

Emacs és Vim fej-fej mellett

Horák György, Török Gábor

Magyarországi Ubuntu Konferencia

2008.09.13.

Miről nem lesz szó?

Why are we
hiding from the
police, daddy?

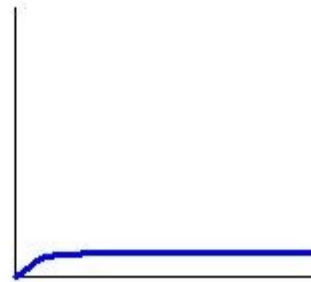
Because we
use vi son,
they use emacs.



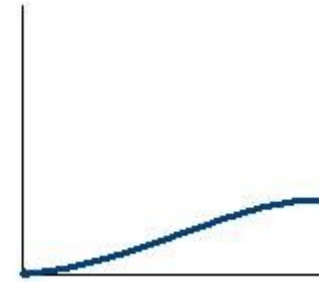
Miről lesz szó?

Classical learning curves for some common editors

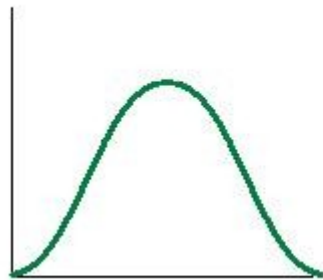
Notepad



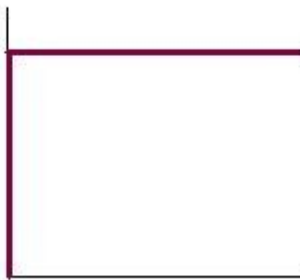
Pico



Visual Studio



vi



emacs



Hogyan indítható el?

emacs

emacs -nw

emacsclient

vi

vim

gvim

Hol fut?

mindenhol telepíthető

emacs csomag

Windows, Linux, OS X stb.

mindenhhol telepíthető

*nix alapból vi

Windows, Linux, OS X stb.

**Milyen nyelven
érhető el?**

csak angol

nincs támogatás többnyelvűséghez

beépített dokumentáció angol

elsődlegesen angol

gvim kezelőfelülete lokalizált

beépített dokumentáció angol

Indulási sebesség

gyári kiszerelés 1-2 mp alatt indul

bekapcsolt funkcióktól függ

emacs -Q

kliens-szerver mód javasolt

byte-compile

„Vim can load and fully syntax-colour a file before Emacs even displays its window.”

azonnal indul

pluginekkel lassítható

Konfiguráció, testreszabás

„An extensible, customizable, self-documenting real-time text editor.”

.emacs

menüből

Emacs Customization

`.vimrc`

menüből (csak session idejére)

Felhasználói felület felépítése



```

text-decoration: underline;
}
div.links span.comment_comments,
div.links li.comment_comments a,
div.links li.comment_add a,
div.links span.comment_forbidden {
padding-left: 20px;
background: url(images/comment.png) left top no-repeat;
}
div.links li.comment_comments a.comments-new {
background: none;
padding-left: 0;
}
.node_read_more, .blogmark_visit {
font-size: 1.1em;
}
.node .meta, .comment .meta {
padding: 0 0 0.5em 0.4em;
margin-bottom: ... 1
margin-left: ... 2
}
.node .margin: ... 3
background: ... 4 top no-repeat;
padding-right: ... 5
float: ... 6
margin-top: ...
width: 50%;
}
.node .content {
margin: 0 0 .5em 0;
padding-bottom: 0.8em;
font-size: 1.05em;
overflow: hidden; /* pl. /archive/all/1999 */
}
span.visitcount, a.statistics_counter {

```

```

Working file: /home/gabor/dev/weblabor/wl-libikoka/root
/sites/all/themes/wl/style.css
-----
r677 | cece | 2008-09-08 14:52:59 +0200 (h, 08 szept 20
08) | 1 line
ez mashol is bejatszott, revert
-----
r675 | gabor | 2008-09-08 14:48:22 +0200 (h, 08 szept 2
008) | 2 lines

```

-U:%%- *vc-change-log* Top (3,67) (SVN-Log-View yas fr
Horák György, ... The GNU Opera...

