

# Uživatelská dokumentace k programu TextEd

MATEMATICKO-FYZIKÁLNÍ FAKULTA UNIVERZITY KARLOVY  
ADAM ŘEŘICHA

# 1 Obsah

2	Anotace .....	2
3	Adresování řádků.....	2
3.1	Zkratky .....	2
3.2	Relativní adresování.....	2
4	Příkazy .....	2
4.1	Struktura příkazu.....	2
4.2	Výpis příkazů.....	3
5	Vypisování chyb .....	4

## 2 Anotace

Program TextEd je textový editor v příkazové řádce, který je podobný Linuxovému textovému editoru [ed](#). Umí základní operace, které by člověk čekal od takového programu jako například načíst soubor, mazat, přepisovat, posouvat, kopírovat řádky a tak dále. Také umí přepisovat slova shodující se se zadaným regulárním výrazem a zarovnat text do bloku. Program se dá spustit s volitelným argumentem, který je název souboru, který se při spuštění programu načte do paměti a je připraven k editaci.

Program se ovládá pomocí příkazů, které se značí jedním písmenem. Můžou mít i volitelný argument a rozsah řádků, ve kterém se příkaz vyvolá. Pro úpravu souboru je potřeba načíst soubor. Po vykonání potřebných editací na textu se musí zapsat zpět do souboru, nebo do jiného, jinak se všechny úpravy ztratí.

## 3 Adresování řádků

U většiny příkazu je možné určit rozsah, ve kterém bude daný příkaz vykonán. Pro to je nutné znát, jak adresovat jednotlivé řádky. Jde to přímo číslem řádku, nebo pomocí zkratky, nebo relativními adresami vzhledem k nějaké adrese.

### 3.1 Zkratky

- Aktuálně vybraný řádek
- \$ Poslední řádek v paměti
- , Rozsah od prvního řádku po poslední. Ekvivalentní zápis je **1,\$**

### 3.2 Relativní adresování

**N+M** Řádek vzdálený **M** řádků níže od řádku **N**

**N-M** Řádek vzdálený **M** řádků výše od řádku **N**

**+** Zkratka pro **.+1**

**-** Zkratka pro **.-1**

**+**, **-** a celkově relativní adresování se dá použít i také pro nastavení vybraného řádku. Tedy pomocí **+** a **-** se dá jednoduše procházet řádky v paměti. Zkratky se také dají použít.

## 4 Příkazy

Zde popisují, jakou strukturu příkazy dodržují a jaké příkazy jsou dostupné v tomto programu.

### 4.1 Struktura příkazu

Obecně příkaz může specifikovat nula nebo více řádků, na kterých se provede, poté následuje písmeno reprezentující jméno příkazu a dále můžou být další argumenty pro příkaz, jako například destinační řádek, nebo soubor, do kterého se bude zapisovat nebo ze kterého se bude číst.

**[ADRESA],[ADRESA][PŘÍKAZ][DESTINAČNÍ\_ADRESA] [ARGUMENT]**

Některé příkazy (např. **a** nebo **i**) po spuštění program přepne do zapisovacího režimu, ve kterém nelze spustit žádný příkaz. Jen lze zadávat text, který se třeba poté připojí na konec textu v paměti nebo jiným způsobem zapíše do paměti. Zapisovací režim se ukončí pomocí tečky na samostatném řádku.

## 4.2 Výpis příkazů

V kulatých závorkách jsou výchozí rozsahy, které se použijí, pokud žádný není specifikovaný. Pokud příkaz má v rozsahu jen jedno číslo, tak bere jen jedno číslo. Pokud se takovému příkazu napíše rozsah, tak se použije levá adresa rozsahu. Napravo od názvu u některých příkazů je v kulatých závorkách napsána výchozí destinační adresa.

**P** – Ukáže/schová hvězdičku, \* na začátku řádku pro rozpoznání příkazového režimu od zapisovacího.

**h** (help) Vypíše chybovou hlášku poslední chyby.

**H** (help) Spuštěním tohoto příkazu se budou chyby vypisovat i s chybovými hláškami.

**(.,.)p** (print) Vypíše řádky z paměti v zadaném rozsahu.

**(.,.)n** (number) Vypíše řádky v zadaném rozsahu s jejich pořadím.

**(\$)r SOUBOR** (read) Přečte zadaný textový **SOUBOR** a přidá přečtené řádky za specifikovaný řádek do paměti a nastaví **SOUBOR** jako výchozí. Pokud není dán **SOUBOR**, tak se použije výchozí soubor.

