

Utilitários GNU

A Ferramenta certa

Saiba um pouco mais sobre duas ferramentas desenvolvidas pelo Projeto GNU e conhecidas por toda comunidade: o editor Emacs e o interpretador de comandos Bash, incluindo algumas funções desenvolvidas por brasileiros para agilizar o trabalho com a linha de comando. **POR CHRISTIANO ANDERSON**



Um sistema operacional completo é composto por várias ferramentas como editores, interpretadores de comando, compiladores e depuradores, entre outras, para tornar possível a interação entre homem e máquina. O Projeto GNU é pioneiro no desenvolvimento de várias destas ferramentas. No decorrer dos meses, esta coluna vai falar um pouco sobre elas e apontar o que os brasileiros vêm desenvolvendo para tornar seu uso ainda mais amigável.

Emacs

O Emacs é conhecido como editor, mas não é um simples editor. Muitas pessoas acham um exagero o pacote do Emacs no Debian ter mais de 20Mb de tama-

nho. É provável que você não necessite de tudo isto: talvez só uma parte do Emacs já baste para fazer o que você precisa. Outra reclamação dos usuários é em relação aos atalhos de teclado; esse fator, porém, deixa de ser um problema se você partir para a personalização. O Emacs é hoje o editor mais personalizável, além de extremamente modular. É possível alterar a aparência do Emacs por completo, inclusive as teclas de atalho, o que torna o editor muito poderoso e adaptável a qualquer usuário. Entretanto, se tal usuário estiver fora de seu ambiente de origem, pode sentir-se desconfortável sem seus próprios atalhos e comandos – nesse caso a personalização acaba sendo uma desvantagem.

As vezes é um pouco complicado descrever o Emacs. O que dizer de um editor de texto que acaba sendo um IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado), em que também é possível ler e-mail, jogar Tetris, bater papo em um canal de IRC, servidor Jabber ou outro sistema de mensagens instantâneas, depurar um código e até mesmo navegar na Internet?

Sim, é complexo. Mas quem se aventura no Emacs acaba não o largando mais, pois a flexibilidade e a liberdade que esse poderoso editor proporciona aos usuários acaba compensando qualquer dificuldade inicial de adaptação. Muitas pessoas dizem ter essa dificuldade com o Elisp [3], uma variante da linguagem de programação Lisp [4], que compõe a base do Emacs. Se você quer aprender personalizar seu Emacs, aconselho uma visita aos vários links mostrados no fim deste artigo.

Escrevendo código

Gosto muito de Python. Quem conhece essa linguagem de programação sabe que ela trabalha com indentação para delimitar blocos. Programar em Python dentro do Emacs é uma tarefa divertida. O editor ajuda o programador a não se perder no código, indenta os blocos automaticamente e insere cores no meio do código separando cada função, bloco, classe e variáveis. Ainda por cima, dá para dividir a janela do Emacs em duas partes: na parte de baixo acompanho o resultado da execução do código que estou digitando em cima. Isso mesmo, o Emacs mergulha no interpretador Python e exibe o resultado de cada bloco de código.

O mesmo ocorre com outras linguagens de programação, como C, C++, Perl, Ada, Lisp, Fortran, Java, etc. O Emacs se adapta às particularidades de cada linguagem, garantindo ao programador muitas horas de “diversão”.

Quadro 1 – Exemplos das funções ZZ

```
prompt$ zzsenna
qyYx5m

prompt$ zzdicabl estorvo
estorvo (ô) s.m. cf. estorvo, do v. estorvar
estorvor (ô) s.m.

prompt$ zzdolar
07/05/2003 compra   venda   hora
Comercial  2.955   2.962   15:20
Paralelo   2.980   3.060   14:57
Turismo    2.930   3.030   14:57

prompt$ zldata 22/12/1999 + 69
29/02/2000
```

E depois de terminar meu código? Ainda dentro do Emacs, posso adicioná-lo a um repositório CVS, criar um patch ou pegar um pedaço desse código e colá-lo a um e-mail.

