

proxima
Veřejná Obchodní Společnost

Monitor a disassembler
Kopírovací programy
Grafický monitor

USER - 1

URČENO PRO POČÍTAČE : DELTA
SINCLAIR ZX - SPECTRUM /+/128
DIDAKTIK GAMA, DIDAKTIK M

INSTRUCTION MANUAL

(c) 1990

POZOR!

Čtete pozorně licenční podmínky firmy **PROXIMA** předtím než porušíte obal diskety (kazety). Počítačový program zaznamenaný na disketě (kazetě) je autorským dílem chráněným ustanovením čs. autorského zákona a mezinárodními smlouvami. Porušením obalu diskety (kazety) se zavazujete dodržovat ustanovení následující smlouvy mezi Vámi a firmou **PROXIMA**.

Podle této smlouvy na Vás nepřechází vlastnická práva k software, ale získáváte právo pouze užívat software po dobu platnosti této smlouvy podle dále uvedených podmínek.

Licenční ujednání

1. Uživatel je oprávněn instalovat a provozovat počítačový program na jediném počítači a smí si pořídit jedinou bezpečnostní kopii nosného média.

2. V případě zakoupení multilicenční dodávky počítačového programu je uživatel oprávněn instalovat a provozovat program na takovém počtu počítačů, jaký je uveden v multilicenční smlouvě.

3. Firma **PROXIMA** neručí za bezvadný chod programu na amatérsky upravených počítačích a počítačích spolupracujících s nestandardními perifériemi včetně interface vlastní výroby.

4. Nehodláte-li respektovat ustanovení této smlouvy, vraťte software v neporušeném obalu tam, kde jste jej získali. Bude Vám vrácena částka kterou jste zaplatili.

Monitor a disassembler

DEVAST ACE

DEVAST ACE je nejnovější monitor a disassembler pro ZX Spectrum a kompatibilní počítače. Pokud programujete ve strojovém kódu Z80, poskytnete Vám **DEVAST** neocenitelné služby, pokud chcete se strojovým kódem začít, pak Vám jej vřele doporučujeme.

Slovo **DEVAST** je sestaveno z prvních písmen slov **D**ream **E**xtension of **V**iewer of **A**ssembly language and **S**creen located **T**racing utility (tento název má z valné části na svědomí Vám jistě nikoliv neznámý František Fuka, původní **VAST** pak Tomáš Rylek neboli T.R.C). Program **DEVAST** vznikl přepracováním **VASTu** - původní program byl zkrácen, mnohé části byly napsány zcela nově, bylo přidáno celkem 14 nových funkcí a možností, úplně nově vznikla relokovatelná verze pro uložení v paměti. Při vytváření **DEVASTu** byl použit ladící systém **PROMETHEUS** a bohaté zkušenosti, které jsem při jeho psaní získal (obzvláště při psaní monitoru k Prometheovi).

Program **DEVAST** je dalším pokračováním známé řady obrazovkových monitorů pocházející z dílny již zmíněného **T.R.C: MIKROMON, MILIMON, PIKOMON** a **VAST**. Na rozdíl od svých předchůdců není **DEVAST** dílem **T.R.C** ale **UNIVERSUM**.

Příponu **ACE** jsme později k programu přidali proto, že je mezi dostupnými programy bezkonkurenční (za další důvod lze považovat skutečnost, že dnes se neustále hovoří o morální, kulturní, ekonomické a dalších devastacích a tak jsme nechtěli zůstat pozadu - vytvořili jsme devastaci programovou).

Mezi uživateli ZX Spectra se pohybuje starší verze programu **DEVAST ACE** pod názvem **DEVAST+**, v této verzi sou dvě chybičky, které se projeví jen velmi zřídka a trvalo několik měsíců, než byly odhaleny. Chyby nemají destruktivní účinky. V programu **DEVAST ACE** jsou chyby samozřejmě opraveny.

Na kazetě máte 2 verze programu:

DEVASTACE+ - tuto verzi můžete nahrát na libovolnou adresu včetně adresy 16384 a z této jej spustit, program se přenesou do obrazovky a v ní pracuje. Pokud jste nesmazali nahraný program v paměti, můžete jej opět spustit (přenesou se znovu do obrazovky). Tuto verzi můžete také nahrát z programu **WLEZLEY 7**, její délka je 4600 Bytů. Pokud budete nahrávat program přímo na adresu 16384, musíte zadat nahrání i spuštění najednou, protože jinak se program poškodí: **LOAD "DEVASTACE+"SCREEN\$:RANDOMIZE USR 16384**

DEVAST ACE - toto je paměťová verze programu, lze ji nahrát a spustit od libovolné adresy větší než 23296. Nahrajte program na zvolenou adresu a odstartujte jej z ní, vymaže se obrazovka a objeví se nějaké nápisy a hlavně **Starting address:adresa**. Toto je adresa, na kterou se program umístí. Nyní můžete tuto adresu přepsat a potvrdit stiskem klávesy **ENTER**. Po potvrzení se provede relokace (přepočítání adres) a spuštění, nyní je možno program spouštět na adrese, kam byl přemístěn. Pokud budete chtít program umístit již na adresu 23296, tedy tak, že přepíše systémové proměnné **BASICu**, nahrajte ho na vyšší adresu a před potvrzením přepište adresu na požadovanou hodnotu (při takovém umístění se samozřejmě nelze vrátit zpět do **BASICu**) Délka této verze je 5700 Bytů, po relokaci však program zabírá asi o 1000 Bytů méně, tedy chcete-li, aby byl program umístěn na konci paměti a pod ním zbylo místo, nahrajte jej na adresu 59800 a před potvrzením přepište adresu na 60800.

Opětne spuštění, obrazovkovou verzi (**DEVASTACE+**) lze spustit v obrazovce, pokud nedošlo k jejímu poškození (výpis hlavičky...). Na adrese 16384 je teplý start (při startu se nezmění hodnoty registrů v monitoru), na adrese 16387 je start studený (při startu se uloží

hodnoty registrů takové, jaké byly při zavolání programu). Po nahrání se provádí studený start. Paměťovou verzi (**DEVAST ACE**) lze také spustit oběma způsoby, od začátku nebo o tři byty dál.



Na obrázku vidíte čelní panel při práci s programem DEVAST ACE, na horním řádku je vypsaná aktuální adresa a jedna disasemblovaná instrukce, pod ní jsou vypsané registry a jejich hodnoty, pod nimi je vypsan stav flag registru a úplně vlevo je výpis stavu přerušeni. Vpravo je vypsané šest adres a jejich obsah jako číslo a jako znak. Jedna z adres je zvýrazněna, je to aktuální adresa. Pod tímto výpisem je editační zóna a kurzor.

Přístup k paměti (aktuální adresa) a pomocné funkce:

M - po stisku této klávesy se objeví otázka **Memory:** a můžete vložit novou hodnotu pro aktuální adresu (po odeslání se obnoví výpis čelního panelu). Adresu zadáváte v takové číselné soustavě, která je nastavena pro výpisy (u šestnáctkové soustavy vkládáte maximálně 4 číslice a znaky **A** až **F**, u desítkové maximálně 5 číslic - pozor na záměnu číselných soustav, může mít nepříjemné následky).

CS+6 - odečtení 1 od aktuální adresy.

CS+7 - k aktuální adrese se přičte délka instrukce na aktuální adrese, ta se tedy posune na další instrukci.

ENTER - pokud je editační oblast prázdná, přičte se k aktuální adrese 1, pokud je v editační zóně napsáno nějaké číslo, bude uloženo do paměti na aktuální adresu a k té adrese se přičte 1. Je-li číslo větší než 255 (nelze ho zapsat do jednoho bytu), uloží se pouze jeho spodní byte (zbytek po dělení 256).

SPACE - funkce podobná té předchozí, liší se tím, že ukládá hodnotu jako dvojbytovou (**word**) a posune aktuální adresu o 2. Pokud je editační zóna prázdná, dojde pouze k posunu aktuální adresy. Hodnota je ukládána obvyklým způsobem, nejprve Low a potom High.

SS+3 - přepínání výpisů čísel a vstupů ze soustavy desítkové do soustavy šestnáctkové a naopak.

CS+1 - přepínání výpisů z malých písmen na velká a zpět.

CS+2 - přepínání výpisů z obyčejných písmen na tlustá a zpět. Tuto možnost jistě oceníte, pokud je obrazovka vaší televize již poněkud unavena.

CS+3 - přepínání třetiny, do níž budou směřovat výpisy, tato funkce není u obrazkové verze, tam je výpis prováděn do spodní třetiny a nastavení nelze změnit.

CS+0 - smazání obrazovky, nastaví bílý papír a černý inkoust. Tato funkce funguje opět jen u paměťové verze.

SS+A - po zadání dvou čísel (**First**, **Last**) se vypíše jejich součet, rozdíl a další informace v desítkové i šestnáctkové soustavě, kladně i záporně.

SS+Q - návrat do BASICu. Pokud byl monitor volán z programu, bude program pokračovat dalším příkazem.

Vnořování a jednotlivé druhy paměťových výpisů:

CS+8 - vnoření. Při prohlížení programu se často objeví potřeba nahlédnout do podprogramu nebo do jiného místa paměti, ale neztratit přitom současnou aktuální adresu. K tomuto účelu slouží vnořování. Při jeho použití uloží program na zásobník současnou aktuální adresu a zeptá se Vás na novou hodnotu (**Memory**). Na zásobník se vejde celkem 8 adres, můžete se tedy vnořit vícekrát.

