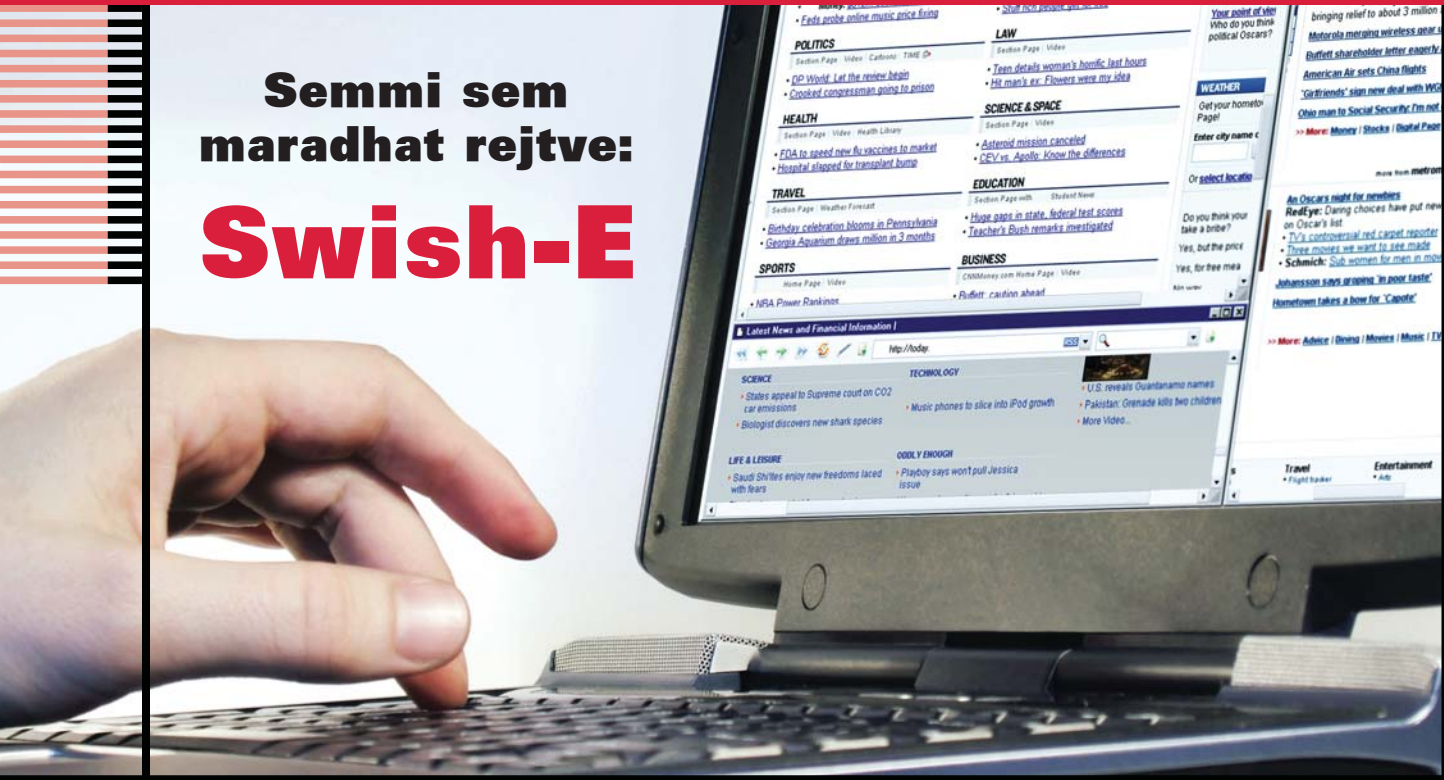


Semmi sem maradhat rejtve: Swish-E



© Kiskapu Kft. Minden jog fenntartva

A swish-e egy egyszerűen használható, rugalmas alkalmazás, amellyel web oldalakat és egyéb fájlokat indexelhetünk. Nem csak szöveges állományokat, de elektronikus leveleket, PDF-, HTML-, XML-, doc-, ppt- és xls fájlokat is képes indexelni, sőt bármit, ami XML/HTML/text formátumba konvertálható...

