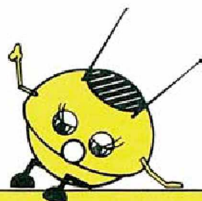
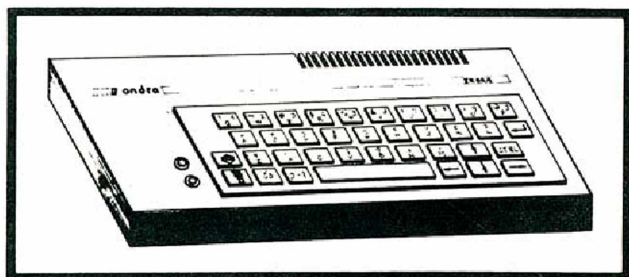


TESLA



UŽIVATELSKÁ PŘÍRUČKA MIKROPOČÍTAČE

ondra



MIKROPOČÍTAČ ONDRA SPO 186

3

TOOL



Programový blok TOOL poskytuje uživatelům mikropočítače ONDRA základní programové prostředky pro práci v jazyce symbolických adres (assembleru). Programový blok obsahuje překladače jazyka symbolických adres, zpětný překladač (překlad strojového kódu do jazyka symbolických adres), editor pro přípravu zdrojových textů – programů v assembleru, trasovací program pro ladění programů v assembleru a pomocné funkce, které usnadňují práci v jazyce symbolických adres. I když mikropočítač ONDRA je vybaven mikroprocesorem U880D (ekvivalent Z80), je v TOOLu zařazen absolutní překladač pro mikroprocesor 8080, jehož instrukční soubor je podmnožinou instrukčního souboru Z80. Protože mikropočítače s procesorem MHB 8080 jsou v ČSSR nejrozšířenější, byl pro mikropočítač ONDRA zvolen tento překladač. Programy připravené na mikropočítači ONDRA (s instrukčním souborem 8080) by měly být přenositelné na jiné typy mikropočítačů, např. SAPI 1. Pochopitelně při přenosu programů musíme respektovat technické odlišnosti jednotlivých mikropočítačů (např. rozdělení paměti, obsazení vstup/výstupních portů atd.).

Vzhledem k tomu, že mikropočítač ONDRA by měl sloužit uživatelům pro seznámení se s výpočetní technikou a měli by se naučit základům programování nejen ve vyšších jazycích, např. BASICu, ale i s jazykem symbolických adres - assemblerem pro nejrozšířenější typ mikropočítačů.

Editor PEDIT slouží pro přípravu programů v assembleru. Je možno ho použít i pro přípravu programů v BASICu a nebo pro přípravu textových souborů. BASIC je ovšem vybaven vlastními prostředky pro přípravu a editaci programů v jazyce BASIC. Pro práci s textovými soubory je mikropočítač vybaven textovým editorem TEDIT, který poskytuje uživatelům podstatně více služeb než PEDIT. Textový editor TEDIT je součástí základního programového vybavení pro mikropočítač ONDRA.



Představuje soubor programů, které umožňují na mikro-
počítači ONDRA připravit, přeložit a odladit program na-
psaný v jazyce symbolických adres (assembler).

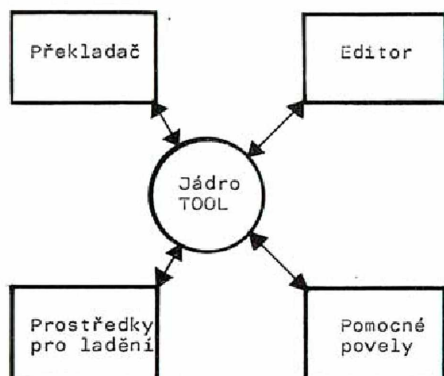
Programování v assembleru již vyžaduje znalost archi-
tektury mikroprocesoru (tj. práci s registry, organizací
paměti atd.), je velmi náročné a vlastně slouží k doplně-
ní programového vybavení Ondry pro ty, kterým nestačí BA-
SIC svojí rychlostí. TOOL je určen také pro seznámení se
základy programování 8-bitových mikroprocesorů. Proto byl
vybrán jazyk symbolických adres používaný na mikroproce-
soru 8080, který se používá na mikropočítačích SAPI 1,
PMD-85, IQ 150, PP-01 atd. Také jej lze použít na Ondrovi.
Takto se sice omezíme na jistou část instrukčního souboru,
který umí mikroprocesor U-880 (ekvivalent Z80). Jako odmě-
na za tento ústupek bude program, který funguje nejen na
Ondrovi, ale i na ostatních mikropočítačích. Mladí pro-
gramátoři budou znát assembler, který se používá i na ři-
zení v průmyslu (např. SAPI-80, SM 50/40).

Na rozdíl od profesionálních vývojových systémů, není ONDRA vybaven vnější pamětí s přímým přístupem (např. disk). Jako vnější paměť má pouze kazetový magnetofon. Data se ukládají na běžně dostupné kazety nebo na kazety pro osobní počítače (označené OP).

Z hlediska uživatele mikropočítače ONDRA ovšem kazeta představuje mnoho omezení. Proto byla zvolena koncepce programu TOOL tak, že překládá program jehož zdrojový text je uložen v operační paměti. Přeložená část programu se ukládá na jiné místo paměti. Tímto odpadne čtení zdrojového programu z kazety a převíjení kazety mezi jednotlivými průchody překladu.

Vlastní doba překladu programu v jazyce symbolických adres je překvapivě krátká a protože ONDRA má poměrně velkou kapacitu paměti, lze snadno přeložit poměrně dlouhý program (až 4 kbyte) přímo v operační paměti.

Program TOOL si můžete představit jako několik programových modulů, které dohromady vytváří integrované programové vybavení pro práci v jazyce symbolických adres.



Obsluha jednotlivých modulů je dále popsána. I přesto, že tento návod není učebnicí programování, je v něm popsán instrukční soubor mikroprocesoru 8080, který je podmnožinou instrukčního souboru Z80 (nebo U880D z NDR, který je použit u mikropočítače ONDRA).

Nyní si částečně popíšeme způsob práce s programem TOOL. Program musíme zavést z kazety do operační paměti, např. povel "L" (Load). Podrobný popis je uveden v Návodu k použití mikropočítače ONDRA. Na obrazovce se vypíše ohlášení programu TOOL

```

.L
TOOL ASM-80 V5.0
+
```

Příkazem "P" vyvoláme obrazovkově orientovaný editor "PEDIT". Ten zaplní polovinu obrazovky nápovědou (Menu). Nyní zavedeme z kazety program, který budeme ladit (# A u PEDITu). Nebo použijeme povel "I" pro zadávání znaků z klávesnice a začneme psát nový program.

```

Např.: TITLE      TEST  překladače          14.10.86
        START:    XRA  A  ;  nuluj akumul.
        ZNOVA:    INR  A  ;  zvýš o 1
                JMP  ZNOVA  ;  opakuj
        END
```

Potom může takto vytvořený program v PEDITu povel "E" uložit na kazetu a povel "J" se vrátíme do TOOLu. Pozor: První řádek se překladačem ignoruje. Slouží k různým poznámkám. Pro vytváření odstavců v programu se používá znak "→".

Nyní můžeme program přeložit.

Poznámka: Když není použita pseudoinstrukce "ORG", systém sám dosadí adresu 1000H; je-li ORG 7000H, potom se program přeloží a uloží od adresy 1000H. Povel "M" (Move) lze potom umístit program na správné místo v operační paměti.

Pozor: Potom se ale přemaže zdrojový text programu!

Proto používáme pro umístění programu prostor od adresy 1000H až 1FFFH.

Požadavek na překlad se zadá povel:

+A

ASSEMBLER Ondra MHB 8080

MODE? X

ASSEMBLER Ondra MHB 8080 PASS=C

Podtržené znaky zadá obsluha. Jejich přesný význam je popsán dále.

Na obrazovce se objeví protokol o překladu (výpis programu, někdy se setkáte s anglickým výrazem "listing") po výzvě PASS zadáme ↵ a překlad je ukončen.

Pomocí příkazů "S", "U", "D" se můžeme přesvědčit, že opravdu od adresy 1000H je uložen přeložený program (nazývaný strojový kód).

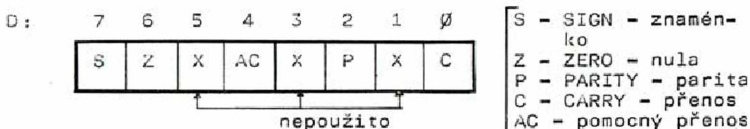
Jako další možnost nám umožňuje TOOL takto přeložený program ověřit (ladit). Tyto služby ocení každý u psaní delších programů, kdy i zkušený programátor se dopouští chyb.

Povелеm `T=1000 <CR>` můžeme spustit trasování našeho textu. Na obrazovce se objeví obsahy registrů a instrukce, která se bude provádět na adrese 1000H. Po stisknutí znaku mezera se provede jedna instrukce programu. Znovu se objeví obsah registrů, další instrukce atd. Po každém stisknutí klávesy mezera se provede další krok. Po stisknutí klávesy "`↵`" se trasování programu ukončí. Povel "W" zrušíme Menu editoru.

Nyní si můžeme odzkoušet různé možnosti trasování a nastavování podmínek. Až zjistíme, jaké úpravy je třeba provést, můžeme povel "P" znovu spustit editor PEDIT. Původní text nám totiž zůstal v operační paměti. Provedeme patřičné modifikace v programu. Potom je dobré zdrojový text uložit na kazetu. Stává se, že chybou při práci se zásobníkem (STACK) se poničí celý obsah operační paměti. Tak bychom si mohli zničit skoro hotový program. Postupně si dále budeme zkoušet další možnosti programu TOOL.

Měli bychom se seznámit s malým rozdílem mezi prováděním instrukcí u mikroprocesorů 8080 a Z80. Jediný rozdíl je v nastavení bitů ve stavovém slově PSW po provedení aritmetických nebo logických operací.

Stavové slovo: 8080



1. Bit D1 u "Z80" slouží pro ADD/SUB operace "N"
 2. Bit D2 u "Z80" se liší po provedení aritmetických operací, kde indikuje přetečení a jenom po logických operacích indikuje paritní bit jako u 8080.
 3. Bit D4 je jinak nastaven po instrukci DAD, rotacích.
- Zde se taky liší navzájem mikroprocesory "8080" a "I8085".

Prakticky jediná závažná změna je v chování bitu D2, kde u "8080" má význam paritního bitu a "Z80" se po aritmetických operacích chová jako indikace přetečení.

Po seznámení se s touto kapitolou jsme získali základní názor, jak pracuje TOOL. V dalších kapitolách se podrobně seznámíme s možnostmi a příkazy editoru PEDIT, syntaxí jazyka symbolických adres všemi povely TOOLu. Kdybychom někdy chtěli využívat instrukci Z80, můžeme pomocí pseudoinstrukce "DB" definovat operační kód.

Doufáme, že programový soubor TOOL se stane vaším cenným pomocníkem a usnadní vám práci a umožní vám vniknout do tajů programování v assembleru. Pro získání základních znalostí doporučujeme se seznámit s příslušnou odbornou literaturou.



Soubor programu "TOOL" obsahuje programy pro práci v jazyku symbolických adres (Assembleru) pro editaci, překlad a ladění uživatelských programů.

Programy TOOLu lze rozdělit do pěti základních skupin:

- 1) Překladače (assembly) - textový, jednořádkový
- 2) Editor - Pedit, příprava zdrojových textů a jejich opravy a modifikace
- 3) Trasovací program - (tracer) pro ladění assemblerových programů
- 4) Zpětný překladač - Deassembler, převod strojového kódu programu do symbolické verze
- 5) Pomocné programy - základní služby monitoru a rozšíření

3.1. Seznam_povelů

V prvním sloupci je uveden kód povelu zadávaný z klávesnice, ve druhém je uveden běžně používaný anglický název povelu a ve třetím je stručný popis funkce. Jak sami můžete vidět, většina povelů je odvozena v prvních písmen anglických názvů tak, jak je u mikropočítačů zvykem.

+...	Ohlášení na displeji programu TOOL
+A ASSEMBLY	Start Assembleru a volba režimu překladač "PASS="
+B BEGIN	Nulování tabulky symbolů, podmí- nek, nastavení zásobníku (STACK) a registrů
+C CONDITION	Nastavení podmínky TRACERu
+D DISPLAY	Zobrazení obsahu paměti (<L.ADR> <H.ADR>)
+E EDITOR	Start editoru (PEDIT)
+F FIND	Vyhledání řetězce znaků v paměti: <L.ADR> , <H.ADR> .. Oblast vy- hledávání <N=1-3> .. Počet vyhle- dávaných bajtů <DATA1> <SP> .. <DATA1> <SP> .. vyhledávaná data <MASKA1> .. <MASKAN> .. Masky (log. součin)
+G GOTO	Start od žadané adresy s volbou přerušení: <START.ADR> , <BREAK ADR1> , <BREAK ADR2>
+H HEXADECIMAL	Hexadecimální součet a rozdíl
+K KAZETA	Příkaz kazetového OS MIKOS (viz návod k použití)
+L LINE - - ASSEMBLER	Jednoduchý přímý překladač (pro krátké programy a opravy při ladění)
+M MOVE	Přesun obsahu paměti do jiné oblasti: <L.ADR> , <H.ADR> , <L.ADR CÍLOVÉ OBLASTI>
+N NULL	Nulování registru a podmínky (viz +B)

+P PEDIT Start programu PEDIT (editoru)
 +Q QUIT Navrat do monitoru
 +R REGISTER Výpis obsahu registrů
 +S SUBSTITUTE Záměna obsahu paměti (ADR):
 "←" posun zpět a výpis adresy
 "." vypíše adresu
 "X" vloží ASCII kód "X" (apostrof+znak)
 +T TRACE Start TRACERu, parametry jako příkaz +G
 a respektuje se nastavená podmínka +C
 +U UNDEFINED CODE Zobrazení obsahu paměti v ASCII kódu:
 <L.ADR> , <H.ADR> 32 znaků na řádek
 +V VERIFY Deassembly (<ADR>) - zpětný symbolický
 překladač
 +X EXAMINE REG. Zobrazení a výměna obsahu registru
 +Y VERIFY PUNCH Deassembly, ale současně запиše na ka-
 zetu zdrojovou část Deassemblovaného
 programu: ukončit <CR> a uzavřít soubor
 příkazem "K_C" (viz Návod k použití)

Poznámka pro VERIFY a FIND:

Po znaku "." se přejde do režimu SUBSTITUTE;

Je možné udělat opravy, změnit adr. a po znaku <CR> se vrátit do původního povelu.

