

Nagy Bence (gimb@freemail.hu)

Huhyphn dokumentáció*

2003. március 24.

1. Huhyphn

A *Huhyphn* projekt célja olyan elválasztómodul kidolgozása, amely hibátlanul választja el az élő magyar nyelv szavait.

Ez az elválasztómodul a \TeX , az *OpenOffice.org* és a *LibHnj* könyvtárat használó alkalmazások számára teszi lehetővé az algoritmus által szabott keretek között magyar nyelvű szövegek elválasztását.

1.1. Előzmények

A fenti rendszerekben jelenleg használt elválasztómodult – *Huhyph 3.12* – tekintjük a hivatalos magyar elválasztómodulnak. Ezt a modult MIKLÓS DEZSŐ hozta létre 1989-ben, majd MAYER GYULA fejlesztette tovább, és jelenleg is ő a hivatalos Huhyph-vonal karbantartója. A legújabb változat – *Huhyph 4.0* – kísérleti jellegű, és noha 2002 júniusában megjelent már, nem került be a disztribúciókba.

A *Huhyph 3.12* és a *Huhyph 4.0* között azonban jelentős szemléletbeli különbség van.

1.1.1. Huhyph 3.12

A magyar nyelv elválasztásában a fonetikus és az összetétel szerinti szabályok játszanak szerepet. Általánosságban elmondhatjuk, hogy minden szót a fonetikai szabályok szerint kell elválasztani, azonban összetett szavak esetében az elválasztási pontnak az összetétel határára kell esnie. Idegen eredetű szóösszetételeknél e két szabály alkalmazása ingadozhat.

* A dokumentum jelen verziója is a *Huhyphn* elválasztási mintáival készült; ha hibás elválasztás van benne, akkor szégyellem magamat.

A *Huhypn 3.12* ezért azt az elvet követi, hogy az elválasztási minták alapját a kézzel rögzített fonetikai szabályok adják, és majdnem minden egyes összetétel elválasztását külön minta szabályozza. Így minden egyes szó elválasztása a fonetikai szabályok szerint történik, hacsak nem került be a kézzel szerkesztett minták közé az ezt szabályzó kivétel.¹ A módszer hátránya nyilvánvaló, hiszen minden egyes összetett szót, amelynél a fonetikai szabályok szerinti elválasztás nem az összetétel határára esik, külön fel kell venni a minták közé.

Olvasáspszichológiailag indokolható, hogy az összetételek határán lévő egy szótagot képző magánhangzók az elválasztás során az őket tartalmazó összetevőben maradjanak, ezért gyakran az amúgy jól elválasztódó szavakra is külön mintát kell alkalmazni. Nyelvtanilag lehetséges a *rádi-óadó* és a *rádióá-dó* forma, de vitán felül áll a *rádió-adó* változat elsőbbsége.

A kézzel szerkesztett mintagyűjtemény esetén felmerülhet, hogy a kollekció kevésbé optimálisan tárolja az egyes kivételeket lekezelő mintákat. Bővítése során erre tekintettel kell lenni, ezért egy-egy minta hozzáadása alapos utánagondolást és a többi minta átnézését igényli.

1.1.2. Huhypn 4.0

A *Huhypn 4.0* már a FRANK LIANG és PETER BREITENLOHNER által fejlesztett *PatGen* programmal készült. A *PatGen* egy elválasztásokat tartalmazó szótár alapján hozza létre az elválasztási mintagyűjteményt, így a *Huhypn 4.0* változatának alapja is egy nagyméretű szótár, mely a a *TypoTeX* Kiadó néhány könyvének szövegét tartalmazza.

A *PatGen*-nel történő mintagenerálás nagy hiányossága, hogy a létrehozott modul teljes bizonyossággal csak a szótárban szereplő szavakon működik. A program optimális eredményt ad a megadott szótárra vonatkoztatva, de ez nem jelenti azt, hogy ez a nyelv szavainak egészére érvényes. Mivel a magyar nyelv agglutináló, ezért rengeteg toldalék, illetve toldalékok kombinációja járulhat egy-egy szóhoz. Nyelvünkben létezik a hangkivetős tövek jelensége is, amikor úgy tűnik, mintha az egyes toldalékok nem az alapszóhoz, hanem annak módosulatához csatlakoznak. Ezért sok esetben hibás elválasztást kapunk egy ragozott szónál akkor is, ha annak töve szerepel a szótárban.