■ A *swish-e* lehetőségeinek teljes listája a http://www.swish-e.org/docs/readme.html#key_features weblapon található meg. Kipróbálásához első lépésben telepítsük az alkalmazást. Töltsük le a legutolsó verziót a <http://swish-e.org/> webhelyről, majd adjuk ki az alábbi utasításokat:

```
tar zxvf latest.tar.gz
cd swish-e-2.4.3
./configure
make
su -c 'make install'
```

Ha web oldalakat is akarunk indexelni, telepítsük a *HTML-Tagset*, *HTML-Parser*, *Compress-Zlib*, *Crypt-SSLeay* (<https://> oldalak eléréséhez) és *libwww-perl* modulokat a CPAN-ról (<http://www.cpan.org/>). Ezek azért szükségesek, mert a *swish-e* a *swishspider Perl* alkalmazást használja web indexelésre. Első példaként indexeljük a saját web oldalunkat! Először azonban készítsük el a *swish-e* konfigurációs állományát. A paraméterek teljes

listája a <http://www.swish-e.org/docs/swish-config.html> címen található.

Adjuk ki a `swish-e -S http -c swish-e.conf -i http://www.mydomain.hu/`

```
1. Lista A swish-e konfigurációs állománya
# kifejezések kereséséhez használja a swish-e BumpPositionCounter
↳ Characters |.
# maximum 10 lépés mélységben keres
MaxDepth 10
# 2 http kérés között 0 másodpercet vár
Delay 0
# az aktuális könyvtárba teszi a spider az ideiglenes állományokat
TmpDir .
# mennyi információt írjon ki a terminálra indexelés közben
IndexReport 2
# ebben a könyvtárban van a swishspider program
SpiderDirectory ./src
```

2. Lista Így indexel

```
Indexing Data Source: "HTTP-
  ↳ Crawler"
Indexing
  ↳ "http://www.mydomain.hu/"
retrieving
  ↳ http://www.mydomain.hu/
(0)...
retrieving
  ↳ http://www.mydomain.hu/
  ↳ howitworks.html (1)...
retrieving
  ↳ http://www.mydomain.hu/
  ↳ install.html (1)...
.... itt még sok html oldal
következik ....

Removing very common words...
no words removed.
Writing main index...
Sorting words ...
Sorting 2,773 words
alphabetically
Writing header ...
Writing index entries ...
  ↳ writing word text: Complete
  ↳ writing word hash: Complete
  ↳ writing word data: Complete
2,773 unique words indexed.
4 properties sorted.
20 files indexed. 97,767
  ↳ total bytes. 13,055 total
  ↳ words.
Elapsed time: 00:00:08 CPU
  ↳ time: 00:00:00
Indexing done!
```

utasítást. Erre a 2. Listában olvasható üzenetek jelennek meg, jelezvén, hogy az elemzés elkezdődött. Ennek hatására az aktuális könyvtárban létrejött két fájl *index.swish-e* és *index.swish-e.prop* néven. A futtatás során (az *IndexReport* változótól függően) több dolgot is kiír, például az éppen indexelt címekeket, hogy éppen az indexet írja, a szavakat rendezzi, hogy hány egyedi szót talált, hány webcímet nézett végig, azok mérete mekkora, illetve hogy mindez meddig tartott. Azokat a webhelyeket amelyekben szerepel a training szó, az alábbi parancs kiadásával kereshetjük meg:

```
swish-e -c swish-e.conf -w
↳ "training"

# SWISH format: 2.4.3
# Search words: training
# Removed stopwords:
# Number of hits: 10
# Search time: 0.003 seconds
# Run time: 0.028 seconds
1000 http://www.mydomain.hu/
↳ training.html "Training the
↳ token database" 5002
410 http://www.mydomain.hu/
↳ install.html "Installation"
↳ 5750
410 http://www.mydomain.hu/
↳ config.html "clapf.conf"
17127
205 http://www.mydomain.hu/
↳ questions.html "Questions"
↳ 6144
205 http://www.mydomain.hu/
↳ "clapf" 3256
.
```

A hashmarkkal kezdődő sorok tartalmazzák a *swish-e* verzió számát, a keresett kifejezést, a találatok számát és a futás illetve keresés idejét. A találatok egyes sorai négy részből állnak: találati relevancia (minél nagyobb ez a szám, annál fontosabb, relevánsabb az adott dokumentum), a webhely, majd annak címe, végül a dokumentum mérete byte-ban. A keresés eredményét egy pont (.) zárja. A *swish-e* csak szöveges állományok indexelésére képes. Segédprogramok (szűrők) segítségével azonban képes akár a .doc, .pdf, stb. állományokat szövegekké alakítani, majd indexelni. Adjuk hozzá az alábbi sort a konfigurációs állományhoz, majd indexeljünk újra a web oldalunkat! Az újabb futtatáskor az indexelt webhelyek között a *PDF* állományok is meg fognak jelenni, illetve a találatok között is, ha megfelel a keresési feltételeknek:

```
FileFilter .pdf
↳ /usr/local/bin/pdftotext
↳ "'%p' -"
```