Znak <CR> je na klávesnici mikropočítače ONDRA označen

"↵". V dalším textu budeme používat označení <CR> .

3.2. Překladače

Součástí souboru TOOL je překladač absolutního assembleru. Překladač jazyka symbolických adres - JSA (Memory Assembler) umožňuje překlad zdrojového textu umístěného v paměti nebo na kazetě. Tento překladač je dvouprůchodový. Podrobný popis je uveden v kapitole 5. Assembler, zde jsou uvedeny příkazy pro řízení překladače.

Vyvolání:

+A ; vyvolání překladače (textového)

Dále soubor TOOL obsahuje jednořádkový, jedno-průchodový překladač Assembleru, který lze použít pro opravy nebo pro jednoduché krátké programy v Assembleru.

Vyvolání:

+L=ADR <ADR 1> ; vyvolání jednořádkového překladače Assembleru, ADR 1 volitelná poč. adresa.

Po vyvolání se ukazatel na obrazovce nastaví do středu následujícího řádku a čeká na vstup řádku Assembleru. Po <CR> provede překlad řádku (strojový kód, předřadí před text) a čeká na vstup nového textu.

Ukončení překladu se provede zadání prázdného řádku.
 (Pouze <CR> nebo pseudoinstrukci END <CR>).
 Povel (+L) lze zapsat strojový kód na libovolnou adresu
 v operační paměti. Při chybě operátora se lze vrátit na
 původní adresu a opakovat řádek. Je možné použít symbolic-
 kých návěstí, ale odkazy jsou možné jen dopředu, tj. na
 nižší adresy.

Poznámka: Po zadání příkazu "L" se vypíše nastavená adre-
 se pro uložení řádku programu. Pokud požadujeme
 jinou adresu, zadáme z klávesnice novou adresu.

Příklad:

```

- START MVI A,10H      ; zdrojový text
1000 3E 10      START MVI A,10H      ; po provedení
-               ; očekává se vstup
... další řádky
- <CR> ... ukončení překladu, (prázdný
                        řádek)
```

Textový překladač JSA (Memory Assembler)

Po povelu +A TOOLu je nutno zadat způsob překladu

A) Zadání modu pro překlad:

-R zdrojový program z vnější paměti (READ). Druhý
 průchod znovu vyžaduje vyhledání souboru na ka-
 zetě.

- X Překlad zdrojového textu uloženého v paměti, zdrojový program je uložen v paměti po edici. Pro MOD "X" musí být program zaveden editorem.

Po zadání modu lze řídit chod překladače

B) Řízení průchodu (PASS):

- 1 První průchod se provede automaticky (MODE:R,X)
- 2 Druhý průchod + výpis programu - (resp. písmeno "C"
- výpis na obrazovku
- P Druhý průchod s výpisem po stránkách (page, listing)
po 20 řádcích (a po <SP> se pokračuje stisknutí tlačítka "mezera" se pokračuje ve výpisu).
- S Vypíše se seznam symbolů s přiřazenými hodnotami
(symbol table)
- N Vypíše pouze chyby při druhém průchodu (No list)
- <CR> Návrat do TOOLu - ukončení činnosti překladače
- K ukládá se výpis programu na kazetu

Soubor je nutné ukončit повеlem K_C v monitoru.

- ^P <CTRL P> řídí připojení a odpojení tiskárny.
(Možno nastavit výpis programu na tiskárnu). Po-
zor! Tiskárna musí být připojena a zapnuta.
- Q Návrat do TOOLu

C) Plášení chyb

- U Nedefinovaný symbol nebo instrukce (UNDEFINED)

- O Chybný operand (OPERAND ERROR)
- M Vícenásobná definice (MULTIPLY DEFINITIONS)
- L Chyba v návěští (LABEL ERROR)
- F Plná tabulka symbolů (FULL SYMBOL TABLE)
- W Neuložil se strojový kód, délka je > 4k BYTE (X MODE) (WARNING)
- X Chyba ve výrazu (EXPRESSION ERROR)

3.3. Editor Pedit

Editor slouží k přípravě, opravě a modifikaci textových souborů, které mohou být uloženy v paměti nebo na vnějším mediu, nebo jejich kombinace (doplnění souboru atd.). Podrobný návod na použití editoru je uveden v kapitole 4 - PEDIT. Program PEDIT lze vyvolat z TOOLu příkazem +P nebo +E. Po ukončení edice a uložení programu na kazetu lze program okamžitě přeložit (povel +A, mode X).

Příklad:

+P ; skok do Peditu

Poznámka: Návrat z Peditu do TOOLu se v Peditu provede příkazem:

*J ↓ ↓

3.4. Trasovací a ladící program TRACER

Trasovací program "TRACER" umožňuje uživateli sledovat chování, chod programu a provádět v programu změny. Velmi zrychluje práci při ladění programu na malých systémech. Způsob práce trasovacího programu se řídí podle nastavených podmínek (pro nastavení podmínky slouží příkaz TOOLU +C), nastavením při spuštění traceru (viz příkaz +T). Dále je možno v průběhu trasování měnit podmínky trasování. Mezi trasovací příkaz G s parametry (body zastavení - break point), kdy dojde k přerušení chodu programu na zadávaných bodech. Úseky mezi body zastavení probíhají v reálném čase!!

+C -CONDITION	Nastavení podmínky, pro řízení trasování může být nastavena jen jedna podmínka. Podmínka N je nastavena po startu TOOLU.
B.. BRIEF:	Výpis PC a instrukce.
C.. CALL:	Trasuje pouze hlavní program, podprogramy provádí v reálném čase.
L.. LOOP:	Trasuje pouze instrukce skoku, umožňuje sledovat program v programových smyčkách
M.. MASK:	Trasování zvolené instrukce s maskou: N=<-3> .. počet bytů sledované instrukce

$\langle DATA1 \rangle$.. $\langle DATA_N \rangle$.. zadané instrukce
 $\langle MASKA1 \rangle$... $\langle MASKA_N \rangle$.. zadané maska
 (log. součin)

Při shodě zadané a sledované instrukce s maskou vypíše zprávu o stavu registrů.

Poznámka: Podrobnější popis a příklady jsou uvedeny u příkazu TOOLu "F" (FIND - viz 3.6.)

- N.. NULL:** Zrušení nastavené podmínky, vypisují se registry, adresa a instrukce a provede se základní nastavení
R.. REGISTER: Provede výpis při shodě dat v registru $\langle JMÉNO REG \rangle$, $\langle DATA \rangle$
S.. STACK: Provede výpis při vybočení zásobníku z oblasti $\langle L.ADR \rangle$, $\langle H.ADR \rangle$
W.. WINDOW: Sleduje program v zadaném úseku a vypisuje: $\langle L.ADR \rangle$, $\langle H.ADR \rangle$

Příklady:

+C: B ; BRIEF výpis PC a instrukcí
 +C: N ; zrušení podmínky trasování
 +C: W=7000H=7100H ; trasování v úseku 7000 až 7100
 +C: M=2 ; 2.bytové instrukce
 =00 =41 (RESP.='A) ; hledané instrukce, které používá
 =00 =FF (MASKA) ; znak "A" tj. hex.41. Vybrat všechny instrukce které v druhém byte

obsahují 41H, tj. znak A.

+C: RA=01

; je-li obsah registru A (akumulátoru) roven 01, potom se trasuje, v ostatních případech probíhá program v reálném čase

Jednotlivé režimy trasování

T.. trasování a provádění I/O

Po prvé je nutno zadat adresu startu programu (1000) a adresu pro přerušení (1006) a popřípadě (1009).

T=1000 \square = 1006 \square = 1009 <CR>

T= <CR> ... pokračuj od adresy v PC (až po prvním přerušení od BREAK POINTU)

T= <SP> = <BREAK> <CR> .. pokračuj od adresy v PC, ale od BREAK zobrazuj adresy

T= <ADR> <CR> ... START.ADR=BREAK.ADR, trasuje se a zobrazují se registry již od udané adresy

T= <START.ADR> = <BREAK.ADR1> = <BREAK.ADR2>

Úsek mezi START.ADR a BREAK.ADR se provádí v reálném čase!!!

Trasuje se po dosažení první adresy z BREAK POINTU. Vypiše obsah registrů adresa a instrukce. Po znaku mezera se provede další krok, znakem CR se končí.

Příklady výpisu trasování:

A) Výpis registru

```
--P-7010 A-D0 F-56 B-01 C-00 D-00 E-00
H-70 L-4E S-4380 :FFFF @CD ZP ← stavové slovo
7010 CD8770 CALL DUMMY <SP> ← pokračuj
--P-7086 A-D0 F-56 B-01 C-00 D-00 E-00
H-70 L-4E S-437E :7313 @CD ZP
7087 3C DUMMY: INR A <SP> ←
--P-7087 A-D1 F-86 B-01 C-00 D-00 E-00
H-70 L-4E S-437E :7313 @CD SP
7088 C9 RET <CR> ← konec
```

Vysvětlivky:

- P PC - program counter (programový čítač instrukcí)
- A-L registry A, B, D, D, E, H, L; A-akumulator (střadač)
- F FLAG - příznaky S, Z, AC, P, CY
- S STACK POINTER - ukazatel zásobníku
- : data (slovo) uložena na vrcholu zásobníku (např. adresa návratu pro RET)
- @ obsah paměťového místa adresovaného dvojicí reg. <HL>

SZPC rozdekódování stavového slova

-S SIGN (minus)
-Z ZERO (nula)
-P PARITY (Sudá parita)
-C CARRY (CY-přenos)

Symbole <SP> , <CR> jsou odpovědi pro trasování, po zadání
<SP> se pokračuje v trasování, <CR> trasování ukončí.

Poznámka:

Trasovat nelze programy v EPROM, protože TRACER zapisuje do
paměti doby přerušení a do paměti ROM nelze zapisovat, lze
paměti ROM projít v reálném čase.

Na displeji se zobrazuje při trasování deasemblovaný program
po řádcích (podle nastavených podmínek +C).

Vlastní řízení průběhu trasování

"SP" ... další instrukce

"CR" ... konec

"," ... (čárka) nová podmínka (viz "C")

"^C" ... přerušení trasování

Příklad:

```
+C: B ; nastavení podmínky trasování-výpis PC a instrukce  
+T=7000 <CR> ; trasuje se od startu  
... výpis instrukce <SP>  
... další instrukce <+>; požaduje se změna podmínky pro  
trasování  
=L ; zastavení na skokových instrukcích  
... běh programu a zastavení (výpis skoku)  
...  
... <CR>; ukončení trasování
```

3.5. Zpětný překladač DEASSEMBLER

Zpětný překladač umožňuje kontrolu strojového kódu tím, že převádí strojový kód do symbolického tvaru. Tak je možno při ladění sledovat, zda došlo ke změně programu apod. Pro zpětný překladač jsou dva příkazy TOOLu:

```
+V zobrazování na obrazovce  
+Y zobrazování na obrazovce a uložení zdrojového  
textu na kazetu
```

Zpětný překladač pracuje po řádcích, další řádek zobrazí po stisknutí klávesy mezera <SP>, klávesa <CR> ukončí překlad.

Příkazy V,Y mají jeden parametr adresu místa paměti, odkud se provádí zpětný překlad.

Příklad: +V = 1000

... výpis řádky (adr., stroj.kód, zdrojový
text)

< SP> ... další řádka

< CR> ... ukončení překladu

< , > přechod do příkazu SUBSTITUTE,
kde lze provést změny a po < CR>
návrat do zpětného překladače.

Poznámka :

Stejný postup platí i pro příkaz Y (se zápisem). Pozor, zdrojový text se zapisuje do vyrovnávací paměti (bufferu) a na kazetu se запиše až po naplnění bufferu a nebo po příkazu K_C (CLOSE). Příkaz "K_C" se musí vždy použít po skončení zpětného překladu tak, aby se ukončil výstupní soubor na kazetě. Jinak by při čtení došlo k chybě.

Tabulka symbolů:

P o z o r ! Při povelu (+B,+N) se zruší tabulka symbolů, pořázená při překladu.

Assembler a deassembler sdílí společnou tabulku symbolů, takže příkazem +L a EQU lze tabulku symbolů během deasemblování doplňovat.

3.6. Pomocné příkazy

Pomocné příkazy umožňují využití služeb monitoru, aniž by bylo nutné vracet se z TOOLU do monitoru a zpět. Dále pak rozšiřuje některé možnosti (např. +H, +F), některé příkazy mají jiný význam než v monitoru (F, D).

Popis příkazů:

+B - počáteční nastavení podmínek a nulování tabulky symbolů - BEGIN

+D - zobrazení obsahu paměti (DUMP) - hex. výpis paměti

Formát: + D =<XXXX><SP>=<YYYY> CR

XXXX ... doplní, YYYY... horní adresa paměti

Příklad:

+D =7000 <SP> =700F <SR>

7000 41 42 43 44 45 46 47 48

7008 50 51 52 53 00 00 00 00

+F - vyhledání zadané instrukce v paměti a možnost modifikace (FIND)

Formát:

+F =<XXXX> <SP> = <YYYY> <SP> ; zadání rozsahu adres paměti pro hledání

```
=N <SP>                ; počet byte instr.
=<DATA1> <SP> =...=<DATAN> <SP> ; N=1-3 hledaný obsah
=<MASK1> <SP> =...=<MASK N> <SP>; - " - MASKA
```

Program provede log. součin obsahu paměti s maskou a výsledek porovná se zadanými daty. Je-li rovnost, provede výpis adresy a zdrojového textu instrukce. Takto je možno volit i třídu instrukcí (např. IN,OUT) nebo najít instrukce pracující s adresou (viz příklady).