1. VERHÁS PÉTER *HiOn* programja is ezt az elvet követi, kiegészítve az igekötők kezelését megoldó véges állapotú automatával.

1.2. Problémák

A fentiek alapján látható, hogy a magyar nyelvű elválasztási minták generálása korántsem egyszerű feladat. A *Huhyp* 3.12 és *Huhyp* 4.0 esetén alkalmazott módszerek akármelyikét vizsgálva jelentős nehézséggel kerülünk szembe.

A kézzel szerkesztett fonetikai bázisra épülő változatnál rengeteg szót kell megvizsgálni és a rájuk alkalmazható mintákat megtalálni. Az új minták felvétele esetén meg kell vizsgálni a régebbieket, hogy azok módosításával elérhető-e a kívánt eredmény, vagy az új minta felvétele esetén találkozunk-e olyan szóval, amelyik eddig helyesen választódott el, de az új minta felvételével már hibásan.

A *PatGen* által készített kollekció esetén pedig akkor érhetünk el optimális eredményt, ha egy szónak valamennyi képzett alapjának elkészítjük az elválasztott formáját, és felvesszük a szótárba. A *Huhyp* 4.0 szókészlete nem haladja meg a 70.000-es méretet, a mintagenerálás folyamata azonban egy 1 GHz-es Pentium IV-es számítógépen majdnem nyolc percet vesz igénybe. Amennyiben minden szónak további változatait képezzük, úgy a művelethez szükséges idő nagyságrendekkel nőhet, és akár órákig is tarthat. Nem lehetünk azonban biztosak ekkor sem abban, hogy a szótárból hiányzó szavak megfelelően választódnak el, csak reménykedhetünk, hogy a szótár növekedésével a fonetikai szabályok és a növekvő számú azonos kivételek átlépik azt a kritikus tömeget, hogy érvényesüljenek a nem felvett szavakra is.

1.3. Megoldások

A *PatGen* kiválóan használható alkalmazás, hogyha meg tudjuk kerülni a használatából származó hátrányait. A fenti problémák alapján két elvárásnak kell megfelelni:

1. A generált minták tartalmazzák teljes egészében a fonetikai szabályokat, hogy a szótárban nem szereplő szótagok esetén is érvényesüljenek.
2. A generált minták ne befolyásolják a fonetikai szabályok szerint választandó szavak elválasztását.

Az első elvárást egyszerű teljesíteni, a *PatGen*-t előre elkészített mintákkal kell meghívni, amelyek tartalmazzák a szótagolás szerinti elválasztás szabályait.

A második elvárás teljesítése a nehezebb, ugyanis itt a *PatGen* működése okozza a problémát. A program ugyanis a bemeneti szótárt alapul véve határozza meg az összetétel szerint elválasztandó szavak esetében a szükséges alkalmazandó mintát. Mivel optimális megoldásra törekszik, ezért ez a minta a lehető legrövidebb lesz, és így könnyen előfordulhat az, hogy egy szótárban nem szereplő, a fonetikai szabályokkal elválasztható szónál hibás elválasztást fogunk kapni.

A *PatGen* megfelelő paraméterezésével érhetjük el, hogy ne törekedjen optimális eredmény generálására, de ebben az esetben is a bemeneti szótár duzzasztására van szükség. Nem agglutináló nyelvek esetén bőven elég lenne a szó felvétele a szótárba, de a magyarban minél több toldalékolt alakot fel kell venni. Erre egy módszer, ha könyvek szavait a *Mispell*-en keresztül átszűrjük. Ez a helyesírás-ellenőrző meg tudja mondani egy szóról, hogy az valamely szótőnek a toldalékolt alakja-e. Ha ismerjük a szótő elválasztását, akkor a toldalékolt alak elválasztását is meghatározhatjuk, mivel a toldalékok minden esetben a fonetikai szabályok szerint választandók el. A szótár növekedése ebben az esetben nem lesz annyira drasztikus, mint egy algoritmus-sal valamennyi létező toldalékolt alakot felvennénk, ugyanis így csak a ténylegesen használt formák kerülnek a szótárba.

