápis, `name` je jméno souboru, který se má otevřít, pokud otevíráte pro zápis již existující soubor, vymaže se jeho obsah.

Typ proměnné `FILE` je ve skutečnosti typedef `int` a je to ukazatel, který funkce vrátí pro potřebu jiných funkcí - potvrzuje typ souboru (zápis, čtení atd), při chybě je nulový.

Zabudovaná funkce.

`int fclose(fp)`
`FILE *fp;`
uzavře soubor označený pointerem `fp`. Po uzavření souboru se uvolní buffer, který systém pro vstup a výstup používal.
Zabudovaná funkce.

`int getc(fp)`
`FILE *fp;`
čte „znak“ ze souboru označeného pointerem `fp`. Ve skutečnosti nevrací `char`, ale `int`, protože potřebuje rozlišit kód `255` od `-1` (EOF, end of file).
Zabudovaná funkce.

`int ungetc(c,fp)`
`int c;`
`FILE *fp;`
vrátí znak `c` zpátky do souboru označeného pointerem `fp`. Jde to jen s jediným znakem, navíc funkce `ungetc` je použita funkcí `scanf`, takže po `scanf` už nejde `ungetc` použít (jde, ale musíte něco mezitím přečíst pomocí `getc`).
Zabudovaná funkce.

`int putc(c,fp)`
`int c;`
`FILE *fp;`
pošle znak `c` do souboru `fp`, znak taky vrátí jako svůj výsledek.
Zabudovaná funkce.

`int getchar()`
vrátí znak ze standardního vstupu „`stdin`“, na Spectru klávesnice. Vstup se děje přes buffer, text je možno opravovat

i použitím delete, vstup ukončuje enter, takže je to spíš takový basicový input.

Zabudovaná funkce.

```
void exit(n)
uzavře všechny soubory a vyleze z programu. Parametr n vyvolá v nule „chybovou“ hlášku OK, jinak odpovídá číslu chybové hlášky, která se má zobrazit. Pro opuštění programu se dá použít i její „podfunkce“ _exit(n), ale vzhledem k tomu, že ta nechává bordel v souborovém systému, používejte jen exit(n), je to jistější.
```

```
char fgets(s,n,fp)
char *s;
int n;
FILE *fp;
čte řetězec s od souborového pointeru fp. Čtení ukončí buď když s dosáhne délky n, nebo když narazí na konec řádku.
```

```
void fputs(s,fp)
char *s;
FILE *fp;
nechá vystoupit řetězec od souborového pointeru fp.
```

```
char *gets(s)
char *s;
čte řetězec ze standardního „stdin“, tedy klávesnice.
```

```
void puts(s)
char *s;
nechá řetězec vystoupit na „stdout“, tedy obrazovku.
```

```
void printf(control,arg1,arg2,...)
char *control;
formátovaný tisk na „stdout“, tedy obrazovku, podle specifikace control. Řetězec control se normálně vytiskne tak, jak je, kromě konverzních údajů, které se uvozují znakem ‚%‘ a udávají, jak se mají vytisknout další argumenty. Znak ‚%‘ je možno normálně vytisknout zadáním ‚%%‘. A teď ty kon(tro)verzní údaje:
```

```
%d - desítkové číslo
%o - oktálové číslo bez znaménka
%x - hexadecimální číslo bez znaménka
%n - desítkové číslo bez znaménka
%c - jeden znak
%s - řetězec ukončený znakem „\0“
%- - zarovnání vlevo (jinak se zarovnává doprava)
%0 - doplnit nulami místo mezer
%999 - (řetězec cifer) určuje minimální rozměr pole
%.999 - maximální počet znaků řetězce, které se vytisknou
%| - „dlouhá data“, tiskne bez omezení
```

Zabudovaná funkce.

```
void fprintf(fp,control,arg1, arg2,...)
FILE *fp;
char *control;
jako printf, ale nikoliv na „stdout“, ale od souborového pointeru fp.
```

Zabudovaná funkce.

```
void sprintf(s,control,arg1,arg2,...)
char *s;
char *control;
jako printf, ale nikoliv na „stdout“, ale do řetězce s. Pozor, abyste si něco nepřepsali.
```

Zabudovaná funkce.

```
int scanf(control,arg1,arg2,...)
char *control;
(!) arg1,arg2... musí být pointery! Je to hodně divoká vstupní funkce, tak pozor: čte znak ze „stdin“, tedy klávesnice, konvertuje podle control a ukládá podle dalších argumentů do paměti.
```

Control obsahuje jednak prázdné znaky (mezera, tabelátor, CR, LF), odpovídající libovolnému počtu prázdných znaků na příslušné pozici (0-nekonečno), běžné znaky, které musí odpovídat následujícímu vstupnímu znaku.

Konverzní specifikace jsou tvořené znakem ‚%‘, případně ‚*‘, který potlačuje přiřazení, případně číslem a konverzním znakem. Konverzní specifikací se určuje, jak se bude interpretovat dané vstupní pole. Výsledek se ukládá do proměnné, na niž ukazuje pointer, argumenty tedy začínají operátorem &. Potlačující znak * způsobí přeskočení daného vstupního pole. Pole je ohraničeno buď prázdným znakem, nebo rozměrem pole (viz %999). Konverzní znaky jsou podobné jako u printf:

```
d - očekává se desítkové číslo, pointer je na int.
o - očekává se oktálové číslo, pointer je na int.
x - očekává se hexadecimální číslo, pointer je na int.
h - obvykle značí short, v HiSoft C je pointer na int.
c - jeden znak, pointer je na char.
s - očekává se string, argumentem je pointer na řetězcové pole (musí být dost velké, aby se tam string vešel i s kódem „\0“).
```

Funkce vrací počet úspěšně přiřazených položek nebo EOF.

```
int n;
char s[20];
scanf(„%d , %19s“,&n,s);
přečte celé číslo do n a řetězec (ne delší jak 19 znaků) do s, při vstupu musí být oba údaje odděleny čárkou. Údaj „%19“ sám ohlídá délku řetězce, argument s se nezadá jako &s, protože pole je vlastně pointerem na svůj začátek! Už to sem píšu poněkolkáté...
```

Funkce scanf ve spectráckém Ččku neodpovídá přesně definici jazyka, je udělané spíš tak, aby odpovídalo této funkci tak, jak je dělána v Ččku pro UNIX.

```
int fscanf(fp,control,arg1,arg2,...)
char *control;
FILE *fp;
jako scanf, ale vstup se neděje ze „stdin“, ale ze souboru podle pointeru fp.
```

```
int sscanf(s,control,arg1,arg2,...)
char *s;
char control;
jako scanf, ale vstup se děje z řetězce s.
```

```
int rawin()
čte znak přímo z klávesnice bez vopíček jako jsou kurzory a echa na obrazovku.
```

Zabudovaná funkce.

```
int keybit()
Po stisknutí klávesy vrací 1 tak dlouho, dokud kód klávesy nepřetete pomocí rawin nebo neshodíte flag v systémových proměnných, oznamující stisknutí klávesy.
```

Zabudovaná funkce.

```
void long_multiply(c,a,b)
char *a,*b,*c;
násobí dvě 32bitová čísla, c=a*b. Čísla jsou uložena jako pole čtyř charů, nejméně významné bity jsou v poli [0], nejvýznamnější v poli[3] a jsou bez znaménka.
```

```
void long_add(c,a,b)
char *a,*b,*c;
sečte dvě 32bitová čísla, c=a+b.
```

```
void long_init(a,n1,n0)
char *a;
unsigned n1,n0;
vytvoří 32bitové číslo. Třeba long_init(a,0x1234,0x5678); vytvoří a=0x12345678.
```

```
void long_set(a,n,d)
char *a;
unsigned n,d;
vytvoří 32bitové číslo. Číslo n je 16 bitů a číslo d udává, kam se mají umístit. Třeba long_set(a,0x1234,1); vytvoří a=0x00123400.
```

```
void long_copy(c,a)
char *a,*c;
provede kopii 32bitového čísla, přiřadí c=a.
```

```
int rand()
vrací 16bitové náhodné číslo (int).
```

```
void srand(n)
n je int, funkce nastavuje seeds generátor pro náhodná čísla.
```

```
void plot(on,x,y)
int on,x,y;
bod na souřadnicích x a y zobrazí nebo smaže podle toho, je-li on nulové čili nic.
```

```
void line(on,dx,dy)
int on,dx,dy;
kreslí nebo maže čáru od poslední pozice do souřadnic dx a dy. Parametr on funguje stejně jako u plotu.
```

```
int ink(color)
```


32 bad type combination
 33 bad operand type
 34 need an lvalue
 35 not a defined member of a structure
 36 expecting a primary here
 37 undefined variable
 38 need a type name
 39 need a constant expression
 40 can only call functions
 41 : does not follow a ? properly
 42 Destination of an assignment must be an lvalue
 43 need a : to follow a ? - check bracketing
 44 need a pointer
 45 illegal parameter type
 46 RESTRICTION: Floating Point not implemented
 47 cannot use this operator with float arguments
 48 bad declaration
 49 storage class not valid in this context
 50 RESTRICTION: can't manage initializers yet
 51 duplicate declaration of structure tag
 52 use a predeclared structure for parameters
 53 structure cannot contain itself
 54 bad declarator
 55 missing , in function declaration
 56 bad formal parameter list
 57 type should be function
 58 generated code area is full
 59 need an lvalue
 60 LIMIT: no more memory
 61 RESTRICTION: use assignment or move() to initialise
 automatics
 62 Cannot initialise this (disallowed storage class)
 63 Cannot initialise this (disallowed type)
 64 too much initialisation data
 65 wrong file, find another header

Reálná čísla

Jak už víte, HiSoft přestal po verzi 1.1 své C dále vyvíjet a tak zůstala spousta věcí, které měly být v dalších verzích doplněny, jen fikcí. Už asi nikdy nebude doplněno zpracování typů float a double.

Snad vám ale pomůže tato knihovna pro plovoucí desetinnou čárku, kterou si napsali Rusové. Není dlouhá, ale opisujte pečlivě...

```

/*****
/*
/*      REAL arithmetics      */
/*
/*      last changed      */
/*      03.01.1991      */
/*
/*
*****/
#list-
typedef char real[5],*p_char,bool;
#define maxin 16
#define end_c 0x38
char s[maxin];
bool rezult;
toreal(rl,str)
real *rl;
char *str;
{ char sign;
  sign=*str;
  if(sign=='+' | | sign=='-')
    str++;
    wpoke(0x5b00,rl);
    wpoke(0x5b02,str);

```

```

inline(0x2a,0x5c5d,0xe5,
  0x2a,0x5b02,0x7e,
  0x22,0x5c5d,
  0xcd,0x2c9b,0xe1,
  0x22,0x5c5d);
if(sign=='-')
  inline(0xef,0x1b,end_c);
pop_a();
}div(a,b,c)
real *a,*b,*c;
{p_poke(a,b,c);
  push_bc();
  inline(0xef,5,end_c);
  pop_a();
}mult(a,b,c)
real *a,*b,*c;
{p_poke(a,b,c);
  push_bc();
  inline(0xef,4,end_c);
  pop_a();
}sub(a,b,c)
real *a,*b,*c;
{p_poke(a,b,c);
  push_bc();
  inline(0xef,3,end_c);
  pop_a();
}add(a,b,c)
real *a,*b,*c;
{p_poke(a,b,c);
  push_bc();
  inline(0xef,0xf,end_c);
  pop_a();
}bool equ(a,b) /* a=b */
real *a,*b;
{ wpoke(0x5b02,a);
  wpoke(0x5b04,b);
  push_bc();
  inline(6,0xe,0xef,0xe,end_c,
    0xcd,0x2dd5,
    0x32,&rezult);
  return rezult;
}bool lt(a,b) /* a<b */
real *a,*b;
{ wpoke(0x5b02,a);
  wpoke(0x5b04,b);
  push_bc();
  inline(6,0xd,0xef,0xd,end_c,
    0xcd,0x2dd5,
    0x32,&rezult);
  return rezult;
}bool ge(a,b) /* a>=b */
real *a,*b;
{ wpoke(0x5b02,a);
  wpoke(0x5b04,b);
  push_bc();
  inline(6,0xa,0xef,0xa,end_c,
    0xcd,0x2dd5,
    0x32,&rezult);
  return rezult;
}r_copy(a,b)
real *a,*b;
{ move(a,b,5);
}tostr(s,r)
char *s; real *r;
{wpoke(0x5b00,s);
  wpoke(0x5b02,r);
  push(0x5b02);
  inline(0xef,0x2e,end_c,
    0xcd,0x2bf1,0x2a,0x5b00,