CS+9 - vnoření; na rozdíl od předchozí funkce se tentokrát program neptá na novou aktuální adresu a vezme si ji z adresní části instrukce na současně aktuální adrese. Například na aktuální adrese je instrukce **call #4C2** a vy se na tuto adresu chcete podívat. Mohli byste stisknout **CS+8** a vložit **#4C2**, ale lepší bude, když stisknete jen **CS+9** a dosáhnete téhož účinku.

CS+5 - vnoření, opačná funkce k předchozím dvěma, program odebere ze zásobníku adresu a vloží ji do aktuální adresy.

SS+C - vypíše zásobník adres pro vnoření a čeká na stisk klávesy.

U - také vypíše zásobník (5 adres), ale tentokrát ten, na který ukazuje registr **SP**. Toto je užitečné při trasování nebo při prohlížení tabulek dvoubytových čísel (nutno nastavit **SP** na její začátek).

SS+4 - výpis disasemblingu od aktuální adresy, vypíše se 8 řádků a čeká se na stisk nějaké klávesy. Podle toho o jakou klávesu se jedná, bude provedeno toto:

CS+1 nebo **SS+4** - návrat zpět k hlavnímu panelu

5 - návrat na začátek výpisu (v tomto případě na aktuální adrese)

6 - výpis další stránky

7 - návrat o řádek zpět ve výpisu

N - vyhledání dalšího výskytu zadané posloupnosti (viz vyhledávání)

SS+3 - přepnutí používané číselné soustavy - pracuje pouze u paměťové verze

jakákoliv jiná klávesa - výpis dalšího řádku

(uvedené funkce platí pro všechny další výpisy)

V - funkce stejná jako předchozí, výpis však nezačíná na aktuální adrese ale na adrese zadané na dotaz **First**.

O - znakový výpis od aktuální adresy, vypisuje se adresa a za ní 25 znaků, znaky jejichž kódy jsou větší než 128, jsou vypisovány inverzně. Znaky, které nemají grafickou podobu, jsou nahrazeny tečkou (kódy 0 až 31 obyčejnou a 128 až 159 inverzní).

SS+0 - stejný výpis jako **O**, pouze počáteční adresa se zadává na dotaz **First**.

L - číselný a znakový výpis od aktuální adresy, vypíše se adresa, hodnoty 5 bytů na této adrese a 5 znaků jim odpovídajících.

SS+L - jako **L**, opět od adresy zadané na dotaz **First:**.

K - výpis BASICu od aktuální adresy. Jsou vypisována klíčová slova, grafické znaky a za kódem 14 je vypsána číselná hodnota, se kterou program v BASICu skutečně pracuje.

SS+K - jako **K** od adresy zadané na dotaz **First:**.

Přesuny a plnění bloků paměti, textový editor:

I - po zadání první adresy bloku (**First:**), poslední adresy bloku (**Last:**) a adresy, na niž má být blok přenesen (**To:**) dojde k přenesení bloku paměti. Počáteční a koncová poloha bloku se může překrývat, program vybere odpovídající způsob přenosu (**ldir, lddr**).

SS+I - doba předchozího příkazu, počáteční blok je určen první adresou bloku a jeho délkou (**Lenght:**).

P - vyplnění bloku paměti zadaného první (**First:**) a poslední (**Last:**) adresou zadaným číslem (**With:**).

SS+P - doba předchozího příkazu, blok je nyní zadán svým začátkem (**First:**) a délkou (**Lenght:**).

T - textový editor, umožňuje vkládat sekvenčně libovolné ASCII znaky, editor se opouští pomocí **CS+1** a adresu lze měnit stiskem **CS+6** a **CS+7**.

Magnetofonové operace:

J - nahrání bloku dat z kazety do paměti, blok je určen první (**First:**) a poslední (**Last:**) adresou. Monitor vynechává při ukládání tu část paměti, ve které je uložen. Leader bloku se nerozlišuje.

SS+J - doba předchozího příkazu, blok je určen první adresou a délkou (**Lenght:**).

S - save; uloží na kazetu blok paměti zadaný první (**First:**) a poslední (**Last:**) adresou pod značkovým bytem (**Leader:**).

SS+S - také save, blok je určen první adresou a délkou (**Lenght:**).

R - zjistí značkový byte (**leader**) prvního souboru na kazetě.

V - načtení hlavičky, vypíše se tyto informace z hlavičky:

typ (**T:Basic, Data, Code**)

jméno v hlavičce (**N:**)

start (**S:**) u Code je počáteční adresa, u Basic jde o startovní řádek

délka bloku dat (**L:**)

délka proměnných pokud se jedná o Basic (**V:**), u Code nevyužito

Na stisk **J** dojde k nahrání bloku dat na adresu **S:** v případě, že se jedná o Code, nebo na adresu **23755** v případě, že se jedná o Basic. Stisknete-li **M**, budou data nahrána na adresu **16384** a z této adresy spuštěna, tato funkce je určena pro nahrání programů **Wlezley 7** nebo **Packmaker**. Při nahrání Basicu se nenastavují systémové proměnné, při návratu nebude

program spustitelný. Pokud chcete nahrát Basic, který se na **MERGE** hroubí, zadejte **LOAD**, po přečtení hlavičky stisknete **BREAK**, nahrajte nebo spusťte DEVAST ACE a znovu načtete hlavičku (**W**). Stiskem **J** nahrajte program a nyní se můžete do programu vrátit bez obtíží.

Vyhledávání v paměti:

C - zadání 5 bytů dlouhé posloupnosti s možností vynechat libovolné byty stiskem **ENTER**. Stisknete-li místo čísla klávesy **SS+Z**, bude vložen kód znaku, který stisknete, stisknete-li klávesu **Z**, nejprve zadáte znak a potom masku (**Maski**). Při hledání se budou porovnávat pouze ty bity, které obsahují v masce jedničku, stisknete-li klávesu **X** místo klávesy **ENTER**, zeptá se program ještě na masku. Pokud Vám není použití masky jasné, klidně na ni zapomeňte, snadno se bez toho obejdete. Zapamatujte si však vkládání znaků, je užitečné při hledání textů. Hledání probíhá od aktuální adresy a po nalezení se její hodnota změní.

N - vyhledání dalšího výskytu hledané posloupnosti. Tuto funkci lze vyvolat také v libovolném druhu výpisu. Hledání probíhá dokud se posloupnost nenajde a ta se najde vždy, alespoň v programu DEVAST ACE, proto nalezení porovnejte s polohou DEVAST ACE, jinak se Vám může stát, že budete přemýšlet zcela zbytečně nad smyslem toho, co jste našli.

SS+C - zobrazí informace o hledané posloupnosti včetně masky.

Krokování, nastavování registrů, breakpoint a volání jiných programů:

SS+Z - provedení jedné instrukce na aktuální adrese a posun na další instrukci (při provádění je přerušeno povoleno nebo zakázáno podle nastavení).

SS+X - provádí jednu instrukci, jako **SS+Z**, pouze instrukce **Call** a **Rst** se provedou i s volanými podprogramy.

SS+V - provádění **SS+Z** až do stisku **BREAK**. Trasování je možno urychlit stiskem klávesy **F**, přestanou se vypisovat hodnoty registrů a opětným stiskem **F** se obnoví předchozí stav.

SS+B - provádění **SS+X** až do stisku **BREAK**, ostatní jako u předchozí funkce.

SS+N - nastavení registrů. Po stisku se jedno jméno registru vypíše inverzně. Nyní můžete stisknout klávesu **ENTER** a vložit novou hodnotu (**With**), u flagů a indikátoru stavu přerušeno se stiskem **ENTERu** okamžitě změní hodnota na opačnou, stiskem **5** a **8** se posunete na předchozí nebo následující registr nebo indikátor. Nastavování ukončíte stiskem **CS+1** nebo **SS+4**.

W - uloží aktuální adresu pro potřeby **SS+U**.

SS+U - funkce vyzvedne tři byty na aktuální adrese, uschová je a na jejich místo vloží skok zpět do monitoru (**BreakPoint**). Potom se nastaví všechny registry na hodnoty v čelním panelu, případně se povolí přerušeno a provede se skok na adresu zadanou stiskem **W**. Až program dojde na místo s Breakpointem, vrátí se zpět do monitoru. Uloží se všechny registry a na místo se vrátí odebrané 3 byty. Z popisu je vidět, že tento způsob nelze použít v oblasti ROM, je také nutno ověřit, že žádný ze tří bytů, které jsou přepsány, není pro běh programu potřebný.

SS+H - volání podprogramu na zvolené adrese (**To:**). Před spuštěním se program zeptá, zda je volba jistá (**Yes:**), v kladném případě se odpovídá stiskem **Y**. Při volání se nastaví hodnoty registrů a případně povolí přerušování, po návratu (pokud k němu dojde) se hodnoty registrů opět uloží. Poznámka v závorce je důležitá, pokud se Vám program při použití tohoto a některých dalších příkazů zhroutí, nehledejte chybu v programu DEVAST ACE ale v sobě. Program DEVAST ACE je velice důkladně a dlouhodobě vyzkoušen a možnost, že byste snad objevili nějakou podstatnou chybu, je vyloučena!

SS+T - přímý skok do programu, nastavení stejné jako u předchozí funkce. Použití přichází v úvahu při spuštění jiného programu, kde se nepředpokládá návrat zpět nebo v případě, že je návrat zajištěn jinak (do programu byl připsán skok zpátky atd.).

