

1 Zanesení korektury.....	1
1.1 Příkaz na zanesení korektury.....	1
1.1.1 Jak to funguje.....	1
1.1.2 Obtížnost.....	1
2 Detaily makra.....	2
2.1 Ruční provádění korektur.....	2
2.2 Jiný způsob umístění kurzoru.....	2
2.3 Dvoj- a jednoznakové rychlovky.....	3
2.4 Změna interpunkce.....	3
2.5 Posun kurzoru.....	3
2.6 Úvodní makro.....	4
2.7 Závěrečné makro.....	5
2.8 Zvýraznění umístění kurzoru.....	5
2.9 Logování příkazů.....	5

# 1 Zanesení korektury

## 1.1 Příkaz na zanesení korektury

### 1.1.1 Jak to funguje

Pokud máme před sebou předlohu, na monitoru text a připojené makro, můžeme ho spustit a začít zadávat příkazy. (Předtím je však ještě dobré spustit úvodní makro, ale o něm až trochu dále.) Pro spuštění makra je přiřazena klávesová zkratka [Alt]+[Mezerník]. Tuto kombinaci lze zmáčknout pomocí palců obou rukou, aniž by se ruce pohnuly ze svého prstokladu. Po spuštění makra se nám objeví okno s příkazovým řádkem.

Umístění tohoto okna s příkazovým řádkem je záměrně posunuto co nejvíce do levého horního rohu. To proto, aby byla vidět pokud možno celá část Wordu, ve které se nachází text.

Nyní je vše připraveno a můžeme začít. Do příkazového řádku zadáme celý příkaz na provedení korektury. Tedy jak část pro hledání, tak část označující

počet (vyžaduje-li to korektura), tak také příkaz korektury. Všechny tři části jsme si popsalí, teď je pouze dáme dohromady a zapíšeme.

Vraťme se k naší ukázkové větě, kterou zde raději napíši znovu a připomenu, že je potřeba podtržené slovo smazat.

Uvádí, že organizace určitě uspěla ve velkém konkurenčním boji.

Do příkazového řádku zapíšeme nejdříve tři znaky pro hledání, tedy znaky uuv značící určitě uspěla ve, čtvrtým znakem bude j nebo 1 pro označení jednoho slova a jako pátý znak zadáme s znamenající příkaz smazat.

Celý příkaz tedy je: uuvjs

Tento příkaz, zapsaný do příkazové řádky makra, odentrujeme a tím se příkaz provede. Během mžiku je příkazová řádka opět připravena pro další zadávání. Soutěžící se tedy dívá jen do předlohy, píše příkazy a ty potvrzuje tlačítkem [Enter].

Některé konkrétní ukázky najdete také v Rozhledech. My se ale nyní podíváme na přílohu A, kde je soutěžní zadání z mistrovství světa ve Vídni. První odstavec (čtyři korektury) by se provedly pomocí těchto příkazů: wnfjp, šztdt, z,bvtyto a vp,js. Druhý odstavec: tvjts, pvapyt a ,žpvnapříklad. Třetí odstavec: k.nvkamna, d.uvtam, dpjjs, nvpdů, 3wnjyd, owstc a jsStránky. Čtvrtý odstavec: žn.jsnikdo, sc,dyj, a(tjk, wtčo, d js a zunjt. Obdobně zbývající odstavce.

### 1.1.2 Obtížnost

Vypadá to velmi složitě, a tak si možná říkáte, že něco tak složitého se nevyplatí učit, že to zabere spoustu času. Pokud děláte korekturu textu jednou za čas a je vám jedno, jestli ji uděláte za pět minut nebo čtyři minuty a 37 vteřin, pak se asi opravdu nevyplatí věnovat několik hodin učení. Pracujete-li s korekturou častěji, může vám makro postupem času ušetřit nejméně minutu.

Asi se také ptáte, jak dlouho trvá naučit se jednotlivé příkazy a hlavně začít vnímat drobné detaily, které jsou ke zvládnutí potřeba (jako příklad si uveďme již zmiňované: způsob hledání, počítání slov, naučit se

jednotlivé zkratky příkazů). Samozřejmě doba potřebná k naučení je velice individuální záležitost. Mohu však uvést jeden poznatek z praxe.