```

```

  0xeb,0xed,0xb0,0xaf,
  0x12);
  tomin();
}scanr(r)
real *r;
{ scanf(„\n%s“,s);
  toreal(r,s);
}tomin()
{inline(0xcd,0x16bf);}
p_poke(a,b,c)
real *a,*b,*c;
{wpoke(0x5b00,a);
  wpoke(0x5b02,b);
  wpoke(0x5b04,c);
}push_bc()
{push(0x5b02);
  push(0x5b04);
}pop_a()
{inline(0xcd,0x2bf1,
  0x2a,0x5b00,0x77,
  0x23,0x73,0x23,
  0x72,0x23,0x71,
  0x23,0x70);
}push(x)
unsigned x;
{wpoke(0x5b10,x);
  inline(0x2a,0x5b10,0x5e,
    0x23,0x56,0xeb,
    0x7e,0x23,0x5e,
    0x23,0x56,0x23,
    0x4e,0x23,0x46,
    0xcd,0x2ab6);
}wpoke(addr,data)
unsigned addr,data;
{poke(addr ,data%256);
  poke(addr+1,data/256);
}poke(addr,val)
{*cast(p_char)adr=val;}
printr(r,lng)
real *r; int lng;
{int i; char es;
  tostr(s,r);
  if(lng)
  {es=1;
    for(i=0;i<lng;i++)
    {if(es) es=s[i];
      fprintf(tek,“%c“,es?s[i]:‘.’);
    }
  }
  else
    fprintf(tek,s);
}#list+
/*****
/*
/*      End Real Arithmetics      */
/*
/*
*****/

```

Příloha: Standardní knihovna funkcí

Následuje „hlavička“ standardní I/O knihovny, obsahuje nejpoužívanější konstanty, definice typů pro knihovnu a funkce „min“ a „max“. Zařazuje se na ZÁČÁTEK programu, který knihovnu používá, povelem #include „stdio.h“. Vlastní knihovna se pak dává na konec programu.

```

/*****
 *      Hisoft C      */
/* Standard Function Library */
/* Header for ZX Spectrum */
/*      */
/* Copyright (C) 1984 Hisoft */
/* Last changed 10 Dec 1984 */
/*****
#list-
#define NULL 0 /* for use with pointers */
#define FALSE 0 /* for Boolean operations */
#define TRUE 1 /* " */
#define EOF -1 /* end of file value */
#define ERROR -1
#define void int /* to use for type of functions which re-
turn no value */
/* File system Structure */
typedef int FILE;
/* Storage Allocation Structure and Variables */
struct _header
{ struct _header * _ptr;
  unsigned _size;
};
typedef struct _header HEADER, * HEADER_PTR;
HEADER _base, * _allocp;
/* Function type forward declarations for non-int library
functions */
extern char *strcat(), *strcpy(), *calloc(),
  *fgets(), *gets(), *sbrk();
extern unsigned strlen();
/* two arithmetic functions which need to be declared
BEFORE they are used, because they are vari-
adic (take any number of arguments). This
means that if you include „stdio.h“ these func-
tions will always be compiled into your program.
You may wish to make a version of this file witho-
ut these two routines if you want to compile a lar-
ge amount of code and don't need them.*/ int
max(param_byte_count) auto
{ static int argc, *argv, max;
  argc = param_byte_count/2 - 1;
  argv = &param_byte_count + argc;
  max = -32767;
  while (argc--)
  {
    if (*argv > max) max = *argv;
    --argv;
  }
  return max;
}
int min(param_byte_count) auto
{ static int argc, *argv, min;
  argc = param_byte_count/2 - 1;
  argv = &param_byte_count + argc;
  min = 32767;
  while (argc--)
  {
    if (*argv < min) min = *argv;
    --argv;
  }
  return min;
}
#list+
/*****
 *      Hisoft C      */
/* Standard Function Library */
/* End Header */
/*****

```

Následuje vlastní standardní I/O knihovna. Obsahuje základní 32bitovou aritmetiku, obsluhy vstupů a výstupů,

beepací a grafické procedury, vazbu na systém...

Umísťuje se na KONEC programu, který knihovnu používá, a to nejlépe příkazem, který knihovnu prohledá a zařadí do kompilace jen ty její části, na které byl v hlavním programu odkaz (#include ?stdio.l?).

```

/*****
 *      Hisoft C      */
/* Standard Function Library */
/* Version for ZX Spectrum */
/*      */
/* Copyright (C) 1984 Hisoft */
/* Last changed 20 Nov 1984 */
/*****
#list-
/* Some arithmetic functions */
/* min and max are in „stdio.h“ because they are varia-
dic */
int abs(n)
{ return n<0 ? -n : n ;
}
int sign(n)
{ return n ?
  ( n<0 ? -1 : 1 ) : 0 ;
}
/* An illustration of how to grub around in the store */
typedef char * __char_ptr;
int peek(address)
{ return *cast(__char_ptr) address;
}
void poke(address, value)
{ *cast(__char_ptr) address = value;
}
/* Format conversion routine - ASCII to binary integer
*/
int atoi(s)
char *s;
{ static int c, value, sign;
  while (isspace(*s)) ++s;
  value = 0;
  sign = 1;
  if (*s == '-') { ++s; sign = -1; }
  else if (*s == '+') ++s;
  while (isdigit(c = *s++)) value = 10 * value + c - '0';
  return sign * value;
}
/* Sorting function - a Shell sort */
void qsort(list, num_items, size, cmp_func)
char *list;
int num_items, size;
int (*cmp_func)();
{ static unsigned gap, byte_gap, i;
  static char *p;
  for (gap = num_items >> 1; gap > 0; gap >>= 1)
  {
    byte_gap = gap * size;
    for (i = gap; i < num_items; ++i)
      for (p = list + i * size - byte_gap; p >= list; p -= by-
te_gap)
      {
        if ((*cmp_func)(p, p + byte_gap) <= 0) break;
        swap(p, p + byte_gap, size);
      }
  }
}
/* String Handling Functions */
char *strcat(base, add)
char *base, *add;
{ static char *result;
  result = base;
  while (*base) ++base;
  while (*base++ = *add++);
  return result;
}

```

```

int strcmp(s, t)
char *s, *t;
{ while (*s == *t)
  {
    if (!*s) return 0;
    ++s; ++t;
  }
  return *s - *t;
}
char *strcpy(dest, source)
char *dest, *source;
{ static char *result;
  result = dest;
  while (*dest++ = *source++);
  return result;
}
unsigned strlen(s)
char *s;
{ static unsigned length;
  length = 0;
  while (*s++) ++length;
  return length;
}
/* Character Test and Manipulate Functions */
/* NB - the common ones are built-in for efficiency */
int ispunct(c)
char c;
{ return isprint(c) & ! isalnum(c);
}
int isalnum(c)
char c;
{ return isalpha(c) | isdigit(c);
}
int isascii(c)
char c;
{ return c < 0x80 ;
}
int iscntrl(c)
char c;
{ return c < , , | c == ,\177' ;
}
int isprint(c)
char c;
{ return c >= , , & c < ,\177' ;
}
/***** FILE SYSTEM *****/
char *fgets(s, n, fp)
char *s;
int n;
FILE *fp;
{ static int c;
  static char *cs;
  cs = s;
  while (--n > 0 && (c = getc(fp)) != EOF)
    if ((*cs++ = c) == ,\n') break;
  *cs = ,\0';
  return ((c == EOF && cs == s) ? NULL : s);
}
char *gets(s)
char *s;
{ static int c;
  static char *cs;
  cs = s;
  while ((c = getchar())
    != EOF && c != ,\n')
    *cs++ = c;
  *cs = ,\0';
  return
    ((c == -1 && cs == s) ?
     NULL : s);
}
void fputs(s, fp)
char *s;
FILE *fp;
{ static int c;
  while (c = *s++) putc(c, fp);
}
void puts(s)
char *s;
{ while (putc(*s++));
}
/* Storage Allocation and Freeing (Heap Management)

```

```

    */
char *calloc(n, size)
    unsigned n, size;
{ static HEADER *p, *q;
  static unsigned nbytes;
  nbytes = (n * size + (sizeof(HEADER) - 1)) / size-
    of(HEADER) + 1;
  if ((q = _allocp) == NULL) /* no free list */
  {
    _base._ptr = _allocp = q = &_base;
    _base._size = 0;
  }
  p = q->_ptr;
  while (TRUE)
  {
    if (p->_size >= nbytes) /* big enough */
    {
      if (p->_size == nbytes) q->_ptr = p->_ptr; /*
        just right size */
      else
      {
        /* split block and allocate tail
        */
        p->_size -= nbytes;
        p += p->_size;
        p->_size = nbytes;
      }
      _allocp = q;
      return cast(__char_ptr) (p+1);
    }
    if (p == _allocp) /* wrapped around free list */
    {
      if ((p = cast(HEADER_PTR) sbrk(nbytes * size-
        of(HEADER))) == ERROR)
        return NULL;
      p->_size = nbytes;
      free(p+1);
      p = _allocp;
    }
    q = p;
    p = p->_ptr;
  } /* end while TRUE */
}void free(block)
char *block;
{ static HEADER *p, *q;
  p = cast(HEADER_PTR) (block - 1);
  for (q = _allocp; !(p > q && p < q->_ptr); q = q->_ptr)
    if (q >= q->_ptr && (p > q || p < q->_ptr)) break;
  if (p + p->_size == q->_ptr)
  {
    p->_size += q->_ptr->_size;
    p->_ptr = q->_ptr->_ptr;
  }
  else p->_ptr = q->_ptr;
  if (q + q->_size == p)
  {
    q->_size += p->_size;
    q->_ptr = p->_ptr;
  }
  else q->_ptr = p;
  _allocp = q;
}#define HEAPSIZ 1000
char *sbrk(n)
    unsigned n;
{ static char p,
  heap[HEAPSIZ],
  heap_ptr=heap;
  if (heap_ptr+n > heap+HEAPSIZ) return ERROR;
  p=heap_ptr;
  heap_ptr += n;
  return p;
}

```

```

}/* Pseudo-Random Number Generator */
/* Adapted from „Learning to Program in C“ by Thomas
  Plum. */
char _rnum[4];
void srand(n)
{ long_init(_rnum, 0,n);
}int rand()
{ static char k[4];
  long_init(k, 0x41c6,0x4e6d);
  long_multiply(_rnum, _rnum, k);
  long_init(k, 0,0x3039);
  long_add(_rnum, _rnum, k);
  return (_rnum[1] << 8) + _rnum[0];
}/* Some Functions for 32 bit integer arithmetic */
void long_multiply(c, a, b)
  char *a, *b, *c;
{ static char x[4], product[4];
  int i, j;
  long_set(product, 0,0);
  for (i = 0; i < 4; ++i)
    for (j = i; j >= 0; --j)
    {
      long_set(x, a[i-j] * b[j], i);
      long_add(product, product, x);
    }
  long_copy(c, product);
}void long_add(c, a, b)
  char *a, *b, *c;
{ unsigned u, i;
  u = 0;
  for (i = 0; i < 4; ++i)
  {
    u += *a++ + *b++;
    *c++ = u & 0xff;
    u >>= 8;
  }
}void long_init(a, n1, n0)
  char *a;
  unsigned n1,n0;
{ a[0] = n0 & 0xff;
  a[1] = n0 >> 8;
  a[2] = n1 & 0xff;
  a[3] = n1 >> 8;
}void long_set(a, n, d)
  char *a;
  unsigned n, d;
{ static int i;
  for (i=0; i<4; ++i) a[i] = 0;
  a[d] = n & 0xff;
  if (d < 3) a[d+1] = n >> 8;
}void long_copy(c, a)
  char *a, *c;
{ move(c, a, 4);
}/** System Interface */
exit(n)
{ inline(0xcd,25236);
  _exit(n);
}_exit(n)
{ inline(0xe1,0xe1,0xe1,
  0x2b,0xc3,0x55,0);
}/* Spectrum Graphics and
  Sound Functions */
plot(on,x,y)
{ _setover(on);
  inline(0xdd,0x46,4,
    0xdd,0x4e,6,
    0xcd,0x22e5);
}line(on,dx,dy)
{ static sx,sy,de,bc;
  _setover(on);
}