Nem csak web oldalakat, de a számítógépünkön tárolt állományokat is indexelhetjük. Ehhez módosítsuk az 1. Listában szereplő konfigurációs állományt úgy, hogy elhagyjuk a következő direktívákat: MaxDepth, Delay és SpiderDirectory, majd indexeljünk!

```
swish-e -c swish-e2.conf -s fs
↳ -i swish-e-2.4.3/src/test
```

Keressük meg azokat az állományokat, amelyekben szerepel a This is just kifejezés:

```
swish-e -c swish-e2.conf -w
↳ 'This is just'

1000 /home/sj/temp/swish-e-
↳ 2.4.3/tests/test_xml.html "If
↳ you are seeing this, ..." 159
778 /home/sj/temp/
↳ swish-e-2.4.3/tests/test.xml
↳ "test.xml" 126
565 /home/sj/temp/swish-e-
↳ 2.4.3/tests/test.txt
↳ "test.txt" 55
```

Ez a módszer nem jó, mert az első kettő találat nem tartalmazza a szóban forgó kifejezést. A kereső logikai VAGY kapcsolatba hozta a „This”, „is” és „just” szavakat. Próbáljuk újra:

```
swish-e -c swish-e2.conf
↳ -w "This is just"

1000 /home/sj/temp/swish-e-
↳ 2.4.3/tests/test.txt
↳ "test.txt" 55
```

Ez már jó. A különbség csak annyi, hogy a This is just kifejezés még külön kettős idézőjelek közé lett téve, és így a *swish-e* már mint kifejezést használta.

Ha az Olvasónak van egy web oldala, amihez keresőt szeretne, akkor mindössze annyit kell tennie, hogy ír egy *CGI* programot, amely csövön (pipe) keresztül lefuttatja például az iménti parancsot, majd annak kimenetét valamilyen kellemes formába öntve visszaadja a böngészőnek. Erre csak abban az esetben biztatom az Olvasót, ha rendkívül körültekintően írja meg a *CGI* input szűrését, hogy ne lehessen érvénytelen adattal a gépünk biztonságát veszélyeztetni. Sokkal biztonságosabb, ha a *swish-e* biztosította *Perl* vagy *C API*-t használjuk, amelyek segítségével *Perl* vagy *C* nyelven fejleszhetünk olyan alkalmazást, amely képes a *swish-e* rendszert használni, majd a tőle kapott eredményt megjeleníteni. Nézzünk meg egy egyszerű programot (az *src/libtest.c* alapján) a *C API* használatára!

3. Lista Egy a swish-et használó program

```

/*
 * fordítás: gcc -O2 -Wall -o kereso kereso.c
 * -lswish-e
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <swish-e.h>

static void display_results(SW_HANDLE
    ↪ swish_handle, SW_RESULTS results){
    SW_RESULT result;

    while((result = SwishNextResult(results))){

        printf("Rank: %ld, Title: %s, Path: %s,
    ↪ Size: %ld\n",
            SwishResultPropertyULONG(result,
    ↪ "swishrank"),
            SwishResultPropertyStr(result,
    ↪ "swishtitle"),
            SwishResultPropertyStr(result,
    ↪ "swishdocpath"),
            SwishResultPropertyULONG(result,
    ↪ "swishdocsize"));
    }
}

int main(int argc, char **argv){
    SW_HANDLE swish_handle=NULL;
    SW_RESULTS res=NULL;

    swish_handle = SwishInit("index.swish-e");
    if(SwishError(swish_handle))
        SwishAbortLastError(swish_handle);

    if(argc < 2){
        printf("hasznalat: kereso kulcsszo\n");
        exit(0);
    }

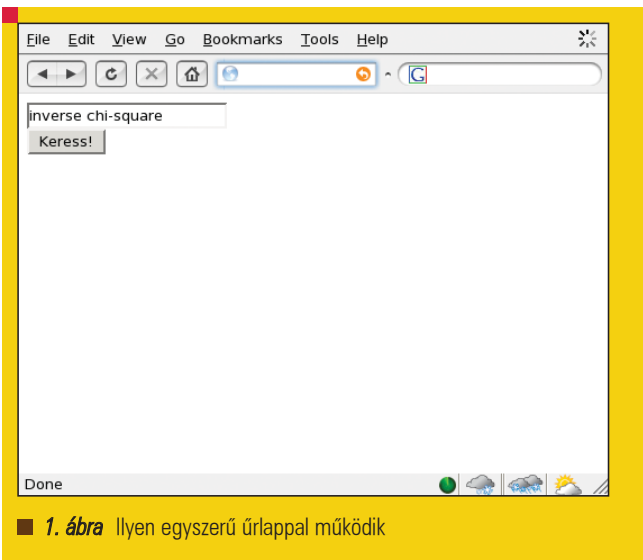
    res = SwishQuery( swish_handle, argv[1]);
    if(SwishError(swish_handle))
    ↪ SwishAbortLastError(swish_handle);

    printf("Talalatok szama: %d\n",
    ↪ SwishHits(res));

    if(res)
        display_results(swish_handle, res);

    Free_Results_Object(res);
    SwishClose(swish_handle);

    return 0;
}
    
```