Disassembler **DEVAST ACE** zobrazuje poloviny **IX** a **IY** jako **HX**, **LX** a **HY**, **LY**. Rotaci **CB30** až **CB37** značí **SLIA**. Ve skupinách **DDCB** a **FDCB** tiskne post kopie, jako např. **SRL A,(IX+5)**. Po instrukci **RST 8** se vypisuje jeden **DEFB byte**, po **RST 28** se tiskne **DEFB** až do **DEFB 38** (konec výpočtu).

Závěrečné doporučení:

Při práci se strojovým kódem platí více než kdekoli v jině přísloví **Třikrát měř, než se jednou řízneš**, snad jen pyrotechnik může jedním jediným chybným pohybem napáchat větší škody. Tento program se výborně hodí k hledání poke do her, k drobným úpravám jiných programů i pro ladění vlastních výtvorů v okamžiku, kdy se pro nedostatek paměti nedá použít žádný jiný program. Není však vhodný pro psaní programů delších než nějakých dvacet bytů, pro tyto účely se hodí program s poněkud většími možnostmi (pokud takový program postrádáte můžeme Vám nabídnout ladičí systém PROMETHEUS - assembler a monitor pro tvorbu programů - nelepší na ZX Spectrum).



PROMETHEUS - ladičí systém určený právě pro Vás!

Assembler - délka 11 KB, relokovatelnost, komprese zdrojového tvaru (poměr 3.5:1, tedy zdrojový text je asi 3.5 krát delší než strojový kód, pro srovnání nerozšířenější assembler GENS má poměr 8:1), tabulka symbolů stále k dispozici k nahlédnutí, rychlost kompilace 2-3 KB/s - tedy prakticky okamžitě, pohodlný editor s kontrolou syntaxe, blokové operace, vyhledávání, kompatibilita s programem GENS, možnost překládat i rozsáhlé programy.

Monitor - délka 5 KB, (pouze spolu s Assemblerem dohromady 16 KB), umí vše, co DEVAST ACE, navíc další trasovací režimy, dokáže kontrolovat instrukce a chránit před přepsáním při trasování sebe i assembler a zdrojový text, možnost chránit zvolené části paměti před zápisem, čtením a během (skok nebo volání kódu v této oblasti okamžitě způsobí chybové hlášení), zcela libovolně přestavitelný čelní panel, jako jediný dokáže při trasování zjistit časovou náročnost programu - počítá cykly procesoru (tzv. T-cykly).

Na tomto systému vznikly všechny nové programy! **DEVAST ACE**, **ORFEUS**, **KRTEK**, **BAD DREAM**, **EXPEDICE**, **CARGANTUA**, **PANTAGRUEL**, **GOLIATH**, **DAVID**, **WLEZLEY**, **DESKTOP** (textový editor, na němž je psán tento text), a mnoho dalších, ostatně sám **PROMETHEUS** vznikl "sám na sobě", do vzniku první provozuschopné verze byl používán GENS a zdrojový text byl dohromady (2 kusy) dlouhý přes 40 KB. Po přenesení se zkrátil na 19 KB, navíc odpadla potřeba asi 6 KB na tabulku symbolů.

Pro začátečníky ale nejen pro ně připravujeme knížku **Assembler a ZX Spectrum**, kde bude vše, co o strojovém kódu a programování v něm víme (během léta 1991).

Grafický monitor a editor

WLEZLEY 7

Program **WLEZLEY** vznikl aby usnadnil různé grafické práce. Lze jej používat k vyrábění fontů, kreslení a opravování obrázků a spritů, prohlížení a editování textů a BASIC programů, k překládání her do češtiny či slovenštiny a mnoha dalším účelům. Jeho velkou výhodou je uložení do obrazovky, nepotřebuje tedy žádný další prostor pro svou práci. Jak napovídá samo jméno - lze jím zjistit mnohá tajemství ostatních programů.

Nahrání a spuštění programu:

Program je na kazetě uložen jako blok CODE o délce 3960 bytů. Můžete si jej nahrát na libovolnou adresu a z této adresy jej spustit - pokud chcete používat program k prohlížení her a nechcete přemazat nic, nahrajte program na adresu 16384 (**LOAD "" SCREEN\$**). Chcete-li například upravit znakový soubor a máte v paměti alespoň 4KB místa, nahrajte raději program do tohoto místa a můžete jej spustit několikrát.

Hlavní menu:

Po spuštění programu se smaže obrazovka - přesněji smaže horních 9 řádků a zbytek se zakryje atributy (pod nimi je nyní běžící program uložen) - a doprostřed horní třetiny se vypíše nápis **Wlezley 7 (C) 1991 UNIVERSUM**. Nyní stiskněte jednu z následujících kláves:

S - save, ukládání na kazetu (**SS+S**)
J - load, čtení z kazety (**SS+J**)
Y - čtení hlavičky
P - picture, editor obrázků a spritů
T - texteditor, editor textů
K - prohlížení BASIC programů
CS+B - návrat do BASICu
CS+R - návrat do volajícího programu

Všechny operace přístupné z hlavního menu jsou podrobně popsány v dalším textu v pořadí, v jakém jsou uvedeny zde.

Kazetové operace:

Program **Wlezley** umožňuje ukládat libovolné úseky paměti na kazetu a zpětně číst opět libovolné bloky z kazety do paměti. Program sám sebe chrání před přeháním a lze tedy nahrávat i bloky začínající od adresy 16384.

Příkazy jsou tyto:

S - save, uložení bloku paměti, který je zadán prvním (**first**) a posledním (**last**) bytem na kazetu, před ukládáním se program ptá na značkový byte (**leader**). Na tuto otázku můžete odpovědět dvěma způsoby: zadáte číslo (obvykle 255) a stisknete ENTER, nebo napíšete dvojtečku a jméno a opět stisknete ENTER. V prvním případě se na kazetu uloží bezhlavičkový blok zadaných parametrů, v druhém případě se na kazetu nejprve uloží

hlavička typu CODE, která obsahuje jméno, startovní adresu a délku, po hlavičce se na kazetu uloží zvolený úsek paměti se značkovým bytem 255. Toto je zcela ekvivalentní příkazu **SAVE "jméno" CODE first,last-first+1** v jazyce BASIC.

S+Symbol Shift - obdoba předchozího příkazu, pouze úsek paměti je nyní zadán prvním bytem a délkou (**length**).

J - load, načtení bloku z kazety do paměti, místo kam se má blok uložit, je určeno prvním (**first**) bytem a posledním (**last**) bytem, značkový byte je ignorován a není nutné jej znát. Pokud neznáte adresu, na niž se bude blok ukládat, vložte adresu vyšší, poslední adresa slouží spíš pro zabezpečení paměti za touto adresou. Po skončení nahrávání se vypíše informace o nahraném bloku: **Leader:255 Length:2000 Parity:Y**, tedy značkový byte, skutečná délka a parita (správnost) nahraných dat. Leader Vás bude zajímat pouze v případě, že budete chtít data opět uložit na kazetu ve stejném tvaru, délka dat Vás může zajímat z obdobných důvodů nebo čistě pro informaci. Parita Vám říká, jestli se data nahrála dobře (**Parity:Y**) nebo špatně (**Parity:N**), pokud řekneme chcete data změnit a uložit zpět na kazetu (při překládání hry atp.), je nutno načíst je bez chyb, pokud se chcete pouze podívat, nemusí vždy chybná parita vadit - lepší je však vždy nahrávku opakovat. Pokud dojde k výpadku při nahrávání, bude údaj **Length**: ukazovat délku dosud nahraného bloku.

J+Symbol Shift - stejně jako u příkazu Save jde o obdobu předchozí možnosti, opět je blok určen prvním (**first**) bytem a délkou (**length**).

Y - čtení hlavičky, protože data na kazetě jsou většinou uložena v blocích s hlavičkou, je možno nahrávat je podle této hlavičky. Po stisku Y se první blok z kazety přečte jako hlavička a vypíše se tyto informace: typ hlavičky (**P-program, N-number array, C-character array a B-bytes**), jméno v hlavičce, start (u typu 3 první adresa, u typu 0 jde o řádek, na kterém program začíná), délka bloku, který za hlavičkou následuje, a ještě jedno číslo, jehož význam je různý podle typu hlavičky - pokud Vás zajímá, můžete jej najít někde v popisu ROM ZX Spectra. Nyní můžete stisknout klávesu **J** a program nahraje data podle informací v hlavičce (start, délka), pokud stisknete **M**, budou data nahrána na adresu 16384 a provede se skok na tuto adresu - taktó lze nahrát do paměti monitor DEVAST ACE, aniž by se přepsal byť jen byte paměti od adresy 23296 nahoru.

Edítace obrázků, spritů a fontů:

P - editor obrázků a spritů. Stisknutím klávesy P se spustí grafický editor, do horní třetiny se zobrazí obrázek a vypíše: **00000 L:000**. První číslo je adresa, od které se obsah paměti chápe jako obrázek a druhé číslo je řádek obrázku, který je na obrazovce úplně nahoře.

V dalším textu budu používat tuto konvenci pro přiřazení kláves a směrů pohybu:

klávesa 6 ... vlevo
klávesa 7 ... vpravo
klávesa 8 ... nahoru
klávesa 9 ... dolů
klávesa 0 ... pal

Toto nastavení odpovídá Sinclair Joysticku a používá se pro posouvání kurzoru po obrazovce, pokud máte Kempston Joystick, můžete ho používat místo kláves.

2 - přepínání mezi módem **Obrázky** a módem **Sprity**. V módu **Obrázky** je vypisován tento informační řádek: **00000 L:000**, v módu **Sprity** obsahuje informační řádek informace o zvolené počáteční adrese, délce, šířce a výšce spritu: **00000 00000 W:000 H:000**. Pokud

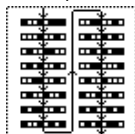
cheete editovat znakové sady (fonty), zvolte mód **Sprity** a nastavte sloupcové sprity (dále) a velikost **W:032 a H:024**.