Příklad: Požaduje se nalezení všech instrukcí

MOV B,D (oper. kód 42) v paměti od 7000 až 7100

```
+F  =7000 =7100
=1          ; jednobytová instrukce
=42         ; hledaný kód instrukce
=FF        ; maska textu všech bitů v paměti
... výpis všech nalezených adres a zdrojů v rozsahu 7000
až 7100 (vypisuje po řádcích a čeká na <SP> pro pokračování
v hledání nebo ", "- skok do SUBSTITUTE (lze provést změnu dat)
a nebo <CR> pro ukončení hledání. Hledání třídy instrukcí IN (DB),
OUT (D3), liší se pouze v jednom bitu.
```

```

+F =7000 <SP> =8000 <SP>           ;:rozsah
=2 <SP>                               ; 2.byťová instrukce
=DB <SP> =00 <SP>                     ; třída instrukcí
=F7 <SP> =00 <SP>                     ; maska pro třídu

7042 DB16    IN    016H <SP>          ; po <SP> pokračuj
7800 D35A    OUT   05AH <CR>         ; ukončeno

```

Vyhledání tříbyťových instrukcí, které pracují s adresou 20H:

```

+F =7000 <SP> -7050 <SP>
=3 <SP>
=C3 <SP> =00 <SP> =20 <SP>
=00 <SP> =FF <SP> =FF <SP>

```

+G - spuštění programu (GO)

Formát: +G =<START ADR> <SP> =<BREAK1> <SP> =<BREAK2> <CR>

Příklad: +G 1003 ; teplý start TOOLu

+H hexadecimální součet a rozdíl. Pomocná funkce při opravách a změnách v programech.

Formát:

+H =<XX> <SP> -<YY> <CR> ; XX, YY hexadecimální čísla
 .SSSS RRRR ; výsledek SSSS=součet, RRRR=rozdíl

Příklad: +H =11 =9
 1A 8

- +J - skok na poslední zavedený program z kazety (JUMP)
- +K Vyvolání systému MIKOS Popis povelů MIKOSu je uveden v příručce Návod k použití.
- +M - přesun úseku paměti (MOVE)

Formát:

+M = <ZDROJ 1> <SP> = <ZDROJ 2> <SP> = <CIL 1> <CR>

ZDROJ 1 - počáteční adresa, ZDROJ 2 - koncová adresa

CIL 1 - adresa místa přemístění

Příklad:

+M =7000_ =700F_ =F000 <CR>

Přepíše obsah paměti od adresy 7000 až do 700F (včetně) na zónu počínaje adresou F000 (tj. do F00F). Přepíše se část video RAM. Změnu je možno vidět na obrazovce.

- +N - nulování registru a podmínky, stejně jako +B (NULL)
Ruší tabulku symbolů pro překladače.

Formát: +N

- +Q - návrat do monitoru (QUIT)

Formát: +Q

- +R - výpis obsahu registrů (REGISTER)

Formát: +R

- +S - záměna obsahu paměti (SUBSTITUTE)

Formát:

+S =XXXX_LYY- ... ; XXXX-ADRESA, YY-OBSAH

lze změnit jak adresu, tak obsah. Změna

se provede zápisem nového HEX. čísla, ponechání <SP> , ukončení <CR> , návrat k adrese <, > po (<←>) znak šipka se zobrazí nižší adresa a její obsah a po znaku apostrof (<'>) lze zadat ASCII znak.

Příklad:

```
+S = 7000 <SP> C3 - <SP> 20 - <10> <SP> 21 - <←>
7001 10- <CR>
```

Pozn.: Hodnotu C3 jsme neměnili (<SP>) hodnotu 20 jsme změnili na 10 a znakem šipka jsme provedli kontrolu zápisu, <CR> ukončeno.

+U - zobrazení obsahu paměti v kódu ASCII (UNDEFINED CODE)

Formát:

```
+U = <ADR 1><SP> = <ADR2> <CR>
... výpis v ASCII jako text ...
```

+X - zobrazení a změna obsahu registru (EXAMINE REG)

Zde se nevypisují názvy registru. Je možné také zadat po apostrofu (<'>) znak ASCII.

Formát:

```
+X YY- <SP>          ZZ... NOVÝ OBSAH
      ZZ              ; YY... OBSAH REG.
+X <CR>              ; CELKOVÝ VÝPIS VŠECH
                      REG. BEZ MOŽNOSTI
                      ZMĚNY
```

Pozor! Při změně reg. příkazem X v monitoru se změny nepromítnou do reg. v TOOLu. Změny se musí provádět přímo příkazem +X. (Totéž platí i naopak). Registry jsou uchovány při skoku do monitoru a při návratu jsou obnoveny.



Textový editor PEDIT (programový editor) je určen k pořizování zdrojových textových souborů pomocí klávesnice a se zobrazováním na televizním displeji. Délka řádku není v PEDITu omezena, přičemž na displeji se zobrazuje 40 znaků. Na obrazovce displeje lze zobrazit 24 řádek. Poslední spodní řádek slouží jako příkazový řádek (pro zadávání povelů PEDITu).

Basic může pracovat s délkou řádku 132 znaků, délka řádku pro assembler je 80 znaků.

Poznámka: Pověly se ukončují tlačítkem "↓" a na obrazovce se zobrazuje znak měny "⌘" nebo "§".

Textový editor pracuje ve dvou režimech:

1. Příímý - "VIDEO MODE", kdy kód povelu je prvním znakem režim na příkazovém řádku. V tomto případě se povel okamžitě provede a zobrazí se modifikovaný text, bez potvrzováním příkazu dvěma znaky "↓", který se na obrazovce zobrazuje jako znak měny "§" nebo "⌘". Takto je možno např. měnit polohu kurzoru na obrazovce atd.

2. Příkazový - "COMMAND MODE" v tomto případě se povel režim okamžitě neprovádí a zobrazuje se v příkazovém řádku, jednotlivé povel y se vzájemně oddělují znakem "↓" a povel y se provedou až po zadání dvou znaků "↓". Jednopísmenné povel y se nemusí odělovat znakem "↓", např. 15L3CV↓↓ . Do této doby je možno příkazy opravovat nebo rušit znakem "←" (šipka vlevo.)

Příkazy, které jsou uvedeny v "přímém režimu - videomodu" pracují i v příkazovém režimu, je-li na prvním místě uveden počet opakování nebo jiný povel, např. "PŘÍMÝ POVEL":
 * ↓ provede okamžitě posun ukazovátka na nový řádek (dolů).

"Příkazový režim":

- * 3L ↓↓ provede posun ukazovátka o tři řádky (směrem dolů) až po zadání dvou znaků "↓".
- * 3LV ↓↓ provede se posun a zobrazí se text
- * TL ↓↓ provede se až po potvrzení

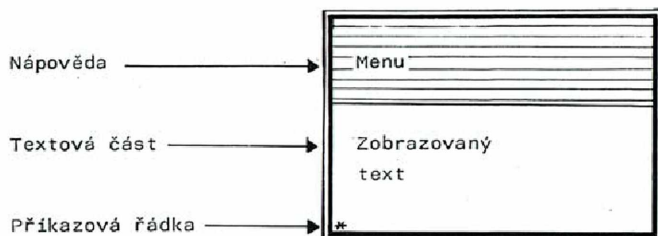
Textový editor "PEDIT" pracuje s dynamickým "oknem", vertikálně i horizontálně. Jak byla uvedeno, řádek může

obsahovat např. 132 znaků, přičemž se zobrazuje "okno" 40-ti znaků na řádku, který se automaticky posunuje při posunu ukazovátka. Ve vertikálním směru se "okno" automaticky posouvá a zobrazuje část editovaného textu. Na konci řádku se zobrazuje znak "↵". Je-li řádek delší a není-li konec řádku zobrazen, potom se znak "↵" nezobrazí. Takto je obsluha informována, že řádek pokračuje.

Poznámky:

V popisu příkazů je použit znak "%" ve významu volitelného čísla opakování (v intervalu - 65535 až + 65535).
 Např. pro příkaz L je uvedeno %L (viz výše uvedený příklad). Uvede-li se v příkazu (tam, kde je uveden znak "%") znak "nerovná se", tj. "#", tak to znamená "všechno" - tj. jako by bylo žádáno největší kladné číslo 65535.
 Např.: *#K <↓>X<↓> zruš všechny řádky od ukazovátka až do konce.

Rozdělení obrazovky



Definice videomodu

Pro urychlení práce s editorem jsou některé povely prováděny v režimu videomodu s okamžitým zobrazením nového stavu textového bufferu. Videomod PEDITu platí pro uvedené povely jen tehdy, je-li znak na prvním místě v příkazovém řádku. Potom není třeba potvrzovat povel dvakrát znakem ↑↑

Povel se provede automaticky včetně zobrazení modifikovaného textu.

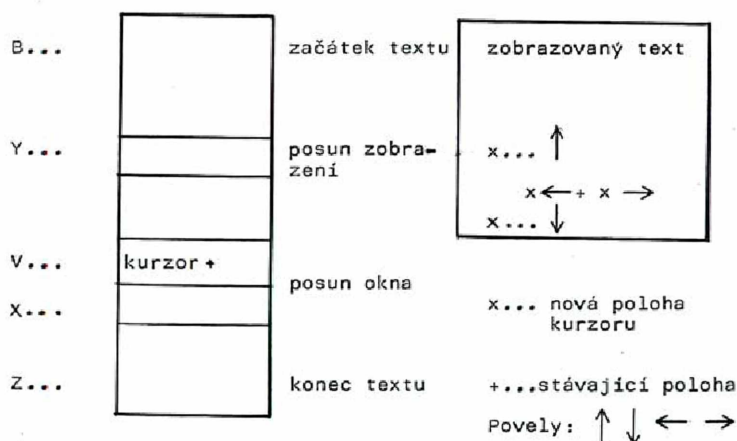
Videomod platí pro příkazy:

B, Z, I, X, Y, V, +(0-9), CTRL D, CTRL I, CTRL K, CTRL S.



Textový buffer

Okno (WINDOW)



4.1. Popis_příkazů_editoru

4.1.1. Příkazy pro řízení videomodu

Vkládání textu:

- I Zobrazí šest řádků předchozího textu a vymaže obrazovku od místa, kde bude vkládán nový text.

* I NOVÝ TEXT <↓> <↓>

Nový text se ukončí dvakrát znakem <↓>, který není součástí vkládaného textu. Po ukončení vkládání se zobrazí nový text doplněný původním textem v okolí kurzoru.

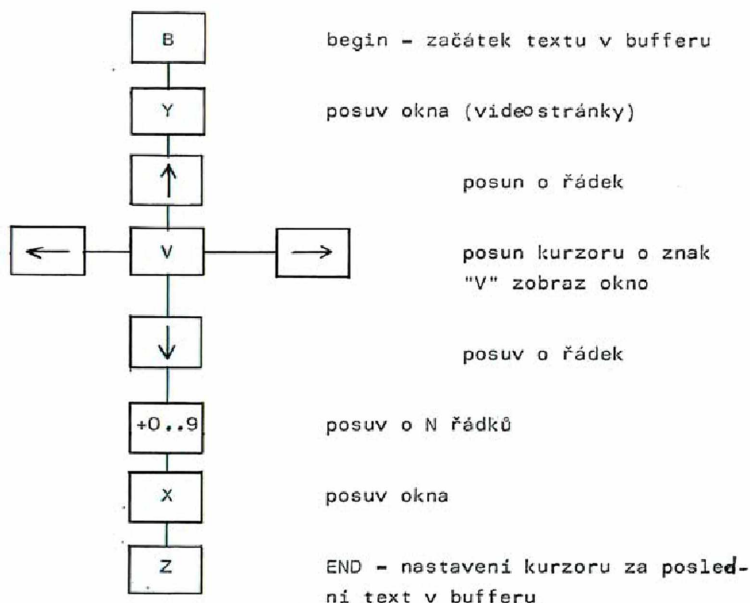
Během vkládání nového textu lze znakem "←" rušit chybně zapsané znaky.

- CTRL I Vložení znaku uvedeného za příkazem CTRL I na místo, kam ukazuje ukazatel (kurzor).



- %A Přečte z vnějšího zařízení - kazetového magnetofonu: % - řádků a uloží je za text v textové vyrovnávací paměti (BUFFER). V případě prázdného bufferu se ukládá od začátku bufferu. Příkaz %A není příkazem videomodu, ale je zde uveden, protože se často používá v kombinaci s příkazy pro vkládání textu.

Přemístění kurzoru

Je možné těmito znaky (znaky jsou vybrány tak, aby bylo možné použít libovolnou klávesnici):

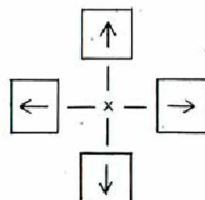


B - nastaví ukazatel na začátku textu a zobrazí text -
BEGIN

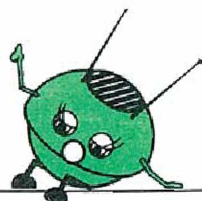
- Z - nastaví ukazatel na konec textu a zobrazí text (END)
- F - vyhledá a automaticky zobrazí hledaný text:
 FPAVEL~~XXX~~. vyhledá a označí text "PAVEL". (FIND)
 Kurzor se umístí za nalezený text. Příkaz není příkazem videomodu (musí se ukončovat znaky<↓>), používá se pro vyhledání textu pro následné vkládání, resp. opravu editovaného textu.
- V - zobrazí text v okolí kurzoru (VIEW). Tento povel se automaticky provádí po ukončení jiných povelů (např. S, F, CTRL D...). (video stránka)
- X - Zobrazí další text (video stránka)
- Y - zobrazí předchozí text (videostránka)
- + - automaticky přesune kurzor o 0-9 řádků směrem dolů (zde X= 0-9 !).
 + ... nastaví kurzor na začátek řádku !

