

Tajemství DESKTOPu

Napište si vlastní Utility !

Welcome to

UNIVERSUM'S
DESKTOP

(C) 1991 PROXIMA

P.Box 24, Pošta 2, 400 21 Ústí nad Labem

Při psaní programů *Disk Operation Manager*, *New Converter*, *Color*, *DTP menu* a taky tiskových utilit jsem potřeboval vědět hodně detailů o Desktopu. Dost mi pomohl článek od Universuma „Vytvořte si utilitu“ (byl myslím na nějakém KUDu). Ovšem některé věci jsem tam nenašel, a tak jsem si je musel najít sám pomocí obrazkové DevastAce (a někdy to trvalo fakt dlouho). A jelikož si myslím, že by se Vám mé poznatky mohly někdy hodit, tak jsem se je rozhodl zveřejnit, resp. opsat ten článek od Universuma a doplnit.

START (Návrat do) Desktopu

Před návratem musí být nastaveno **IM 1** a povoleno, taky registr **IY** musí být nastaven na **23610** !!!

jp 48200

studený start Desktopu (pozor! vymaže se i celý text)

jp 48206

teplý start Desktopu (text se nemaže)

jp 48203

teplý start s konverzí textu (měl by se použít vždy, když něco provedete s fonty nebo šířkou textu!)

jp 24240

teplý start s výběrem obrázku

ret

to lze použít pouze po EXT p (viz dále)

Utility (1680 bytes, od adresy 46520)

EXT p = 46520

Po tomto příkazu Desktop převede jednotlivé řádky nastaveného bloku do grafické podoby a pro každý řádek zavolá podprogram na adrese 46520. Návrat z tohoto podprogramu by měl být zajištěn instrukcí RET (pokud utilita není tisková, tak návrat zaleží jen na Vás...). Pokud se ale během tisku stiskne BREAK, mělo by se skočit na 48206 (teplý start). Grafická podoba jednoho řádku se nachází na adrese uložené na 48209 (tzn. NE na adrese 48209, ale na té, která je na 48209 a 48210 uložena!). Grafický mikrořádek je široký 97 bajtů, a řádků je 12.

EXT i = 46523

Zavolá podprogram na adrese 46523.

Podprogramy

54916,SETL

Nastaví tiskovou pozici na levou stranu nejspodnějšího řádku na obrazovce (informační řádek v Desktopu).

54906,SETL2

Totéž co SETL, ale provede se i výmaz tohoto řádku.

54903,SETL3

Totéž co SETL2, ale navíc nastaví barvu Borderu.

57291,TT

Vytiskne na informační řádek text, na který ukazuje registr HL. Text musí být ukončen invertovaným znakem (+128). Na adrese 57310 si můžete nastavit tiskovou pozici. Nezapomeňte ale na to, že není ošetřen přechod na další řádek!

57300,SPOUT

Vytiskne mezeru na informační řádek.

57354,NUM5

Vytiskne na informační řádek obsah registru HL jako pěticiferné číslo.

57360,NUM4

Vytiskne na informační řádek obsah registru HL jako čtyřciferné číslo. Nesmí být větší než 9999!

57366,NUM3

Vytiskne na informační řádek obsah registru HL jako trojciferné číslo. Nesmí být větší než 999!

55070,INPCOM

Vstup textu/čísel (max. 10 znaků). Při volání musí registr HL ukazovat na nějaký text (ukončený invertovaným znakem). Tento text se vypíše před vlastní vstup.

Adresy textů v Desktopu:

```
55133 „FONT:“  
55138 „How many:“  
55147 „Name:“  
55164 „Are you sure (y/n)?“  
55184 „Verify (y/n)“  
55196 „Found:“
```

Text, který byl zadán, je od adresy **55152**.

Pokud je při vkládání stisknuto EDIT (CS+1) je po návratu z podprogramu nastaveno **z**. Při normálním ukončení – stisk ENTER, je nastaveno **nz**.