- Location: <http://konf.ubuntu.hu/2008/program/emacs-es-v>
- [kezdőlap](#)
 - [regisztráció](#)
 - [program](#)
 - [helyszín](#)
 - [sajtó](#)
 - [kapcsolat](#)

Horák György, Török Gábor: Emacs és vim fej fej mellett

Ha szövegszerkesztőre, fejlesztő-környezetre terelődik a szó, gyakran kerül említésre az Emacs-Vim páros, többnyire mint a nehezen tanulható, keménydió eszközök, amelyeket a megszállottakon kívül úgy sem használ senki – különben is Eclipse és mcedit rulez. Az előadás során célunk az említett két eszköz, az Emacs és Vim lehetőségeinek, szolgáltatásainak



4 /u//p/s/d/c/context_processors.py .vimrc ~/.bashrc

{not in Vi}

1. Starting diff mode

The easiest way to start editing in diff mode is with the "vimdiff" command. This starts Vim as usual, and additionally sets up for viewing the differences between the arguments.

```
vimdiff file1 file2 [file3 [file4]]
```

This is equivalent to:

diff.txtH [unix:utf-8:help]

notab|ic [c] 25, 33 4%

```

+ 1 +-- 34 lines: """"-----+ 1 +-- 34 lines: """"-----
- context_processors.py (/opt 35 context_extras['debug'] = True 35 context_extras['debug'] = True
|- class 36 from django.db import connectio 36 from django.db import connection
|| PermLookupDict 37 context_extras['sql_queries'] = 37 context_extras['sql_queries'] =
|| PermWrapper 38 return context_extras 38 return context_extras
39 39
- member 40 def i18n(request): 40 def i18n(request):
|| __init__ [PermLookupDic 41 from django.utils import translatio 41
|| __repr__ [PermLookupDic 42 42
|| __getitem__ [PermLookup 43 context_extras = {} 41 context_extras = {}
|| __nonzero__ [PermLookup 44 context_extras['LANGUAGES'] = setti 42 context_extras['LANGUAGES'] = settin
|| __init__ [PermWrapper] 45 context_extras['LANGUAGE_CODE'] = t 43 if hasattr(request, 'LANGUAGE_CODE')
|| __getitem__ [PermWrapp 44 44
|| __iter__ [PermWrapper] 45 else:
46 46
- function 46 context_extras['LANGUAGE_BIDI'] = t 47
|| auth 47 47
|| debug 48 return context_extras 48 from django.utils import translation
|| i18n 49 49
|| media 50 50
|| request 51 return context_extras 51
52 52
- context_processors.py (/usr 50 def media(request): 53 def media(request):
|- class 51 """ 54 """
|| PermLookupDict + 52 +-- 24 lines: Adds media-related contex + 55 +-- 24 lines: Adds media-related context
|| PermWrapper 76 class PermWrapper(object): 79 class PermWrapper(object):
77 def __init__(self, user): 80 def __init__(self, user):
78 self.user = user 81 self.user = user
79 82
__Tag_List__ <] 24, 1 6% [ 85] context_processors.py < 41, 1 Top [ 84] context_processors.py < 43, 34 Top
/language

```

Bilentyűkombinációk

GNU stílusú (bash, most)

Escape/Meta/Alt-Control-Shift

major mode, minor mode

M-x

segítség: C-h c, C-h k

Unix stílusú (bash -v, less, more)

vim mode-ok:

normal

insert (i, a, o, c, s, r)

command (:)

visual (v)

[select (gh), ex (Q)]

Mozgás a pufferben

GNU stílusú (C-a, C-e, C-n, C-v)

C-<karakter>, M-<karakter>

M-g M-g

C-l

insert mode-ban nem mozgunk

h j k l

d, c

:<szám>, gg, G

z Z

Puffer pozíció mentése

register

bookmark

C-u C-space

marks

m<karakter>, '<karakter>,

`<karakter>

! , ` .

jumps

viminfo

Inkrementális és regex-alapú keresés

C-s, C-r

„grep”-szerű regex tudás

M-x grep

kisbetűs keresőszó case insensitive

inkrementális keresés néha lassú

M-/ (csere akár regexen)