Posuny ukazovátka (kurzoru)

- ↓ - posun dolů o jeden řádek
 ↑ - posun nahoru o jeden řádek
 ← - posun doleva o jeden znak
 → - posun doprava o jeden znak

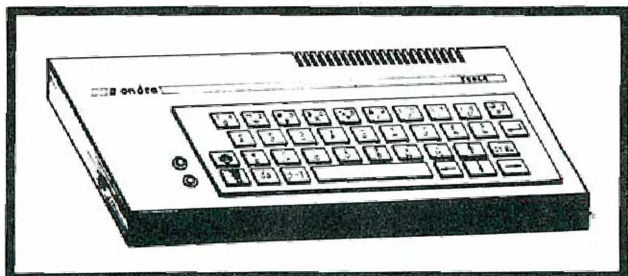


TESLA



UŽIVATELSKÁ PŘÍRUČKA MIKROPOČÍTAČE

ondra



MIKROPOČÍTAČ ONDRA SPO 186

2

TEDIT



Textový editor TEDIT rozšiřuje možnosti mikropočítače ONDRA pro přípravu textových souborů. Umožňuje všechny základní funkce editorů pro přípravu programů v assembleru a Basicu. Basic má ovšem vlastní možnosti přípravy a editace programů, ale je možné použít pro přípravu programů editoru. Hlavní předností nového editoru TEDIT je možnost přípravy textových souborů a jejich formátování. TEDIT umožňuje pracovat s celým souborem znaků KOI-8/ čs 2, tj. velká a malá písmena a znaky české abecedy.

Technické řešení mikropočítače ONDRA a zejména membránová klávesnice omezují možnosti použití mikropočítače ONDRA pro automatizaci kancelářských prací, mikropočítač však pro toto použití nebyl určen. Textový editor TEDIT Vás seznámí se základy práce s textovými soubory, možnostmi vytváření a úprav textů a jejich formátováním. Pod pojmem formátování rozumíme stanovení počtu znaků na řádek, počet řádek na stránce, stanovení levého okraje, číslování stránek, nastavení tabulátorů atd. Při formátování TEDIT zarovnává řádky tak, aby poslední znak slova na řádku byl na

na pravém okraji řádku. Provádí to doplňováním mezer mezi slovy nebo rozdělením posledního slova.

Této vlastnosti TEDITu můžeme využít při psaní textů například návodů, příruček, dopisů atd. Nemusíme sami dbát na úpravu stránek (tj. věty nemusí být až do konce řádků), zpětně můžeme do napsaného textu vsunout další text nebo část textu zrušit nebo ho změnit. Editor umožňuje slučování různých textových souborů do nového souboru. Tím si mnohdy ušetříme čas při přípravě textů. Když píšete na psacím stroji a zjistíte, že jste část textu zapomněli napsat, nezbyvá Vám nic jiného, než danou stránku přepsat nebo vložit další list se vsuvkou. Častým případem je, že máte téměř stejný text a potřebujete změnit pouze malou část (např. napíšete pozvánku na schůzi a změní se datum nebo čas konání) nebo text je stejný a mění se pouze záhlaví (např. rozesíláte stejný dopis různým adresátům). Toto všechno velmi jednoduše zvládnete pomocí textového editoru TEDIT.



Textový editor TEDIT 1 je program, který umožňuje pořizovat a modifikovat textové soubory tvořené alfanumerickými znaky. TEDIT 1 je určen pro mikropočítačový systém ONDRA s programovým vybavením V5.

Textový soubor se vytváří zápisem jednotlivých znaků z klávesnice do textové vyrovnávací paměti při současném zobrazení pořizovaného souboru na displeji. Soubor lze modifikovat jak v průběhu jeho pořizování (edice), tak i po ukončení edice. Soubor lze uložit na vnější paměťové médium (kazeta) nebo vypisovat na tiskárně.

Modifikační schopnosti textového editoru TEDIT 1 zahrnují možnost úprav nebo doplňování textu a jeho formátování. Formátování spočívá ve vytvoření nových textových souborů, jejichž obsah je shodný s původním souborem, ale liší se šířkou a délkou stranky, mají zarovnaný pravý okraj. Modifikace textu se uskutečňuje prostřednictvím povelů zadávaných z klávesnice. Jednotlivými povely nebo jejich kombinacemi lze např. ovládat:

- vkládání a mazání jednotlivých znaků nebo textových řetězců (částí textu např. slovo, věta aj.)
- náhradu původních řetězců z původního místa na nové místo v rámci celého souboru
- formátování textu, při kterém je možno volit šířku a délku stránky, šířku mezery na levém okraji textu, číslování stránek, vkládání nového textu při výpisu textového souboru na tiskárnu aj.

Činnost TEDITu je rozdělena do dvou módů. Mód označený MAIN (hlavní) je určen hlavně pro vkládání a úpravu textu, zatímco mód označený HELP (pomocný) slouží především k jeho formátování. Povelů módu MAIN se ve většině případů shodují s editorem PEDIT, který je již široce využíván při pořizování zdrojových programů pro mikropočítačové systémy SAPI 1 a ONDRA.

Uvedené vlastnosti textového editoru TEDIT 1 umožňují jeho široké využití při zpracování různých textů (zprávy, dopisy, technické textové dokumentace aj.). Odpadá zdlouhavé přepisování celých stran textu při výskytu chyb nebo v případě, kdy je nutno část textu změnit. Text zpracovaný TEDITem se blíží tištěnému textu, má zarovnaný pravý okraj a jeho přehlednost lze zlepšit tučným tiskem.

2.1. Klávesnice

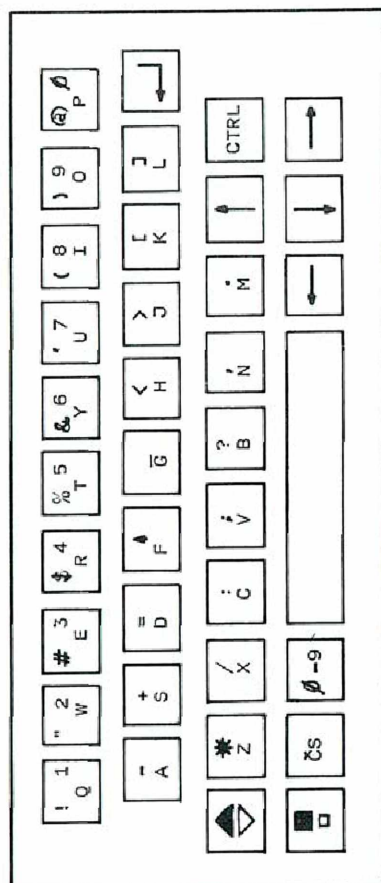
Mikropočítač ONDRA je vybaven membránovou klávesnicí s tlačítky, která umožňuje vstup všech alfanumerických znaků a znaků české abecedy. Protože tato klávesnice má pouze omezený počet tlačítek, je nutno provádět nastavování významu tlačítek podle okamžité potřeby. Horní řada tlačítek má tři významy a ostatní dva významy (kromě speciálních tlačítek), kromě toho písmena mohou být velká a malá a znaky české abecedy. Tato klávesnice není určena pro profesionální pořizování dat a textů.

Pro orientaci nastaveného přepnutí klávesnice je mikropočítač doplněn dvěma signálními LED diodami. Bližší význam jednotlivých signálů a způsob práce s klávesnicí je podrobně popsán v příručce Návod k použití mikropočítače ONDRA.

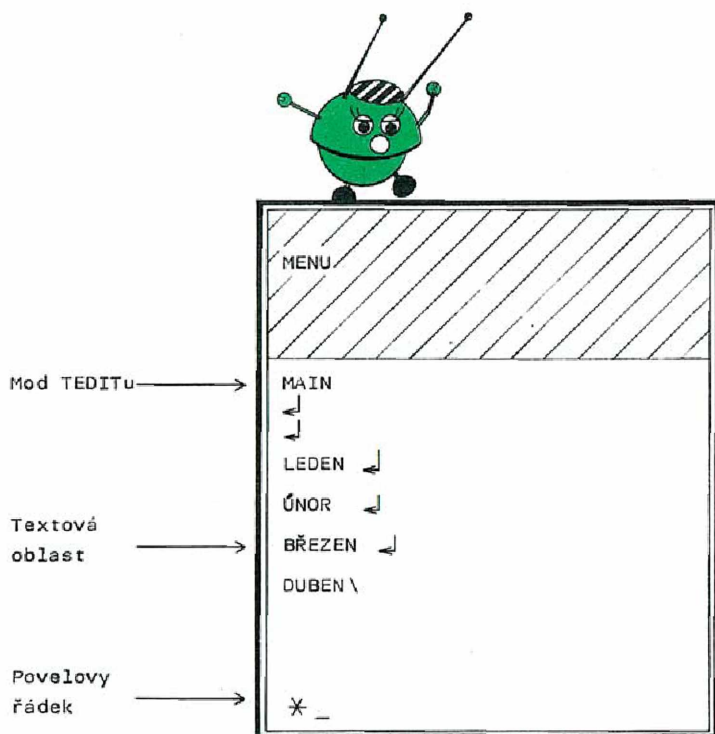
Na obrázku 1 je rozmístění znaků na klávesnici.

2.2. Zobrazení na displeji

Zobrazení na displeji při práci s textovým editorem TEDIT je znázorněno na obr. 2.



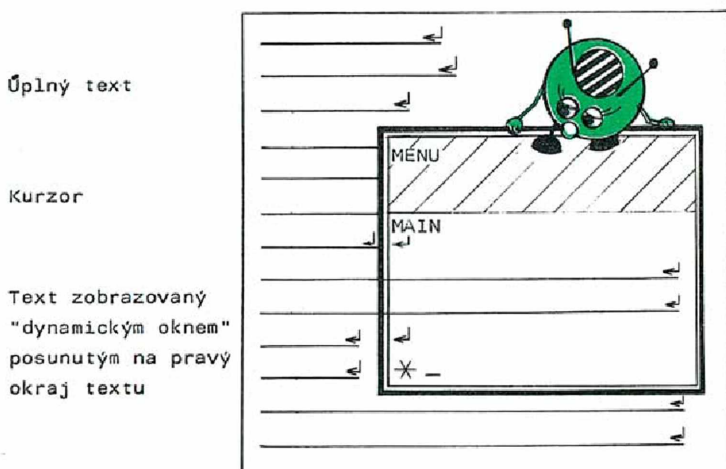
Obr. 1 Klávesnice ONDRA



Obr. 2 Displej textového editoru
TEDIT 1 v modu MAIN

V horních řádcích obrazovky je po dobu práce s TEDITem trvale zobrazováno "MENU" typ modu, tj. MAIN nebo HELP. Následuje jeden volný řádek. Další řádky na obrazovce zaujímá pořizovaný text. Konec každého řádku textu je vyznačen znakem "␣", který se vypisuje po stisknutí klávesy "CR". Konec textu je vyznačen znakem "\", který se automaticky posouvá při pokračujícím vkládání textu. Poslední řádek je povelový, začíná v modu MAIN znakem "*", za nímž následuje znak " _ " (kurzor, podtržení), který označuje pozici, na níž se vypisuje první znak povelu zadávaného z klávesnice. V modu HELP začíná povelový řádek znakem ">". Poloha kurzoru v textu je zobrazována inverzním znakem.

Délka textového řádku TEDITu není omezena. Na displeji je však možno zobrazit maximálně 40 znaků na řádku. Překročili-li se uvedený počet znaků, zobrazení textu se posouvá vpravo tak, že je zobrazeno 40 znaků od pravého okraje textu. Obdobně se text posouvá směrem nahoru, překročili-li se při vkládání textu maximálně zobrazovaných 14 řádek (při zachování "MENU"). Tento způsob zobrazení označovaný jako "dynamické okno" se využívá i při prohlížení editovaného textu pomocí povelů TEDITu, které umožňují libovolný posun "okna" 40 znaků x 9 řádků v celém rozsahu vloženého textového souboru.



Obr. 3 Zobrazení textového souboru pomocí "dynamického okna"

Z obr. 3 je zřejmé, že znak konec řádku "↵" je zobrazován pouze v tom případě, že "okno" je nastaveno na pravý okraj textu.

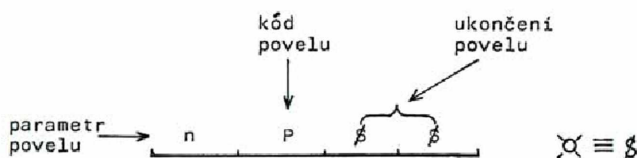
V textové oblasti obrazovky se nachází kurzor (ukazatel), který označuje místo, k němuž se vztahují veškeré manipulace s textem (úpravy, doplňování, výpis text aj.). Umístění kurzoru je vyznačeno inverzním znakem textu.



Povely pro řízení činnosti TEDITu se zadávají prostřednictvím klávesnice a jsou zobrazovány na povelovém řádku v modu MAIN za znakem "X" a v modu HELP za znakem ">".

3.1. Tvar_povelu

Obecný tvar povelu TEDITu:



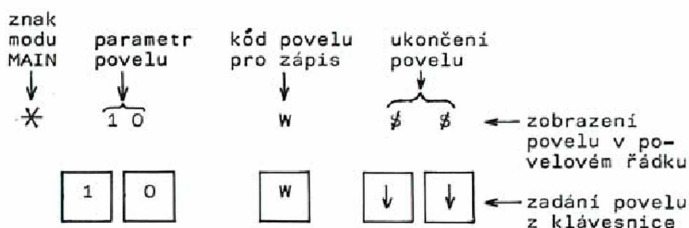
Kód povelu "P" obsahuje většinou jedno písmeno, specifikující povel, který má být proveden. Parametr "n" se může pohybovat v rozmezí od -65535 do +65535. Na místě parametru lze použít znak "#", který je ekvivalentní maximálně přípustnému kladnému číslu (+65535). Zápis povelu je ukončen dvojnásobným stisknutím klávesy "↓". Na displeji zobrazují znaky §.