Pokud potřebujete vytisknout **číslo**, které bylo zadáno, tak po návratu z tohoto podprogramu **zavolejte ještě 53476 a pak NUM5** (nebo NUM4,3).

Příklad:

```
ld hl,55138 ;How many?  
call 55070 ;tisk  
jp z,... ;stisknuto EDIT = skok pryč  
call 53476 ;převod čísla z ASCII kódu  
;na normální  
call 57354 ;NUM5
```

54976,BLOCK

Vrací v **HL** adresu prvního a v **DE** adresu posledního řádku nastaveného bloku.

Pozor na to, že tento podprogram **někdy** vrací v DE špatnou hodnotu! Pokud totiž nějak upravíte označený blok (např. změňte fonty), tak se většinou musí skočit na adresu teplého startu s konverzí (jp 48203). A tady je kámen úrazu. Po dokončení konverze někdy zůstane označený blok, tak jak byl předtím, ale text může být (a většinou taky je) kratší (nemyslím v Desktopu, ale v paměti), a když teď zavoláte 54976,BLOCK tak v DE bude chybná hodnota, která ukazuje na místo, které už jako konec bloku neplatí (to platilo před konverzí). Na tohle jsem doplatil v DTPmenu1.0 (2.0 s tím už počítá...). Chyba se dá opravit dvěma způsoby. Buď vždy před skokem na teplý start s konverzí nastavíte v Desktopu začátek i konec bloku na 1. řádek v Desktopu (musíte poopravit podprogram BLOCK – na adresu 54976+1 a 54979+1 dejte hodnotu 24304), nebo (a tak jsem to vyřešil i já) při práci s blokem testujete konec textu (sekvence 13,0,13) a pokud nastane, tak program skočí na teplý start s konverzí (a můžete nastavit i blokovou značku jako u prvního způsobu).

55507,UPHL**55596,DOWNHL****55430,DRAWBOX**

Nakreslí (xorovaně) rámeček

HL – adresa levého horního rohu rámečku

55432 – šířka rámečku v osmicích bodů

55445 – výška rámečku v bodech

Šířku a výšku (adresa 55432 a 55445) musíte pak vrátit do původního stavu!!

56282,ROLLS

Rolování celé obrazovky o jeden bod doleva nebo doprava. Směr rolování udává registr **A** – **49** (kód klávesy 1) = doleva, **50** (kód klávesy 2) = doprava.

49533,PICTURE (tak jsem to pojmenoval já)

Vybrání obrázku (příp. části obrázku) pomocí okénka. Nevýhodou tohoto podprogramu je, že se po stisku SPACE vrací do Desktopu. Dá se to ale obejít – musíte **nahradit instrukci na adrese 49632 jr... na instrukci ret** (pak to ale nezapomeňte dát do původního stavu). Ani toto řešení však není dokonalé, protože pokud už se obrázek do paměti nevejde, dojde k návratu do Desktopu. Vyřešit to můžete tak, že před zavoláním podprogramu provedete test na velikost volné paměti (mělo by tam být raději dost místa), a pak teprve zavoláte výběr obrázku (49533).

Příklad:

```
ld hl,(54890) ;nejprve zjistíme
ld de,(54893) ;volné místo
or a ;v Desktopu
sbc hl,de ;(budeme ho mít v hl)
ld a,h ;teď otestujeme
cp 36 ;jestli se obrázek
; vejde do paměti
jr c,... ;odskoč, je-li tam
; málo místa
ld a,201 ;kód instrukce RET
ld (49632),a ;
ld a,0 ;můžete použít i xor a
ld (49633),a ;
call 49533 ;výběr obrázku
ld a,24 ;vrácení pův. hodnot
ld (49632),a ;
ld a,155 ;
ld (49633),a ;
```

Barvu BORDERu a atributů si DESKTOP (PAPER, INK, BRIGHT, FLASH) nastaví podle systémových proměnných:

23624 pro **BORDER** (8*barva),

23693 pro **atributy** (128*blikání + 64*jas + 8*barva papíru + barva inkoustu).