A szótár gyarapodásával együtt jár a feldolgozási idő növekedése, de ezt az árat meg kell fizetnünk.

1.4. Jelentőség

Amíg a \TeX -rendszer főleg tudományos körökben használt eszköz maradt fejlesztésének évei alatt, a Liang-féle elválasztó algoritmus problémái csak kevés embert érintettek. Mivel a \TeX nyílt rendszer, ezért bárki készíthet hozzá kiegészítéseket, javításokat, így korábban a szakértő felhasználók is orvosolni tudták gondjaikat, és elmondhatjuk, hogy a \TeX szellemiségébe beleillik ez a fajta felhasználói változtatás.

Az *OpenOffice.org* általános célú irodai programcsomag, amelybe szintén ezt az elválasztó algoritmust építették be, azonban ez a rendszer már a szélesebb közönséget célozza meg. A program a „középiskolás fokon” oktató számítástechnikai ismeretekkel is egyszerűen használható, ezért elterjedése nem lehet kétséges. Mivel GPL-licence révén szabadon terjeszthető, ezért a Linux operációs rendszer elsőszámú irodai programcsomagjává vált, és része a legtöbb disztribúciónak – aki egy modern Linux disztribúciót telepít, az előbb-utóbb találkozik a programmal. Az asztali gépen Linuxot használók száma még csekély a Microsoft Windows felhasználóihoz képest, de valószínűleg ez utóbbi platformon is gyorsan fog nőni a programot használók száma, nem beszélve a többi operációs rendszerről, amelyen elérhető.

Az *OpenOffice.org*-ba épített *LibHnj* programkönyvtárat RAPH LEVIEN írta 1998-ban (2001-től *ALTLinux* néven érhető el), ennek a magas szintű elválasztás és sor-kiegyenlítés a feladata. Forráskódja szintén szabadon felhasználható, ezért várható, hogy újabb programokba is be fogják építeni. Az egyik ilyen ismert alkalmazás, a *Scribus* DTP-rendszer, melynek jelenleg még csak béta-verziója létezik.

A helyes elválasztás kérdésével várhatóan egyre több felhasználó fog találkozni, és a jelenleg hivatalosnak mondható *Huhypn 3.12* valószínűleg már nem fogja kielégíteni az igényeiket.

1.5. Felhasznált projekteredmények

A *Huhyphn* fejlesztése során több nyílt forráskódú projekt eredményének felhasználása történt meg.

1.5.1. ElMe

Az *ElMe* az Egyszerű Linuxos Morfológiai Elemző rövidítése. A projektet NAGY VIKTOR indította 1999-ben, a programnak mindössze egy 0.1-es változata készült el. A program egyszerű morfológiai elemzőként szolgál, egy szóról leválasztva annak toldalékait meghatározza a szófaját.

A program része egy nagyméretű szógyűjtemény, mely a *Szóvéghmutató szótár* adatbázisát használja, és mintegy 58.000 szót tartalmaz. Az adatbázis a szavakon kívül nyelvtani információkat is tartalmaz, melyek alapján meg lehet határozni azok szófaját és szóösszetételek esetén az összetevők számát.

Az összetevők számának ismeretében egy keresőprogrammal sikerült azokat összetevőkre bontani, a szó különböző darabokra bontott részeit kerestük a szótár nem összetett szavai között, és ez csekély hibaszázalék és jól meghatározható hibajelenségek mellett jó eredményt adott. Az összetevőket ezek után már a fonetikai szabályok szerint el lehetett választani. Mivel az idegen eredetű összetételek más jelölést kaptak, ezért azok feldolgozását elkülönítve lehetett elvégezni, és a többi összetétel esetén biztonsággal lehetett használni a fonetikai szabályokat.