Počáteční adresa se nastavuje buď přímo tím, že stisknete klávesu **I** a vložíte zvolenou hodnotu, nebo pomocí kláves pro směry takto: doprava (doleva) přičtení (odečtení) 1, doprava (doleva) a pal, přičtení (odečtení) 32 a nahoru (dolů) a pal znamená přičtení (odečtení) 256, pouze nahoru (dolů) vybírá zobrazenou část, toto platí pro obrázky. U spritů je doprava (doleva) s palem i bez něj určeno k přičtení (odečtení) 1, nahoru (dolů) přičítá (odečítá) šířku spritu, nahoru (dolů) se stiskem pal přičítá (odečítá) délku spritu.

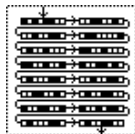
A - zapnutí (vypnutí) zobrazení atributů. Některé obrázky a takřka všechny sprity nemají atributy.

M - stiskem této klávesy a současným stiskem některého ze směrů se mění velikost spritu ve směru pohybu.

T - změna typu spritu. Sprity ve hrách jsou obvykle řádkového typu (druhý obrázek), takto jsou uloženy také velké fonty (znakové sady). Druhou možností jak uložit sprity (respektive jak posloupnost bytů v paměti vykreslit na obrazovku) je po sloupcích (první obrázek). Tento způsob je mnohem méně častý, používá se hlavně pro uložení normálních fontů, z her se s ním můžete setkat třeba u hry **Mystery of the Nile**. Přepnutím z řádkového způsobu zobrazení na sloupcový se přehodí výška a šířka spritu a odpovídajícím způsobem také nastavování velikosti a posuny adresy.



(respektive jak posloupnost bytů v paměti vykreslit na obrazovku) je po sloupcích (první obrázek). Tento způsob je mnohem méně častý, používá se hlavně pro uložení normálních fontů, z her se s ním můžete setkat třeba u hry **Mystery of the Nile**. Přepnutím z řádkového způsobu zobrazení na sloupcový se přehodí výška a šířka spritu a odpovídajícím způsobem také nastavování velikosti a posuny adresy.

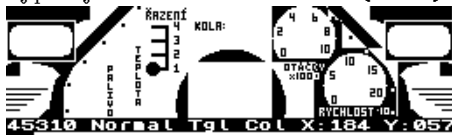


E - přepnutí na vynechání každého druhého bytu. Většina spritů používá při kreslení masku (informace o tom, co se má přepsat a co ne) - viz obrázek. Obvykle bývá maska uložena za spritem, některé hry (opět **Mystery of the Nile** ale i další) mají uloženy vždy byty spritu a za ním vždy odpovídající byty masky - použitím přepínače **E** můžete jeden z nich vždy vypustit a prohlížet druhý.



C - nastavení znakového souboru pro Texteditor, přepnete hledání na sloupcové sprity, nastavte mezeru do levého horního rohu a stisknete **C**; od této chvíle bude Texteditor psát jiným fontem. Pokud chcete vrátit zpět normální písmena, nastavte adresu do ROM na adresu 15616 a opět stisknete **C** (nastavíte font z ROM).

1 - Přepnutí z hledání na editaci obrázků. Při přepnutí se do informačního řádku vypíší tyto informace: **00000 Normal (Invert) Tgl (Res, Set) Non (Col) X:000 Y:000**, které



znamenají: **00000** je adresa obrázku, **Normal (Invert)** je nastavený mód pro kurzor, v módu **Normal** se akce provádějí pouze při stisku **Pal**. V módu **Invert** se akce provádějí trvale, stiskem **Pal** je možno je přerušit. **Tgl (Res, Set)** - druh akce, kterou kurzor provádí - stejně jako u **ART STUDIO**: **Toggle** - změň, **Reset** - smaž, **Set** - nakresli, **Non (Col)** je stav operací s atributy: **Non** - neměň atributy, **Col** - nastav atributy podle atributu levého horního rohu kurzoru, **X:000 Y:000** - pozice kurzoru, počátek je levý dolní roh obrázku.

RÁZENÍ
4

Editační funkce:

Po přepnutí na editaci lze používat tyto funkce:

A - zapnutí nebo vypnutí atributů, jde o stejnou akci jako při vyhledávání obrázků nebo spritů v paměti.

M - a současným stiskem směru se mění velikost kurzoru v daném směru; levý horní roh zůstává na místě, ostatní se pohybují.

E - změna způsobu, jakým kurzor reaguje na stisk pal. Mění režim **Normal** a **Invert**. První režim provádí akci při stisku pal, druhý ji naopak při stisku pal přerušuje. Režim **Invert** je vhodný při kreslení čar nebo mazání větších ploch.

T - změna typu kurzoru: mění, maže, kreslí (**Tgl**, **Res** a **Set** jako v ART STUDIUM).

R - současným stiskem některého směru se provádí rolování části obrázku pokryté kurzorem ve zvoleném směru. Pomocí této funkce lze posunovat části obrázku, naklánět písma a podobné akce.

G - zapnutí a vypnutí jasové mřížky, na rozdíl od ART STUDIA při vypnutí nemaže jas.

I, P, B, F - změna inkoustu, papíru, jasu (**bright**) či blikání (**flash**). Proveďte se buď v atributovém čtverci, kde je levý horní roh (režim **Non**), nebo ve všech čtvercích, kam zasahuje kurzor (režim **Col**).

C - přepínání režimu práce s barvami (**Non, Col**). Při posunování kurzoru v režimu **Col** se nová plocha obarvuje zvolenými barvami, takto lze snadno vybarvit velkou plochu.

S (+SS) - uložení zvoleného bloku paměti na adresu, můžete použít k nahrání spritu jako obrázku (**16384, 2048**).

J (+SS) - nahrání libovolných dat do paměti, lze použít k získání spritu z obrázku.

ENTER - uložení obrázku nebo spritu zpět do paměti, všechny změny se provádí na obrazovce a chcete-li je potvrdit, stiskněte **ENTER**. A naopak, podařilo-li se Vám upravit obrázek nežádoucím způsobem, můžete dvakrát stisknout klávesu **1** a začít znovu.

BREAK (Caps Shift + Space) - návrat do hlavního menu.

Texteditor - editace textů:

Obrazovka při práci s texteditorem vypadá podobně jako na obrázku vpravo. V informačním řádku je vypsaná adresa, relativní posun ASCII kódu, typ kurzoru C nebo L, kódy znaku s kurzorem a jeho sousedů (vlevo a vpravo) a případně znak nebo klíčové slovo BASICu.

BREAK - návrat z texteditoru do hlavního menu.

SS+ENTER - nastavení adresy výpisu, také lze měnit pomocí šipek (**5,6,7,8+CS**) a Kempston Joysticku.

CS+1 - povolení nebo zakázání zápisu, je signalizováno barvou informačního řádku, při povoleném zápisu je informační řádek zelený, při zakázaném je červený.

CS+2 - zapnutí nebo vypnutí **CAPS LOCK**. Na rozdíl od BASICu funguje **CAPS LOCK** tak, že mění malá písmena na velká a opačně.

CS+3, 4 - změna kódu znaku na pozici kurzoru. Tuto funkci použijete v okamžiku, kdy budete potřebovat vložit do textu znak, který nelze zadat z klávesnice.

CS+9, 0 - nastavení posunu ASCII kódu, některé hry nemají znaky s obvyklými kódy ale většinou zachovávají alespoň abecedu, například některé hry firmy ULTIMATE mají kód posunutý tak, že velké **A** má kód **0**.

SS+E - přepnutí na vkládání invertovaných znaků. Tyto znaky mají kód vyšší o 128 a ve výpisu jsou napsány inverzně, obvykle se v programech používají pro ukončení textu. Pokud upravujete texty ve hře, která je má takto ukončeny, musíte je také tak ukončit.

Ostatní klávesy při povoleném zápisu vyvolají zápis na pozici kurzoru a posun kurzoru na další adresu.

Pokud budete překládat hry do češtiny nebo slovenštiny, musíte používat stejně dlouhé nebo kratší texty, pokud má hra vlastní znaky, udělejte si z nepoužívaných znaků písmena s diakritikou a použijte je.

Pokud hra vlastní znakovou sadu nepoužívá, pokuste se najít volné místo a vyrobit znakovou sadu novou, potom budete muset najít tiskový podprogram a opravit odkaz na nový znakový soubor.

Jen nepřekládejte hry do Cestiny, jak to dělá jistý **Luxus Software House** z Ostravy, to raději nechte hru v původním jazyce. Není nic horšího, než když v programu naleznete perličky typu **pasak** a nevíte, jestli je to **pasák**, **pašák**, případně **pásák**, zkratka používejte vedle své druhé gramotnosti (programování) také svou první!

BASIC – prohlížení programu:

K - stiskem této klávesy v hlavním menu se vypíše **Address:23296 Extend:No**, což znamená, že výpis je nastaven od adresy 23296 a nebudou se vypisovat neviditelné znaky, jako jsou definice barev, inverse... Pokud chcete měnit adresu výpisu, stisknete klávesu **I** a vložte požadované číslo, nebo pomocí směrů můžete měnit adresu pomalu. Způsob výpisu změňte stiskem klávesy **ENTER**, vlastní výpis vyvoláte stiskem klávesy **SPACE**.