```

```

sx=sy=1;
if (dx<0)
{
  dx= -dx;
  sx= -1;
}
if (dy<0)
{
  dy= -dy;
  sy= -1;
}
de= (sy<<8)+sx;
bc=(dy<<8)+dx;
inline(0xed,0x5b,&de,
  0xed,0x4b,&bc,
  0xcd,0x24ba);
}beep(duration,pitch)
  unsigned duration,pitch;
{ static ft;
  if (! pitch)
    for(ft=0;ft<duration;++ft)
      for(pitch=4630;++pitch);
  else {
    ft=duration*pitch/10;
    pitch=pitch/10;
    _beep(ft,cast(unsigned)43750/pitch-30);
  }
}paper(i)
{ return _colour(17,i);
}ink(i)
{ return _colour(16,i);
}cls()
{ inline(0xcd,0xd6b);
  _setover(on)
{ printf(„\025%c“,on?0:1);
  }_beep(DE,HL)
{ static de,hl;
  de=DE;
  hl=HL;
  inline(0xdd,0xe5,
    0xed,0x5b,&de,
    0x2a,&hl,
    0xcd,0x3b5,
    0xdd,0xe1);
  }_colour(h,i)
{ if (i<0 || i>7) return -1;
  putc(h,2);
  putc(i,2);
  return i;
}#list+
/*****
/* Hisoft C */
/* Standard Function Library */
/* End */
*****/

```

Název: HiSoft C 1.1
Typ: programovací jazyk
Producent: Hi Soft
Hodnocení:

75 %

George K.'s TOLKIEN

+GAMA

Vzpomeňte si na ZX M 1/93, kde George K. psal „...texty jsou zapakovány programem Tolkien 4.0, což se samo o sobě rovná ochraně. Pro přečtení by se musely vybalit, bylo by jich asi o 9 kB víc...“, nebo na ZX M 4/93, kde vzpomíná na Tolkiena 5.0 - „Jen tak z matematické zvědavosti jsem naprogramoval... ...pakuje až na 50% původní délky... ...text je naprosto nečitelný oku pirátovu a navíc ho před použitím nemusíte rozbalovat - stačí asi 50ti bajtový prográmek, který vám ho přečte i takhle zahuštěný...“. Ano, je to tak. Jedna z posledních verzí textového pressoru Tolkien (ano, přesně TOHO legendárního Tolkiena, o němž George psal) se vám právě dostává do parátů (a tak skončí každý program, který se dostane omylem mezi lidi).

Než ho ale začnete používat, přečtěte si tento skromný manuál. Proxima totiž Tolkiena nikdy neprodávala, George K. ho psal jen pro potřebu svou a svých kamarádů, takže žádný manuál nikdy neexistoval, a když jsem Tolkiena dostal do ruky, ani bratři Flaškovi nevěděli, jak se používá (to prý věděl jen sám George K.). Když jsem ho neuměl použít, zkoušel jsem psát packery vlastní (Caesar), jen tak z matematické zvědavosti (a hlavně potřeby), pakovalo to, ale kvůli nepřehlednosti nebyly výsledky tak ohromující (bodejť ne, když George má dvojnásob velkou tabulku a nepakuje nic, co se skutečně nezkrátí, ale to předbílám), nakonec jsem vzal disassembler a dva roky disassembloval, krokoval, zkoušel, testoval a teprve 6.4.1998 (ten den jsem napsal i tento článek - je to nechuť, od napsání k vydání přes dva roky - pozn. +GAMA) jsem plně pochopil, jak se to celé vlastně používá (je to v podstatě jednoduché, jen se to špatně vysvětluje). Vedlejším efektem bylo, že jsem pochopil skoro celý systém pakování. No nádherná. Snad se pamatujete i na ZX Mag, kde se v Intru psalo o tom, že George K. Tolkiena urychlil, aby mu zbytečně celý den neblokoval počítač. Asi se mu to skutečně podařilo, protože déle než 5-10 minut se mi pakovat nepodařilo.

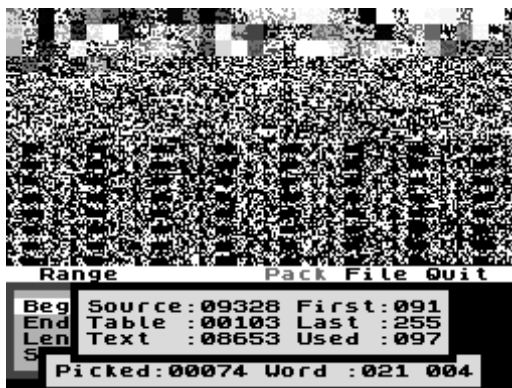
Upozornění! Tolkien i v kazetové verzi zuřivě outuje (autodetekce přítomnosti D40), na 128 Spectru je nezbytně nutné zakázat stránkování!!! (OUT 32765,48).

Po spuštění si jistě všimnete, že packer se ovládá přes podobná menu, jako Mr-Pack. Vlevo je Range, kde si nastavujete rozsah pakovaného textu a adresu, kde se bude pakovaný blok spouštět. Když ovšem

loadujete a joinujete (vysvětlím později) přímo z Tolkiena, nastavuje si všechno inteligentně sám. Pak je volba packu módem 0, 1 nebo 2. Vysvětlím později. No a pak je Load (nahraje text), Join (připojí k nahranému textu další), Save spakovaného bloku (ten ovšem zvolíte až tehdy, až ho skutečně spakujete, jasné?) a volba zařízení (kazeta nebo D40, kazeta výborně funguje na MB-02+). Poslední jsou funkce Reset (reset..), Return (vrátí se do volajícího programu) a Basic (vyskočí do Basicu, při poškození systému během packu provede i NEW, zachováva přitom data nad ramtopem).

To ovšem byly celkem standardní volby, známé více či méně z MrPacka. Teď k módům: Mode 0 zapakuje nativní text (strukturu zmíním později), Mode 1 zapakuje text z Desktopu (strukturu až později) a Mode 2 zapakuje text z Desktopu tak, že všechna písmena budou malá (ale výsledný packovaný kód je pak nejkratší).

Módy 1 a 2 automaticky převedou Desktopový text do nativního formátu (pro-



stě ho rozkomprimují (Desktop přece komprimuje, ne?) a vyhážou všechny řídicí znaky (a všechny Entery převedou na mezery), takže zůstane skutečně asi jen to, co v Desktopu při psaní doopravdy vidíte). Mode 2 navíc zkonvertuje i písmenka - a teprve nativní text se pak pakuje.

Celý spakovaný blok se skládá z menších jednotek - textů - které mají pořadové číslo od 1 do... podle toho, kolik jich chcete mít.

Jde o to, že PRVNÍ znak nativního textu (tedy první znak, který napíšete do Desktopu!) slouží jako oddělovač jednotlivých textů (v praxi je asi nejlepší použít zavináč - oddělovat texty zavináčem nebo podobným paznakem), a musí být zároveň umístěn na úplném konci celého bloku textů (na konci posledního textu).

Je jasné, že, vzhledem k tomu, jak pac-

ker postupuje, nejde Joinovat nativní text s desktopovým a pakovat desktopový text v Mode 0 je blbost. No v podstatě jsme text zabalili. Nemyslete si, že packování skončí v okamžiku, kdy je text zabaleno. Houby s dršťkovou. Pack skončí v okamžiku, kdy se zaplní tabulka odkazů (většinou se totiž přeplní dříve, než k úplnému spakování dojde, úplné spakování se vám přihodí skutečně jen u textů krátkých a obtížně balitelných).

Nastavení Start v menu Range určuje, na jakou adresu je třeba spakovaný text nahrát, chcete-li ho použít. Tam taky začíná přiložená čtecí rutina. Jak ji volat?

Nejprve vezměte devastaci nebo jinou pomůcku a podívejte se na rutinu. najdete tam veselou destruktivní instrukci JP 00000. Nu tož si ji přepište, dejte tam skok na svou rutinu pro tisk znaku (geniální je rutina, která zarovnává slova do řádek, ale pro zkoušku stačí dát tam JP 16). Pozor na chyták - do rutiny tisku znaku se skáče jumpem, ale vrací se returnem!!!

Když je tato úprava hotová, stačí už jen LD DE, CISLOTXTU : CALL TOLKIEN (za návěští CISLOTXTU si dosadíte pořadové číslo textu, který chcete vytisknout, viz výše, návěští TOLKIEN má tady hodnotu začátku rutiny (to, co bylo zobrazeno jako START)).

No a to už je skutečně všechno. Chcete-li vědět, jak Tolkien pracuje tak je to takhle: Projede si pakovaný blok a jednotlivým hodnotám (bajtů, of což) přiřazuje čísla v tom pořadí, jak je potkal - začíná ale od nuly a nula pak slouží jako oddělovač textů, nejdříve se tedy seznámí s oddělovníkem textů, že. Skončí u počtu, kolik různých hodnot bajtů se v bloku vyskytuje. Je-li jich moc, tak pakovat přestane. Jednotlivá písmena v textu nahradí jejich hodnotou (proto jsou texty nečitelné, to, co můžete považovat za zapakovaný text, je ve skutečnosti tabulka hodnot, dlouhá jen asi 100 bajtů) a hodnoty nepoužité se při dalším pakování použijí jako odkazy pro volání rekurzivního stromečku (xor za terminologii, teorii pakování jsem v životě neviděl a ve škole ji z pochopitelných důvodů nebereme).

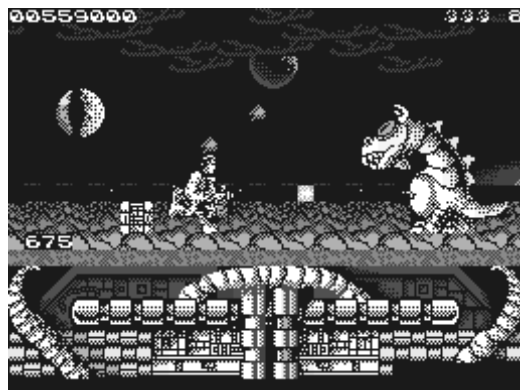
A to je skutečně všechno, víc mne zatím nenapadá.

O grafice

+GAMA

Tož je vám asi jasno, o čem bude řeč (mně ne, a to jsem autor). Asi si řeknete, co na tom vlastně je, když grafik načmáře sprajty, programátor (a většinou to stejně bývá jedna a ta samá osoba) na to ušije nějakou rutinu, případně obšlehne George K.'s AnimAce a je to. Maximálně by obráceně mohl programátor spáchat rutinu a grafik se do ní snažil napasovat nějaké sprajty či jinou šílenou grafiku. Blah!

No a pak to ovšem vypadá tak, jak to vypadá. Monochromaticky. Proč každá nová bomba už od dřevních dob Oceanu musí nutně postrádat barvy? Worms,

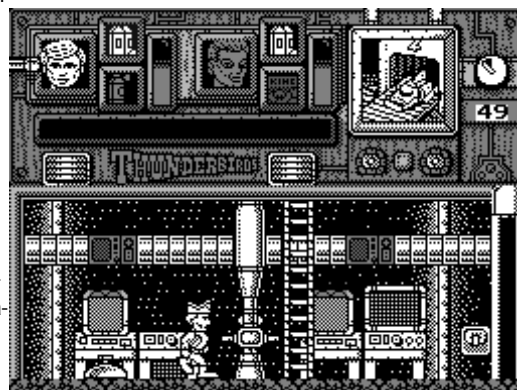


SimCity je colourfull...

Přitom historicky bylo Spectrum rájem barev, to se dočtete už u Franty Fuky, jak vychvaluje JetPac za krásné barvy, ani ten Manic miner nebyl zrovna ČB a i ty pitomé simulátory se snaží barvy dodržovat. S monochromem tedy zřejmě teprva nedávno přišel nějaký degén (rozuměj denegere.. degene-generova... deregenova... vy víte co...) pravděpodobně od Oceánu, poté, co Ultimate a další začali dělat monochromatické rohovky (ne že by to barevně nešlo...).

No dobře. Tož jsem si od plic (žaludku, střev, ledvin, varlat(?)...) zana-dával a teď co s tím. Způsobů je mnoho a taky jich už v minulosti

některé novější věci od Oceánu a příbuzných - Navy Seals, Total recall aj., kde hrdina mění barvy jak chameleon podle momentální nálady grafika a pozadí (přitom ta okolnost je super, dokonce v Totáči jsou i



Black raven, (H)Robocop, Badman the movie, Mordal kombat, avizovaný Walker a mnozí další mne děsí ve snech, kde se prohánějí po černobílé televizi Pluto, honění z IQčka nebo PMDčka a rozdíl v barvách nevidět! Přitom my přece máme k DisPozici patnáct (slovy patnáct!) krásných spectristických barev (na PeCi jich sice na VGA mívají 256, ale krásné spectristické jsou z toho tak dvě, tři...), kterými se pro změnu může chlubit Doom (já vím, že fialový Imp není asi to,



animované větráky, blikající nápisy SEX SHOW GIRLS a (h)různé, ovšem víceméně plynulé, scrolly.