Recado da vovó

Enquanto desenvolvo meu código em Python, deixo um outro buffer (como são chamadas as janelas de edição no Emacs) aberto verificando meus e-mails de tempos em tempos. Quando chega um e-mail com alguma característica pré-determinada, o filtro entra em ação e pode, por exemplo, me avisar que recebi um e-mail da minha avó. Sem sair do editor, posso acessar minha Inbox utilizando o módulo Gnus[5], ler a mensagem e respondê-la. O mesmo pode ser feito com um canal em um servidor IRC, já que posso dividir a tela em duas e acompanhar uma discussão enquanto programo no buffer superior.

O Emacs é um canivete suíço para quem procura agilidade e não quer perder tempo usando diversos aplicativos para realizar tarefas diferentes. Ainda por cima, o editor corrige as palavras que digitei errado, com o dicionário ibrazilian! O EmacsWiki[6] é um site interessante para obter informações sobre o Emacs, suas extensões e exemplos de uso. Para quem quer se aventurar a ler e-mails no Emacs, sugiro consultar o site My Gnus[7], que possui vários exemplos de configuração. Por último, se você utiliza Debian, aconselho a instalação dos pacotes *emacs21*, *emacs21-el*, *emacs-goodies-el* e *emacs-goodies-extra-el*. Divirta-se!

Bash

Se você já mexeu com alguma distribuição GNU/Linux, certamente já usou o Bash, mesmo sem saber. Aquela “tela preta”, como alguns o chamam, é uma forma poderosa de trabalhar com o sistema. Muitas pessoas utilizam o Bash da maneira como ele veio na distribuição, isto é, não utilizam configurações especiais que o tornam ainda mais útil.

Na distribuição Debian, é possível habilitar o recurso *bash_completion* editando os arquivos *.bashrc* e *.bash_profile* no diretório pessoal de seu usuário. O *bash_completion* é uma das inúmeras ferramentas que estão embutidas

no Bash, que acaba funcionando como linguagem de programação (veja a série “Papo de Botequim” nesta edição).

Funções ZZ

Aurélio Jargas[8], conhecido por ter editado livros e documentos sobre expressões regulares, criou uma série de funções em Bash para facilitar a vida daqueles que gostam de uma linha de comando. As Funções ZZ [9] são resultado do trabalho do Aurélio e de outros colaboradores para criar extensões úteis para toda a comunidade. Todas as funções começam com as letras *zz<função>*. Entre outros, há utilitários para apagar linhas repetidas em um arquivo, apagar linhas em branco, consultar dicionários online, coletar notícias de sites da Internet, ver o resultados de loterias, ver a grade de programação de um canal de TV, consultar mecanismos de busca, ver cotações de moedas e fazer cálculos com horas, datas, etc. Alguns exemplos de uso (extraídos do site), são mostrados no Quadro 1.

O funcionamento das funções ZZ é bastante interessante, pois elas chamam aplicativos diversos que se baseiam muitas vezes em expressões regulares para coletar informações na Internet. Algumas ferramentas da Free Software Foundation, como o *wget*, são utilizadas em conjunto com o Bash. ■

SOBRE O AUTOR

Christiano Anderson (anderson@gnu.org) é desenvolvedor autônomo de Software Livre, participa do Projeto Software Livre Brasil (PSL-BR) e contribui com o Projeto GNU. Sua principal missão é difundir o software livre e sua filosofia, principalmente no mercado educacional.

INFO

- [1] <http://www.gnu.org/software/emacs/>
- [2] <http://www.gnu.org/software/bash/>
- [3] <http://www.gnu.org/software/emacs/emacs-lisp-intro/emacs-lisp-intro.html>
- [4] <http://www.lisp.org/table/contents.htm>
- [5] <http://www.gnus.org>
- [6] <http://www.emacswiki.org>
- [7] <http://my.gnus.org/>
- [8] <http://www.aurelio.net>
- [9] <http://www.aurelio.net/zz/>