3.2. Opravy_povelů

Pokud ještě není zápis povelu ukončen zapsáním druhého znaku \$, je možno dříve zapsané znaky kódu povelu a parametru mazat klávesami "←" a nahrazovat novými znaky. Provádění povelu začíná až po ukončení povelu (2x "↓").

3.3. Příklady_povelů

Příklad povelu pro vtištění deseti řádků textu od řádku, v němž je umístěn kurzor na kazetu (mod MAIN).



Obr. 4 Zadávání a zobrazování povelu

Pro zápis textu od řádku, v němž je umístěn kurzor až do konce textového souboru lze použít povel ve tvaru:

$\times \# w \$ \$$

Znak # je zde užit ve smyslu "vše", to znamená, že risk bude probíhat až do dosažení konce textového souboru.

3.4. Videomod

Pro urychlení práce s editorem TEDIT 1 jsou některé povely vykonávány v tzv. videomodu sokamžitým zobrazením nového stavu textové paměti. Videomod platí pro dále uvedené povely pouze v tom případě, je-li kód povelu na prvním místě povelového řádku. Potom není třeba v provedení povelu používat kláves 2x "↓". Povel se provede automaticky včetně zobrazení modifikovaného textu.

Videomod platí pro povely B, Z, I, X, Y, V, +/0-9/, CTRL D, CTRL I, CTRL K, CTRL S (viz kap. 4, 5 a dodatek č. 3) nebo povely uvedené v menu.

3.5. Povelové řetězce a jejich opakování

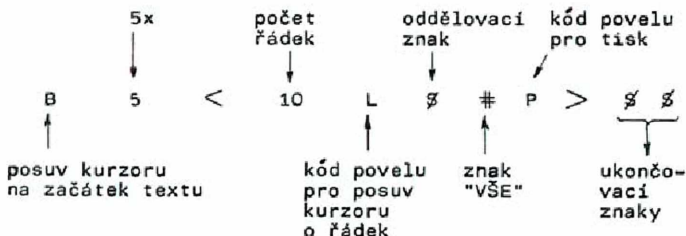
Povely je možno řadit do řetězců, tj. za sebou a zadávat tak několik povelů současně. Jednotlivé povely v řetězci musí být odděleny znakem "↓" s výjimkou povelů k jejichž definici postačuje pouze jedno písmeno (např. B, Z).

Následující příklad ukazuje povelový řetězec, který posune kurzor o 5 znaků vpravo a vymaže 2 znaky vpravo od nové polohy kurzoru:



Má-li být povelový řetězec vykonán několikrát, musí být uzavřen v ostrých závorkách, před nimiž je parametr opakování.

Následující příklad ukazuje povel pro tisk pěti kopií textového souboru s vynecháním prvních deseti řádků:





Mod MAIN je určen především k pořizování textových souborů, které se dále upravují v modu HELP. Mod MAIN je vybaven velkým množstvím povelů, které buď samostatně nebo ve spojení do řetězců poskytují široké možnosti k vytváření a modifikaci textových souborů.

V dalším textu jsou povelý, které pracují ve videomodu, označovány pouze kódem povelu (např. B, Z). Povelý vyžadující ukončení dvojnásobným stisknutím klávesy "↓" mají za kódem povelu uveden znak \oint (např. nP \oint).

Povelý v modu MAIN je možno ovládat:

- nastavení kurzoru a zobrazení textu
- vkládání a modifikaci textu
- vstup a výstup textu na periferní zařízení

4.1. Povelý pro nastavení kurzoru a zobrazení textu



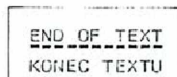
BEGINNING OF TEXT
ZAČÁTEK TEXTU

Povel B nastaví kurzor na první znak textového souboru a zobrazí začátek textu.

Povel se používá

- ke vkládání nového textu před začátek původního textu
- k přesunu kurzoru na začátek textu při tisku celého souboru (tisk začíná vždy od pozice, na kterou je nastaven kurzor) při vyhledávání žádaného textového řetězce v celém souboru nebo při substituci původního řetězce novým.

Povel B pracuje ve videomodu a provádí se tedy okamžitě po stisknutí klávesy B, je-li na prvním místě povolového řádku.



Povel Z nastaví kurzor za poslední znak souboru a zobrazí konec textu. Povel se používá při vkládání dalšího textu za původní text. Provádí se ve videomodu.



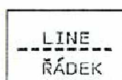
Povel "↓" posune kurzor na začátek následujícího řádku a zobrazí se nová poloha kurzoru. Povel pracuje ve videomodu a nezobrazuje se tedy v povelovém řádku.



Povel "↑" posune kurzor na začátek předchozího řádku a zobrazí se nová poloha kurzoru. Povel pracuje ve videomodu.



§



Povel nL přesune kurzor na začátek n-tého řádku od původního postavení kurzoru. Má-li kladnou hodnotu, posune se kurzor dolů. Záporná hodnota n způsobí posuv nahoru. Pro $n = 0$ se kurzor posune na začátek téhož řádku, ve kterém byl původně umístěn. Povel nepracuje ve videomodu, zobra-

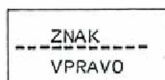
zuje se v příkazovém řádku a má-li být nová poloha zobrazena, musí se použít ještě povel V, který zobrazuje text v okolí kurzoru nebo povel "T".

Posuv kurzoru o 8 řádků nahoru a zobrazení nového stavu vykoná povelový řetězec:

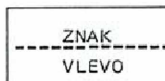
✕ - 8 L V § §

Posuv kurzoru o 10 řádků dolů a zobrazení nového stavu vykoná povelový řetězec

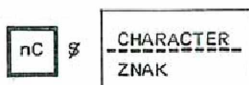
✕ 10 L V § §



Povel "→" posune kurzor o jeden znak vpravo a zobrazí se nová poloha kurzoru. Povel pracuje pouze ve videomodu.



Povel "←" posune kurzor o jeden znak vlevo a zobrazí se nová poloha kurzoru. Povel pracuje ve videomodu.



Povel nC přesune kurzor o n znaků od původního postavení kurzoru. Má-li n kladnou hodnotu, posune se kurzor vpravo. Záporná hodnota n způsobí posuv kurzoru vlevo. K zobrazení nové polohy kurzoru je nutno použít povel "V" nebo povel "T". Povel "C" není součástí videomodu.

Posuv kurzoru o 5 znaků vlevo a zobrazení nového stavu

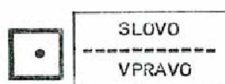
✕ - 5 C V § §

Posuv kurzoru o 20 znaků vpravo a zobrazení nového stavu vykoná povel:

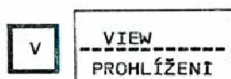
✕ 20 C § §



Povel " 9 " posune kurzor o jedno slovo vlevo a zobrazí se nová poloha kurzoru. Povel pracuje ve videomodu.



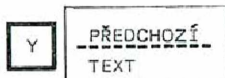
Povel " • " posune kurzor o jedno slovo vpravo a zobrazí se nová poloha kurzoru. Povel pracuje ve videomodu.



Povel V zobrazí text v okolí kurzoru, pracuje ve videomodu a používá se k verifikaci provedených změn v textu.



Povel X zobrazí následující text souboru a přemístí na nový řádek kurzor. Povel pracuje ve videomodu a slouží k prohlížení textu.



Povel Y zobrazí předcházející text souboru a přemístí na nový text kurzor. Povel pracuje ve videomodu a slouží k prohlížení textu.



Povel F "text" vyhledává řetězec znaků "text". Vyhledávání začíná od pozice, na které se nachází kurzor a pokračuje až do nalezení řetězce "text" nebo do konce souboru, pokud se v prohledávané části souboru "text" nevyskytuje.

Je-li řetězec znaků "text" nalezen a potvrdí se provedení povelu klávesou "Y", přesune se kurzor na první znak za hledaným řetězcem.

Pokud není "text" nalezen, zůstává kurzor na původní pozici a vypíše se hlášení:

N O F I N D "text"

Povel se používá pro vyhledávání textových řetězců, za nimiž má být vložen nebo opraven text.

Vyhledávání textového řetězce EDITOR začínající od počátku souboru a přemístění kurzoru za nalezený řetězec se dosáhne následujícím zápisem z klávesnice

* B F E D I T O R \$ \$

4.2. Povely pro vkládání a modifikaci textu

I text	INSERT TEXT ----- VKLÁDÁNÍ TEXTU
--------	--

Povel I text umožňuje vkládat text z klávesnice od pozice, na které se nachází kurzor. Vkládání textu se ukončí dvěma znaky "↓". Během vkládání textu je možno rušit chybně vložené znaky znakem "←". Po ukončení vkládání znaky "↓" se zobrazí text v okolí kurzoru.

CTRL I "znak"	VLOŽENÍ ----- 1. ZNAKU
---------------	------------------------------

Povel CTRL I "znak" provede vložení znaku na místo označené kurzorem. Povel pracuje ve videomodu

a slouží pro jednoduché opravy textu

CTRL I A vložení do textu znaku A

Poznámka: Potřebujeme-li vložit do textu znak tabulátoru
ve videomodu, provede se příkazem:

✕ CTRL I CTRL I
povel "znak" = 09H (TAB)

Vložení textu "ÚVOD" na začátek již existujícího
textu se dosáhne následujícím zápisem z klávesnice:

✕ B I U V O D \$ \$

V povelu "I" má tlačítko "→" význam tabulátoru.

CTRL S "znak"

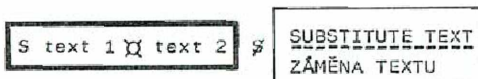
ZÁMĚNA
1. ZNAKU

Povel CTRL S "znak" zamění znak, na němž je umístěn
kurzor novým znakem zadaným z klávesnice.

Je-li např. kurzor umístěn v textu na znaku "A" a
požadujeme-li přepsání tohoto znaku na "B", je k této
změně potřebný následující zápis z klávesnice:

✕ CTRL S B

Povel pracuje ve videomodu a je určen pro opravy přepisu znaků.



Povel S text 1, text 2 vyhledá v původním textu řetězec "text 1" a po potvrzení znakem "Y" nahradí řetězec "text 1" novým řetězcem "text 2". Hledání v textu začíná od pozice, na které je umístěn kurzor. Povel se ukončí zobrazením textu v okolí kurzoru.

Záměna textového řetězce "PAVEL" novým textovým řetězcem "PETR" se uskuteční po následujícím zápisu do klávesnice:

S P A V E L \$ P E T R Y \$

Příklad: Záměna v textu programu instrukce MOV za instrukci LD

* 99 < S MOV \$ LD \$ > \$ \$

Pomocí dotazů a potvrzení znakem Y lze provést záměnu.



Povel CTRL D vymaže znak, na němž se nachází kurzor a zobrazí provedenou opravu.



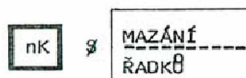
Povel nD maže počet znaků specifikovaný hodnotou n. Má-li n kladnou hodnotu, vymaže se n znaků za kurzorem. Záporná hodnota n se maže n znaků před kurzorem.

Vymazání 5 znaků před kurzorem a automatického zobrazení opraveného textu se dosáhne povel:

* - 50 § §



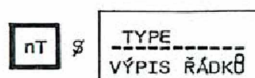
Povel CTRL K vymaže zbytek řádku (včetně CR, LF), na kterém se nachází kurzor a zobrazí provedenou opravu.



Povel nK maže počet řádků specifikovaný hodnotou n. Má-li n kladnou hodnotu, vymaže se n řádků směrem dolů od kurzoru včetně části řádku, která se nachází za kurzorem. Záporná hodnota n znamená, že se vymaže n řádků směrem nahoru včetně části řádku před kurzorem. Povel ØK maže od začátku řádku až do kurzoru.

Povel $\times 2 K \text{ } \text{⌘} \text{ } \text{⌘}$ způsobí vymazání dvou řádků za kurzorem a zobrazí se provedená oprava.

4.3. Povely pro vstup a výstup textu, součinnost s periferním zařízením



Povel nT vypíše na displeji počet řádků specifikovaných hodnotou "n". Má-li "n" kladnou hodnotu, začíná výpis od kurzoru a pokračuje na displeji směrem

dolů. Při záporné hodnotě n začíná výpis na n -tém řádku před kurzorem a pokračuje až po dosažení kurzoru.

Má-li být vypsán celý řádek bez ohledu na to, v které jeho části se nachází kurzor, je třeba zadat následující povelový řetězec:

* ØT § §

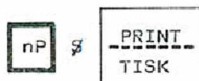
První část řetězce ØT způsobí výpis části řádku od jeho začátku po kurzor a druhá část: T vypíše zbytek řádku, tj. od kurzoru do konce řádku.

Pro výpis celého souboru na displeji lze použít povelového řetězce *B

* # T § §



Povel $nØ$ zapíše n řádků textového souboru počínaje řádkem, na němž je umístěn kurzor z paměti na kazetu. Po zadání povelu se řízení předává operačnímu systému MIKOS, jehož prostřednictvím se uloží text pod definovaným názvem na kazetu. Text zůstává v paměti, na rozdíl od povelu "W".



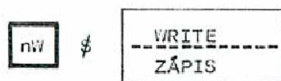
Povel nP umožňuje tisk textu na tiskárnu. Parametr n udává počet řádků, který bude vytištěn. Tisk začíná od řádku, na kterém se nachází kurzor.

Poznámka: Pro použití povelu "P" musí být v mikropočítači ONDRA připojena tiskárna, která musí být připravena k tisku. Není-li tiskárna připojena, pak při použití povelu P čeká program na připravenost tiskárny.

U povelu "K" v módu HELP je v poznámce uvedena úprava ovladače tiskárny na prázdný ovladač.

Povel pro nastavení kurzoru na začátek textu a tisk celého souboru

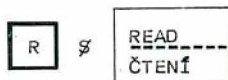
* B # P \$ \$



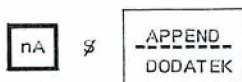
Povel nW zapiše n řádků textového souboru z paměti na kazetu prostřednictvím operačního systému MIKOS. Text

v paměti se zruší!