/ ?, n N

regex

set incsearch

set ignorecase

s///

Menekülés

C-g

ESC ESC ESC

jelzi a lenyomott billentyűket

set showcmd

ESC, de inkább ^[

**Egyidejű szerkesztés
több fél között**

M-x make-frame-on-display
tty-mode, kliens-szerver

nem igazán támogatott

vim –remote*

Verziókezelés

elterjedt VCS-ek támogatása
közös felület valamennyi VCS-hez
aktív VCS választható

vcscommand plugin

(CSV, SVN, SVK, git)

http://www.vim.org/scripts/script.php?script_id=90

clearcase plugin

http://www.vim.org/scripts/script.php?script_id=15

Undo-redo

undo: C-

redo: C-

az undók is undozhatók

végtelen előzménylista

redo-mode

u, Ctrl-r

:earlier <idő>, :later <idő>

g-, g+

:undolist

Parancsok ismétlése, makrók

C-u <szám> <parancs>

F3, F4, M-<szám> F4

makrók tárolhatók, módosíthatók

Emacs Lisp

<parancs> <szám> <mozgás>

q <karakter>, [műveletek], q

@ <karakter>

**Dolgozás kijelölt
területen**

kijelöléssel dolgozhatunk*

*többnyire

narrowing: szűkített munkaterület

kijelöléssel dolgozhatunk
narrowing plugin

Shell elérése

parancs futtatása: M-/

parancs futtatása kijelölésen: M-|

M-x term: shell indítása pufferben

M-x eshell: Emacs Lisp Shell

:!<parancs>, :.!<parancs>

:[tartomány]!<parancs> (filter)

:shell

vimshell plugin

**Más felhasználó
birtokolta file-ok
szerkesztése**

tramp (sudo, su)

C-x C-f /sudo:/<file>

sudo vim :)

:e scp://root@localhost/<file>

Távoli file-ok elérése

tramp (rcp, scp, ftp stb.)

C-x C-f /ssh:user@host:/<file>

távoli könyvtár és file böngészés

rcp, scp, ftp stb.

:e scp://user@host/<file>

könyvtárakat is lehet böngészni

**Bővíthetőség,
skriptelhetőség**

„An extensible, customizable, self-documenting real-time text editor.”

Emacs Lisp (.el)

nincs hivatalos tároló

emacs-wiki.org-on összegyűjtve

VimScript

TCL, Python, Perl, Ruby, MzScheme

getscript, :GLVS

www.vim.org-on összegyűjtve,

rangsorolhatóan

.vba

Fejlesztést támogató eszközök

szintaxis kiemelés, auto-indent

CEDET

debugger

ctags/etags

flymake

rope, emacs-rails, jde, slime

skeleton, yasnippet

szintaxis kiemelés

auto-indent

vimdiff

Omni complete (Ctrl-X Ctrl-O)

:make, quickfix

taglist plugin

snippetsEmu plugin

Átjárás

viper-mode

vile

NetBeans, Eclipse

conkeror

Firemacs

vimacs

cream

MS VisualStudio, Eclipse

**Legnagyobb
megnyitható file**

bármekkora*

bizonyos mode-ok lassíthatják

bármekkora

Memória foglalás

gyári kiszereles: 15M

enyem: 43M

szerver-kliens

kb. 9M

Beépített segítségek

beépített tutorial: C-h t

M-x help

Elisp referencia

vimtutor

:help <bármí>

Kilépés

C-x C-c

C-5 0

:q
ZQ

Mire használom?

fejlesztés (JS, Python, PHP, Elisp)
mindennapi élet (:

konfigurációs file-ok

fejlesztés (Python, régebben C++)

mindig, amikor szöveget kell

szerkeszteni :)

Köszönjük a figyelmet

x Horák György

gyuri@horak.hu

x Török Gábor

gabor@20y.hu

Felhasznált irodalom

John Dierdorf: *Vi and Emacs*

<http://www.io.com/~dierdorf/emacsvi.html>

<http://www.gnu.org/>

<http://www.vim.org/>