54923,BORDER

Nastaví barvu borderu podle 23624.

57170,CLRSCR

Nastaví atributy podle 23693.

57233,DIV8HL

Provede $HL=HL/8$ (vydělí registr HL osmi).

56868,DIVMOD

Podprogram pro 16bitové dělení, **Dělenec je v HL, dělitel v DE**. Po provedení je **HL** zbytek po dělení a v **AC** podíl. Při dělení nulou je výsledek 65535.

55560,ADRESSES

Adresy jednotlivých řádků na obrazovce (celkem 15 adres) – 16384, 17440, 16480, 17536, 16576, 17632, 18464, 19520, 18560, 19616, 18656, 21504, 20544, 21600, 20640

54932,WAITK

Čeká na stisk klávesy nebo pohyb joystickem (pokud byl v Desktopu zapojen) – tedy PAUSE 0

55965,IN31

Testování klávesnice a portu 31 (kempston)

z = nic nestištěno

nz = stištěna klávesa nebo pohyb joystickem

56105,CLICK

Pípnutí – délka se nastavuje sys. proměnnou na adrese **23609**.

57074,IQLDIR

Ldir s inteligencí – neprovádí se v případě, že je v BC 0 (nula). Pokud je nutné volí se místo instrukce *ldir* instrukce *lddr* (když by mohlo dojít k chybnému přenosu při překryvu obou oblastí).

54944,ROMLD

Volání podprogramu pro LDBYTES v ROMce tak, že se při chybě nebo stisku BREAK nevypisuje chybové hlášení (řízení se vrátí zpět do volajícího programu s **nc**). Podprogram tedy nahraje z kazety blok, jehož LEADER je v A, délka v DE a počáteční adresa v IX.

Simulace příkazu EXT

Uvedený podprogram dokáže nasimulovat jakýkoli příkaz DESKTOPu, který se vyvolává stisknutím EXT *něco* v editoru.

```
DTPEXT push af
        ld  hl, (55003)
        call 52564
        pop af
        ld  sp, (48243)
        set 1, (iy+118)
        jp  48452
```

Před zavoláním musí být v **registru A** hodnota, která říká, **jaký příkaz se má provést**. Např. pro EXT *i* musí být v A znak " i ", pro EXT *SS+k* zase " + ", u EXT *x* bude " x ", a podobně pro ostatní příkazy.

Tento podprogram jsem objevil při „prohlížení“ Utility *PullDown* (byla na nějakém KUDu) a ušetřil mi hodně práce při psaní *DTPmenu*...

Utility+

Jak jistě víte Utilitka+ je program, který DOM nahraje místo sebe (*ani nevíte kolik lidí (staodvacetosmičkářů) mi to už vytklo*) a spustí. Utilita+ by měla počítat s tím, že může být spuštěna i na Gamě 80K (DOM před spuštěním utility vloží číslo do registru A (80=Gama, 128=128K)).

Návrat z Utility+ zajišťuje program na adrese 23300 (pokud si ho ovšem něčím nepřepíšete...).

Utilita+ se pozná podle toho, že 10 znak v jejím jméně je @.

Jelikož se mě samotnému taky zrovna moc nelíbí, že Utilita+ přepíše DOMa a ten se pak musí nahrávat znovu, zavedl jsem Utility+II.