Výpis vypadá takto: adresa, číslo řádku, text až po kód 13 (Enter), prázdný řádek a další řádek. Pokud zvolíte mód Extend vypisují se inverzně neviditelné části textu, jako skrytá čísla (binární vyjádření čísla v BASICu následuje za znakovým výpisem a při listingu se nevypisuje, mělo by být stejné ale nemusí a počítač při provádění programu bere v úvahu pouze binární vyjádření; takto jsou zakrývány skutečné hodnoty v některých programech), barevné kódy (je vysáno co se nastavuje a jak), znaky způsobující posun tisku o pozici vlevo, vpravo, nahoru a dolů (vypsány Left, Right, Up, Down) a další, všechny inverzně.

Výpis se ukončí stiskem **BREAK (Caps Shift + Space)**.

Návrat do Basicu a do volacího programu:
.....

CS+B - návrat do BASICu, pokud bude volání programu WLEZLEY 7 součástí programu, bude tento program po návratu pokračovat dalším příkazem.

CS+R - návrat do volacího programu. Při tomto návratu se obnoví SP registr na hodnotu, jakou měl při volání programu, obnoví se stav přerušeni (Wlezley 7 pracuje se zakázaným přerušeni) a provede se návrat zpět do programu, ze kterého byl Wlezley 7 volán. Pokud budete chtít připojit program Wlezley 7 do svého programu ve strojovém kódu, můžete to udělat, ale vracet se z něj musíte pomocí CS+R, nejlépe bude, když přepíšete test pro návrat z **CS+R** na **CS+B** a návrat do BASICu prostě zrušíte úplně (jedná se tu o instrukce **cp 82** a **cp 66**).

Program **WLEZLEY 7** je mnohostranně využitelný, jistě sami přijdete na další možnosti, jistě jsem zde neuvedl zdaleka vše, k čemu lze program využít. Pokud nemáte Kempston a dělá Vám potíže ovládání, nahrajte si nějakou oblíbenou hru, zvolte si Sinclair Joystick a hrajte ji s tímto ovládáním, zvyknete si a nebude Vám již potom činit potíže.

Pokud máte počítač **DIDAKTIK** a nepoužíváte **KEMPSTON JOYSTICK**, může program chybně pracovat protože nepozná, že joystick není připojen. V takovém případě proveďte:

LOAD "WLEZLEY 7+" CODE 6e4
POKE 60490, 175
SAVE "WLEZLEY 7+" CODE 6e4, 3960

Doufáme, že se tento program stane Vaším trvalým pomocníkem v tvorbě vlastních programů a že Vám ušetří alespoň trochu práce. Pokud budete program používat třeba jen k prohlížení her či jiných programů i zde Vám poskytne kvalitní služby.

Mnoho úspěchů Vám přeji **Proxima** a **UNIVERSUM**

Kopírovací programy
GARGANTUA
a
PANTAGRUEL
DAVID
a
GOLIATH

Vznik těchto kopírovacích programů byl podnícen dvěma podstatnými zjištěními:

- 1) v nahrávací rutině ROM je spousta času
- 2) existencí velmi špatného programu **O-copy**

K bodu dva, pokud jste si toho sami nevšimli, tento program vytváří do nahrávaného signálu tak nádherné pulsy, že je nelze přehlédnout (nahrájte nějaký blok ze samých nul a překopírujte si jej jinam, pak si kopii poslechněte, stojí to za to), což by nevadilo ani tak uchu uživatele, ale bohužel to občas vadí programům a zatvrzele se odmítají nahrát bez chyby.

Nové kopírovací programy **GARGANTUA**, **PANTAGRUEL** a **GOLIATH** navíc dokáží hry nejen kopírovat, ale pokud jsou to komprimované verze (vyrobené pomocí programu **PRESSOR** a **PACKMAKER**), tak také spustit!

Kapacita kopírovacích programů (všechny také komprimují a pojmu mnohdy mnohem více), v závorce je uvedena hodnota po odpojení spouštěcí části:

Gargantua ... 44454 (44800)
Pantagruel ... 45910 (46260)
Goliath ... 46506 (46858)
David ... 47200

Nahrání a spuštění:
.....

Programy nahrajete příkazem **LOAD**, ale to by Vás asi nenapadlo, po nahrání se provádí odkódování, po odkódování se objeví jisté texty. Můžete stisknout libovolnou klávesu. Přepíná se šířka písma, nastavte si tu, kterou chcete a stiskněte buď **ENTER** nebo **E**, v prvním případě se odpojí spouštěcí část programu (**SS+R**) a zvětší se kapacita, v případě druhém bude program spuštěn i se spouštěcí částí.

Ovládání programu:
.....

Filosofie tohoto programu je odlišná od ostatních kopíráků (vyjma starších verzí těchto programů), obvykle se nejprve volí akce a teprve potom s čím bude prováděna. Zde se nejprve volí s čím a teprve potom co, podobně jako u textových editorů. Nastavte tedy nejprve blok a potom stiskněte odpovídající klávesu.

L - přihrání dalších souborů do paměti. Budou uloženy za již nahrané soubory.

H - podobný příkaz, očekává, že první soubor bude hlavička a za ní odpovídající (délkou) soubor dat, znovu hlavička, soubor dat a stále dokola. Tento příkaz používejte v případě, že je hlášena chyba při nahrání souboru a délka přečteného souboru je delší, než je uvedeno v hlavičce. Některé magnetofony totiž mají tendenci vyrábět tak vysoký šum, že program nezjistí konec souboru a čte ještě několik bytů a pak teprve ohlásí chybu.

S - uložení nastaveného bloku souborů na kazetu.

V - kontrola nahrávky na kazetě. Nastavte kazetu tak, aby první soubor, který chcete verifikovat byl také prvním souborem, jenž půjde do počítače z magnetofonu, jinak bude samozřejmě signalizována chyba.

ENTER - nastavení okraje bloku.

6, 7 (šípky) - listování seznamem. Na tomto místě se zmíním podrobněji o způsobu nastavování bloku: jeden okraj bloku určíte tak, že nastavíte vybraný soubor do významného řádku (barevně odlišený řádek uprostřed) a stisknete **ENTER**, druhý okraj je vždy soubor ve význačném řádku. Najeďte tedy druhým okrajem do význačného řádku a můžete provést vybranou akci (**save, verify, delete**). Blok je označen znakem “,” mezi číslem a jménem. Tento způsob nastavování se liší od ostatních kopírovacích programů, je bližší textovým editorům a poněkud úspornější (jednou nastavený blok můžete do omrzení ukládat na kazetu a verifikovat stiskem pouze jedné klávesy). O funkci **delete** tato slova samozřejmě neplatí - lze ji za stejných podmínek provést pouze jednou. Většinou nebudete muset blok vůbec nastavovat protože již bude nastaven od minulé akce. K zaholení není ani fakt, že máte přímou vizuální kontrolu o tom, že jste se nespletli při nastavování bloku, nelze se prostě přepsat jako třeba u **TF COPY**.

A - nastaví blok tak, že obsahuje všechny soubory.

SS+D - delete; smaže nastavený blok souborů.

SS+L - smaže poslední soubor - obvykle nepatřičná hlavička nebo chybně nahraný soubor, je to rychlejší než pomocí **SS+D**.

SS+B - reset počítače (návrat do Basic s vymazáním paměti).

N - vypnutí/zapnutí pípání na konci nahrávání.

M - vypnutí/zapnutí zobrazování počítadel volné paměti a délky při kopírování.

I - přepínání prohlížečícího režimu (**nic, znaky, obrázky**), **GARGANTUA** může při kopírování ukazovat data ve formě znaků nebo obrázku v horní třetině.

R - rename; program umožňuje přejmenovat soubory nahrané do paměti. Tuto možnost používejte tak, abyste nenapáchali zmatek mezi Spectristy, není nic lepšího, než když má hra deset názvů a každý z nich se pokud možno ještě co nejvíce liší od názvu skutečného...

CS+A až **CS+Z** - nastavení pauzy mezi soubory při Save. Při **CS+A** je pauza minimální a plynule roste až k **CS+Z**.

F - u programů **GOLIATH** a **DAVID** ukáže velikost zbývajících paměti, u programu **GARGANTUA** je vidět neustále, u programu **PANTAGRUEL** jen když je první soubor ve význačném řádku.

SS+R - spuštění komprimovaného programu (obvykle hry) upraveného pomocí programů **PRESSOR** a **PACKMAKER**. Jsou to programy skládající se z Basicu s obrázkem a

bloku dat začínajícího na adrese 24200 a majícího většinou v hlavičce kóde na posledních dvou místech písmena "cd". Takto upravené programy jsou obvykle mnohem kratší než původní verze (až o polovinu) a jsou často vybaveny volbou **Poke?**. Pokud máte takto upravené hry (a existuje jich už mnoho - my jiné ani nemáme, resp. všechny upravujeme), můžete používat tuto novinku. V paměti musí být pouze 4 soubory (basic a code s hlavičkami), po stisku **SS+R** dojde k umístění programu na správné místo a ke spuštění. Pokud je ve hře možnost volby **Poke?** a chcete ji využít, stiskněte klávesu "**p**", jinak stiskněte něco jiného a hra se spustí. Pro zajímavost a pro zkušenější programátory uvedu co vlastně v tomto případě znamená umístit hru na správné místo: dekomprese, otočení bloků - jsou ukládány odzadu, přečtení startovací adresy a možných poke z basicu, obnovení systémových proměnných, dekomprese obrázku, dekomprese hlavního bloku, čekání na stisk klávesy a případné vložení poke do hry a nakonec skok do programu - všechno dohromady zabírá 350 bytů.

