



VI (VIM)

Introducere

- *vi* este un editor disponibil pe TOATE sistemele UNIX (consuma extrem de putine resurse)
- *vim* ("Vi Improved") este o clona din *vi*
- interfata grafica pentru *vi* este *gvim*

Ce putem face cu vim?

- *vim* este un editor de texte care poate fi configurat pentru diverse necesitati
- ofera *syntax highlight* pentru numeroase limbaje de programare, scripting, hipertext si fisiere de configurare
- auto completion
- crearea de *macro*-uri
- etc.

Moduri

- modul *ECRAN* Modul in care vim porni by default. Folosit pentru navigare dar si pentru executarea de comenzi.
- modul *INSERT* Din modul ecran se intra cu taste predefinite (vezi urmatorul paragraf). Acest mod este folosit pentru inserarea de caractere in fisier.
- modul *COMANDA* Din modul ecran se intra cu ":"
- modul *VISUAL* Din modul ecran se intra cu "v". Folosit pentru selectarea la nivel de caracter (copy/paste pe portiuni de text). Dupa selectarea portiunii din text folositi **y** pentru a face yank (copy) sau **d** (delete) pentru a sterge. Acel text va fi pastrat in buffer, putand fi inserat cu ajutorul tastei **p** (paste).

Comenzi de baza

Deplasarea in text

- **h**, **l**, **k**, **j** pentru a muta cursorul (se pot folosi si sagetile)
- **w** deplasare inainte cu un cuvant (word)
- **b** deplasare inapoi cu un cuvant (back)
- **0** deplasare la inceputul randului
- **\$** deplasare la sfarsitul randului
- **H** deplasare la inceputul primei linii din pagina(High)
- **M** deplasare la inceputul liniei din mijlocul paginii (Medium)
- **L** deplasare la inceputul ultimei din pagina (Low)

Stergere si editare

- **x** sterge caracterul marcat de cursor (del in Windows)
- **X** sterge caracterul de dinaintea cursor (backspace in Windows)
- **dd** va sterge o linie
- **dw** sterge de la pozitia cursorului pana la sfarsitul cuvântului
- **db** sterge de la pozitia cursorului pana la inceputul cuvântului
- **d0** sterge de la pozitia cursorului pana la inceputul randului
- **d\$** sterge de la pozitia cursorului pana la sfarsitul randului
- **yy** va copia o linie (in buffer. *yum=copy*)
- **p** va face *paste*
- **u** undo
- **Ctrl+R** redo
- **i** pentru a intra in modul insert inainte de pozitia cursorului
- **a** INSERT dupa pozitia cursorului



- ○ INSERT pe linie noua

Salvare, Cautare, Iesire din/in fisier

- `:wq` va salva fisierul si va iesi din program (write and quit)
- `:q!` va iesi fortat din fisier, fara a salva schimbarile
- `/string` va cauta sirul de caractere "string" in fisier (cautare inainte)
- `?string` cautare inapoi
- `!:comanda` executa comenzi shell. De exemplu puteti rula comanda `less` pe alt fisier fara a iesi din vim. **FOARTE IMPORTANT**

Management ferestre

- `:split` imparte ecranul in doua ferestre separate orizontal
- `:vsplit` imparte ecranul in doua ferestre separate vertical
- `Ctrl-W` pentru a naviga intre ferestre
- `:close`, `:quit` inchide fereastra curenta
- `:only` inchide toate ferestrele mai putin cea curenta

TASK

- Introduceti la consola comanda `vim test_text`, pentru a deschide fisierul `test_text`.
- Observati ca modul in care a pornit este modul ecran.
- Folositi comenzile de navigare pentru a muta cursorul (`h,l,j,k`).
- Intrati in modul INSERT folosind tasta `i`. Scrieti un caracter, apoi iesiti din modul insert cu `ESC` si intrati la loc folosind tasta `o`. Ce observati?
- Din modul *ECRAN* stergeti prima linie folosind `dd` si lipiti-o (paste) dupa primul paragraf folosind `p` apoi faceti **yank** (copy) pe aceasta linie si copiat-o unde era initial. Observati ca toate aceste comenzi au fost facute din modul *ECRAN*.
- Apasati `/`. In partea de jos-stanga a ecranului a aparut caracterul `/`. Tastati cuvantul **machine** si apasati `ENTER`. Cursorul se va muta imediat pe prima aparitie a cuvantului **machine**. Pentru a repeta cautarea acestui sir apasati `n` (next).
- Intrati in modul **COMANDA** folosind `:`
- In partea de jos-stanga a ecranului au aparut `:`, aceasta inseamna ca sunteti in modul COMANDA. Orice sir de caractere va fi interpretat drept comanda.
- Tastati `help`, apoi apasati `ENTER`. Acum va aflati in "help". Tastele din modul ecran sunt valabile si in **help**.
- Cautati sirul de caractere "pattern". Cursorul va fi pozitionat pe prima aparitie a cuvantului "pattern". La inceputul liniei curente apare `/usr_27.txt/`, acesta este un **hyperlink**.
- Plasati cursorul pe acest hyperlink si apasati `CTRL-]`. Cursorul este trimis la explicatiile pentru cautare avansata. Aici gasiti informatii despre moduri de cautare cu diferite filtre. Pentru a "iesi" din hyperlink se foloseste `CTRL-o`.
- Urmariti hyperlink-ul legat de Character ranges (`/27.6/`) si folositi informatiile de acolo pentru a gasi toate cuvintele care incep cu una din majusculele de la A la G.