Ez a program megkeresi az argumentumként megadott szót vagy kifejezést az *index.swish-e* állományban, kiírja a találatok számát, illetve az egyes találatokhoz tartozó relevanciát, a dokumentum címét, webhelyét és méretét.

Tegyük fel, hogy az olvasó egy webkiszolgálón akar keresés funkciót készíteni. Az előbbi kód kiegészítésével gyerekjáték az egész, mindössze arra van szükség, hogy átadjuk a keresett kifejezést egy *HTML* űrlap segítségével, dekódoljuk a *CGI* adatot, majd

valami dizájnolt köritéssel kiírjuk a keresés eredményét. Ha az olvasó meg akarja spórolni ezt az ujjgyakorlatot, akkor használja a kibővített programot, amely egy tetszőleges *HTML* sablon állományba illeszti be az eredményt, ill. sok találat esetén lapozni

4. Lista A swish-e „protokollja”

```
Path-Name: keddi_tv_musor
Content-Length: 18
Last-Mtime: 1143554804
Document-Type: HTML*
```

Ez itt az adat...

tud közöttük. A program a (http://dev.acts.hu/swish_e_kereso-0.1.tar.gz) címről tölthető le. Az 1. és 2. ábrán az említett program látható működés közben.

Web oldalak indexelése során a *swish-e* nem veszi figyelembe az idegen oldalakra mutató hivatkozásokat, szigorúan csak a megadott web helyen belül marad. Egyszer azonban úgy kellett egy népszerű oldalt indexelnem, hogy bizonyos megadott idegen hivatkozásokat is követnem kellett. Erre azonban nem képes a *swishspider*, a *swish-e* gyári modulja. Mi sem mutatja azonban jobban a *swish-e* rugalmasságát, hogy

magunk is készíthetünk saját *spider* alkalmazást. Így kiindulásként vettem a *swishspider Perl* programot, és ez alapján megírtam a saját pókomat, ami úgy indexel egy web oldalt, ahogy én fütyülök. A *swish-e* készítői szerint megfelelő pókkal bármit indexelhetünk, nem csak a fájlrendszert vagy a honlapunkat, de akár egy *MySQL* adatbázist, a *Thunderbird* postaládánkat, az *MP3* gyűjteményünket éppúgy, mint a jövő heti tv-műsort. Ehhez mindössze egy olyan program szükséges, amely – az előző példákat tekintve – lekérdezi az adatbázist, feldolgozza postafiókunkat vagy megszerzi valahonnan a tv-műsort, és azt átadja a *swish-e* programnak.

Ha elkészültünk saját pókunkkal (szuperpok.pl), futtassuk a *swish-e*-t az alábbi módon:

```
swish-e -c 3.conf -s prog -i
↳ szuperpok.pl
```

A külső program paraméterezése:

```
# ezzel a direktívával lehet
parametereket megadni a külső
```

```
spider programnak, pl.
SwishProgParameters
mysql://localhost/
```

A külső segédprogram a *HTTP* protokollhoz hasonló módon kommunikál a *swish-e* alkalmazással, az alábbi 4. Lista önmagáért beszél.

A *swish-e* egy rendkívül sokoldalú indexelő alkalmazás, amely bármit képes feldolgozni, amit (akár különféle segédprogramokkal is) szöveges formában át lehet neki adni. A cikkben leírt példák mind olyan feladatok nyomán keletkeztek, amelyeket valahogyan meg kellett oldanom. A *swish-e*-nél alkalmasabb eszközt aligha találhattam volna.



Sütő János
(jsuto@freemail.hu)

1997 óta használ Slackware Linux-ot. Szabadidejében a postfix clap nevű vírus- és spam-szűrőjét polírozza.