Kombinací prvního a druhého způsobu je ten, kde si postavička tahá barvu s sebou, většinou černý ink a barevný papír, s tím se setkáme ve hrách Everyone's a Wally nebo je to jeden ze způsobů zobrazení ve Three Weeks in paradise a jiných hrách.

Třetí způsob: postavička skáče po attributech a barvy si tahá s sebou, většinou barevný ink a černý paper, sprajt je tedy



bylo mnoho použito. Abych začal nějak rozumně.

První způsob je házet sprajty inkoustové normálně po pixelech na černý papír a k nim aproximativně prostě ukrutně přibližně nastavovat správný INK v attributech, při přiblížení nebo překryvání spritů tu máme veleoblíbenou attr kolizi čili (d)efekt. Viz Galaxians, JetPac, erm... určitě ještě něco existuje...

Nebo se v pestrobarevné krajině (většinou černý papír, barevný ink) pohybuje klasický maskovaný pixlíkový a nevybarvovaný spra(j)tek. Klasická ukázka Dizzy a klony (proč všechny dizzovky hrome přebírají tenhle defekt?), inkoust sprajtu prostě přebírá barvu toho místa, kudama (mo)mentálně prolézá. Sem patří i

na co jste zvyklí z Pece, ale nebude to spíš tím, že PeC takhle krásnou fialovou vůbec neumí?), Dan Dáre drei (slovy 3), Astro Marine Corps, Sausage (nebo Savage?), Bad Dream i tolikrát proklínané Inferno, Extréme, Želvy Ninja 1 (dvojka byl zas odporný monochrom) a dokonce i to

zbarvený a pohybuje-li se po stejném pozadí (velké černé paperové plochy, barevné objekty inkoustu), je to geni(t)ální (nebude to sice úplně pravda, ale ten Dan Dáre III tento způsob připomíná, dále sem patří Astro Marine a Inferno), kolidují-li spolu ob-

Návod

jekty - třeba v Infernu velké budo-
vy a výbuchy -
anebo pokud je
pozadí převážně
barevné (tedy vět-
šinou inkoustové),
jsou kolem posta-
viček prima černé
zuby (krásný pří-
klad Golden Axe,
zejména vyšší le-
vely).

Čtvrtý: Posta-
vička je stále založena na attributech, ale
na barvě papíru či inkoustu už moc nezále-
ží, protože si s sebou tahá jen inkoust, za-
tímco papír přejímá od pozadí (Bad Dream,
pár věcí od kluků Probáckých jako Želvy
Nindža I, Extreme, Turrican - ale Robocop 3
sem rozhodně nepatří!!!).

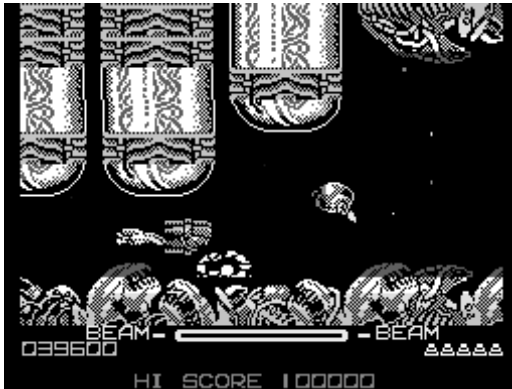
Pátý je nějaký složitý a nedobře se mi
chápe, v podstatě je to odvozením čtvrté-
ho, až na to, že některé atributy nepřebírají
nic (ty připomínají nejvíce způsob třetí, viz
třeba Berkovy oči), některé přebírají papír
či inkoust od pozadí podle toho, co je právě
potřeba, sem patří hry od Dona Priestleyho
(Popeye, TrapDoor, Trough the Trap Door).

Navíc si neodpus-
tím ještě jednu ma-
ličnost - nahrajte
Trap Door, spusťte
hru a na nic neša-
hejte; jděte s Ber-
kem o dvě místnos-
ti doprava, v míst-
nosti se závaží se
můžete trošičku
projít (doleva a do-
prava kolem záva-
ží), vyskočte do Ba-
sicu (Multifacem,
NMIčkem, máte-li

Busyho ROMku, tak stačí RESETem, to je
holt výhoda) bez smazání obrazovky!) a
přebarvíte všechny atributy na černobílo
(OVER 1 enter PAPER 0 enter INK 7 enter
PRINT AT 0,0; enter FOR A=1 TO
703:PRINT „ „;(to je mezera): NEXT A enter)
a pokud spatříte to, co já, pravděpodobně
nedožijete rána... (u hry POPEYE to ale ne-
funguje).

Vůbec hry od Dona jsou překvapivé.
Grafika je přímá obrovská, postavičky jsou
přímá obrovská, dokonce slušně animova-
né, a to skorem ve všech barvách (hlavně
Popeye), na přímou obrovskou plochu! A
být to na 128ičce
přes dvě VRAM, bylo
by to i úžasně plynu-
lé (to ono je i teď,
ale stejně, ty hloupé
LDIRy ukrutně zdr-
žují).

No ale to není
vše, je tu ještě je-
den způsob. Šestý
(zatr. číslování)! Bar-



ní téměř absolutně všude (aby všude, ku-
dama se bude šípka pohybovat, byl černý
ink, ono není moc hezké, když při šmejdě-
ní mění příliš barvy, kor podle inkoustu po-
zadí)! Pokud vám nedochází, o čem to
vlastně melu, tak si vzpomeňte na dobývá-
ní pevnosti v North & South... Další výhoda
tohoto způsobu je zároveň ta, že maska
nemusí kolem sprajtu vytvářet známou
auru (obvykle širokou 1 pixel, je to skorem
všude, viz Dizzy) ani připitoměle zubance,
ale prostě jen kryje prázdné (tedy rovná se
barevné) plochy uvnitř sprajtu, výsledkem
je vyšší realističtvo, například černé holinky
v černém stínu jsou skutečně holinky ve

stínu a žádná pi-
tomá aura (kde
se vzala, tu se
vzala) nikde ne-
překáží.

Platí tedy nej-
prve přemýšlet,
shodnout se na
něčem s progra-
mátorem a pak
teprva čmárat.
Samozřejmostí
je, že k příslušné-
mu sprajtu se na-
čmáruje i při-

slučná maska, ovšem pozor, tady potřebu-
jeme (čas od času, tedy pořád) maskovat i
atributy... (já to dělám tak, že si v atribu-
tech, které nechci zobrazovat, nastavím
flash, mám pak aspoň k dispozici celou pa-
letu brightů atd., šlo by to sice dělat i jinak,
ale protože zapnutý flash ve spritu asi po-
třebuje málokdo, není to zas tak moc bad
idea), vlastní kreslení snad ani nemusím
vysvětlovat (nastavit malý rastr, transpa-
rentní barvy, načmárat podklad, pak ho vy-
kolorovat - Artstudio, v Artistu 2 pak po-
dobně, neřkuli úplně stejně). Po menších

krocích než přes-
ně po atributu by
se se sprajtem
asi hýbat nemě-
lo, ale řekněte
sami, jestli vám
to na Probáckých
gamesách vadilo,
navíc se pohyb
dá plynuleji roz-
kreslit, takže vý-
sledný pohyb po
attributech zanik-

ne - třeba by se tomu dal připodobnit po-
hyb raketky v R-Type nahoru a dolů, no
možná někde existuje typičtější příklad...

No a na konec tohoto sloupku (jó, ně-
které rozumy by šly roztahat na dvoustrán-
ku, ale zas tolik o ně asi nestojíte) vám po-
přeju, abyste svou ďábelsky barevnou gra-
fikou natrhli klukům Probáckým prdel a
aby tu vznikalo víc takových dílek jako tře-
ba... (no nevím, Twilight byl sice pestroba-
revný, ale ta bezatributová postavička dost
vadí...), ony totiž bůhvíproč ty graficky nej-
nej věci bývají celkem i dost blbosti (nepo-
čítám-li pár zcela ukázkových, koneckonců
zde průběžně uváděných), i když, že ano,
proč hrát něco, co je sice supr zábavné a s
prima muzikou, když grafix pokulhává (na
obě nohy a s ortézou...)! ■

Freeware ZX Magazínu

Freeware 1:

Booter
Pekleso
Gambler
Poklady
Falling Down demo
Metal Massacre
Oradot 2 demo

Freeware 2:

Erotika 1
Erotika 2
21 erotic
Kostky strip
Počty strip

Freeware 3:

Dizzy 3.5
Dizzy 8
Seymour take one
Zabijáci číhají
Smagly 2
Crime Santa Claus

Freeware 4:

Doom pre-release
Worms demo (ne na disku 5.25")
Prince of Persia

Freeware 5:

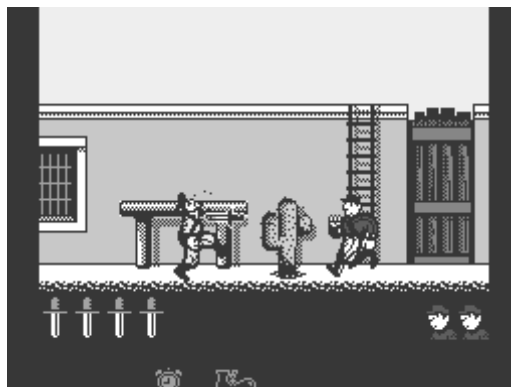
File viewer
Personal talk

Freeware 6:

Normal life 1
Normal life 2
X-System pro

Freeware objednávejte na adrese:

Matěj Kryndler
Lotyšská 8
160 00 Praha 6
nebo na redakčním mobilním telefonu
0604/803 853



OPAT

(Open Program Architecture)

Studio 22, Tritol

OPAT přichází, aby umožnil programátorům a uživatelům provozovat programy, psané s podporou OPATu, na libovolném diskovém systému, včetně libovolného systému budoucnosti.

OPAT kernel začíná od adresy 23867, smí být max. 1536 bytes dlouhý.

OPAT - software interface:

vstupní bod = 23867 (MUSÍ obsahovat instrukci JP xxxx!)

COM = 23870 (1) - povel OPATu (dolní 4 bity povel, horní 4 bity určení)

SEL = 23871 (1) - diskové zařízení

(A, B, C, D, M jako RAMdisk, 1-8 microdrive, F síťový disk;

X = poslední volané zařízení)

(tape se doporučuje označovat jako T, ale jeho obsluhu nezajišťuje OPAT, nýbrž aplikace sama)

ADD = 23872 (3) - adresa začátku bloku (nejnižší, vyšší, nejvyšší bajt)

LEN = 23875 (3) - délka dat

NAM = 23878 (12) - adresa jména (viz INI)

ERR = 23890 (1) - číslo chyby (dolní 4 bity skupina chyby, horní 4 bity bližší upřesnění)

Chyby: 0 - OK

1 - Not supported (příkaz nebo zařízení)

2 - Parameter error

3 - disk error 1 (DOS)

0 - (viz jednotlivé příkazy)

1 - File not found

2 - Disk full

3 - Dir full

4 - Unknown disk

5 - Device unavailable

6 - Device I/O error

7 - Device damaged

8 - Drive error

9 - CRC error

10 - Disk is write protected

11 - Unknown error

4 - disk error 2 (OPAT)

0 - (viz jednotlivé příkazy)

1 - File exists

2 - File is read protected

3 - File is write protected

4 - File is delete protected

5 - Stream is not available

6 - Stream doesn't exist

7 - Read after end

8 - Read/write before begin



OPAT

compatible

9 - Disk changed

10 - Verify error

5 - Internal OPAT error

(to je spousta chyb, v praxi stačí i jen rozlišení hlavních skupin)

Obsuha pásky není předmětem služeb OPATu. Tu si musí aplikace zajistit sama, doporučujeme používat standardní rutiny ROM (SA-BYTES, LD-BYTES) kvůli kompatibilitě (tape emulátory atd.). Podpora pásky není pochopitelně povinná.

Co se jména souboru týče, jeho zpracování z libovolné podoby (i maska) zadané uživatelem aplikaci si zajišťuje OPAT, ne aplikace!