Už asi patnáct soutěžících (převážně reprezentanti České republiky na Mistrovství světa ve zpracování textu ve Vídni, 2005) se makro učilo, naučilo a používá. Prvně se o něm dozvěděli na soustředění v Karviné. Zde se soutěžící poprvé, během půlhodiny, kterou jsem věnoval přednášce o makru, dozvěděli základní principy. Národní trenér Jaroslav Zaviačič pak nechal rozdat tréninkové předlohy a byl konec teoriím. Makro se začalo zkoušet prakticky. To, že první kroky jsou vždy obtížné, víme všichni. Důležitá je ale informace, jak dlouho tyto kroky trvají. Že stačí jen desítky minut, abychom se naučili základy, dokládá Jaroslav Zaviačič: „... někteří soutěžící při prvních desetiminutovkách s potížemi zvládali několik desítek korektur, jak si na nový způsob zvykali a povedli si postupně mechanizovali, počty zpracovaných korektur se zvyšovaly několikanásobně.“

Důležité je podotknout, že toto soustředění bylo jen víkendové a hlavně nebylo zaměřeno jen na korektury, ale i na opis a diktáty. Korekturám tedy nebylo věnováno příliš času. Ale i to stačilo!

Na druhém soustředění, o necelé dva měsíce později, v Havlíčkově Brodě, už soutěžící jen pilovali své výkony. Občas byly dotazy na konkrétní problémy a jak je efektivně řešit. Ale většinu operací už zvládali bez problémů. Důležitým poznatkem tedy je, že soutěžícím stačilo jen několik desetiminutovek pro pochopení práce s makrem, pak několik desetiminutovek na procvičování. Z toho vyplývá, že základy se lze naučit řádově během několika hodin. Časovou náročnost naučení nevidím jako nepřekonatelnou, ba naopak.

## 2 Detaily makra

Jak už jsme si řekli, je třeba na soutěžích dbát na každý detail. Proto i makro má spoustu drobných vylepšení, které lze využít.

### 2.1 Ruční provádění korektur

I když makro zvládá všechny korektury, které se v dané soutěžní předloze mohou objevit, i přesto se může stát, že soutěžící z nějakého důvodu dá přednost ručnímu zanesení korektury. Buď se domnívá, že ručně to půjde rychleji než za pomoci makra, nebo, což je častěji, si soutěžící zrovna nemůže vybavit, jak to za pomoci makra provést.

Makro lze ukončit stisknutím tlačítka [Escape] a lze pracovat s Wordem tak, jak každý umí. Jak jsem již uvedl, většinou se jedná o případ, kdy soutěžící neví, jak korekturu za pomoci makra provést. Byla by tak škoda mu nepomoci v oblasti, kterou zná – tedy pomoci mu s vyhledáním a umístěním kurzoru, případně ještě i s označením části textu.

Pokud tedy narazí na korekturu, pro kterou zapomněl příkaz, může do příkazové řádky zadat pouze tři, resp. čtyři znaky (které musí potvrdit klávesou [Enter]). Zadá-li pouze tři znaky – což jsou znaky pro lokalizaci umístění kurzoru – a odentruje, kurzor se na dané místo přemístí a makro se automaticky ukončí. Pokud soutěžící zadá čtyři znaky (a nejedná-li se o nějaký čtyřznakový příkaz), makro přesune kurzor na určené místo a označí daný počet slov. Vzápětí se makro ukončí. Pak soutěžící pracuje normálně s Wordem a poté může opět makro spustit pomocí [Alt]+[Mezerník].

### 2.2 Jiný způsob umístění kurzoru

V soutěžní předloze můžeme narazit na případ, kdy jsou dvě korektury v textu hodně blízko sebe. Po provedení první korektury se druhá nachází buď hned za ní, nebo jen o několik slov za ní. Sami, když předlohu vidíte, si můžete říct: Ta další korektura je dvě slova za tou, kterou jsem zrovna teď provedl. A pokud si něco takového může říct soutěžící, bylo by škoda, kdyby se tato informace nedala předat makru.