Poznámka: Povel W zapiše n řádků od počátku textové paměti (nikoliv od pozice kurzoru). Po ukončení práce s editorem je nutno soubor na kazetě uzavřít. Příkazem editoru "E" se uzavře výstupní soubor nebo můžeme soubor uzavřít příkazem MIKOSu K-C.



Povelem R se nastavují příznaky pro čtení nového souboru z kazety. Používá se ve spojení s povelům A. Používá se pro spojování několika vstupních souborů.



Povel nA přečte n řádků z kazety a uloží je za dosavadní text v paměti. V případě, že je textová paměť prázdná, ukládá se od začátku paměti.

* # A § § přečte a uloží do paměti vstupní soubor čili text, který budeme zpracovávat

4.4. Ostatní_povely

nG § GET-----
OZNAČENÍ

Povel nG označí a pro kontrolu vypíše n řádků na displeji, které se budou kopírovat na jiné místo v paměti.

U § UNSAVE-----
PŘEMÍSTĚNÍ

Povel U přemístí n řádků označených n G na místo, kde se nachází kurzor. Mezi povelům "G" a "U" se nesmí používat žádný povel typu "K", "D", "I", který mění počet znaků v paměti.

E	§	<u>END OF FILE</u> KONEC SOUBORU
---	---	-------------------------------------

Povel E uloží text z operační paměti na kazetu a uzavře textový soubor, řízení se předá systému MIKOS, který uloží pod definovaným názvem soubor na kazetu.

nJ	§	<u>JUMP</u> SKOK
----	---	---------------------

Povelem nJ se zadává okamžitá poloha kurzoru. Povelem J se tato poloha později vyhledává. Od definování místa nJ se nesmí použít povelů typu "K", "I" a jiné.

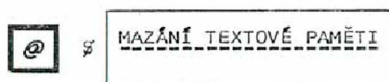
Q	§	<u>QUIT</u> NÁVRAT
---	---	-----------------------

Povelem Q se přechází po potvrzení znakem Y do monitoru.

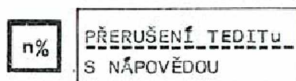


Povellem H se přechází do modu HELP.

- H § přejde do modu HELP a zruší Menu.



Povellem @ se maže obsah textové paměti a provede se studený start editoru.



Povel "%" se v TEDITu používá ve dvou případech.

V prvním případě "Ø%" slouží k uložení nápovědného textu do paměti editoru.

Příklad: * Ø % Nastav papír na sloupec 15 § §

Ve druhém případě "%" (n=1) se příkaz používá v příkazovém řetězci. Při provádění příkazového řádku znak % způsobí výpis zadaného textu na obrazovku a provádění pří-

kazů se zastaví. Na dalším řádku se vypíše text "HOTOVO?" a editor čeká na stisk libovolné klávesy. Po stisknutí pokračuje v provádění příkazového řádku.

Příklad: * - 15 P % 20 P \$ %

Poznámka: Příkaz "%" pracuje stejně v obou módech



ZÁPIS TEXTU

DO VYROVNÁVACÍ PAMĚTI

Povel " [", resp. "{ " lze zapsat do vyrovnávací paměti text až do 80ti znaků. Povel je ukončen dvakrát znakem "↓",

Zapsaný text lze vyvolat příkazem CTRL B.



VYVOLÁNÍ TEXTU

Z VYROVNÁVACÍ PAMĚTI

Povel CTRL B umožňuje použít text zapsaný ve vyrovnávací paměti dvěma způsoby:

- a) generování často se opakujícího textu při použití povelu "I" (vkládání textu)

- b) text ve vyrovnávací paměti mimo povelu "I" se bere jako příkazový řádek. Po stisknutí (2x ↓) se tento řádek provede,

Př. * [I Mikropočítač ONDRA \$ \$

Stiskem klávesy CTRL B se vloží text (mikropočítač ONDRA) do místa, na které ukazuje kurzor. Jsme v povelu "I" a tudíž můžeme vkládat další text nebo povel ukončit (2x ↓).

Př. * [12 T 12 L \$ \$

Po stisknutí CTRL B se na příkazovém řádku vypíše povel 12 T 12 L, který potvrdíme dvakrát stiskem klávesy "↓".



Povely editoru TEDIT 1 v modu HELP jsou určeny především pro formátování textového souboru pořízeného v modu MAIN a uloženého v textové paměti.

Povely editoru umožňují:

- nastavení kurzoru a zobrazení textu
- formátování textu, tj. nastavení šířky stránky včetně zarovnávání pravého okraje textu a automatického rozdělování slov, nastavení délky stránky, číslování stránek, nastavení délky tabulátoru aj.
- nastavení tučného tisku (BOLDFACE), změnu barvy tisku, vkládání nového textu z klávesnice nebo z kazetového magnetofonu do textu vypisovaného na tiskárně nebo na displeji a jiné služby.

5.1. Povely pro nastavení kurzoru a zobrazení textu

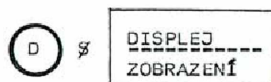
V modu HELP lze používat následující povely pro nastavení kurzoru a zobrazení textu:

- B - nastavení kurzoru na začátek textu

- Z - nastavení kurzoru na konec textu
- V - zobrazení textu v okolí kurzoru
- X - zobrazení následujícího textu a přesun kurzoru na nový text
- Y - zobrazení předcházejícího textu a přesun kurzoru na nový text
- . - posuv kurzoru o slovo vpravo
- , - posuv kurzoru o slovo vlevo

Povel se provádí ve videomodu a jejich funkce je stejná jako v modu MAIN.

5.2. Povely pro zobrazení a nastavení formátovacích parametrů a pro formátování textu



Povel D nejprve zobrazí mapu textové paměti ve tvaru

← obsazená paměť (# = 1 K B)
 ← volná paměť (. = 1 K B)

Dále se zobrazí formátovací parametry:

W = šířka stránky

L = délka stránky

S = nastavení levého okraje

G = nastavení tabulátoru

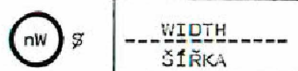
N = počáteční číslo stránky

U = nastavení typu písma a vstupu znaků

Dále se vypíše záhlaví stránky a zobrazí se řídící řádek, který udává nastavení velikosti levého okraje, nastavení tabulátoru znakem "!" a šířku stránky znaky "-".

Zadání povelu z klávesnice

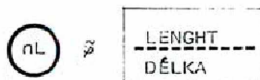
> 0 \$ \$



Povелеm nW se nastavuje požadovaná šířka stránky. Maximální hodnota W = 255. V případě, že W = 0 je vyřazeno zarovnávání pravého okraje textu (zdvojení mezer, rozdělování slov) a průběžné číslo stránky se vypisuje do pozice 40.

Povel pro nastavení šířky textu stránky na 40 znaků

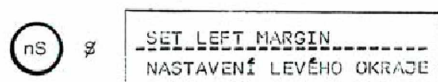
> 40 W § §



Povelem nL se nastavuje požadovaná délka stránky, tj. počet řádků na stránce.

Povel pro formátování textu na délku stránky 60 řádků

> 60 L § §



Povelem nS se nastavuje počet mezer na levém okraji textu.

Povel pro formátování textu tak, aby na levém okraji bylo 5 mezer

> 5 S § §

Zrušení mezer se na levém okraji provádí příkazem Ø S § §

nG §

GET TABULATOR

NASTAVENÍ TABULÁTORU

Povelcm nG se nastavuje délka tabulátoru, přičemž
n = počet znaků, který může být 4 nebo 8.

Povel pro formátování textu na délku tabulátoru 8 znaků

> 8 G § §

nN §

NUMBER

ČÍSLO STRÁNKY

Povelcm nN se nastavuje číslování stránek, n udává po-
čáteční číslo stránky a může se pohybovat v rozmezí 1 - 99.
Zadá-li se n = 0, nejsou stránky číslovány.

Povel pro číslování stránek začínající stránkou 10

> 10 N § §

nU §

UPPER

TYP PISMA A VSTUP ZNAKŮ

Povelcm nU se nastavuje přepínání typu písma a vstupu

znaků. Hodnota $n = 0$ znamená vstup nebo výstup malých i velkých písmen. Pro $n = 1$ vstupují i vystupují pouze velká písmena a pro $n = 2$ se při čtení nového textu z kazety potlačují znaky 0 až 20 H mimo CR, LF a HT.

Povel pro vstup a výstup jen velkých písmen

> 1 U § §

Povely pro nastavení a zobrazení formátovaných parametrů je vhodné spojovat do povelových řetězců, např. povelový řetězec

> 60 W § 00 L § 4 G § 1 N § D § §

nastaví požadované parametry a současně je zobrazí, takže je možná kontrola těchto parametrů.



§

FORMAT ----- FORMATOVÁNÍ

Povel nF přeformátuje text souboru podle zadaných parametrů v délce n řádků, počínaje řádkem, na které je umístěn kurzor. Pravý okraj textu se zarovnává zdvojeňáním mezer a automatickým rozdělováním slov. Původní rozdělení slov se ruší. Odstavce v textu se zachovávají pro parametr

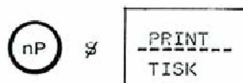
$S = 0$. V případě, že S je větší než 0, vloží se na začátek každého řádku stanovený počet mezer. Po přeformátování textu se zobrazí začátek nového textu. Formátování se ukončí před zadaným počtem řádků, není-li v paměti dostatek místa pro přeformátovaný text.

Povel pro přeformátování 12 řádků:

>12 F § §

Poznámka: Povel "F" zachová původní členění textu na odstavce, když nový řádek textu začíná mezerou nebo tabulátorem. Rovněž zachovává odstavce v případě, že řádku textu předchází několik prázdných řádků.

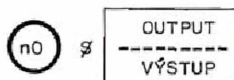
5.3. Povely pro vstup a výstup textu, součinnost s periferním zařízením



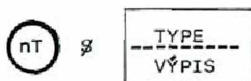
Povel nP umožňuje tisk přeformátovaného textu na tiskárnu. Parametr n udává počet řádků, který bude vy-

tištěn. Tisk začíná od řádku, na které se nachází kurzor.

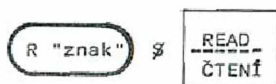
Povel pro nastavení kurzoru na začátek textu a tisk
 10 řádků: > B
 > 10 P § §



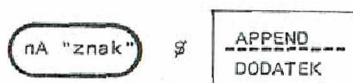
Povel nO umožňuje zápis n řádků přeformátovaného textu na kazetu. Po zadání povelu přechází program do operačního systému MIKOS, jehož prostřednictvím se požadovaný text uloží pod definovaným názvem na kazetu (viz Návod k obsluze mikropočítače ONDRA). Takto uložený text ale není vhodný pro další zpracování, je vhodný pouze pro tisk.



Povel nT umožňuje zápis n řádků přeformátovaného textu na displej. Řádky se zobrazují v celém rozsahu bez omezení "oknem".



Povel R se nastavují příznaky pro čtení nového souboru z kazety a zároveň se vynechává text do zadaného znaku, je-li použit. Používá se ve spojení s povelu A.



Povel nA umožňuje kopírování textu z kazety na tiskárnu nebo displej. Pro n = 1 se kopíruje text na tiskárnu, n = 0 znamená výpis na displej. Jsou-li v textu na kazetě použity vhodné rozdělovací znaky, které netvoří text, např. znak "/", je možno samostatně kopírovat jednotlivé úseky textu oddělené rozdělovacími znaky. Povel nA tedy umožňuje vkládat do tištěného souboru nebo souboru zobrazovaného na displeji text uložený na kazetě. Po zadání povelu se předává řízení operačnímu systému MIKOS, ve kterém se uskutečňuje komunikace s kazetovým magnetofonem. Po ukončení komunikace se řízení vrátí do TEDITu.

Text uložený na kazetě po znak "/" se kopíruje povel-
ly: > R 1 A / Š Š

Pokud text neobsahuje znak "/", vypíše se celý soubor.



KEYBOARD ----- KLÁVESNICE

Povel nK umožňuje vkládat nový text do souboru, který se vypisuje na tiskárnu nebo zobrazuje na displeji. Po $n = 1$ se vkládá nový text do tištěného souboru, pro $n = 0$ se doplňuje text na displeji. Nový text se nejprve ukládá v paměti a vypisuje se až po zadání znak "CR" nebo po dosažení 80ti znaků. Během vkládání textu z klávesnice do paměti je možno znakem "←" rušit chybně zapsaný text. Vkládání textu se ukončuje znakem "↓". Na začátku každého řádku se pro obsluhu vypisuje hlášení "INPUT", které signalizuje vkládání textu z klávesnice.

Pozor: Je-li zadáno pouze písmeno K, pak se dosadí $n = 1$ což je tisk na tiskárnu. Není-li tiskárna připojena, pak systém přestává zobrazovat další text, protože čeká na komunikaci s tiskárnou.

Stane-li se Vám tato chyba, je možné následujícím postupem zachránit text:

1. CTRL W ... předá řízení monitoru
2. Příkazem SUBSTITUTE zrušíme obsluhu tiskárny v Monitoru. Takže zrušíme volání ovladače tiskárny. Na adrese 10F změním instrukci C3 na C9 (RET).
`.S0316_10F_C3 C9 ↵`
3. Příkazem .B nebo G1003 znovu spustíme editor TEDIT a můžeme pokračovat.



§

<u>INSERT DECIMAL</u> ----- VLOŽENÍ DEKADICKÝCH ZNAKŮ

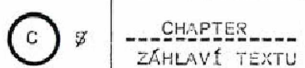
Povелем nI lze vkládat do textu dekadické znaky, které umožňují řízení tisku. Při n = 1 se nastaví tiskárna na tučný tisk (BOLDFACE). Toto nastavení trvá až po druhý vložený znak 1 nebo do konce řádku. Je-li n = 12, pokračuje tisk na nové straně. Vložením n = 18 nebo n = 17, se mění barva tisku u tiskárny Consul C 2111.