EMACS

Introducere

- EMACS este un editor de text pentru multe tipuri de fisiere
- Este utilizat de catre cei mai multi programatori deoarece permite editarea de fisiere, compilarea si debug-ul, spre deosebire de vim care nu permite decat editarea cu syntax highlight



Cu ce incep?

- Prescurtari: *C=CONTROL* (CTRL pe tastatura), *M=ALT*. Exemplu *C-x f* inseamna *CONTROL+x f*
 - Lansarea programului se face din consola tastand `emacs nume_fis`
 - Lansarea tutorialului `C-h t`
 - Comenzi pentru "deplasarea" in fisier
 - `C-f` (forward) deplasare inainte cu un caracter
 - `C-b` (backward) deplasare inapoi cu un caracter
 - `M-f` deplasare inainte cu un cuvint
 - `M-b` deplasare inapoi cu un cuvint
 - `C-n` (next) deplasare pe linia urmatoare
 - `C-p` (previous) deplasare pe linia precedenta
 - `C-v` (*v* seamana cu o sageata orientata in jos) deplasare inainte cu un "ecran"
 - `M-v` deplasare inapoi cu un ecran. La deplasarea cu un ecran se vor pastra ultimele doua linii ale ecranului precedent pentru a asigura continuitatea textului.
 - Alte comenzi:
 - `C-x C-f` creare/deschidere fisier
 - `C-x C-s` salvare fisier
 - `C-x 2` impartirea ecranului orizontal
 - `C-x 3` impartirea ecranului vertical
 - `C-x C-c` iesire din program
 - `C-s` cautare
 - Comenzi repetitive `C-n x comanda`, unde *x* reprezinta numarul de repetari pentru o comanda
 - **Pentru a ANULA orice comanda folositi `C-g`**

Extensii ale programului

- *Calc* - calculator
- *Calendar - mode*
- *Doctor* - psihoterapie :)
- *Emacs/W3* - web browser
- *ERC* - IRC client
- *Tetris*
- etc.

Moduri majore din Emacs

- Fundamental Mode
- Text Mode
- Outline Mode
- Lisp Mode
- C Mode

TASK

1. Deschideti Emacs `emacs`
2. Intrati in tutorial `C-h t`
3. Folositi comenzile pentru deplasare in interiorul textului pentru a va familiariza cu ele
4. Creati o fereastră separata printr-o linie orizontala de cea veche si mutati cursorul in noua fereastră folosind `C-x o`.
 - Din noua fereastră deschideti fisierul `test_text`
 - Deschideti fereastră care indica ce alt buffer (fereastră/fisier) mai sunt deschise `C-x C-b`
 - Inchideti toate ferestrele mai putin cea cu `test_text` `C-x 1`
 - Schimbati bufferul curent in cel care contine tutorialul `C-x b`



- Creati o noua fereastră despartita orizontal in care sa avem tutorialul si parcurgeti cat mai mult din el.

5. ACASA: de terminat de citit tutorial.

Emacs vs. Vim http://en.wikipedia.org/wiki/Editor_war

VARIABLE DE MEDIU

Introducere

- Orice **shell** activ memoreaza informatiile pe care le foloseste in *VARIABLELE DE MEDIU*
- O *variabila de mediu* poate contine informatii privitoare la locatiile fisierelor de configurare, mailboxes, directoarele din path, promptul shell-ului respectiv, tipul sistemului de operare etc.
- Folositi comanda `declare | less` pentru a afisa variabilele de mediu setate pentru shell-ul curent (bash in cazul nostru)
- `set`
- Pentru a afisa valoarea unei variabile de mediu, de exemplu `$USER` tastati `echo $USER` si va aparea numele utilizatorului logat. La fel se procedeaza pentru orice alta variabila de mediu.

Variabile de mediu standard (Exemple)

- Orice variabila are un *nume* si o *valoare*. Atunci cand folosim caracterul "\$" inseamna ca ne referim la *valoarea* variabilei.
 - Exemplu: `echo HOME` va afisa la numele variabilei HOME
 - `echo $HOME` va afisa valoarea variabilei HOME, si anume calea directorului "home" al userului care introduce comanda.
- `$BASH` - contine calea completa a comenzii `bash`, in cele mai multe cazuri `/bin/bash`
- `$HOME` - directorul home al utilizatorului logat
- `$PS1` - modul in care apare promptul (data, zi, username, etc.)
- `$UID` - user id, id-ul asignat userului logat (toate uid se gasesc in `/etc/passwd`)

Comanda *export*

- Putem crea sau modifica variabilele de sistem folosind comanda `export`
- Exemplu: vrem sa adaugam un director la variabila `$PATH`, pentru a putea executa un script (program) fara sa folosim calea catre acel script
`export PATH=$PATH:/home/student/director_program`
(Inainte de egal NU folosim caracterul "\$" deoarece ne referim la numele variabilei pe care vrem sa o schimbam. Dupa egal FOLOSIM caracterul "\$" deoarece schimbam **valoarea** variabilei de sistem)
- **SCHIMBARILE VARIABLELOR FOLOSIND COMANDA EXPORT SUNT TEMPORARE, DECI ELE SE PIERD DUPA LOGOUT**
- Pentru a face aceasta schimbare permanenta, trebuie editate fisierele `$HOME/.bashrc` si `/etc/profile`, in functie de variabila pe care dorim sa o schimbam
- Daca dorim sa resetam o variabila (modificata cu `export`) la valoarea ei default, folosim comanda `unset $VARIABILA_DE_MEDIU`

TASK

- Creeati o variabila de mediu, apoi modificati-o cu ajutorul comenzii `export`