Pokud jsou dvě korektury hned za sebou, je zbytečné zadávat tři znaky pro hledání. Výhoda je nejen v ušetřených znacích, ale také v tom, že takové určení místa je jednoznačné.

Pro klasické hledání jsou potřeba tři znaky. To nám dává možnost makro dost rozšířit. Když zadáme místo tří znaků hledání jakýkoliv jiný kratší řetězec (tedy místo oněch tří znaků dva, nebo jeden), můžeme tím určit úplně jiné příkazy.

Pokud místo oněch tří znaků (ostatní znaky zůstávají stejné) zadáme mezeru (jeden znak), řekneme tím, že korekturu má provést na místě, kde stojí kurzor. Slouží pro případy, kdy dvě korektury jsou hned za sebou.

Můžeme také zadat číslo a mezeru (což jsou dva znaky) místo tří znaků pro hledání. Tím řekneme: Posuň kurzor o daný počet slov doprava a pak proved korekturu. Tohoto využijeme v případě, kdy jsou dvě korektury blízko sebe, třeba jen dvě či tři slova od sebe.

Uveďme si to raději opět na nějakém příkladě. Máme ukázkovou větu, kde je potřeba první podtržené slovo nahradit za čárku, druhé podtržené slovo je potřeba smazat.

Uvádí, že organizace určitě uspěla ve velkém konkurenčním boji.

Pokud by korektury nebyly blízko sebe, použijeme dva klasické příkazy: .žojs, a uuvjs

Protože ale korektury stojí blízko sebe, můžeme použít zrychlení v podobě hledání umístění kurzoru pro druhou korekturu. První příkaz .žojs, zůstane stejný, druhý se ale změní, bude d js.

## 2.3 Dvoj- a jednoznakové rychlovky

Pokud do příkazu zadáme jen dva znaky, případně pouze jeden znak, je jasné, že se nejedná o plnohodnotný příkaz. Nejsou splněny minimální tři znaky pro hledání.

Tohoto je opět využito ku prospěchu věci. Příkazy, které lze zadat jedním nebo dvěma znaky (počítáme bez znaků na hledání) lze do příkazové řádky zapsat a odentrovat. Daný příkaz se provede v místě, kde stojí kurzor. Je to tedy obdobné, jako bychom místo

hledání napsali mezeru (viz předchozí kapitola). Tuto mezeru si však můžeme ušetřit, jde-li o jedno- či dvouznakový příkaz. Takové korektury mohou být např.: změna řezu písma, smazání slov, vložení odstavce aj.

## 2.4 Změna interpunkce

V soutěžní předloze z Vídně se začala objevovat svým způsobem typická korektura. Jednalo se o změnu interpunkčního znaménka za jiné (objevila se zde celkem 15krát). Např. místo tečky se má napsat čárka, místo středníku čárka, místo tečky vykřičník a jiné varianty.

Značný výskyt takové specifické korektury vedl k zamýšlení, jestli nelze příkazy nějak optimalizovat. Vždy se jedná o nahrazení jednoho slova (slova ve smyslu tak, jak ho chápe Word).

Uvádí, že organizace určitě uspěla ve velkém konkurenčním boji.

Normální příkaz v této ukázkové větě by byl .žojs,. Optimalizace proběhla tak, že se v příkazu vynechá js, ostatní zůstává, tedy příkaz ve finále bude .žo,.

Jak je toto možné? Na čtvrté pozici v příkazu se většinou vyskytuje označení počtu slov/znaků. Jak lze daný počet označit, je přesně definováno.

Je tedy možné na čtvrté pozici využít jakékoliv jiné znaky, které nejsou v tabulce uvedeny, a patřičně naprogramovat, aby se provedl nějaký příkaz. Jak je vidět, v tabulce žádný interpunkční znak není, nic tedy nebrání v jejich použití.

Ušetříme tak v jednom příkazu dva znaky, což je obrovská úspora. Nejen ve zkrácení době zápisu, ale čím méně znaků, tím menší šance přepsání se a následného provedení špatné korektury, nebo neprovedení žádné a potřeby zapsat příkaz znovu.