Poznámka: Povel "nI" ukládá znak, jehož kód odpovídá dekadické hodnotě n. Není-li n zadáno, TEDIT dosadí

hodnotu $n = 1$ (zdvojený tisk).

Další znaky uvedené za příkazem "I" se normálně uloží do paměti! Pro n nelze použít hodnotu 27 (ESC).

5.4. Ostatní povel



Povel C umožňuje zadat záhlaví stránky formátovaného textu do délky 80 znaků. V záhlaví je možno používat i znaky "CR" a "LF" a nastavit tak velikost horního okraje. Povel pro zadání záhlaví "KAPITOLA 1" :

```
> C KAPITOLA 1 § §
```

Zrušení záhlaví: > C § §

Začíná-li text znakem "%", potom před výpisem "Záhlaví textu" na novou stránku se vypíše na obrazovce "Zaloz dalsi list HOTOVO?" Teprve po stisknutí klávesy se pokračuje ve výpisu.



§

END OF FILE

KONEC SOUBORU

Povelen E se uzavře právě rozpracovaný výstupní soubor na kazetě. Má význam po povelu nO. Na rozdíl od modu MAIN se nekopíruje na kazetu text v paměti!



§

JUMP

SKOK

Povelen nJ se zadává okamžitá poloha kurzoru. Povel J se tato později vyhledává. Pozor, povel INSERT, DELETE ovlivňují polohu kurzoru!



MAIN

HLAVNÍ

Povelen M se přechází do modu MAIN.
- M zruší Menu.

Vložení poznámky pro obsluhu

Je-li v příkazovém řádku znak "%", potom se vypíše na obrazovku "HOTOVO?". Pozastaví se provádění programu a editor čeká, až obsluha provede činnost podle poznámky. Potom se stisknutím libovolné klávesy pustí pokračování činnosti editoru.

Příklad:

Vlož nový papír na položku č. 15

 zadání poznámky

20 < % # P § > § § vytištění dvaceti kopií, ale vždy před tiskem se čeká, až obsluha nastaví formulář.

Povелеm " [" nebo "{ " lze zapsat do vyrovnávací paměti text o délce až 80 znaků, který můžeme kdykoliv vypsat po stisknutí znaku CTRL B. Jsme-li v povelu INSERT a často používáme v nově psaném textu určité slovo, např. "KONDENZATOR", může toto slovo uložit do zapisovaného textu jedním stisknutím znaku CTRL B. Nebo chceme-li vložit do textu určitou větu, musíme do bufferu uložit povel a text:

* [I Ondra je mikropočítač bez dovozních součástí § §.

Potom po stisknutí CTRL B se z bufferu provede povel "I", který zapiše text na zvolené místo a po ukončení dvěma znaky "↓" je text zapsán.



Tato kapitola zahrnuje podrobný popis základních úkonů nezbytných pro používání editoru TEDIT 1 k nahrávání editoru do paměti RAM mikropočítače ONDRA, pořizování textových souborů a jejich formátování. Dále jsou uvedeny některé typické aplikace editoru.

Postup při používání editoru je znázorněn ve formě tabulek, kde sloupec označený K zahrnuje data zadávaná z klávesnice. Ve sloupci označeném D je uveden výpis na displeji a sloupec P obsahuje komentář.

6.1. Nahrávání editoru do paměti

Textový editor TEDIT 1 je dodáván na kazetě a jeho použití v systému ONDRA je možné až po nahrání z kazety do paměti RAM. Ke ztrátě programu v paměti dochází při každém vypnutí nebo při výpadku sítě. V uvedených případech je nutno editor znova nahrát z kazety.

K nahrávání editoru se používá operační systém MIKOS, který je uložen v paměti EPROM a je tedy k dispozici okamžitě po zapnutí mikropočítače ONDRA. Podrobné údaje o operačním systému MIKOS jsou uvedeny v Návodu k použití mikropočítače ONDRA.

Před začátkem nahrávání je třeba převinout pásek v kazetě na začátek. Je vhodné mít program TEDIT uložený na začátku kazety a až za ním ukládat pořizovací texty. Při používání dalších programů kromě TEDITu je výhodné používat dvě kazety, jednu systémovou pro programy a druhou pro texty. S ohledem na možné poškození pásku v kazetě je účelné po skončení práce překopírovat pořízené texty na další tzv. archivní kazetu.

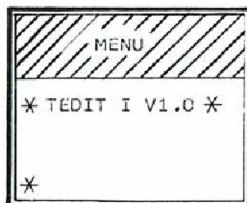
Postup při nahrávání editoru názorně ukazuje následující tabulka:

klávesnice:	displej:	poznámka:
1.	ONDRA V.5	Vypíše se po zapnutí mikropočítače ONDRA
2. L	.L čtení-hotovo?	Povel pro zavedení systémového programu do paměti. Nastavte mgf na čtení a stiskněte libovolnou klávesu

3. 

zhasnutí obrazovky


Provádí sečtení dat z kazety. Po nahrání programu se provede automatický start TEDITu.



Z monitoru lze TEDIT spustit příkazem "B" nebo G=1000 (studený start), G=1003 (teplý start).

6.2. Pořizování textového souboru

Je-li v paměti mikropočítače ONDRA nahrán textový editor TEDIT 1, je možno přikročit k pořizování žádaného textového souboru. Dále je uveden postup pořizování textového souboru s označením TEXT 1, jež plně znění je uvedeno v příloze č. 4

	K	D	P
1.		MAIN	Volba modu MAIN z klávesnice. Znak "\" udává začátek prostoru pro text (je nastaveno)

```

2.  I  B  SP  A  SP  S
    SP  I  SP  C  ↓  ↓
    SP  SP  J  E  D
    .  .  .  .
    .  .  .  O  B  V
    O  D  U  .  ↓  ↓  ..OBVCDU.
  
```

* _

MAIN

BASIC ↓

JED...

Po zadání povelu I se postupně zapisují jednotlivé znaky textu do paměti a současně se text zobrazuje na displeji. Chybně vložené znaky lze mazat klávesou "←". Znak "↓" označuje posuv na nový řádek a je provázen krátkým akustickým signálem. Znak " \ " udává konec vloženého textu. Pokud je třeba provést větší úpravy v již vloženém textu (záměna části textu, mazání řádků aj.), lze použít povel pro modifikaci textu uvedené v kap. 3.2. V tomto případě je však třeba nejprve vkládání textu ukončit 2x " ↓ ", provést požadovanou změnu textu a vkládání dalšího textu začít novým povelom "I".

3. poslední název Uzavření souboru, přechod do MIKOSu a požadavek MIKOSu na jméno souboru, za který má být pořízený soubor TEXT 1, na kazetu zapsán.

4. :DOPIS1 Nový soubor TEXT1 má být zapsán za soubor s názvem DOPIS1.

5. ctení hotovo? Dotaz MIKOSu, je-li požadavek na vyhledání místa na kazetě za již existujícím souborem DOPIS 1.

6. ctení hotovo?Y Odpověď z klávesnice stisknutím libovolného tlačítka. Při stisknutí tlačítka přehrávání na kazetovém mgf se vyhledá na kazetě místo za souborem DOPIS1 a magnetofon se zastaví.

7. zapis hotovo? Dotaz MIKOSu, požadavek na nastavení magnetofonu pro nahrání nového souboru.

- | | | | |
|-----|---|-----------------|--|
| 8. | <input type="checkbox"/> Y | zápis hotovo?Y | Odpověď z klávesnice (libovolné tlačítko) |
| 9. | | nový název
: | Požadavek MIKOSu na
název nového souboru |
| 10. | <input type="checkbox"/> T <input type="checkbox"/> E <input type="checkbox"/> X <input type="checkbox"/> T <input type="checkbox"/> 1 <input type="checkbox"/> J | :TEXT1 | Při stisknutých tlačítkách pro nahrávání na mgf - zápis nového souboru s názvem TEXT1 za již existující soubor DOPIS1, zastavení magnetofonu |

6.3. Formátování textového souboru

Na kazetě je zapsán textový soubor nazvaný TEXT1. Následující tabulka ukazuje postup při načtení souboru a výpis přeformátovaného souboru na tiskárnu v módu HELP. Požaduje se přeformátování textového souboru TEXT1 podle následujících požadavků:

- šířka stránky 60 znaků
- délka stránky 20 řádek
- stránky nečíslovat

K	D	P
1. <input type="text" value="R"/> <input type="text" value="#"/> <input type="text" value="A"/> <input type="text" value="↓"/> <input type="text" value="↓"/>	MAIN * R#A	Požadavek z klávesnice na otevření vstupního souboru a nahrání celého souboru do textové paměti R = otevření A-- načtení všech řádek souboru přechod do MIKOSu
2.	název :	Požadavek MIKOSu na název souboru, který má být na- hrán do textové paměti.
3. <input type="text" value="T"/> <input type="text" value="E"/> <input type="text" value="X"/> <input type="text" value="T"/> <input type="text" value="1"/> <input type="text" value=" "/> :	TEXT1	Zadání názvu souboru z klávesnice
4.	čtení hotovo?	Dotaz MIKOSu na nahrání souboru
5. <input type="text" value="Y"/>	MAIN BASIC ↓	Odpověď z klávesnice, nahrání souboru do texto- vé paměti a návrat do TEDITu.
6. <input type="text" value="H"/>	*- HELP BASIC\n > -	Přechod do modu HELP z modu MAIN.

7.

6	Ø	W	↓	2	Ø
L	↓	Ø	N	↓	D
↓	↓				

D 1

Po stisknutí předposlední klávesy "E↓C" je na displeji zobrazeno

HELP

BASIC ↵

.....

...^

>6ØX Ø 2ØL Ø ØN Ø D Ø Ø

C 1

Povelovým řetězcem byla zadána šířka stránky W=60, délka stránky L=20, stránky nečíslovat(N=0) a povelom D požadavek na zobrazení formátovacích parametrů a mapy textové paměti.

D 2

Po stisknutí poslední klávesy "E↓C" se na displeji zobrazí

#

.....

W=60 L=20 S=00 G=08 N=00 U=00

-----!-----!-----!-----!


C 2

Znak "# " udává, že je obsazen LKB textové paměti.

Znaky "...." udávají délku neobsazené textové paměti.

Znaky "--" je zobrazena šířka stránky a znaky "!" nastavení tabulátoru (G=8)

Význam ostatních formátovacích parametrů je uveden v kap. 4.2. - popis povelu D.

8. Text formátovaného F - formátování všech řádek souboru
 souboru
 T - zobrazení celého souboru na displeji

Výpis přeformátovaného souboru TEXT1 je uveden v příloze č. 5.

6.4. Doplňování textových souborů z klávesnice

Při pořizování dopisů, pozvánek apod. se velmi často vyskytuje případ, kdy má být dopis stejného nebo jen velmi málo odlišného znění odeslán různým adresátům. Takový dopis lze velmi efektivně pořídít kombinací základního textu dopisu uloženého v textové paměti s dalším textem, který je pro jednotlivé

adresáty rozdílný, z klávesnice.

Předpokládáme, že v textové paměti je uložen následující soubor s názvem DOPIS1.

Obvodní národní výbor Brno 5, odbor bytového hospodářství
Palackého 95, B r n o

Brno 18. 5. 1986

P o z v ě n k a

Žádáme vás, abyste se dostavil na ONV Brno V, I. poschodí, dveře č. 100 dne

Jednání se bude týkat Vaší žádosti o přidělení bytu.

Pozvánku je třeba poslat různým adresátům s různým datem a hodinou.

Nastaví se mod HELP, povelom S se nastaví kurzor na začátek souboru a zadá se povelový řetězec

>SP1K $\&$ $\&$ tisk na tiskárnu

nebo >STPK $\&$ $\&$ odzkoušení na displeji,

kterým se uskuteční tisk devíti řádků od začátku souboru a na displeji se vypíše zpráva "INPUT", která udává, že systé-
m je připraven přijímat znaky z klávesnice.

Na klávesnici se napíše první řádek adresy

" HT JIRI A D A M E C ↵ "

Klávesa "HT" je horizontální tabulátor (→). Po zadání znaku (nový řádek) se text vypíše na tiskárně a na displeji se opět zobrazí zpráva "INPUT". Pokračuje se zápisem dalších řádků adresy z klávesnice

"HT BOZETECHOVA 175 ↵ "

"HT 612 00 B R N O ↵ "

Vkládání znaků z klávesnice se ukončí znakem "↵".

Následuje přechod do modu MAIN, ve kterém se povelom

*12L ↵ ↵

přesune kurzor na řádek, od kterého bude pokračovat další výpis textu na tiskárně.

Přepne se opět do modu HELP a povelovým řetězcem

>9P1K ↵ ↵

se vytiskne devět řádků od řádku, na který je nastaven kurzor a na displeji se opět vypíše zpráva "INPUT". Z klávesnice se vloží

"HT HT 30. 6. 86 V 9.00 ↵ "

a vkládání se ukončí znakem "↵".

Přepne se do modu MAIN, povelom B se nastaví
kurzor na začátek textu a zadá se povelový řetězec:

* 22L \$#P \$ \$,

kterým se dotiskne soubor do konce.

Výpis dopisu pořízeného popsáním postupem je u-
veden v příloze č. 6.

6.5. Doplňování textových souborů z kazety

V administrativní praxi je velmi častým požadav-
kem pořizování dopisů nebo pozvánek stejného znění,
které jsou rozesílány stálému okruhu adresátů. V to-
to případě je vhodné pořídit textový soubor obsahují-
cí jména a adresy všech adresátů, uložit tento soubor
na kazetu a doplňování základního textu dopisu prová-
dět z kazety.

Předpokládáme, že v textové paměti je uložen ná-
sledující soubor s označením DOPIS2.