Seznam příkazů:

0 - INI

1 - LDF

2 - SVT

3 - CAT

4 - DEL

5 - VER

6 - FIN

7 - DRI

8 - FCR

9 - FOP

10 - FCL

11 - RDF

12 - WRT

13 - SEK

14 - CPY

15 - neobsazeno

INI

initialisation

In:

ADD (0) - 0 = 48k memory

jinak jednotlivé bity označují používané 128 stránek.

Out:

LEN - adresa, kde jsou vyjmenována dostupná zařízení (ASCII)

ADD - (0,1) - adresa, která ukazuje na informace o OPATu:

LEVEL (např. "A1"), chr\$

13

(písmeno ukazuje paměťovou verzi, A=48, B=128, C=256, D=512 a více)

SYSTEM (např. "MDOS"),

chr\$ 13

VERSION (např. "1.0"),

chr\$ 13

Autor (např. "Tritolsoft"),

chr\$ 13

NAME a EXTENSION

LENGTH (jeden bajt; dolní 4

bity délka jména, nejvyšší 2 bity délka přípony; NAME + EXTENSION + oddělovací tečka = max. 12 znaků)

COMMAND LIST - seznam podporovaných povelů OPATu, chr\$ 255

LDF

Load from

rozříšení: 0 - celý soubor, 1 - část (max. 512 bytes)

In:

SEL - zařízení

NAM - jméno (MDOS: 1234567890.B, TRDOS:12345678.C,

+3DOS:12345678.COD etc)

ADD - adresa

LEN - rozříšení 1: délka (max. 512); rozříšení 0 čte délku souboru z jeho hlavičky

Out:

3 - File doesn't exists

SVT

Save to

0 - nepřepisovat starý, 1 - přepíše existující soubor

In:

SEL - zařízení

NAM - jméno

ADD - adresa

LEN - délka

Out:

4 - file exists

3 - dir, fat, disk full

Programování

CAT

catalogue
inicializace katalogu, načtení první položky, zvolení masky
0 - pro buffer použij paměť vymezenou aplikací (1536 bytes)
1 - použij interní DOS RAM (pokud ji systém má - D40, MB02+)
2 - použij paměť 128
3 - použij jinou paměť (nespecifikováno) (zvolený mód se vztahuje na ostatní příkazy katalogu)
+8 - setříd' katalog (podle ASCII)
In:
SEL - zařízení
NAM - maska
ADD - rozšíření 0 - adresa bufferu
rozšíření 2 - ADD (0,1) adresa, ADD (2) číslo 128 stránky (0-7)
Out:
NAM - položka katalogu
3 - not found

rozšíření 13 - CAT init
nastavení pointeru na první položku
Out:
NAM - položka katalogu
3 - not found

rozšíření 14 - CAT previous
nastavení pointeru na předchozí položku
Out:
NAM - položka katalogu
3 - not found

rozšíření 15 - CAT next
nastavení pointeru na další položku katalogu
Out:
NAM - položka katalogu
3 - not found

DEL (ERA)
Delete (Erase)
rozšíření: 0 - při delete protected error
4.4,

1 - vždy smazat

In:
SEL - zařízení
NAM - jméno nebo maska
Out:
3 - not found
4 - delete protected

FIN

File info
In:
SEL - zařízení
NAM - jméno
Out:
LEN - délka
ADD - start
NAM - třetí parametr
první byte=0: není třetí parametr
1: textová informace (11 znaků)
2: číselná informace (3 byty)
3 - not found

DRI

Drive info
In:
SEL - zařízení
Out:
NAM - jméno disku (NAM(0)=255 - není disk nebo neznámé jméno)
ADD - adresa informací
byte 1: 0 - fyzicky dostupné zařízení,
255 - logicky dostupné zařízení
další byty:
0 - formát (1 byte stopy, 1 byte sektory, 1 byte povrchy)
1 - celková kapacita (3 bytes)
2 - volné místo (3 bytes)
3 - obsazené místo (3 bytes)
4 - celkový počet položek v adresáři (3 bytes)
5 - volných položek (3 bytes)
6 - obsazených položek (3 bytes)
7 - délka clusteru v bytech (3 bytes)
255 - konec

Příkazy nepodporované levellem A1:

VER

Verify
rozšíření: 0 - celý soubor, 1 - část
In:
SEL - zařízení
NAM - jméno
ADD - adresa
LEN - rozšíření 1: délka; rozšíření 0 bere délku z hlavičky souboru
Out:
3 - not found
4 - difference found
FCR
File create
vytvoří soubor pro sekvenční přístup
rozšíření: 0 - nepřepisuj starý soubor
1 - přepiš (vytvoř soubor s nulovou

délkou)
In:
SEL - zařízení
NAM - jméno
LEN (0) - atributy: bit 0 - zápis, bit 1 - čtení,
bit 2 - chráněný soubor (až do uzavření kanálu nepovolí žádný přístup)
Out:
NAM (0) - číslo kanálu (0..15)
2 - no free channel
3 - dir full
4 - file exists

FOP

File open
In:
SEL - zařízení
NAM - jméno
LEN (0) - atributy: bit 0 - zápis, bit 1 - čtení,
bit 2 - chráněný soubor (až do uzavření kanálu nepovolí žádný přístup)
Out:
NAM (0) - číslo kanálu (0..15)
2 - no free channel
3 - dir full
4 - file exists
FCL

File close

zavře kanál, smaže buffery
In:
NAM (0) - číslo kanálu (0..15)
Out:
2 - channel not open
3 - disk full (zavřeno bez uložení)

RDF

Read from
In:
NAM (0) - číslo kanálu (0..15)
LEN - délka bloku
ADR - adresa uložení v paměti
Out:
2 - channel not open
3 - read out of file (LEN = o kolik)

WRT

Write to
In:
NAM (0) - číslo kanálu (0..15)
LEN - délka bloku
ADR - adresa uložení v paměti
Out:
2 - channel not open
3 - disk full

SEK

SEEK, přesun pointeru na jinou pozici v souboru

In:
NAM (0) - číslo kanálu (0..15)
LEN - relativní posun od aktuální pozice
LEN (0,1) - kolik bajtů
LEN(2) - 0=dopředu, 1=zpět
ADR - přesun na absolutní adresu (při LEN (0,1)=0)
Out:
2 - channel not open
3 - seek out of file (LEN=o kolik)

CPY

COPY, kopírování souboru mezi zařízeními
In:
NAM - jméno nebo maska
SEL - zdrojové zařízení
ADR (0) - cílové zařízení
Out:
2 - device not found
3 - no free space
4 - file not found

(Za tu kresbičku opata se omlouvám, ale originální obrázek nemám a scanner jsem líný používat... pozn.+GAMA)

Doufáme, že se standard O.P.A.T.u ujme, protože vyvíjet ho kvůli jedinému programu, CLIPu, by nemělo valného významu.

Otázka je, jak a čím lidi donutit, aby při psaní svých programů pamatovali na to, že O.P.A.T. existuje a je vhodné jej na dílko aplikovat.

Druhá otázka je, jak lidi donutit, aby si napsali O.P.A.T.a pro svůj systém. Třeba na MBčko žádný není, BETASHITový zůstal nedokončený...

Pokud začnou OPATa používat Rusové, jsme zachráněni.

FOTO reportáž

Foto: Starsoft

Text: +GAMA

Jaký byl a o čem byl Zlíncon '99, poslední spectristická akce loňského roku



Spectrum. Tak o tohle asi šlo. Tenhle stroj je asi Chrysagonův, ale to nevíme jistě. Takových ovšem bylo ve Zlíně mraky.

Všimněte si Amigy (původně v tomto článku bylo "všimněte si hAMIž-né, Matěj si ale myslel, že by tomu nikdo nerozuměl- pozn.+GAMA) v levém rohu. Jen se krč, příšero! (furt lepší jak PeC, ne?)



DiHalt. Aneb spectristické tričko do povinné výbavy české armády. Bez těch dvou písmen ZX se prostě moderní uniforma neobejde. Snad to na příštím Zlínconu bude mít už za sebou, bude z něj muž a bude mít na Spectrum víc času.

(a darmo zní naše nabádání: Neblázni, emigruj.)



DiHalt, Marwin, Chrysagon. (kdybych někoho identifikoval blbě, tak kéž se dotčný dotčený nerozčiluje, ničil by si zbytečně zdraví...). No, to je dohromady kousek CI5, kousek K3L, kousek Naughty Crew. Dokonce kousek Phantasy, nemýlím-li se...

Všichni obyvatelé a pelešníci Zlínconu se tu pochopitelně neobjeví, ale bylo jich tam taky požehnaně. Schválně, koho bylo víc - lidí, či strojů?



8 Bit Company. Pravda, na obrázku nejsou všichni, ale tradičním atributem nejsou ani tak oni, jako spíše gulášovka :)



Prašivka čili popelnice: Ve Zlíně sněžilo. A sice sněžilo tak, že sněžilo hodně. Velmi hodně moc. Což kromě sněhových bitek (kam se hrabou na North&South ve dvou hráčích!) a jiných zimních sportů (kam se na ně hrabe Winter Games!) přineslo i disciplíny z jiného soudku (tak kdypak ti Rusové dodělají ten slibovaný Crazy Cars III, už se těším na sněžení...), jako třeba sněžný wrestling.

Pokud se do Zlína vydáte tuto zimu, máme pro vás příjemnou zprávu: Dům Astra v Příluku je vybaven sněhovým hrablem, lopatou a koštětem. Ozkoušeno na vlastní kůži (končetiny, svalstvo, ...). A letní gumy se nejeví jako dobrý nápad (taky vyzkoušeno, na druhou stranu to byly první hodiny, které jsou zcela **Y2K** rezistentní a nevyžadují **RTC** obvod).



POL of Phantasy. Aspoň doufám, že je to on a osobně.

A ten druhý je ZX Magazin. No jéje, to je ale reklama... Zdá se, že velké dvojčíslo (poslední, které sázel Tritol) bylo vhodně načasováno a zaujalo i obsahem (to se tomuto dvojčíslu asi bohužel stát nemůže...).

Nemajitelé D40 se měli možnost seznámit s jeho File Viewerem, mnohé, hlavně milovníky modemů, zaujal Personal Talk. Je to sice jen taková hříčka (nejzajímavější aplikace jsou tradičně ty, jež mají v praxi nejmenšího či nižádného využití), provedená však tak precizně, že mnozí zatoužili po pořádné Spectrácké síti. Kdeže skončil ZX Net z Interface 1, kde je konec síti Disciple, nejsou ethernetové karty, D-Net (Dubnický net) od grupy 3SC se také nechytil, ruské periferie (třeba X-Trade modem) u nás nesnadno potkat...

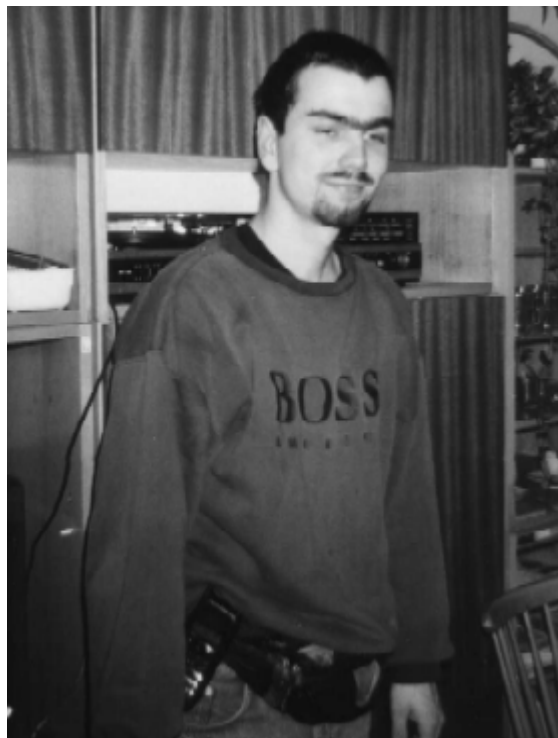
Ale co my víme. Třeba na příštím Zlinconu vezmeme všichni dráty, propojíme své 8255ky nebo XTMka, nahrajeme Personal Talk nebo jeho dokonalejší verzi a budeme tlachat jen skrzevá klávesnice a televize svých Specter a Tikťaktiků.