## 2.5 Posun kurzoru

Na tomto místě musím podotknout jednu podstatnou věc. Část makra obstarávající hledání, tedy lokalizaci

umístění kurzoru, prohledává část textu od současně-  
ho umístění kurzoru vždy doprava.

Pokud tedy potřebujeme provést nějakou korekturu před současným umístěním kurzoru, makro nám toto místo nenajde. Taková situace nastává nejčastěji, pokud přehlédneme nějakou korekturní značku v předloze a všimneme si jí, až provedeme korekturu následující (hůře, pokud si toho nevšimneme vůbec, penalizace je obrovská). Jediné, co nám stačí, abychom mohli nerušeně zadávat příkazy, je, abychom kurzor dostali před korekturu, kterou jsme opomněli. Není potřeba kurzor umístit přímo před danou zapomenutou korekturu, stačí ho umístit někde před ni v rozumném rozmezí.

Bylo by zbytečné a zdlouhavé ukončovat makro, zmáčknout několikrát šipku nahoru, aby se kurzor přemístil o patřičný počet řádek výše a pak zase makro spouštět. Elegantnější by bylo, kdyby posunutí kurzoru obstaralo přímo makro. Byl proto navržen další dvouznakový příkaz.

První znak uvádí počet řádků, o kolik se má posunout kurzor. Číslování je stejné jako všude jinde, stejné, jako jsme si dříve uvedli. Takže se nejedná o nic nového. Druhým znakem uvedeme, zda chceme kurzor přesunout doleva (resp. nahoru), nebo doprava (resp. dolů). Není potřeba pro toto rozlišení vymýšlet nová písmena. Použijeme již známé znaky y a -, které známe např. z přesunu slov. Klávesa [y] z levé části klávesnice přesouvá kurzor doleva, klávesa [-] z pravé části klávesnice přesouvá doprava. Kdo tedy zná základy a příkazy, není to pro něj nic nového.

Provedeme-li nějakou korekturu a v předloze si všimneme, že na předchozí řádce je korektura, kterou jsme přehlédli, stačí vrátit kurzor. Není dobré ho posouvat o jednu řádku, mohlo by se stát, že ani tak se kurzor nedostane před patřičné místo. Ideální je zadat vždy o jeden až dva řádky více. Je-li tedy korektura na předchozím řádku, posuneme kurzor o tři řádky, tedy zadáme příkaz ty. Pak už zadáme normální příkaz na provedení korektury.

Posun kurzoru budeme potřebovat, i když budeme pracovat pečlivě a žádnou korekturu nepřehlédneme. V předlohách se objevuje takový případ: Máme určitý blok textu, tento blok je potřeba přesunout a zároveň je potřeba v tomto bloku zanést korektury. Postup může být následující: provedeme korektury

uvnitř bloku, tím je kurzor někde v bloku a my ho potřebujeme dostat před blok, abychom ho mohli označit a přesunout. Tady je právě potřeba přesunout kurzor a pak můžeme korekturu přesunu bez problémů použít. Kdybychom kurzor neposunuli, při označení bloku by došlo k problému, že by nebyl nalezen začátek bloku.

## 2.6 Úvodní makro

Před zahájením soutěže je vždy výhodné, ba dokonce nutné spustit úvodní makro. To provede několik nastavení. Jako nejviditelnější a pro soutěžícího nejdůležitější je nastavení řádkování na 1,5 a nastavení fontu na Courier New. Takové nastavení zpřehlední text na monitoru a v případě, že je potřeba na monitoru něco zkontrolovat, lze daný text snáze číst.