Mivel a szógyűjtemény a 70-es évekből származik, ezért sok helyen a nyelv változása miatt helyesírási hibát és néhol az ékezetes betűknél elgépelést tartalmaz. Ezért a szavak szűrésén és javításon is átmentek.

1.5.2. Magyar Ispell

A *Magyar Ispell* projektet NÉMETH LÁSZLÓ indította el. A célja, hogy szabad szoftverek felhasználásával magyar nyelvű helyesírás-ellenőrzőt készítsenek. A projekt eredményei már ma is lenyűgözőek, noha még mindig nincsen belőle hivatalosan kiadott verzió.

Ezt a helyesírás-ellenőrzőt használja a magyar *OpenOffice.org*, és parancssorból használva \TeX -források, HTML- és XML-dokumentumok is ellenőrizhetőek vele.

A *Huhyphn* projekt céljaként szerepel a *Magyar Ispell* által nyelvtanilag megfelelőnek ítélt szavak hibátlan elválasztása. A *Magyar Ispell* szóállománya folyamatosan bővül, felépítésének köszönhetően egyszerű hozzá szaknyelvi modulok létrehozása és karbantartása.

Mivel a célrendszerek megegyeznek a *Huhypn* által kitűzötttekkel, ezért a *Magyar Ispell* szóállományának felhasználása folyamatosan meg fog történni.

A szótárba felvett toldalékolt alakok képzését is a *Magyar Ispell* projekt keretében készülő *Mispell* alkalmazás egyszerűsíti meg.

1.5.3. HiOn

A *HiOn* elválasztóprogramot VERHÁS PÉTER írta. A program a fonetikai szabályokat felhasználva működik, és ezt kiegészítendő egy összetett szavakat tartalmazó kivételszótárat tartalmaz. Ebben sok olyan szó megtalálható, amely máshol nem szerepel, ezért ezek felvétele is megtörtént.

1.5.4. Huhyp 4.0

A megfelelő magyar elválasztómodul létrehozására indított eddigi legnagyobb projekt MAYER GYULA érdeme. Az általa létrehozott szótár minimális számú hibát és az általa alkalmazott elvektől való eltérést tartalmaz. A szótár sok olyan idegen eredetű nevet és szóösszetételt tartalmaz, melyek elválasztásához tudományos szintű nyelvészeti ismeretek szükségesek, ezeknél mindenképpen érdemes a szótár általi változathoz ragaszkodni.

1.6. Telepítés T_EX alá

A T_EX-rendszerhez rendelkezésre álló mintagyűjtemény a *huhypn.tex* nevű fájlban található. A fájlban található karakterek az EC-, T1- vagy más néven Cork-kódolás szerint szerepelnek. Ez a szokásos magyar nyelvű L^AT_EX használat mellett eredményezi a megfelelő működést. A fájlt a texmf-fa */tex/generic/hyphen* könyvtárába kell másolni, majd a *mktexlsr* programmal frissíteni a fájlnyilvántartást.

A fájl megfelelő helyre másolása és a konfigurációs fájlok szükséges módosítása után szükség van a formátumfájlok legenerálására, mivel a szótárak feldolgozása nem futásidőben történik. A teT_EX-rendszeren a beállítások végrehajthatóak a *texconfig* programmal, amely a különböző makrócsomagoknál teszi lehetővé az eltérő beállítás használatát, és gondoskodik a formátumfájlok elkészítéséről is. Az alábbiakban a kézzel történő beállítások találhatók.

1.6.1. PlainT_EX és L^AT_EX

A PlainT_EX és L^AT_EX makrócsomagok esetén a betöltendő elválasztási minták a *language.dat* fájlban találhatóak. Ennek szokásos helye a texmf-fa

```
/tex/generic/config
```

könyvtárában található. Szerepelnie kell benne egy

```
magyar huhypn.tex
```

tartalmú sornak, esetleg százalékjellel az elején. Ezt a sort tegyük megjegyzésbe, és egy másik sorba írjuk a következőt:

```
magyar huhypn.tex
```

A formátumfájl legenerálásakor már a *Huhypn* elválasztási mintái épülnek be.