Kopírovačí programy umožňují také vysílat na libovolný port zvolenou hodnotu při skončení nahrávání, při **SAVE** a při **LOAD (VERIFY)**. Pokud tuto možnost chcete využívat, najdete (pomocí **DEVAST ACE**) v programu nahraném pomocí **MERGE ""** z BASICu tuto sekvenci instrukcí:

```
ld b,0  
ld c,0  
ld d,0  
ld e,0,
```

kde v registru **B** je adresa portu, v registru **C** hodnota vyslaná při **SAVE**, v registru **D** hodnota vyslaná při **LOAD** a v registru **E** hodnota vyslaná při skončení akce. Přepište tyto hodnoty na požadované, vraťte se zpět do BASIC a uložte program na kazetu povelom **SAVE "jméno" LINE 1**, kde **jméno** je samozřejmě jméno upraveného kopíráku.

Doufáme, že budete s těmito programy spokojeni a že Vám budou dobře sloužit.

Obsah příručky k programům DEVAST ACE, WLEZLEY, GARGANTUA
.....

Monitor a disassembler DEVAST ACE	1
Přístup k paměti (aktuální adresa) a pomocné funkce	2
Vnořování a jednotlivé druhy paměťových výpisů	3
Přesuny a plnění bloků paměti, textový editor	4
Magnetofonové operace	4
Vyhledávání v paměti	5
Krokování, nastavování registrů, breakpoint a volání jiných programů	5
Závěrečná doporučení	6
Grafický Monitor a editor WLEZLEY 7	7
Nahrání a spuštění programu	7
Hlavní menu	7
Kazetové operace	7
Editace obrázků, spritů a fontů	8
Editační funkce	10
Texteditor – editace textů	10
BASIC – prohlížení programu	11
Návrat do Basicu a do volajícího programu	12
Kopírovací programy GARGANTUA, PANTAGRUEL, GOLIATH a DAVID	13
Nahrání a spuštění	13
Ovládání programu	13
Obsah příručky k programům DEVAST ACE, WLEZLEY 7, GARGANTUA	16

KLÁVESNICE NA ZX SPECTRU

Testování klávesnice na ZX Spectru lze provádět mnoha způsoby. Postupně se dozvíte všechny. Ke každému způsobu uvidíte program, který vrací v registru A kód stisknuté klávesy. Pro snazší vyzkoušení klávesnicových podprogramů vložte nejprve jednoduchý obrazkovový editor:

```
ent $

START
TESTER  ld  a,2           ; otevření kanálu pro tisk
        call #1601      ; do horní obrazovky

        xor  a
        ld  (LINE),a
        ld  (COLUMN),a
        ld  hl,22528
        ld  (CURSOR),hl } počáteční inicializace

MAINLOOP call 8020      ; otestuj klávesu BREAK
        ret  nc         ; vrať se, je-li stištna

        ld  a,22
        rst 16
        ld  a,(LINE)
        rst 16
        ld  a,(COLUMN)
        rst 16 } nastav pozici pro tisk

        ld  hl,(CURSOR) ; namaluj kurzor na
        ld  (hl),180    ; obrazovku

        call INKEY1     ; čekej na stisk klávesy

        cp  ""         ; test typu znaku
        jr  c,CONTROLS ; odskok pro kód < 32

        rst 16         ; vytiskni znak

        ld  hl,(CURSOR)
        inc hl
        ld  a,(COLUMN)
        inc a
        cp  32
        jr  c,CHAR1
        ld  a,(LINE)
        inc a
        cp  22
        jr  c,CHAR2
        xor a
        ld  hl,22528
        ld  (LINE),a
        xor a
        ld  (COLUMN),a
        ld  (CURSOR),hl } posun na další pozici
        jr  MAINLOOP

CHAR2
CHAR1
```

CONTROLS	ld	b,a	}	smazání kurzoru
	ld	a,(23695)		
	ld	hl,(CURSOR)		
	ld	(hl),a		
	ld	a,b		
	cp	8	}	kurzor doleva
	jr	nz,CTRL1		
	dec	hl		
	ld	a,(COLUMN)		
	dec	a		
	cp	255		
	jr	nz,CLEFTRGT		
	ld	bc,32		
	add	hl,bc		
	ld	a,31		
CLEFTRGT	ld	(COLUMN),a	}	
	ld	(CURSOR),hl		
	jr	MAINLOOP		
CTRL1	cp	10	}	kurzor dolů
	jr	nz,CTRL2		
	ld	bc,32		
	add	hl,bc		
	ld	a,(LINE)		
	inc	a		
	cp	22		
	jr	nz,DOWN		
	ld	bc,22*32		
	or	a		
	sbc	hl,bc	}	
	xor	a		
CDOWN	jr	CUPDOWN		
CTRL2	cp	11	}	kurzor nahoru
	jr	nz,CTRL3		
	ld	bc,32		
	or	a		
	sbc	hl,bc		
	ld	a,(LINE)		
	dec	a		
	cp	255		
	jr	nz,CUPDOWN		
	ld	bc,22*32		
	add	hl,bc	}	
	ld	a,21		
CUPDOWN	ld	(LINE),a	}	
	ld	(CURSOR),hl		
	jr	MAINLOOP		
CTRL3	cp	9	}	kurzor doprava
	jr	nz,CTRL4		
	inc	hl		
	ld	a,(COLUMN)		
	inc	a		
	cp	32		
	jr	nz,CRIGHT		
	ld	bc,32		
	or	a		
	sbc	hl,bc		
	xor	a		
CRIGHT	jr	CLEFTRGT		

```

CTRL4      JP      MAINLOOP      ; další klávesy nejsou

LINE       defb  0
COLUMN     defb  0
CURSOR     defw  0
} - uložení pozice kurzoru

INKEY1     ei                ; povolení přerušení
           halt              ; čekání na přerušení
           bit  5, (iy+1)     ; test na stisk klávesy
           jr   z, INKEY1     ; není-li stisk, vrať se
           res  5, (iy+1)     ; zruš příznak stisku
           ld   a, (23560)    ; přečti kód klávesy
           ret                ; vrať se s kódem v B

A0LENGTH  equ  #-START      ; v A0LENGTH je délka

```

První způsob (podprogram INKEY1) plně využívá možnosti, které poskytuje operační systém Spectra. Počítač každou padesátinu sekundy provede otestování klávesnice a pokud zaznamená stisk klávesy nebo klávesy a nějakého shiftu, zapíše na adresu 23560 kód stisknuté klávesy a nastaví pátý bit na adrese 23611 (neboli IY+1). Při použití tohoto podprogramu musíte mít povolené přerušení v módu IM 1 nebo IM 2. V druhém případě musí Váš obslužný program pro přerušení volat podprogram na adrese 56 - nejjednodušší je na konci místo instrukcí EI a RET vložit instrukci JP 56 - přerušení se obnovuje v tomto podprogramu. Registr IY musí obsahovat hodnotu 23610 (#5C3A). Na testování mají vliv některé systémové proměnné:

23561 - doba (v padesátinách sekundy), která uplyne, než se začne u stále stisknuté klávesy opakovat vrácení jejího kódu - autorepeat. Hodnota se inicializuje na 35.

23562 - interval, v jakém se bude opakovat kód stále stisknuté klávesy. Na počátku je nastavena na 5 padesátin sekundy.

Význam těchto konstant je následující pokud stisknete a budete držet klávesu, vrátí se její kód v okamžiku stisku, potom 35 padesátin sekundy nevrátí počítač nic a potom bude stejný kód vracet každých 5 padesátin sekundy a to dokud klávesu nepustíte. Obě konstanty můžete nastavit na libovolnou hodnotu v rozmezí 0 až 255.

System umí číst klávesnici ve všech módech - . Nastavení, v jakém módu bude klávesnice čtena, se provádí na třech adresách - 23611, 23617 a 23658. Jednotlivé módy (jde o módy kurzoru v BASICu) nastavíte takto:

```

klávesový mód  - ld (iy+1), 204
                  ld (iy+7), 0
                  ld (iy+48), 8

```

```

klávesový mód  - ld (iy+7), 1

```

```

klávesový mód  - ld (iy+7), 2

```

```

klávesový mód  - ld (iy+1), 196
                  ld (iy+7), 0

```

```

klávesový mód  - ld (iy+1), 204
                  ld (iy+7), 0
                  ld (iy+48), 0

```

Registr IY musí opět ukazovat na adresu 23610 (#5C3A). Používáte-li ve svých programech přerušení v módu IM 1, nepoužívejte raději registr IY a nechte jej nastavený na uvedenou hodnotu. Pokud se přece jen ukáže použití IY jako vhodné, musíte zakázat přerušení po dobu, kdy je v registru IY jiná hodnota.

Uvedená nastavení lze na pro čtení klávesnice používat i v BASICu, nesmíte se však na kód klávesy dotazovat pomocí INKEY\$ ale přímo pomocí PEEK 23560. Podprogram pro INKEY\$ totiž povoluje čtení pouze v módech **█** a **█**.

Občas se může stát, že Vás nezajímá ani kód klávesy podle módu ani to, jestli je stisknutý CAPS SHIFT či SYMBOL SHIFT, a chcete pouze zjistit, která klávesa je stisknutá. V tomto případě stačí číst adresu 23556 a na ní je zapsán tzv. hlavní kód klávesy a jeho hodnota je obnovována každou padesátinu sekundy bez ohledu na to, jestli klávesa byla stisknuta nebo ne. Pokud klávesa stisknuta nebyla, je zde uložena hodnota 255. Jinak je tu uložen odpovídající kód číslice, velkého písmene, ENTERu (13), SPACE (32) nebo EXTEND MODE (14).

```

INKEY2   e i                ; povolení přerušeni
         h a l t            ; čekání na přerušeni
         l d   a , (23556)   ; načtení hodnoty do A
         c p   255          ; test na nestištění
         j r   z , INKEY2    ; skok zpět když platí
         r e t              ; návrat z podprogramu

```

Ve zkušebním programu přepište volání INKEY1 na INKEY2 a můžete vyzkoušet funkci. Tentokrát Vám nebudou pracovat pohyby kurzoru.