Hood of Crazytronic. Chudák chlapec, na téhle fotce vypadá jak blbeček. Ve skutečnosti jsou ale blázni z Crazytronicu jedni z neaktivnějších Spectristů poslední doby (chm, to vypadá jako pochvala... jen aby je nepostihla pýcha, neb ta, jak známo, předchází rum...). Dokonce sestavili speciální demo pro Zlincon, byť ho vypustili mezi lid už o pár dní dříve.

Ve Zlíně se sice zjevil jen tento zástupce skupiny, takové drobnosti, jako MBčková verze ProTrackeru 3.3, jsou však dostatečnou omluvou za neúčast ostatních.

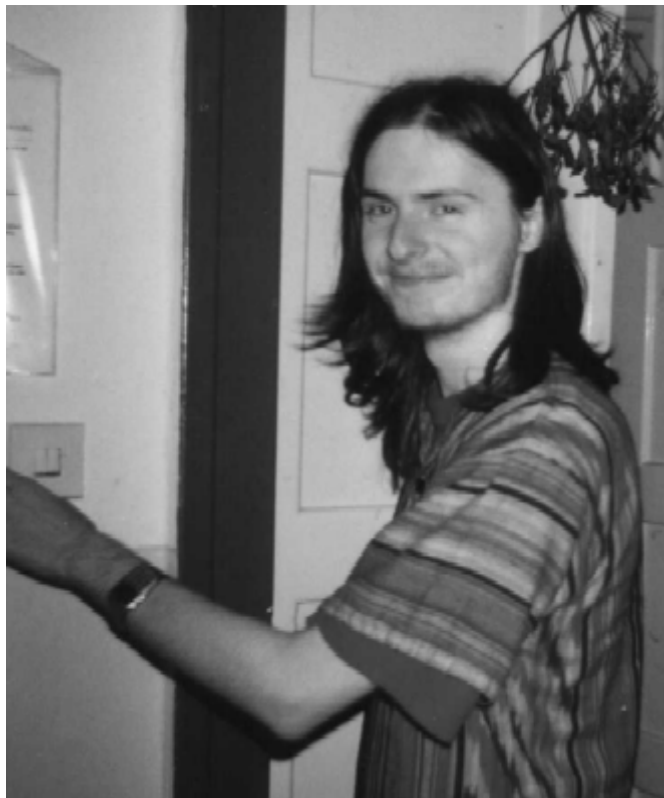
A to kdybyste věděli, že s Mr B0Bem už crackuje Pro Tracker 3.4, nebo jaká dema chystají na léto... chystají-li vůbec nějaká...



Matsoft of Naughty Crew. Šéfredaktor a vydavatel ZXM. Jak zajisté vidíte, je plně elektronizován. U dolního okraje zřítte redakční debilní telefon, vzadu kabely, externí záznamové zařízení moderního typu čili gramofon... A co je nejdůležitější, nekompromisně se dozvídáte, kdo že je to tu vlastně ten BOSS.

Tento rok zřejmě zrušil svou tradici v noci crackovat a hýkat na půl baráku a ráno se pak nechat nalézt na klávesnici Spectra ve stavu bezvědomém, ale jistě to tvrdit nikdo nemůžeme, leda snad on.

Dema se od něj nedočkal nikdo, ale cracky ruských věcí se díky němu naštěstí šíří. Kéž by jich bylo víc, kéž by nebyl líný je každému nahrát! (Tak a teď jsem šéfa pomluvil, to jsem si to asi pěkně polepil...)



Dron of K3L. Velikán. Jak je vidno ze snímku, s elektrickými zařízeními to prostě umí. I když jsou pod proudem. A víte, jak poznáte, že je pod proudem Dron? Podle vlasů. Schválně si nechal narůst biologický indikátor elektrického náboje. A není sám. Tady je ovšem vybitý. Doufejme, že mi za tyto šplechty nenabije.

Na Zlínconu se mohli nevěřivci přesvědčit o existenci HDD MB-02+ patche, který z jeho dílny vzešel. A opět si mohli poslechnout, že General Sound opravdu v českých máme a že už Dron dhalil, v čem vězelo to čertovo kopýtko, ta chyba zavševná, a z nouze ho donutil housti.

Těšíme se na další bombu.



Dizzy. V civilu se maskuje jako systémový programátor na burze nebo snad v nějakém brokerském domě, ale na Zlínconu byl. Přijel, viděl, věděl, znal. K přehlédnutí nebyl. Kámošům doma pouští na videu Cybernoida a chlubí se, jak daleko se dostal a kde mistrně uhnul před kterou potvorou. Jeho hlavní zálibou je ale hardware.



P.V.L. Předvedl všem, kteří nyní jen tiše závidí funkční harddisk (pokud si nekoupili jeho geniální ZX IDE Face), že to jde.

Předvedl všem uživatelům D40, že kazetové loadery nebo sejvy pozic netřeba crackovat a převádět pro náš český diskový systém. Alespoň pokud používají jeho TAPE Emulátor pro D40. Předvedl taky, že NMI menu není špatná věc, že DevastAce se do ROMky vejde, že adresáře na D40 nejsou utopie, nýbrž holá skutečnost, že lze snapovat i pouhé screeny a bez výpisu počátečních adres souborů v katalogu rozumně pracovat prostě nejde.

Nebudu se o tom víc rozepisovat, spolu s Tritolem se na stranách ZXM o svých úspěších šířili již dost.

A co třeba, že bychom se mohli těšit na Unix pro Spectrum? Přece nebudeš pořád používat Linux jen na PeCi, člověče!



Johny-X of Naughty Crew. A ZX Magazin. Aspoň vidíme, co čte elita. Tentokrát bez gumídka a tapezatoru, zato s MBčkem a Spectrem 128 +2. Víme, že by rád dal dohromady demo, hru, cokoliv... nezbývá nám, než se těšit.



Tritol. Vysoký spectrista (a hele, taky velikán).

Tou dobou ještě pracoval u jedné nejmenované počítačové firmy a na firemních strojích sázel a na firemní ležrovce a šičce tiskl a sestavoval ZX Magazin.

Dnes pracuje jinde, kde nemají tak nízko stropy (*tedy stropy jsou o něco níže, ale Y2K bug je Y2K bug, viz minulé Intro - pozn. Matsoft*).

Co předvedl ve Zlíně on?

Na jeho černé sestavě s CD ROM-kou a hadrem jako již běžnou spectráckou periferií běžel krom jiného celý jeden díl Simpsonů (Akta S), obraz i zvuk přehrávané ze Spectra. Kupodivu se na to dalo koukat. Obrázky byly vyrenderované slušně, i písmenka se dala číst. Zajímavé, co? A to furt slyšíme něco o nedostačujícím rozlišení obrazu.

K vidění byl i MB commander, neustále slibovaný v časopise Your Spectrum od 8 Bit Company. Tritol se nezapře (jak to napsal Matsoft? Softwarový bůh Tritol?). Program je to pěkný a skvělý, rychlý, a efektivní. MBčkáři se mají na co těšit.

Navic už delší čas má Tritol rozpracován D40 commander, to je ten, který na D40 automaticky pozná výměnu diskety a jiné vymoženosti, nevidané ani na PeCi (ještě by to chtělo, aby ji po poklepání na ikonu diskety vyhodila D40 z laufwerku).

Na další spectristické akci se ale bude muset svými výdobytky chlubit někdo jiný. Tritol je tak zaneprázdněn, že neví, kde mu barák stojí a netrefí tak často ani domů. A ono by to změnu chtělo. Vždyť v nás ostatních je přeci také jakýsi potenciál. Máme šanci. Nepromarněme ji.

(to jsou mi řeči...)

Ve Zlíně byla pochopitelně i spousta jiných lidí a o žádném se snad nedalo říct, že by to nebyl spectrista jak se sluší a patří.

I návštěva ze zahraničí byla. Spokojený uživatel z Doxyconu, Tiger's Claw, se dostavil i sem (zajistiv dopravu Hoodovi), stejně jako poslední z F.B.I. a jeho němečtí soukmenovci otevřeli krátek se spectráckými periferiemi, z nichž nejvíce zaujala světelná pistole a hra s hudbou od Tima Follina, která uchvátila hlavně T.D.M. (ale sám v tom nebyl). Lidé si objednali konektory do MIDI/RS 232 konektoru 128ky, občas si někdo koupil nějakou tu kazetu s hrou a +GAMA se opět ukázal jako šťastlivec, když již poněkoličkáte úspěšně odolal pokušení koupit si Multifuck 128.

Nezaháleli ani 8 Bit Company.

Úspěšně se jim podařilo zatajit existenci nové Uly PRO s rozšířenou grafikou (každý bajt pixelů má svůj bajt atributů - tedy celoobrazovkový multicolor, viz Your Spectrum), neviděl ji snad opravdu nikdo.

Stejně tak nikdo neviděl ani demoverzi detektivky The Devil Inside. Asi budeme muset poslat Matsofta, aby to ukradnul, cracknul a pustil do oběhu.

A.I.D.S. získal nějaké ty zakázky na oživení chciplých Specter (no sláva, aspoň k tomu je Eight Bit dobrá). Velká reklama na BS DOS 400 a SFS zabránila bohužel Busymu se zúčastnit - bál se o svůj bídný život, o nějž by jej rozrušení spectristé byli dozajista připravili, oznámili by jim, že práce kvapná málo platná a BS DOS si ještě nějaký ten čásek tvrdě makačky vyžádá.

Aragorn se předvedl se svou kastlí z PeCe, skrývající Tikťaktik M.

A to by mohlo stačit, pokud chcete nějakou pravou Speccy akci zažít na vlastní uši, zúčastněte se už teď v létě nějaké!

LETOS BUDE:

Kaplicon '00

Kaplicon se koná v jižních Čechách, v Kaplici, v Domě dětí a mládeže (nebo jak se to dnes asi jmenuje).

Datem konání je 4. - 6. srpen.

Předpokládají se demo a jiné soutěže a především setkání co největšího počtu žijících spectristů.

Pořádá ho snad skupina CI5 Amateurs nebo IJK (je on členem CI5, nebo není?).

ZX Party 2000

ZX Party 2000 se koná za hranicemi naší vlasti, v polské Wroclawi, tak si přichystejte pas.

Datem konání je 25. - 27. srpen.

Očekávají se demosoutěže s mezinárodní účastí a setkání spectristů na tak trochu vyšší úrovni (rozuměj internacionální, otevřeno je pochopitelně každému).

Pořádajícím je tentokrát skupina Hooy Program.

Zlincon 2000/+2

Zlincon 2000/+2 se koná tam, kde vždycky. Ve Zlíně, nejspíš zas v Domě dětí Astra v Příluku.

Datem konání je poslední víkend před vánoci, 15. - 17. prosinec.

Kromě tradičních demosoutěží je plánována i maxisoutěž MillenniumB, ale jinak půjde nejspíš o zimní období Kapliconu. Přijďte prostě všichni.

MillenniumB

velká soutěž o 3x MB-02+ 512KB RTC proběhne na Zlinconu 2000/+2

MillenniumB má **3 hlavní kategorie**:

- nejlepší MultiTech grafická hra
- nejlepší MultiTech demo
- nejlepší uživatelský program pro MB-02+ MillenniumB má též **4 standardní kategorie**:

galerie:

- nejlepší obrázek (SCREEN\$)
- nejlepší AY/ULA hudba
- nejlepší 4K intro demo
- nejlepší demo

Podmínky soutěže:

Přihlášení účastníci mají pobyt zdarma!

• Přihlásit se můžete DO termínu konání na e-mailovou adresu 8bc@8bc.com. Přihláška: jméno, adresa, kategorie, název a charakteristika projektu.

• Finální projekt musí být prezentován na Zlinconu, posléze bude volně k distribuci.

• Je třeba alespoň tří přihlášených do kategorie.

• MB-02+ budou vydána výhercům ihned.

Brokimsoft dokončil po dvouleté práci hru CRIME SANTA CLAUS DE JA VU. K dispozici máme zatím pouze poslední demo.



Yuri Nazarenko vyrobil nového DIZZYho, jeho výtvar pochází ze začátku roku a nese název DIZZY A.

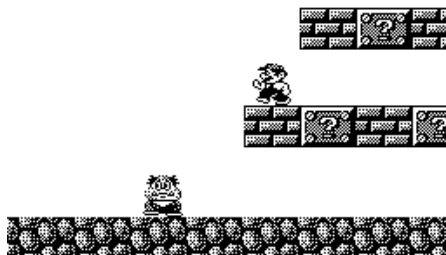


X-Project Association vydala 2in1 neboli dvě hry v jednom balení, první hrou je SNAKE - hadí bitva pro 4 hráče a druhou je předělávka hry SUPAPLEX z PC, pokud někdo neví o co jde, tak to je cosi jako boulder dash. Hra má nevidaných 333 levelů.