1.6.2. ConT_EXt

A ConT_EXt más megközelítést használ. A texmf-fában található a

```
/tex/context/config/cont-usr.tex
```

nevű fájl, melyben a rendszer minden egyes elválasztási mintához definiál egy egy-
séges szinonimát. A magyar nyelv definícióját a következő sor tartalmazza.

```
\definefilesynonym [lang-hu.pat] [huhypn.tex]
```

Ez kell módosítani az alábbira:

```
\definefilesynonym [lang-hu.pat] [huhypn.tex]
```

Ha még nem tettük volna meg, töröljük a százalékjelet a következő sor elejéről, ezzel
érhetjük el, hogy a magyar elválasztási minták beforduljanak a formátumba.

```
% \installlanguage [\s!hu] [\s!status=\v!start] % hungarian
```

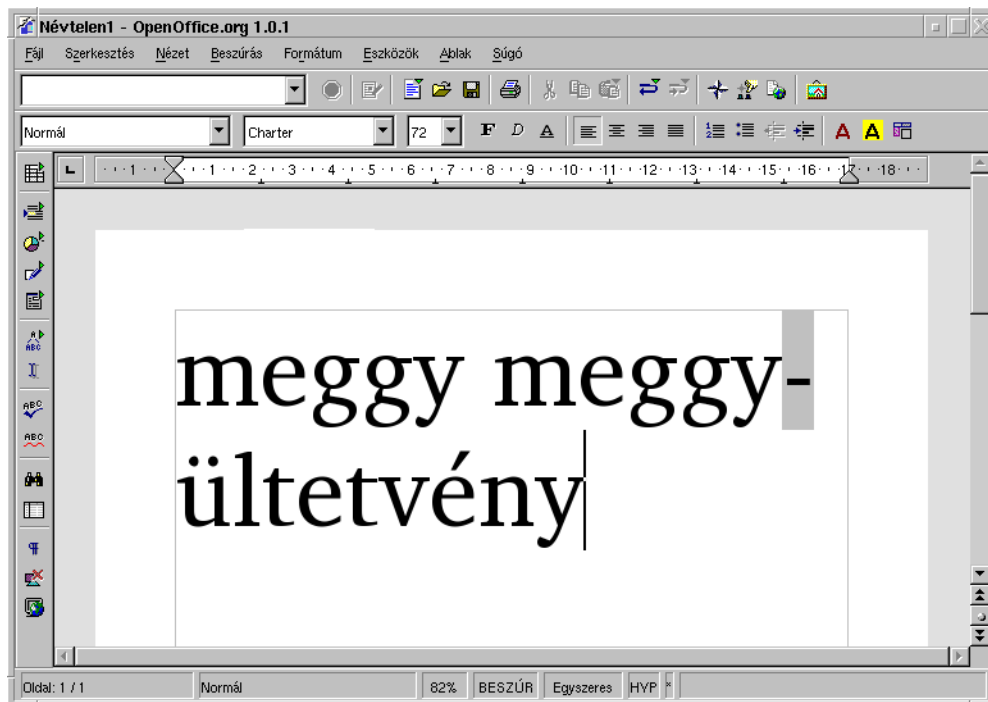
Ezek után generáljuk le a formátumfájlt.

1.7. Telepítés OpenOffice.org alá

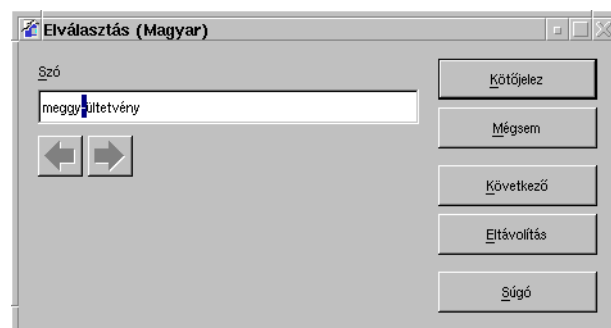
Az *OpenOffice.org* irodai programcsomaghoz használható elválasztási mintafájl a
`hyph_hu.dic` nevet viseli. Ezt a következő könyvtárba másolva írjuk felül az eredeti
változatot (a könyvtár disztribúciótól függően változhat):

```
/usr/lib/OpenOffice.org/share/dict/ooo/
```

A program ezek után már a *Huhypn* elválasztási mintáit használja.