Instrukce EI v podprogramu nemusí být pokud máte jistotu, že při volání je přerušeni povoleno. Instrukce HALT také není pro funkci programu bezpodmínečně nutná.

Tento i předchozí podprogramy jsou udělány tak, že čekají, až nějaká klávesa stisknuta bude. Pokud budete chtít, aby byl signalizován stav, kdy žádná klávesa stisknuta není, stačí provést jednoduché úpravy - místo skoku zpět do testu vložit do registru A třeba nulu (kód, který říká, že žádná klávesa stisknuta nebyla) a vrátit se zpět.

Dosud popsané způsoby testování vyžadují pro práci povolené přerušeni - tato vlastnost však může být občas nežádoucí. Vyzkoušejte tento program:

```

INKEY3   c a l l  654        ; volání KEY-SCAN v ROM
         j r   n z , INKEY3  ; více kláves, skok zpět
         c a l l  798        ; volání K-TEST v ROM
         j r   n c , INKEY3  ; nevyhovuje, skok zpět
         d e c   d           ; nastavení █ módu
         l d   e , a         ; hlavní kód do E
         j p   819          ; skok do dekodování

```

Tato rutina vrací zpět hodnoty jako INKEY1 v módu **█**. Na tento program má vliv to, kam ukazuje registr IY, pokud je totiž na adrese IY+48 nastaven 3 bit, jde o mód **█**, pokud je tento bit nulový, jde o mód **█**. Pokud tedy IY ukazuje do systémových proměnných (23610 - #5C3A), jde o takový mód, který byl naposledy nastaven. Chcete-li mít test určitě v módu **█**, nastavte registr IY na hodnotu 39 před voláním KEY-SCAN. Více podrobností se můžete dozvědět z komentovaného výpisu ROM.

Předchozí způsoby vracejí kódy kláves stejné, jako jsou při práci v BASICu. Někdy se hodí, aby byly vráceny kódy jiné. Můžete si vyrobit tabulku změn a nevhodné kódy nahradit požadovanými - je vhodné pokud se nejedná o příliš mnoho kláves. Chcete-li klávesnici doslova převrátit naruby, je vhodnější následující program:

```

INKEY4   c a l l  654        ; volání KEY-SCAN v ROM
         j r   z , INKEY4    ; více kláves, opět test
         l d   a , e         ; test jestli byla vůbec
         c p   255          ; nějaká klávesa stištna
         j r   z , INKEY4    ; pokud ne, testuj znovu

         l d   a , d         ; do A případný SHIFT

```

```

        ld    hl, SYMBTAB
        cp    #18
        jr    z, INKEY4A      } stišťen SYMBOL SHIFT

        ld    hl, CAPSTAB
        cp    #27
        jr    z, INKEY4A      } stišťen CAPS SHIFT

        ld    hl, NORMTAB     ; nebylo stišťeno nic

INKEY4A  ld    d, 0
        add   hl, de
        ld    a, (hl)
        ret                                } přečtení kódu z tabulky

SYMBTAB  def m  " *† [ & % > ? / "
        def m  " , - ] ' # < { ? "
        def m  " . ( # "
        def b  143
        def m  " \_ "
        def b  0
        def m  " = ; ) @ "
        def b  131
        def m  " | : "
        def b  " ", 13, 34
        def m  " _ ! "
        def b  130
        def m  " ^ " , 0                                } tabulka pro SYMBOL

CAPSTAB  def m  " B H Y "
        def b  10, 8
        def m  " T G U "
        def m  " N J U "
        def b  11, 5
        def m  " R F C "
        def m  " M K I "
        def b  9, 4
        def m  " E D X "
        def b  0
        def m  " L O "
        def b  15, 6
        def m  " W S Z "
        def b  " ", 13, " P "
        def b  12, 7
        def m  " Q A "                                } tabulka pro CAPS

NORMTAB  def m  " b h y 6 5 t g w "
        def m  " n j u 7 4 r f c "
        def m  " m k i 8 3 e d x "
        def b  0
        def m  " l o 9 2 w s z "
        def b  " ", 13
        def m  " p 0 1 q a "
        def b  0                                } tabulka bez shiftu

```

Podprogram KEY-SCAN vrací zpět v registru E hodnotu v rozsahu 0 až 39 - pokud byla stisknuta nějaká klávesa, nebo hodnotu 255 - pokud nebyla stisknuta žádná klávesa. Je-li současně stisknut nějaký SHIFT, je jeho hodnota uložena v registru D (#18 pro SYMBOL SHIFT a #27 pro CAPS SHIFT). Při neúspěšném testu (bylo stišťeno více kláves a ani jedna nebyl SHIFT) je nastavena podmínka NZ, v opačném případě pak platí Z. Stisknete-li

současně CAPS SHIFT a SYMBOL SHIFT, bude kód pro CAPS SHIFT v registru D a pro SYMBOL SHIFT v E.

Pomocí této rutiny můžete provádět i takové změny, které nelze dosáhnout tabulkou změn - testovat např. CAPS SHIFT a ENTER nebo SYMBOL SHIFT a SPACE a podobné kombinace. Stačí na vhodném místě tabulky vložit požadovaný kód. Například klávesa ENTER vrací kód 13 ať je stisknuta sama nebo spolu s nějakým SHIFTEM. Ve všech tabulkách je na tomtéž místě napsána hodnota 13, obdobně klávesa SPACE vrací vždy kód 32.

Pokud budete chtít testovat klávesu bez ohledu na to, jestli je stišťena ještě nějaká další klávesa, musíte číst přímo jednotlivé porty klávesnice. Nejprve ukázkový testovací program:

```

ent      #

MAINLOOP call  SCANNER
         call  SHOWKEYS
         call  8020
         jr    c,MAINLOOP
         ret

SHOWKEYS ld    hl,KEYBOARD ;buffer s klávesami
         ld    ix,22528+33 ;adresa v atributech
         ld    b,4         ;čtyři řádky
SHOW1    ld    c,10        ;kláves je deset v řadě
SHOW0    ld    a,(hl)
         inc   hl
         ld    (ix+0),a
         ld    (ix+1),a
         ld    (ix+32),a
         ld    (ix+33),a
         inc   ix
         inc   ix
         inc   ix
         dec   c           ;vnitřní cyklus přes
         jr    nz,SHOW0    ;sloupce a skok zpět
         ld    de,66
         add   ix,de
         djnz SHOW1
         ret

SCANNER  ld    hl,PORTTAB  ;tabulka adres portů
         ld    ix,KEYBOARD ;buffer pro klávesy
         ld    e,4         ;čtyři řádky klávesnice
         ld    c,254       ;dolní byte adresy portu

SCAN1    ld    b,(hl)      ;horní byte adresy portu
         inc   hl          ;posun na další položku
         push  hl          ;uložení pro další použití
         ld    d,5         ;kláves je 5 na portu
         ld    hl,BITTAB   ;tabulka masek pro bity

SCAN2    in    a,(c)       ;hodnota portu BC do A
         cpl             ;komplement registru A
         and   (hl)        ;ponech pouze žádaný bit
         inc   hl          ;další položka v tabulce
         jr    z,SCAN2B    ;odskok, není stišťeno
         ld    a,255       ;signál - je stišťena

SCAN2B   ld    (ix+0),a    ;zapiš výsledek
         inc   ix          ;posun na další klávesu
         dec   d           ;opaku j celkem 5 krát
         jr    nz,SCAN2    ;skok na začátek cyklu

```


Poslední informace, které potřebujete vědět pro testování libovolné klávesy, jsou adresy portů a bity pro jednotlivé klávesy. Všechno naleznete v následující tabulce:

Adresa portu			0	1	2	3	4
254	11111110	65278	C	S	X	C	U
253	11111101	65022	A	S	D	F	G
251	11111011	64510	Q	W	E	R	T
247	11110111	63486	1	2	3	4	5
239	11101111	61438	0	9	8	7	6
223	11011111	57342	P	O	I	U	Y
191	10111111	49150	ENT	L	K	J	H
127	01111111	32766	SPC	SSH	M	N	B

První tři sloupce v tabulce jsou adresa portu vypsaná třemi způsoby. První dva obsahují horní byte adresy vypsaný nejprve v desítkové a potom ve dvojkové soustavě. Třetí pak celou adresu (dolní byte je vždy 254). V pravé části tabulky jsou jednotlivé bity a klávesy, které jim odpovídají. Pokud je vybraná klávesa stisknuta, bude mít odpovídající bit hodnotu 0, v opačném případě pak hodnotu 1.