Dalším krokem, které úvodní makro provede, je uložení. Nejde jen o uložená data, ale i o to, že v případě, kdy pracujeme s novým (ještě ani jednou neuloženým) dokumentem, tak se nás zeptá na název souboru a umístění na disku, kam chceme uložit. Proč se toto děje? Má to jasný cíl. Protože i závěrečné makro (viz následující kapitola) ukládá, je nutné zajistit, aby název souboru byl definován dopředu. Závěrečné makro se spouští zároveň s hvizdem píšťalky označující konec soutěže. Po tomto hvizdu se už nesmí sahat na klávesnici ani myš. A jak bychom pak zadali název souboru bez toho, abychom se dotkli klávesnice? Proto se ukládá už v úvodním makru a v případě neznámého názvu souboru ho lze takto jednoduše a bez hrozby diskvalifikace zadat.

Úvodní makro také nastaví v celém dokumentu zarovnání k levému okraji (takzvaně na praporek vpravo).

Z pohledu soutěže je asi nejdůležitější záměna pevných mezer za normální. Pokud by v dokumentu zůstaly pevné mezery, může se stát, že některé korektury nebudou provedeny zcela správně. Z tohoto důvodu je úvodní makro nutné spustit.

Úvodní makro se spustí klávesou [F5] a o jeho provedení je soutěžící utvrzen informačním oknem, které potvrdí tlačítkem OK.

## 2.7 Závěrečné makro

Závěrečné makro je neméně důležité, jak to úvodní, a to hned z několika důvodů:

- V místě, kde jsme skončili, vloží pět prázdných řádků. Toto je vyžadováno pravidly soutěže a je to z důvodu, aby bylo poznat, kde soutěžící skončil.
- V celém dokumentu nastaví písmo Courier New. Toto písmo je nastaveno i úvodním makrem, ale protože vyhodnocovací program ZAV-Jury měl problémy, když mu byl dodán soubor s jiným písmem, tak proto je pro jistotu nastavení písma i v závěru celé soutěže.
- Odstraní dvojité mezery, které mohly během některých korektur vzniknout. Např. při vložení slova, přesunu atd. Tyto dvojité mezery by jinak byly považovány za chyby, což by ve výsledku mohlo znamenat poslední místo v umístění i pro toho nejlepšího soutěžícího.

Závěrečné makro se spustí klávesou [F12]. Je možné použít i klávesovou zkratku [Alt]+[k], která vychází z dřívějších šablon maker.

## 2.8 Zvýraznění umístění kurzoru

I když makro je navrženo tak, aby se soutěžící na monitor nemusel skoro nikdy podívat, nemůžeme tuto možnost vyloučit. I soutěžící je jenom člověk, který dělá chyby, a je občas potřeba se na monitoru přesvědčit o výsledku. Pokud se podíváte na monitor, vůbec netušíte, kde se nachází kurzor (opomenuli případ, kdy se na monitor podíváte před provedením první korektury, kdy kurzor je na začátku textu).

K lepší orientaci tedy makro barevně označí vždy čtyři slova za kurzorem. Hned na první pohled je vidět, kde se kurzor nachází. Bez této drobnosti by hledání kurzoru bylo na dlouho. Zvlášť, když při spuštění makru se kurzor na monitoru neobjeví (neblíká na své pozici, není vůbec vidět).

## 2.9 Logování příkazů

Potřeba logu vychází z doby tvorby a testování makra, ale dobře slouží i nyní. Mockrát se mi stalo, že mi někdo hlásil, že některý příkaz byl špatně proveden. Pokud se několika lidem provede dobře a jednomu špatně, je hodně pravděpodobné, že je chyba na jeho straně. Stačí se podívat do souboru vl.txt, který je umístěn v kořenové části na disku C:. Lze tak jednoduše dopátrat, jaký příkaz byl zadán a kde se soutěžící přepsal, případně zadal něco zcela špatně.

Protože se všechny záznamy ukládají do jednoho souboru, tak je potřeba nějak odlišit, ke které předloze dané příkazy v logu patří. Proto se vždy před příkazy objeví kopie několika slov začátku předlohy (to zaznamenává úvodní makro). Na dalších řádcích je už vidět pořadové číslo, datum a čas a hlavně patřičný příkaz, který byl zadán. Zaznamenávají se sem i nestandardní věci jako např. zadání špatného příkazu, zvuková signalizace při nalezení více výskytů atd.

Vyhotoveno dne 2007-10-22