1.1. ábra: Az *OpenOffice.org* ablakában látható immár helyes elválasztás.



1.2. ábra: Az elválasztás engedélyezésének ablaka.

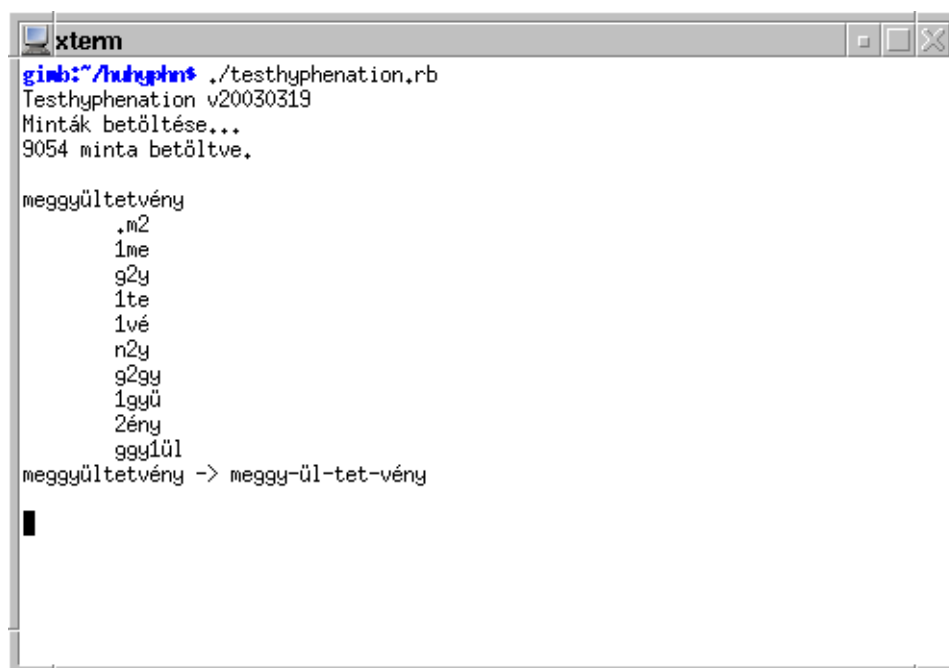
1.8. Az elválasztómodul tesztelése²

Az elválasztómodul tesztelésére külön alkalmazás szolgál. A *Testhyphenation* Ruby nyelven írt program, mindössze egy fájlból áll, mely a `testhyphenation.rb` nevet viseli. A Ruby programnyelvről bővebb információt a www.ruby-lang.org oldalon lehet találni, innen tölthető le a rendszer forráskódja is.

2. A példán látható „meggyültetvény” szót NÉMETH LÁSZLÓ küldte nekem, ezt egyik korábbi magyar modul se választotta el helyesen.

A programot elindítva szavakat írhatunk be szöveges terminálon, melyeket kiír elválasztva, valamint az illeszkedő elválasztási mintákat is felsorolja. Így hibás elválasztás esetén egyszerű megállapítani, hogy azt mely helytelen minta okozhatta. Egy üres enter lenyomásával léphetünk ki a programból.

A program működéséhez a T_EX-hez készült elválasztási mintafájlt (`huhypn.tex`) használja. Mivel ebben a fájlban a hungarumlautos ékezetes betűk az EC-kódolás szerintiek, a mintákat beolvasáskor a Latin2-es kódkészletre konvertálja át.



```
xterm
giub:~/huhypn$ ./testhyphenation.rb
Testhyphenation v20030319
Minták betöltése...
9054 minta betöltve.

meggyültetvény
  .m2
  1me
  g2y
  1te
  1vé
  n2y
  g2gy
  1gyű
  2ény
  ggy1ül
meggyültetvény -> meggy-ül-tet-vény
```

1.3. ábra: A *Testhyphenation* futása szöveges terminálban, az alkalmazott minták egy tabulátorhellyel beljebb íródnak ki az elválasztott szó felett.