Například: chcete testovat klávesu Q. V tabulce je napsáno, že tato klávesa je na portu 64510 (horní byte je 251) a jedná se o nultý bit tohoto portu. Testovací program může vypadat například takto:

```

ld    bc,64510      ; adresa portu do BC
in    a,(c)         ; přečti port do A
bit   0,a           ; platí Z je-li stisknuta

ld    bc,64510      ; adresa portu do BC
in    a,(c)         ; přečte port do A
rra   ; platí NC je-li stisknuta

ld    a,251         ; horní byte adresy do A
in    a,(254)       ; přečti port 254 do A
bit   0,a           ; platí Z je-li stisknuta

ld    a,251         ; horní byte adresy do A
in    a,(254)       ; přečti port 254 do A
rra   ; platí NC je-li stisknuta

```

Zde máte čtyři způsoby testování klávesy Q, nejkratší z nich je poslední program. Pokud budete testovat jiné, než krajní klávesy, nebude použití rotací vhodné - neplatí v případě, že testujete několik kláves na jednom portu, například rozeskok podle klávesy - například rozeskok v menu lze napsat takto:

```

ld    a,247         ; horní byte adresy do A
in    a,(254)       ; přečti port 254 do A
rra   ; zarotuj A (0 bit)
jp    nc,PRESSED1  ; platí NC při stisku 1
rra   ; zarotuj A (1 bit)
jp    nc,PRESSED2  ; platí NC při stisku 2
rra   ; zarotuj A (2 bit)
jp    nc,PRESSED3  ; platí NC při stisku 3
rra   ; zarotuj A (3 bit)
jp    nc,PRESSED4  ; platí NC při stisku 4
jp    NOPRESS      ; nebylo stisknuto nic

```

Při definici ovládání ve hrách se často používá program pro zjištění bitu a portu stisknuté klávesy. Těmito hodnotami se potom modifikuje ta část programu, která je volána pro řízení pohybu.

```

LOOP      ld    bc, #FEFE      ; do BC adresa portu
          in    a, (c)        ; přečti do A hodnotu
          cpl                    ; komplement A registru
          and   31             ; ponechej bity 0 až 4
          ret   nz            ; vrať se při stisku
          rlc   b              ; posun na další port
          jr   c, LOOP        ; není-li poslední, cykluj
          scf                    ; nastav C a vrať se
          ret

```

Pokud byla při zavolání programu stisknuta libovolná klávesa, vrátí program v registru BC adresu portu a v registru A nastavený odpovídající bit - platí podmínka NC. Pokud nebude nalezeno nic, vrátí program podmínku C. Bude-li stisknuto více kláves, nemusí program pracovat správně.

Posledním programem bude čekání na stisk libovolné klávesy. Uvedeme zde celkem tři způsoby:

```

WAITKEY   call  654           ; volání KEY-SCAN v ROM
          inc  e              ; pokud je v registru E
          jr   z, WAITKEY    ; číslo 255, testuj znovu

```

Toto je nejkratší způsob, jak napsat čekání na stisk klávesy, má jen tu nevýhodu, že mění registry AF, BC, DE, HL. Tuto nevýhodu nemá následující program (mění pouze AF):

```

WAITKEY   xor   a            ; vynuluje registr A
          in   a, (254)       ; přečti port 254 do A
          cpl                    ; komplement A
          and  31             ; testuj bity 0 až 4
          jr   z, WAITKEY    ; není stisk, opakuj

```

Tento program potřebuje vysvětlení. Pokud je horní adresa portu rovna 0, testují se najednou všechny klávesové porty. Každý bit vlastně reprezentuje celý sloupec kláves z tabulky. Je-li alespoň jedna klávesa stisknuta, je hodnota bitu rovna 0, jinak je rovna 1.

Pro čekání na klávesu lze použít také podprogram příkazu PAUSE v ROM. Na rozdíl od předchozích vyžaduje povolené přerušení a tedy i registr IY nastavený na 23610 (#5C3A), umožňuje však navíc po zadaném časovém intervalu pokračování i bez stisku klávesy. Program mění registry AF a BC.

```

          res  5, (iy+1)     ; klávesa není stisknuta
          ld  bc, 100        ; čekej maximálně 2 sec.
          ei                    ; povolení přerušení
          call #1F3D        ; podprogram PAUSE

```

Doba, po kterou se na klávesu čeká, je uložena v registru BC a měří se v padesátinách sekundy. Pokud do BC vložíte 0, bude čekání ukončeno pouze stiskem klávesy.

Popisované způsoby vyčerpávají všechny možnosti, jak testovat klávesnici. Každý program, který komunikuje přes klávesnici, používá nějaký z těchto způsobů.

Vyberte si ten, který nejlépe splňuje Vaše požadavky.

ZÁVĚREM

Článek, který jste nejspíš právě přečetli, je kapitola z připravované knihy s pracovním názvem **ASSEMBLER a ZX SPECTRUM**. Knížka bude obsahovat množství příkladů, které pokryjí všechny oblasti použití assembleru na ZX Spectru. Chceme všechny kapitoly pojmout stejně vyčerpávajícím způsobem (i za cenu našeho naprostého vyčerpání), jako kapitolu s testováním klávesnice. Pro informaci uvádíme několik pracovních názvů dalších kapitol:

Tisk testu na obrazovku - od obyčejného až k doporučenému, zvětšenému, zesílenému a dalším způsobům úprav.

Všechno o LOAD a SAVE - jak si vyrobit vlastní loader, jak používat rutiny z ROM...

Celocíselná aritmetika - násobení a dělení šestnáctibytových čísel, jak udělat počítadlo skóre, tabulku rekordů, počítání životů a různé další "finty".

Hýbáme obrazovkou (SCROLL a ROLL) - způsoby, jak lze pohybovat obrazovkou nebo její částí. Použití rotací a posunů.

Jak udělat sprite - neboli jak na obrazovku nakreslit letadlo, raketu, strašáka a jiné pohybuující se objekty. Jak provádět jejich animaci.

Použití tabulek a testů - použití rozsokových a dalších tabulek. Uložení textů a práce s nimi. Jak vše uložit v co nejmenším rozsahu paměti.

Kompresie dat - jak zkrátit obrázky, texty a ostatní pro uložení na kazetu, použití ve vlastních programech atd.

Grafika - body, čáry a jiné geometrické obrazce. Rychlá čárová grafika. Vyplňování obrazců.

Ke knížce bude patřit kazeta se zdrojovými texty složitějších příkladů. Příklady jsou odladěny na assembleru PROMETHEUS. Také text je psán s ohledem na široké možnosti tohoto ladičského systému. Komplet bude stát asi 150 Kčs (knížka + kazeta).

Budete-li se chtít assemblerem vážně zabývat, stanou se pro Vás takřka nepostradatelné některé programy z naší nabídky:

Psaní a ladění programů assembler **PROMETHEUS (199)**.
Ladění a prohlížení jiných programů (i grafika) **USER I (131)**.
Kompresi všeho možného připravovaný **USER II (cca 130)**.

Objednávky na programy a předběžné objednávky na **ASSEMBLER a ZX SPECTRUM** na adrese:

Proxima Software
P. box 24, pošta 2
400 21 Ústí nad Labem



si Vám dovoluje nabídnout programové vybavení pro počítače řady

ZX-SPECTRUM

a kompatibilní (Delta, Didaktik GAMA, Didaktik M, ...)

UŽIVATELSKÉ PROGRAMY

1. **ORFEUS** - hudební program - umožňuje vytvořit hudbu do Vašich programů (131,-).
2. **DESKTOP** - „textový procesor“ - kombinace obrázků s texty, proporcionalní tisk (210,-).
3. **PROMETHEUS** - integrovaný ladící systém pro programování v assembleru Z80 (199,-).
4. **USER I** - Devast Ace, Wlezley 7, Gargantua 4, Pantagruel 4, David 4, Goliath 4 (130,-).
5. **EDIT SAMPLER** - program pro vytváření zvukových efektů a digitalizaci zvuku (90,-).
6. **ZX-TOOLS** - univerzální program pro obsluhu disketové jednotky DIDAKTIK 40 (200,-).

Všechny námi nabízené programy jsou původní. Součástí programů jsou obsáhlé manuály.

HRY - HRY - HRY

Nabízíme následující soubory her (cena souboru 120,-).

7. **BAD DREAM** (aneb Zlý sen Františka Koudelky), **ATP tour simulator**, **KRTEK** a **autíčko**, **AKNADACH** a **prémie TOM JONES**.
8. **LETRIS**, **STAR DRAGON** („střílečka“), **ATOMIX**, **DOUBLE DASH** a jako **prémie BEERLAND**.
9. **První EXPEDICE** na divnou planetu, **HEXAGONIA** (ATOMIX II) a opět **prémie MUSICLOGIC**.

Všechny programy jsou nahrány na kvalitních zahraničních kazetách na profesionálním kopírovacím zařízení (Na zaslané kazety programy už nenahráváme).

PŘIPRAVUJEME

Již tradičně jsou to především hry **Bláznivá mise** (atraktivní textová hra) a **Agent W4C** (detektivní hra). Dále pak **USER II** (soubor nových komprimačních komprimačních programů) a řada dalších her a užitkových programů.

Připravujeme také několik programů pro soukromé podnikatele např. jednoduché a podvojně účetnictví, vedení skladu, výuka pro autoškoly aj. (dle přání zákazníka vypracujeme i programy na zakázku).

V neposlední řadě se jedná o programy pro disketovou jednotku DIDAKTIK 40 (výrobce Didaktik Skalica). Připravujeme disketové verze všech námi distribuovaných programů. Ve většině případů se tím značně zvýší komfort obsluhy (zejména u DESKTOPu a dalších uživatelských programů).

PROXIMA - software
post box 24 pošta 2
400 21 Ústí nad Labem

Tisk : firma SLON, Jiří Bartoš, U chemičky 18, 400 01 Ústí nad Labem, tel.:2187-3066

Elektronická verzia: 22.12.2006

OCR, sadzba a grafika: Peter Turányi alias Softhouse

Skenovanie: Luděk Sladký alias Sweet