**(1,\$)w SOUBOR** (write) Zapiše řádky v zadaném rozsahu z paměti do **SOUBORu**. Pokud není specifikovaný **SOUBOR**, tak se použije výchozí soubor.

**f SOUBOR** (file) Nastaví výchozí soubor na **SOUBOR**. Pokud není zadán **SOUBOR**, vypíše se aktuálně nastavený výchozí soubor.

**N** Změní aktuálně vybraný řádek na **N**, kde **N** je číslo řádku nebo některá z adresovacích zkratk a vypíše obsah daného řádku.

**(.)a** (append) Po zadání tohoto příkazu začne zapisovací režim (pokud je zapnutý **P** příkaz, tak zapisovací režim lze poznat tak, že na začátku řádku není hvězdička, \*), kde se zadává text, který se připojí za danou adresu. Zapisovací režim se přeruší napsáním samostatné tečky na řádek.

**(.)i** (insert) Po zadání příkazu začne zapisovací režim a napsané řádky textu se vloží na zadaný řádek.

**(.,.)d** (delete) Smaže řádky v zadaném rozsahu.

**q** (quit) Ukončí program, pokud nejsou neuložené změny. Pro ukončení programu nehledě na změny je vhodný příkaz **Q**.

**Q** (quit) Ukončí program nehledě na změny na řádcích v paměti.

**(.,+1)j** (join) Spojí řádky v zadaném rozsahu do jednoho.

**(.,.)c** (change) Po zadání příkazu začne zapisovací režim a napsané řádky přepíše řádky v zadaném rozsahu.

**e SOUBOR** (edit) Smaže všechny řádky v paměti a načte do ní zadaný **SOUBOR**. Ten se nastaví jako výchozí soubor. Pokud není specifikovaný, tak se použije výchozí soubor. Pokud byly udělány změny na řádcích v paměti, tak se vypíše varování, že jsou v paměti neuložené změny. Aby se provedl tento příkaz, tak je potřeba zapsat řádky z paměti do souboru pomocí příkazu **w**, nebo použít příkaz **E**, který smaže a načte soubor neohledně na změny na řádcích.

**E SOUBOR** (edit) Dělá to samé, co **e**, ale vykoná se bez ohledu na změnách na řádcích.

**(.,.)s/REGEX/NAHRAZENÍ/PŘÍPONY** (substitute) Na řádcích v paměti v zadaném rozsahu se přepíše první výskyt textu, který se shoduje s regulárním výrazem **REGEX**, na text **NAHRAZENÍ**. Po nahrazení textu se vypíše řádek s nahrazeným textem. Pokud se nenajde shoda s regulárním výrazem, tak se vypíše chyba. Volitelně se můžou zadat přípony, které mění chování příkazu.

Použitelné přípony:

**g** (global) – příkaz nahradí všechny výskyty shodujícího textu na **NAHRAZENÍ** v zadaném rozsahu řádků. Na konci se vypíše řádek s poslední shodou.

**i** (case-insensitive) – příkaz nahradí shodující text neohledně na malá a velká písmena.

**n** (number) – příkaz vypíše řádek se změněným textem s jeho číslem.

**(.,.)m(.)** (move) Posune řádky v paměti v daném rozsahu na dané číslo řádku.

**(.,.)t(.)** (transfer) Kopíruje řádky v zadaném rozsahu z paměti a vloží je na specifikovaný řádek.

**u** (undo) Vráť řádky v paměti do stavu před posledním vykonaným příkazem, který nějakým způsobem řádky v paměti změnil. Příkaz **u** je sám k sobě inverzní, takže použitím příkazu dvakrát řádky v paměti dostane do stavu před prvním použitím příkazu.

**b DÉLKA\_ŘÁDKU** (block) Zarovná řádky v paměti do bloku s délkou řádku **DÉLKA\_ŘÁDKU**. Pokud **DÉLKA\_ŘÁDKU** není specifikována, použije se jako délka řádku číslo 72.

## 5 Vypisování chyb

Pokud nastane nějaká chyba, tak se vypíše otazník **?**. Pro vypisování chybových hlášek je potřeba použít příkaz **H**. Pro vypisování chybové hlášky poslední chyby je příkaz **h**.