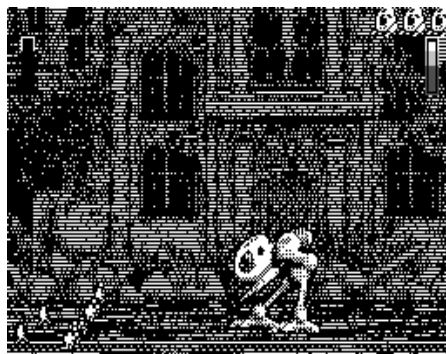


Master Home Computers Group pracuje na hře MARIO, poslední demo, které se k nám dostalo je 0.02. Ke hře je dodáván editor levelů.

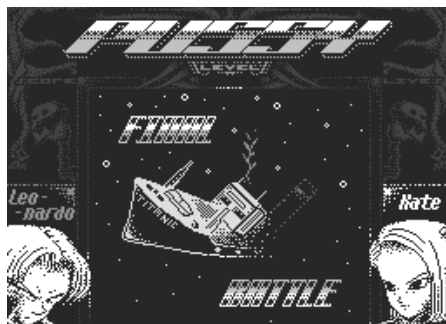
Alien Factory ještě tento rok vydají plnou verzi WALKERA (však už bylo na čase po třech letech).



HW požadavky: ZXS 128 - bez zvuku
PENTAGON nebo SCORPION 256, lépe
512 - se zvukem na COVOX a GS

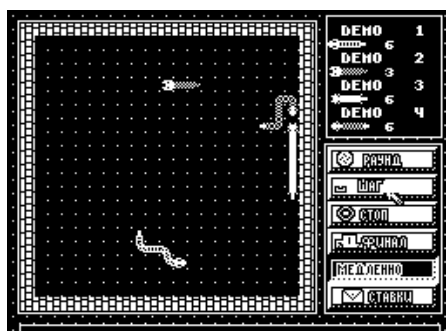


Pravděpodobně posledním dokončeným produktem firmy Fatality bude hra PUSSY - The Titanic Story. SNAPSHOT posledního (?) levelu máme k dispozici, z něj jsou také přiložené screenshoty.



Černá kronika:

Digital Reality se vy...i na DooM.



NoWinky

Black Raven2 od Copper Feeta je také již minulostí. Z krátkého hovoru, který jsem s ním vedl na IRC vyjímám (nebrat zcela doslovně):

Mat: Btw:Kdy bude plná verze BR2?

C_Feet: BR2 nikdy nebude plný.

Mat: Proč? Málo uživatelů ZXS?

C_Feet: No moc jich není, ale to by nevadilo tolik, jako spíš fakt, že málo lidí hry kupuje a většina si je zkopíruje.

Mat: Hmm, to je ale problém, který tu je hodně dlouho, a nebude ani nějaké nové demo? (poslední demo je 0.04)

C_Feet: Asi ne, 0.04 jsem dal dohromady narychlo a už se tím nechci moc zabývat.

Mat: To je škoda, u nás je BR poměrně oblíbená hra.

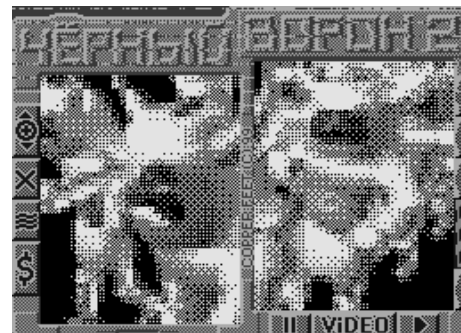
C_Feet: Někde na internetu se dají stáhnout nové mise a editor levelů (tuším, že na www.zx.ru/echo v oddílu games).

Mat: Dík, ale myslím si, že to moc Spectristů neuspokojí, těšili se na BR2.

C_Feet: Jejich chyba, nemají tolik „pirátů“.

Mat: Tak nic, díky za info a BYE.

Obrázek je z animovaného intra.



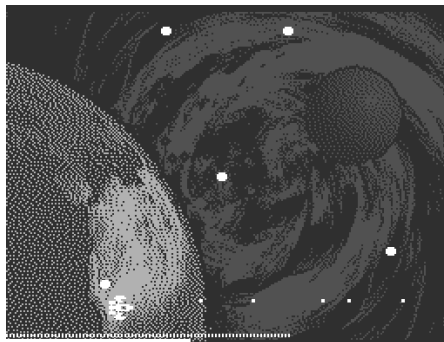
NoWinky sestavil
Matsoft^NC



Rage Technologies vydali třetí díl hry KILL PC. Tentokrát se opět jedná o střelečku, kde pomocí mrštné rakety sestřelujete malé kulaté mršky, které představují PC.

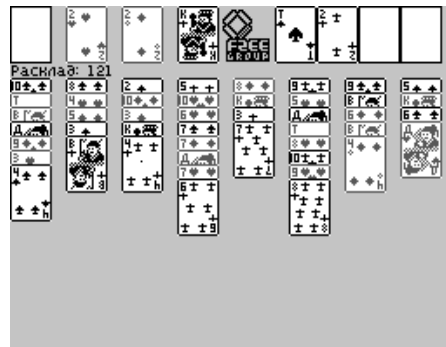


Před každým levelem se objevuje info-screen, kde se dovídáte jaký šrot zrovna likvi-



dujete (Všimněte si úžasné hlášky PC/XT 286, ta se opatuje až do 486). Celá hra běží přes 2 VRAM, v jedné je akční okno, ve druhé pozadí (obrázky byly sesazeny, abyste viděli jak hra opravdu vypadá bez odporného blikání).

Free Group vydala předělávku oblíbené Win95/98/2e3 hry Freecell, takže pokud někdo z vašich rodičů náruživě páří karetní hry, neukazujte mu Freecell, neboť by se mohlo stát, že Speccy bude okupováno třeba 5 hodin v kuse někým jiným než jste vy.



Wild Chars

P.V.L.

Rádi bychom vám ukázali jednu rutinu z dílny P.V.L., ne že bychom si mysleli, že byste ji taky nezvládli napsat, ale když už to jednou na světě je a funguje to, proč vymýšlet už vynalezené aneb lenost je matkou pokroku.

Jedná se o rutinu pro vyhodnocování masky s hvězdičkovou konvencí (pokud to ještě nevíte, tak wildchars jsou „*“ a „?“). Bude se vám určitě hodit třeba budete-li chtít napsat pro svůj diskový systém OPA-Ta, ale i jakýkoliv jiný program pracující s diskem, případně textem (takové hledání v textu nebo v databázi a podobně). Je ozkoušená, jednak Tritolem byla použita v MB Commanderu, druhak ji použil +GAMA ve Fast Commanderu. Oba mohou potvrdit, že funguje a je skvělá.

Od toho, na co jsou uživatelé D40 zvyklí z MDOSu, se ale v jedné podstatné věci liší. Hvězdička značí sice libovolný počet znaků, ale znamená to i žádný znak. Nevíme, zda sebral P.V.L. tento nápad z UNIXu nebo odjinud, ale +Gamovi a Tritolovi se to děsně líbí, je to použitelnější než MDOSová hvězdička.

A začíná se nám zdát, že v podstatě není o čem víc psát. Rutina samotná nepotřebuje ostatně nižší žádné další nápovědy, je jasná a snadno pochopitelná. Nechtě tedy promluví sama za sebe. Si ji užíjte.

```
;wildchars
; ? Jeden libovolný znak
; * libovolný počet znaku
;
;Pr: *?? = min 2 znaky
;
;vstup/
;DE vzor, HL k porovnání
;C počet znaku v DE, B počet znaku v HL

;/vystup
; z vyhovuje, nz ne
;
WILDCHAR inc c
          inc b
          ld a,1
          push af
          jr WCST
WCZP     pop af
          pop af
          pop af
WCHVOP2 dec c
          ret z
          ld a,(de)
          inc de
          cp "*"
          jr z,WCHVOP2
          dec de
          inc c
WCHVOP  push bc
WCST     push hl
          push de
WCHVOP4 ld a,(de)
          inc de
          dec c
          jr z,KON5
```

```
cp      "*"
jr      z,WCPZ
dec     b
jr      z,KON3
cp      "?"
jr      z,WCHVDAL2
cp      (hl)
WCHVDAL2 inc hl
          jr z,WCHVOP4
WCHVZN2 pop de
          pop hl
          pop bc
          inc hl
          djnz WCHVOP
          dec b
          ret
KON3     dec b
          jr WCHVZP2
KON5     dec b
          jr nz,WCHVZN2
WCHVZP2 pop hl
          pop hl
          pop hl
          ret
LENWC    equ $-WILDCHAR
```



Towdieho zná každý, Towdieho hrál každý. Někdo ho i dohrál. Ten někdo jsou šoumeni (a šoumenka) z grupy Crazytronic. A ti vám přináší stručný návod, jak se hrou prokousat.

Towdie

CrazyTronic



Po spuštění hry se před vámi objeví malá zelená příšerka (*já téhle barvě říkám světle modrá, ale nic proti - pozn. +GAMA*) (No, já tedy opravdu nevím, ale +Gama je



asi barvoslepy, neboť Towdie je ve skutečnosti bílý. - pozn. Matsoft). Ničeho se však nebojte a zachovejte klid. I přes svou oblidnost je velmi mírumilovná a poslušná. Jak už někteří jistě správně tuší, není to nikdo jiný než Towdie.



No a teď, když už jste se úspěšně seznámili, můžeme začít hru. Hned na začátku vezme Towdie zelené jablko a lopuchy. Má-li obě věci, skočí o patro níž, kde se ocitne vedle spícího kováře. Sebere kovářský meč a vydá se doleva. Cestou se doporučuje promluvit si s katem a princem. U nich najde Towdie tužidlo, které tam samozřejmě nenechá ležet. Pokračuje v cestě, až dojde ke kuchaři. Ten potřebuje do své polévky zeleninu a tak dostane lopuchy (*zajímavé je, že princezna považuje za ze-*

leninu spíše jablko, vřele vám doporučuji i jiné nesprávné použití předmětů, sice pak hru nedohrajete, ale moc se pobavíte - pozn. +GAMA). Za ně dá Towdiemu mýdlovou vodu. Ten ji však zatím nepotřebuje a může si ji nechat u kuchaře.

Zelené jablko věnuje hladovému ježkovi (*to je ta věc, co vypadá jako námořní mina a nejde se přes ni dostat, hlavně ne poté, co vám jablko zblajzla princezna - pozn. +GAMA*), který mu uvolní cestu. Ve věži osvobodí od prachu hodiny, hodinový skřítek (*ano, přesně ten, co si vytrhal zo zúfalstva obidva vlasy - pozn. +GAMA*) mu z vděčnosti daruje živou vodu (*fakt dobrá*



proti uhrům - pozn. +GAMA). Nechá ji vedle věže a vydá se k princezně (*už dlouho túžila po takomto tužidle - pozn. +GAMA*). U ní najde krém na opalování, cestou k drakovi



Návody

pytlík se zrním a u draka (*je to všechno na draka - pozn. +GAMA*) zdechlou rybu. Rybou Towdie usmrtí pracující trpaslíky (*správně, stihl je trest, a tak zatočíme s každým - pozn. +GAMA*) a projde do další obrazovky. Tou proletí čáp a vytratil pírkó (*ještě že ne dítě, to by byla moc morbidní hra - pozn. +GAMA*), kterým později Towdie probudí kováře.

Vlaštovka dostane krém na opalování, aby mohla odletět do teplých krajin. Za tento dobrý skutek odmění Towdieho prstenem. V této obrazovce si Towdie může zatím ponechat tužidlo a pytlík se zrním. Po rozmluvě s šachistou získá příručku morseovky. Prstenem obdaruje princeznu a po polibku, který za prsten požaduje, zůstane po princezně jen zrcátko (a žába). To musí Towdie napravit a tak jde pro živou vodu (*tak tohle jsou ty uhry? -*

pozn. +GAMA). Polije princeznu (žabu) a tím ji osvobodí.