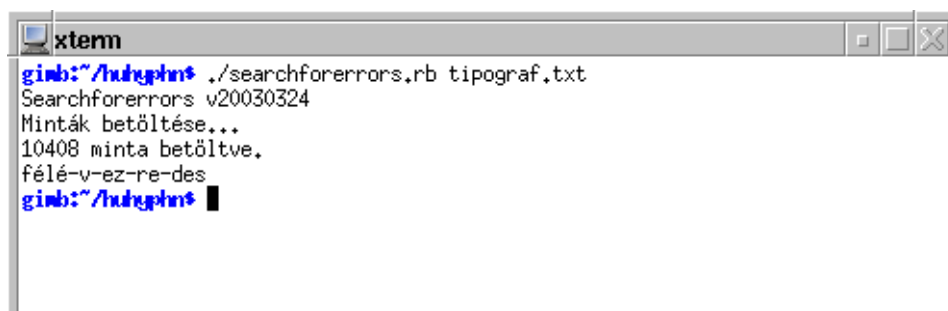
1.9. Hibakeresés

A *Searchforerrors* alkalmazás a `searchforerrors.rb` fájl futtatásával indítható. A programnak bemenetként egy fájlnevet kell megadni, melyből a csak betűkből álló szavakat elválasztja, és ezekben hibák után kutat. Ez a program szintén a T_EX-hez készült elválasztási mintafájlt használja fel.

Háromféle hibát lehet kiszűrni vele:

1. Csak egy mássalhangzó kerül két elválasztójel közé.³ Ezt a hibajelenséget egy túlságosan optimalizált minta okozza, mely a szó szótárba vételével egy hosszabb alakot fog felvenni.
2. Csak egy magánhangzó kerül két elválasztójel közé, melyek másik oldalán mássalhangzók állnak. Ez csak abban az esetben lehetséges, hogyha a magánhangzó szóösszetétel határán áll. Mivel összetett szavak határánál az egybetűs szótagok elválasztása nem ajánlott, ezért ezt is hibának tekintjük.
3. Az egyszerűsítve kettőzött hosszú mássalhangzók mellett mássalhangzó vagy eltérő hangrendű magánhangzók⁴ állnak. Ebben az esetben minden bizonnyal összetett szóról van szó, így egy elválasztójel elhelyezésére mindenképpen szükség van.

Eddigi tapasztalataim alapján úgy tűnik, hogy egy fájlban mindössze egy-két ezreléki ilyen jellegű hiba fordul elő. A leggyakoribb hibafajta – egy összetett szó a fonetikus szabályok szerint formailag helyesen, de egyébként az összetételt tekintve hibásan van elválasztva – ezzel a programmal nem fedezhető fel, így az ilyen szavak megtalálása a továbbiakban sem nélkülözheti az emberi tényezőt.



```
xterm
gimb:~/huhypn$ ./searchforerrors.rb tipograf.txt
Searchforerrors v20030324
Minták betöltése...
10408 minta betöltve.
félél-v-ez-re-des
gimb:~/huhypn$
```

1.4. ábra: A *Searchforerror* futása szöveges terminálban. SZÁNTÓ TIBOR *A tipográfia nyelve* című cikkét a MEK-ről töltöttem le, ebben a fájlban mindössze egy hibát talált.

1.10. Elérhetőség

Jelenleg a gimb.freeweb.hu/huhypn oldalon érhetőek el a projekt keretében létrehozott elválasztómodulok és programok, melyeket a GPL-licenc feltételei szerint lehet felhasználni.

3. Ilyen legfeljebb idegen szavak vagy régies nevek esetén és csak kétjegyű mássalhangzóknál fordulhat elő, de ez igen ritka.
4. Az *i* és *í* magánhangzók hangrendje ingadozik, ezért ezeket nem vizsgálja, mert a legtöbb esetben csak téves risztást adna.