Utility+II

Půjdu na to trochu od lesa. Každý majitel 128K nyní Desktop používá tak, že v 17. stránce má DOMa, v 19. DTPmenu a ve 20. je takový odpadkový koš (DTP menu si tam odkládá kopii obrazovky). Ovšem 22 a 23 jsou zcela prázdné. A právě toho jsem využil pro Utility+II. Už při nahrávání Desktopu si ji (je) tam uživatel nahraje a když ji pak bude chtít spustit, použije nové (v2.0) DTPmenu (anebo klidně pěkně ručně - vyskočí do BASICu, přestránkuje a spustí) Spouštěcí adresa Utility+II ovšem nesmí být 49152, ale **49200** (pokud bude Utility+ nahrána v 22. stránce) nebo **49210** (pokud bude v 23. stránce). Po takovémto spuštění se musí Utilita+II sama přesunout do stránky 20, dát si na 23300 vlastní program na návrat do Desktopu (a třeba i zásobník) a ve 20. stránce se spustit.

Utilita+II je (musí být) plně kompatibilní z Utilitou+. Je nutné počítat s tím, že ji klidně může nahrát DOM a spustit (JP 49152). V tomto případě bude pracovat ve stránce 17 nebo ve druhé bance na Didaktiku Gama 80K. **Pamatujte na to** (a né tak, že tam dáte suchou hlášku only Utility+II, či only 128K) **!!!**

Příklad:

```
org 49152

START jp START2

defm 'Utility+II:název'

org 49200

ld a,22
jr GO
nop
nop
nop
nop
nop
nop
ld a,23
GO ld (PAGE1+1),a
ld (PAGE2+1),a
```

podprogram PRESUN hodíme
do obrazovky (atributů)

```
ld hl,PRESUN
ld de,22528
ld bc,PRELEN
ldir
```

na adresu 23300 musíme dát
vlastní program pro přestránkování
a návrat do DESKTOPu

```
ld hl,A23300
ld de,23300
ld bc,30
ldir
jp 22528
```

přestránkování a simulace GOTO 1

```
A23300 ld a,16
ld bc,32765
out (c),a
xor a
ld (23619),a
inc a
ld (23618),a
ld (23620),a
ld sp,(23613)
jp 7030
```

přesun ze stránky 22 nebo 23
do stránky 20 (přes obrazovku)

```
PRESUN ld hl,49152
ld de,16384
ld bc,6144
ldir
ld a,20
ld bc,32765
out (c),a
ld hl,16384
ld de,49152
ld bc,6144
ldir
PAGE1 ld a,0
ld bc,32765
out (c),a
ld hl,49152+6144
```

```
ld de,16384
ld bc,6144
ldir
ld a,20
ld bc,32765
out (c),a
ld hl,16384
ld de,49152+6144
ld bc,6144
ldir
PAGE2 ld a,0
ld bc,32765
out (c),a
ld hl,49152+6144+6144
ld de,16384
ld bc,4096
ldir
ld a,20
ld bc,32765
out (c),a
ld hl,16384
ld de,49152+6144+6144
ld bc,4096
ldir
jp START3

nop
PRELEN equ $-PRESUN

START3 ld a,20
přepsat stránkování v Utilitě+
(normálně tam je stránka 17)

START2 ld (GAMA+1),a
di
ld sp,23296+242
im 1
ei
...
GAMA ld a,128
cp 80
call z,ZMENGAMU

ld hl,MENU
call PRINTAT
...
```

Doufám, že jste to pochopili. Možná se to někomu může zdát zbytečně komplikované, ale věřte mi, že toto řešení je nezbytné kvůli tomu, aby to mohlo fungovat i jako normální Utilita+ (17/GAMA) a taky, aby Utilita+II byla funkční jak ve stránce 22 tak i 23 (z BASICu).

Hlavně dodržujte standardní začátek každé Utility+II (od adresy 49155 **musí být Utility+II: její název**). Je to nezbytně nutné, protože DTPmenu vypisuje názvy právě odtud. A aby uživatel rozlišil, že se jedná o Utilitu+II, tak by na disketě měla být pojmenována takto: ????????@.B (logika v tom je - 1 zavináč pro Utilitu+ a 2 pro Utilitu+II).

Přeji Vám hodně štěstí a pevné nervy při psaní programků pro Desktop.

MTs (a UNIVERSUM)