Obvodní národní výbor Brno 5, odbor bytového hospodářství
Palackého 95, Brno

Brno 16. 6. 86

P o z v á n k a

na plenární zasedání bytové komise, které se koná dne
8. 7. 86 v 9.00
v zasedací místnosti ONV Brno 5, I. poschodí, dveře č. 126

Do souboru je třeba doplnit jména a adresy jednotlivých členů komise. Předpokládejme dále, že kazeta obsahuje soubor nazvaný ADRESY, v němž jsou postupně uložena jména a adresy v následujícím tvaru

HT JOSEF N O V O T N Ý

HT TYLOVA 85

HT 612 00 B R N O /

HT FRANTIŠEK B U B L A

HT HUSITSKÁ 27

HT 612 00 B R N O

.
.
.
.

Znak "/" slouží jako oddělovač jednotlivých adres.
Tisk pozvánek s různými adresami se provede následujícím postupem.

Kazetový magnetofon se nastaví před soubor ADRESY, nastaví se mód HELP a povelom B se nastaví kurzor na začátek souboru.

Povelovým řetězcem

>11PR1A/ \$ \$

se vytiskne jedenáct řádek textu od začátku, otevře se soubor na kazetě /R/ a operační systém MIKOS výpisem "navez" na displeji vyžaduje název souboru na kazetě, z něhož se bude tisknout.

Na klávesnici se napíše název souboru ADRESY a po potvrzení výpisu "cteni hotovo?" známkem "Y" (resp. stiskem libovolné klávesy) z klávesnice se po vyhledání souboru ADRESY na kazetě vytiskne první adresa souboru.
Zbytek textu se dotiskne v modu MAIN povelom

*13L \$ # P \$ \$

Tisk další pozvánky s druhou adresou uloženou v souboru ADRESY se začne opět posunutím kurzoru na začátek textu /B/ a po zadání povelu v modu HELP

> 11P1A/ \$ \$

se vytiskne jedenáct řádek textu a druhá adresa. Přepne se do módu MAIN a povel

* 13L Z # P Z Z

se dotiskne zbytek textu. Uvedeným postupem je pak možno pokračovat v pořizování dalších pozvánek.

Pozvánky pořizené popsáním postupem jsou uvedeny v přílohách č. 7 a 8.



Textový editor TEDIT patří mezi základní programové vybavení pro mikropočítač ONDRA. Editor pracuje ve dvou základních režimech - MAIN a HELP. Režim MAIN je pro pořízení a úpravy textů a jejich uložení na výstupní zařízení. Režim HELP je určen pro formátování textových souborů a speciální editační funkce. Výstup v tomto režimu je převážně určen na tiskárnu.

Editor pracuje ve dvou módech. Prvním módem je video-mód, ve kterém se příkazy okamžitě po zadání provádí a není nutné je potvrzovat. Druhým módem je "normální" příkazový mód, kdy příkazy se zadávají v příkazovém řádku. Příkazy se začínou provádět až po zadání ukončovacích znaků příkazové řádky (tj. dvakrát znak "↓", který se na obrazovce zobrazuje znakem "§").

Textový editor TEDIT byl vyvinut z editoru PEDIT, který je součástí bloku programů pro práci v assembleru nazvaného TOOL. Převážná část příkazů v režimu MAIN je obdobná s příkazy editoru PEDIT. Proto, když zvládnete TEDIT, můžete bez problému pracovat i s editorem PEDIT.

Věříme, že textový editor TEDIT se stane platným pomocníkem při Vaší práci a že oceníte možnosti, které nám poskytuje při práci s textovými soubory.

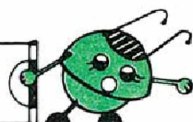


Přehled povelů textového editoru TEDIT 1 v modu MAIN

Kód povelu	Funkce
A	Čtení textu z kazety
B	Nastavení kurzoru na začátek textu
C	Posuv kurzoru o specifikovaný počet znaků
D	Vymazání specifikovaného počtu znaků
E	Ukončení edice a zápis textu na kazetu
F	Vyhledání specifikovaného textového řetězce
G	Vyznačení a výpis textu určeného ke kopírování do jiného místa souboru
H	Přechod do modu HELP
I	Vložení textu
J	Zadání nebo vyhledání okamžité polohy kurzoru
K	Vymazání specifikovaného počtu řádků
L	Posuv kurzoru a specifikovaný počet řádků
M	Výpis počtu volných bytů v paměti
O	Výpis textu na kazetu
P	Výpis textu na tiskárnu

Q	Návrat do MONITORU
R	Otevření vstupního souboru na kazetě
S	Náhrada původního textového řetězce nově specifikovaným řetězcem
T	Výpis textu na displej
U	Přemístění textu označeného příkazem S na pozici vyznačenou kurzorem
V	Zobrazení textu v okolí kurzoru
W	Výpis textu na kazetu
X	Zobrazení následujícího textu
Y	Zobrazení předchozího textu
Z	Nastavení kurzoru na konec textu
.	Posuv kurzoru o slovo vpravo
,	Posuv kurzoru o slovo vlevo
CTRL B	V příkazu "I" vložení vyrovnávací paměti do psaného textu
CTRL C	Přerušování činnosti
CTRL D	Vymazání znaku v místě kurzoru
CTRL I	Vložení znaku na místo označené kurzorem
CTRL K	Vymazání řádku od znaku na němž se nachází kurzor
CTRL S	Nahrazení znaku v místě kurzoru novým znakem
5	Zadání nebo vyvolání textu pro obsluhu
I	Vložení části textu (až 80 znaků) do vyrovná- vací paměti

PŘÍLOHA Č. 2



Přehled povelů textového editoru TEDIT 1 v modu HELP

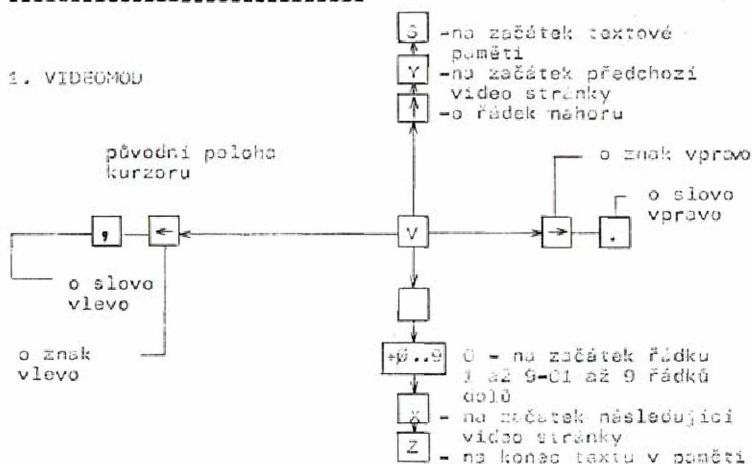
Kód_povelu	Funkce
A	Vkládání textu z kazety do textu vypisovaného na tiskárně nebo na displeji
B	Nastavení kurzoru na začátek textu
C	Zadání textu záhlaví stránky
D	Zobrazení mapy textové paměti a parametrů formátování
E	Uzavření souboru
F	Zadání počtu formátovaných řádků
G	Nastavení délky tabulátoru
I	Zadání znaků pro změnu barvy tisku, tučný tisk a nastavení nové stránky
J	Zadání nebo vyhledání okamžité polohy kurzoru
K	Vkládání textu z klávesnice přímo do textu vypisovaného na tiskárně nebo na displeji
L	Nastavení délky stránky
M	Přechod do modu MAIN

N	Číslování stránek
O	Výpis textu na kazetu
P	Výpis textu na tiskárnu
Q	Návrat do modu MAIN
R	Otevření vstupního souboru na kazetě
S	Nastavení velikosti levého okraje
T	Výpis textu na displej
U	Přepínání velkých nebo malých písmen
V	Zobrazení textu v okolí kurzoru
W	Nastavení šířky stránky
X	Zobrazení následujícího textu
Y	Zobrazení předcházejícího textu
Z	Nastavení kurzoru na konec textu
.	Posuv kurzoru o slovo vpravo
,	Posuv kurzoru o slovo vlevo
%	Zadání nebo vyvolání textu pro obsluhu
[Vložení části textu do vyrovnávací paměti
CTRL C	Přerušení činnosti

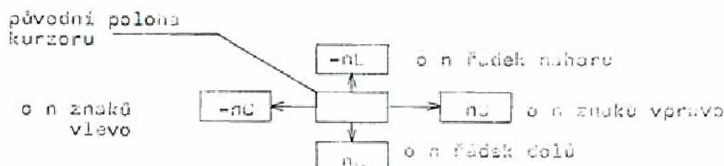


Povelů pro přemístění kurzoru

1. VIDEOMOD



2. POKELV, KTERÉ VYŽADUJÍ UKONČENÍ



Poznámka: YL - posuv na začátek téhož řádku



Textový soubor "TEXT1"

B A S I C

Jedním z prvních programovacích jazyků použitých v systémech se sdílením času byl BASIC (název jazyka je zkratkou v BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE) byl vyvinut J. Kemenym a T. Kurtzem v šedesátých letech v USA právě s cílem poskytnout velkému počtu uživatelů přístup k počítači prostřednictvím jednoduchého jazyka. Spolu s vývojem počítačové techniky byly definice (specifikace) jazyka několikrát měněny a jazyk stále rozšiřován.

Ačkoliv se BASIC stal známým prostřednictvím počítačových systémů pracujících se sdílením času, masové rozšíření jazyka nastalo až v sedmdesátých letech, kdy se s prudkým rozvojem elektroniky rozšířily počítače s vysokým stupněm integrovanosti elektronických obvodů.



Textový soubor "TEXT1" přeformátovaný
na šířku stránky W=60 znaků

B A S I C

Jedním z prvních programovacích jazyků použitých v systémech se sdílením času byl BASIC (název jazyka je zkratkou z BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE) byl vyvinut J. Kemenym a T. Kurtzem v šedesátých letech v USA právě s cílem poskytnout velkému počtu uživatelů přístup k počítači prostřednictvím jednoduchého jazyka. Spolu s vývojem počítačové techniky byly definice (specifikace) jazyka několikrát měněny a jazyk je stále rozšiřován.

Ačkoliv se BASIC stal známým prostřednictvím počítačových systémů pracujících se sdílením času, masové rozšíření jazyka nastalo až v sedmdesátých letech, kdy se s prudkým rozvojem elektroniky rozšířily počítače s vysokým stupněm integrovanosti elektronických obvodů.

PŘÍLOHA Č. 6



Textový soubor "DOPIS1" doplněný z klávesnice

Obvodní národní výbor Brno 5, odbor bytového hospodářství
Palackého 95 B R N O

Brno 16. 6. 86

Jiří A D A M E C

Božetěchova 175

612 00 B R N O

P o z v á n k a

Žádáme Vás, abyste se dostavil na ONV Brno 5, I. poschodí dveře č. 100 dne

30. 6. 86 v 9.00

Jednání se bude týkat Vaší žádosti o přidělení bytu.

PŘÍLOHA Č. 7



Textový soubor "DOPIŠ2" doplněný
první adresou z kazety

Obvodní národní výbor Brno 5, odbor bytového hospodářství
Palackého 95, B R N O

Brno 10. 6. 86

Josef Novotný
Tylova 86
612 00 B R N O

P o z v á n k a

na plenární zasedání bytové komise, které se koná dne
8. 7. 86 v 9.00
v zasedací místnosti ONV Brno 5, I. poschodí, dveře čis. 126.

PŘÍLOHA Č. 8



Textový soubor "DOPIS2" doplněný druhou adresou z kazety

Obvodní národní výbor Brno 5, odbor bytového hospodářství
Palackého 95, B R N O

Brno 10. 6. 86

František D u b l a
Husitská 27
612 00 B R N O

P o z v á n k a

na plenární zasedání bytové komise, které se koná dne

8. 7. 86 v 9.00

v zasedací místnosti ONV Brno 5, I. poschodí, dveře čis. 126.



Převodní tabulka klávesnice

ONDRA	MEZINÁR. OZNAČ. /ASCII/	HEX , HODNOTA
	CR	0D
	ESC	1B
CTRL C	STX	03
	HT	09
CTRL I	HT	09
mezera	SP	20
CTRL D	EOT	04
CTRL K	VT	0B
CTRL B	STX	02
	BS	08



1.	Úvod	3
2.	Textový editor TEDIT 1	5
2.1.	Klávesnice	7
2.2.	Zobrazení na displeji	7
3.	Povelý pro řízení textového editoru TEDIT 1	12
3.1.	Tvar povelů	12
3.2.	Opravy povelů	13
3.3.	Příklady povelů	13
3.4.	Videomod	14
3.5.	Povelové řetězce a jejich opakování	14
4.	Povelý editoru TEDIT 1 v modu MAIN	16
4.1.	Povelý pro nastavení kurzoru a zobrazení textu	16
4.2.	Povelý pro vkládání a modifikaci textu	23
4.3.	Povelý pro vstup a výstup textu, sou- činnost s periferním zařízením	27
4.4.	Ostatní povelý	31
5.	Povelý editoru TEDIT 1 v modu HELP	36
5.1.	Povelý pro nastavení kurzoru a zobrazení textu	36

5.2.	Povely pro zobrazení a nastavení formátovacích parametrů a proformátování textu	37
5.3.	Povely pro vstup a výstup textu, součinnost s periferním zařízením	42
5.4.	Ostatní povely	47
6.	Použití textového editoru TEDIT 1	51
6.1.	Nahrání editoru do paměti	51
6.2.	Pořízení textového souboru	53
6.3.	Formátování textového souboru	56
6.4.	Doplňování textových souborů z klávesnice	59
6.5.	Doplňování textových souborů z kazety	62
7.	Závěr	66

Přílohy:

1.	Přehled povelů textového editoru TEDIT 1 v modu MAIN	69
2.	Přehled povelů textového editoru TEDIT 1 v modu HELP	71
3.	Povely pro přemístění kurzoru	73
4.	Textový soubor TEXT 1	74
5.	Textový soubor TEXT 1 přeformátovaný na šířku stránky W = 60 znaků	75
6.	Textový soubor DOPIS1 doplněný z klávesnice	76
7.	Textový soubor DOPIS2 doplněný první adresou z kazety	77

8.	Textový soubor DOPIS2 doplněný druhou adresou z kazety	78
9.	Převodní tabulka klávesnice	79
Obsah		80