Nyní si vezme toaletní papír, který potkal na cestě k trpaslíkům, a odnese ho

princi. Ten na něj napíše vzkaz, popsany lístek vrátí Towdiemu a přidá ještě talisman. Obě věci si tady zatím ponechá. S příručkou morseovky se vydá k majáku, rozluští šifru a získá lístek s číslem. Donese ho katovi, který jej však již nepotřebuje, ale hodil by se mu od kováře pilník. Towdie se tedy vydá ke kováři, vzbudí ho pošimrá-

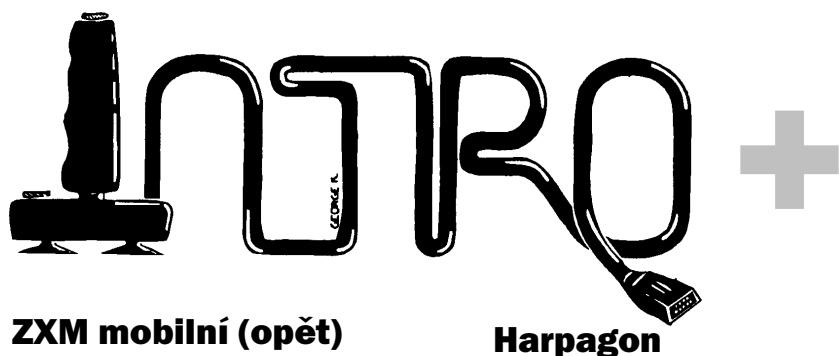
die talisman a popsany lístek a rozběhne se k šachistovi. Předá mu talisman, za který mu šťastný šachista věnuje plynovou masku. S její pomocí snadno přejde palouček smradlavých květin a na jejím konci si jednu utrhne. Teď už má vše potřebné, aby se mohl vydat do věže u vlaštovky (ta už se samozřejmě někde opaluje). Na duchy po-

užije tužidlo, těm ztuhne úsměv na tváři. V další věži předá holubici popsany lístek a protože je zesláblá hladem, tak i pytlík se zrním (*sypte ptáčkům - pozn. +GAMA*). Valymu dá čuchnout květiny, jenže on nemá smradlavé kytky rád, proto raději změni lokaci svého výskytu a zůstane po něm jen dělová koule. Pomocí zrcátka od princezny zničí Towdie čaroděje, jenž po něm rozzuřeně vrhá blesky. Potom donese Valymu, jenž utekl k drakovi, mýdlovou vodu, aby divoké sani zblblnul hlavu.



ním pírkem pod nosem, ale pilník nemůže dostat, pokud kovář nebude mít křemeny. Towdie mu je donese a pilník dostane. Předá ho katovi a ten mu za to dá pro kováře kladivo. Netřeba je zatím vracet, lépe je si toto nářadí ponechat. Teď už vezme Tow-

die No a nakonec, jak už to v pohádkách bývá, zabuší na zvonec (*jasně, kladivo - pozn. +GAMA*) a Towdieho je konec.



Hosana!

Při sazbě ZXM se podařila věc nevídaná, Spectristickými božstvy kladně přijímaná. Počet "článků", tedy sázecích oddílů textu, dosáhl v okamžiku dokončení rovných 128, slovy einhundertachtundzwanzig... (bledé tváře majitelů Pentashitů 512 teď ještě více blednou).

Zdrcnutí

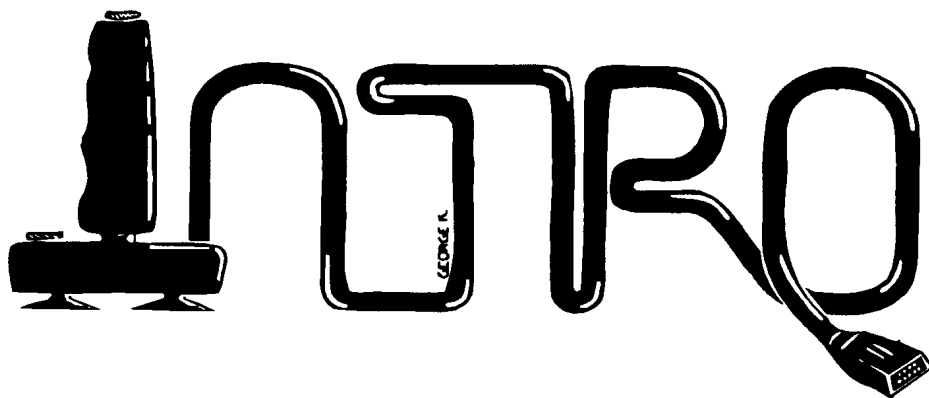
Matsoft při jedné ze svých výprav do různých koutů Prahy zabloudil i na Palmovku, při pěší cestě na jedno z blízkých škodováckých vrakovišť narazil na dvě známé tváře, kterými byly xichty nechvalně známých vůdců 8BC, tedy Martina Blažka a Honzy Kučery. Sražení se neobešlo bez konzultace o obou časopisech, ale hlavní věci, která by uchvátila asi každého (rozumného) Spectristu byla tato věta vyřčená 8BC ihned po pozdravu: "Konečně vidíme nějakého normálního člověka"

Matsoft si stihl udělat papíry, avšak nikoli na hlavu (ty by potřeboval, aby ho přestali otravovat vojáci), nýbrž na auto, je tudíž držitelem řidičského průkazu (*pokud ovšem bude stále jezdit jako prase, tak o něj brzy přijde. - pozn. Dopravního Inspektorátu*) Předtím si ale stačil pořídit další Škodivku, tentokrát 120L, tu ale po ujetí 3500 km s jednou malou nehodou prodal a pořídil 105S z roku 1977, tou pravděpodobně přírčí i na Kaplicon"00. (*Já ji viděl - nechceš mi ji, Matěji, v nějakém záchvatu štědrosti darovat? - pozn. +GAMA*)

Houby dostaneš, a ne škodovku! Marně se těšíš na záchvat! - pozn. Matsoft

Dement totální

Matsoft od Zlínconu"99 (pořadatelé ho v záchvatu euforie z blížících se nul v leto-počtu přejmenovali na Zlíncon"00, takže nám jeden ročník této sleziny jaksi chybí) nevybalil Speccy, tedy on ho vybalil (slovy jedenkrát) na FOREVERu 2e3, kde na něm něco málo udělal a před odjezdem zapackoval opět do tašky, kde leží dodnes.



Ví, jak na to

Po přečtení chvály na svou osobu se P.V.L. rozhodl zopakovat své úspěchy na poli packovacích programů. Jak nám sdělil, jeho nový DELETE PACKER 4.21 je "revoluční aplikací ztrátové komprese na ZX Spectru".

K jeho principu dodal jen tolik, že "název odpovídá funkci".

Ví, jak na něj

Jak nám sdělil Tritol, jeho dosud nedokončený MB Commander dokáže smazat vaše data mnohem rychleji než nějaký DELETE PACKER 4.21 - smazání 40 000 souborů na pokusné MB-disketě mu trvalo asi 15 minut.

Hardware

Jak nám sdělil Matsoft, on dokáže s pomocí jednoduchého hardwarového přípravku (magnet) mazat ještě rychleji - smazání 40 000 souborů na +GAMově disketě se zdrojčky do Promethea mu trvalo asi 5 sekund.

Hell

Podle toho, co nám sdělil Starsoft, nastoupil AIDS of 8BC do Green Hell. Stejně jako jiní před ním, ani on neuposlechl dobře míněnou radu "neblázni, emigruj". Protože kvůli utajení nesmí posílat dopisy a balíčky, nemůže poslat Starsoftovi jeho ZX +2A, které mu opravil.

Green Hell nebo jeho mírnější obdobu absolvovali takoví velikáni, jako M.B.&D.G., Bell, Aragorn, Factor 6, nyní se očekává návrat DiHalta.

Matsoft s ohledem na svůj zhoršený zdravotní stav (familiární hyperbilirubinémie Gilbertova typu) poukazuje na to, že jeho Blue Hell je mnohem horší. Kéž by se ho opravdu dočkal...

Je to blbec

Poté, co Zilog alias Stupidius pustil zdroják svého packeru mezi lidi (viz jinde v tomto čísle), zmocnil se jednoho exempláře D.J. z P. Pochopitelně, jak je jeho dob-

rým zvykem, si ho kousek umazal (ne zaspínal, ale smahl), a tudíž ho nemá ani celý, ani funkční.

Johnny-X ho prosil, ať mu packer pošle. Nic nedostal.

Hood ho prosil, ať mu packer pošle. Nic nedostal.

D.J. z P. nyní vzývá Alláha, aby Zilogovu záložní kopii zachoval ráčil a jemu ji poslal.

F-19 128

Tento simulátor je ve 48 verzi dost rozšířen, bohužel jeho na Internetu se vyskytující 128 verze, která obsahuje totálně boží zvuky, je nefunkční.

Johnny-X jen tak mimochodem vytvořil funkční 128 verzi.

D.J. z P. mimochodem si jí kousek umazal a teď vzývá Alláha, aby Johnnyho kopii zachoval ráčil a jemu ji poslal.

Johnny-X mu ji asi pošle. Asi. Ale až dostane packer!

Zabodoval

Tritol si zaslouží pochvalu.

Poté, co Matsoft instaloval půl hodiny fonty, zjistilo se, že ani jeden z několika set těchto fontů není pro ZXM použitelný. Inteligentní Windows problém vyřešily hravě a všude dosadily krásný a estetický font Courier, kterýžto sliboval, že čtenáře ZXM při pohledu na tuto úpravu stihne tranzitorní ataka nevolnosti.

Tritol slíbil +GAMovi smrt, vysázeli ZXM tímto fontem, a aby se moc neušpínal od krve, poslal okamžitě všechny potřebné fonty e-mailem.

Word

Je to pytel sraček a dál nemá cenu se tímto softwarovým kancelářským nástrojem zabývat.

Už to bude!

Jak nám donesl nejmenovaný donašeč, Busy dokončil další část BS-DOSu 400, který nejmenovaná osmibitová firma slibuje už asi pět let.

Kolik částí ještě zbývá dokončit prý ne-

tuší ani sám Busy.

Hlavně nemá na programování čas.

Busy je busy.

Stále tvoří

Starsoft nelení a nezláká se. Je softwarově činný. Sice si pořídil domů PC (=Pomocný Computer ke Spectru), ale také si ke svým dvěma Spectrům vypůjčil třetí, na němž hodlá dokončit návod na hry Hero Quest 1 a 2.

Zatím má hotové intro (neověřené informace tvrdí, že zaplní natříkrát celou 128ku) a usilovně paří, aby se mu podařilo především druhý díl Hero Questa dokonale zmapovat.

Stále netvoří

Johnny-X poté, co převedl ruský dungeon Last Hero of the Light Force na MB-02+, plánuje tvorbu dungeonu českého.

Zatím kreslí netvory. Samé netvory. Prostě a jednoduše netvoří.

(Možná je přihlásí do Graphic compa na některé letní ZX party.)

AY-Tracker

Johnny-X prohlásil ruský tracker ASM (ASC Sound Machine) za hnusný. Podobně prohlásil za hnusný i SQ-Tracker, Sound Tracker Pro, Pro Tracker 3.3 a Pro Tracker 3.4f. Dále prohlásil za hnusný Master Tracker, všechny ruské trackery a všechny trackery vůbec.

Asi si bude muset napsat vlastní tracker.

DEMENTI

+GAMA prohlásil za dementia Matsofta.

Ani ne tak kvůli jeho hardwarovým pokusům s disketou a magnetem (ty zdrojčky prý byly stejně dementní - tím se řadí také mezi dementy), jako spíš proto, že dlouho nebyl schopen dodat důležité materiály pro tvorbu ZXM.

A především byl Matsoft prohlášen za dementia proto, že mu je úplně jedno, nominoval-li jej někdo na dementia, čili nic.

Za zcela tradičního dementia byl prohlášen Bill Gates.

Za dementia netradičního byl Matsoftem prohlášen +GAMA.

Má to být nejen pomsta za jmenování dementem, ale také za to, že se dotýčný schválně nechává ve škole vyhazovat od zkoušek, aby se musel učit a neměl tak čas se věnovat ZX Spectru.

Pokud Johnny-X nenapíše do ZX Magazínu návod na Word Master, bude také jmenován čestným dementem.

Magorem (ve zkratce mgr.) byl jmenován Hood, ale za to my fakt nemůžeme.

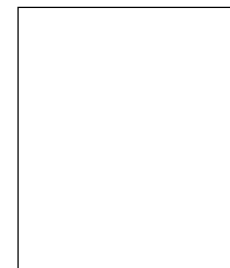
Odesílatel:

Matěj Kryndler

Lotyšská 8

Praha 6

160 00



Příjemce: